

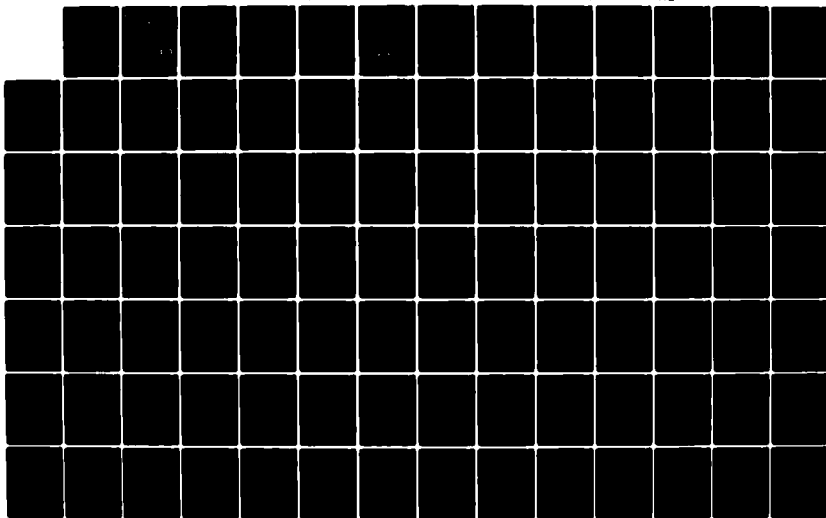
AD-A148 363 A GENERALIZED ESCAPE SYSTEM SIMULATION (GESSI) COMPUTER
PROGRAM VOLUME 2 G. (U) KETRON INC WARMINSTER PA
L A D'AULERIO ET AL. APR 84 NADC-84068-60

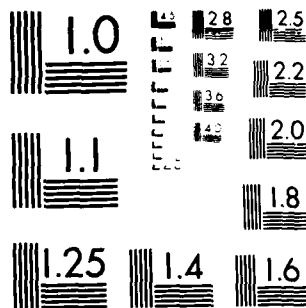
1/4

UNCLASSIFIED N62269-81-C-0206

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

REPORT NO. NADC-84068-60



**A GENERALIZED ESCAPE SYSTEM SIMULATION (GESS)
COMPUTER PROGRAM: GESS PROGRAMMER'S MANUAL
VERSION II — VOLUME II**

AD-A148 363

Louis A. D'Aulero
Aircraft and Crew Systems Technology Directorate (Code 6032)
NAVAL AIR DEVELOPMENT CENTER
Warminster, PA 18974

and

David A. Fender
KETRON, INC.
444 Jacksonville Road
Warminster, PA 18974

APRIL 1984

FINAL REPORT
AIRTASK NO. A5315312-001D-1531W1100
Work Unit No. WC530
Contract No. N62269-81-C-0206

**DTIC
ELECTE**
S DEC 6 1984 **D**
B

Approved for Public Release; Distribution is Unlimited

Prepared for
NAVAL AIR SYSTEMS COMMAND (AIR-5312)
Department of the Navy
Washington, DC 20361

84 11 07 031

DTIC FILE COPY

NOTICES

REPORT NUMBERING SYSTEM — The numbering of technical project reports issued by the Naval Air Development Center is arranged for specific identification purposes. Each number consists of the Center acronym, the calendar year in which the number was assigned, the sequence number of the report within the specific calendar year, and the official 2-digit correspondence code of the Command Office or the Functional Directorate responsible for the report. For example: Report No. NADC-78015-20 indicates the fifteenth Center report for the year 1978, and prepared by the Systems Directorate. The numerical codes are as follows:

CODE	OFFICE OR DIRECTORATE
00	Commander, Naval Air Development Center
01	Technical Director, Naval Air Development Center
02	Comptroller
10	Directorate Command Projects
20	Systems Directorate
30	Sensors & Avionics Technology Directorate
40	Communication & Navigation Technology Directorate
50	Software Computer Directorate
60	Aircraft & Crew Systems Technology Directorate
70	Planning Assessment Resources
80	Engineering Support Group

PRODUCT ENDORSEMENT — The discussion or instructions concerning commercial products herein do not constitute an endorsement by the Government nor do they convey or imply the license or right to use such products.

APPROVED BY:


V. J. GALLAGHER
CAPT, MSC, USN

DATE:

13 September 1994

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM															
1. REPORT NUMBER NADC-84068-60	2. GOVT ACCESSION NO. 10-1148	3. RECIPIENT'S CATALOG NUMBER 363															
4. TITLE (and Subtitle) A GENERALIZED ESCAPE SYSTEM SIMULATION (GESS) COMPUTER PROGRAM: GESS PROGRAMMER'S MANUAL VERSION II - VOLUME II		5. TYPE OF REPORT & PERIOD COVERED FINAL REPORT															
7. AUTHOR(s) Louis A. D'Aulerio David A. Fender		8. CONTRACT OR GRANT NUMBER(s) N62269-81-J-0206 Task No. 630-1944															
9. PERFORMING ORGANIZATION NAME AND ADDRESS See Reverse		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS AIRTASK NO. A5315312-001D-1531W1100 Work Unit No. WC530															
11. CONTROLLING OFFICE NAME AND ADDRESS NAVAL AIR SYSTEMS COMMAND (AIR-5312) Department of the Navy Washington, DC 20361		12. REPORT DATE APRIL 1984															
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Aircraft and Crew Systems Technology Directorate (Code 6032) Naval Air Development Center Warminster, PA 18974		13. NUMBER OF PAGES 351															
		15. SECURITY CLASS. (of this report) UNCLASSIFIED															
16. DISTRIBUTION STATEMENT (of this Report) Approved for Public Release; Distribution is Unlimited		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE															
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) <table border="0"> <tr> <td>Coordinate Systems</td> <td>Parachute</td> <td>Runge-Kutta</td> </tr> <tr> <td>Equations of Motion</td> <td>Rails</td> <td>DART</td> </tr> <tr> <td>Forces and Moments</td> <td>Stabilization</td> <td>DRI</td> </tr> <tr> <td>Catapult</td> <td>Aerodynamic Drag</td> <td>Dynamic CG</td> </tr> <tr> <td>Rockets</td> <td>Droque Slug/Container</td> <td></td> </tr> </table>			Coordinate Systems	Parachute	Runge-Kutta	Equations of Motion	Rails	DART	Forces and Moments	Stabilization	DRI	Catapult	Aerodynamic Drag	Dynamic CG	Rockets	Droque Slug/Container	
Coordinate Systems	Parachute	Runge-Kutta															
Equations of Motion	Rails	DART															
Forces and Moments	Stabilization	DRI															
Catapult	Aerodynamic Drag	Dynamic CG															
Rockets	Droque Slug/Container																
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The Generalized Escape System Simulation (GESS) program is a computerized mathematical model for dynamically simulating the performance of existing or developmental aircraft ejection seat systems. The program generates six-dimensional trajectory predictions of the aircraft, seat/occupant, occupant alone, and seat alone by calculating the forces and moments imposed on these elements by the seat catapults, rails, rockets, stabilization, and recovery systems included in most escape system configurations. User options																	

DD FORM 1473

JAN 72

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

9. Continued

Louis A. D'Aulerio
Aircraft and Crew Systems Technology Directorate (Code 6032)
Naval Air Development Center
Warminster, PA 18974

David A. Fender
Ketron, Inc.
444 Jacksonville Road
Warminster, PA 18974

20. Continued

are provided to simulate the performance of all conventional escape system designs under most environmental conditions and aircraft attitudes or trajectories. This GESS Programmer's Manual summarizes program conventions, lists the annotated FORTRAN IV program code, and represents the second of two volumes of GESS documentation. Volume I, the GESS User's Guide, describes the elements occurring in typical escape systems, the theory and formulation of the simulation model, and the procedures necessary to successfully prepare, execute and use this and the related ACT and DRAS programs.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

FOREWORD

This final report represents the second of two volumes of documentation for the Generalized Escape System Simulation (GESS) computer program. This GESS Programmer's Manual summarizes program conventions, lists the annotated FORTRAN IV program code and describes the procedures necessary to successfully use the GESS computer program. Volume I, the GESS User's Guide, describes the elements present and events occurring in typical escape systems, the theory and formulation of the simulation model, and the procedures for preparing and executing the GESS program and the related ACT and DRAS programs. A portion of this work was performed by Ketron, Inc., in accordance with NADC Contract N62269-81-C-0206, Task No. 630-1944.

This work was sponsored by the Naval Air Systems Command (AIR-5312) under Airtask No. A5315312-001D-1531W1100, Work Unit No. WC530.

DTIC
ELECTE
DEC 6 1984
B



Accession For		✓
NTIS		
DTIC		
Unannounced		
Justification		
By		
Distribution		
Availability Codes		
Dist	Avail and/or	Special
A-1		

ABSTRACT

The Generalized Escape System Simulation (GESS) program is a computerized mathematical model for dynamically simulating the performance of existing or developmental aircraft ejection seat systems. The program generates six-dimensional trajectory predictions of the aircraft, seat/occupant, occupant alone, and seat alone by calculating the forces and moments imposed on these elements by the seat catapults, rails, rockets, stabilization, and recovery systems included in most escape system configurations. User options are provided to simulate the performance of all conventional escape system designs under most environmental conditions and aircraft attitudes or trajectories. This *GESS Programmer's Manual*, summarizes program conventions, lists the annotated FORTRAN IV program code, and represents the second of two volumes of GESS documentation. Volume I, the *GESS User's Guide*, describes the elements present and events occurring in typical escape systems, the theory and formulation of the simulation model, and the procedures necessary to successfully prepare, execute, and utilize this and the related ACT and DRAS programs.

TABLE OF CONTENTS

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
	FOREWORD.....	i
	ABSTRACT.....	ii
	TABLE OF CONTENTS.....	iii
	LIST OF FIGURES.....	vi
	LIST OF TABLES.....	vi
1.0	INTRODUCTION.....	1-1
2.0	PROGRAM CONVENTIONS.....	2-1
2.1	Coordinate Systems.....	2-1
2.2	Variable Identification.....	2-4
2.3	Integration Arrays.....	2-5
2.3.1	Aircraft Trajectory Array.....	2-7
2.3.2	Occupant Alone Trajectory Array.....	2-8
2.3.3	Seat Alone Trajectory Array.....	2-9
2.3.4	Seat/Occupant Trajectory Array.....	2-10
2.3.5	Parachute Trajectory Array.....	2-11
2.3.6	Thrust Vector Control Array.....	2-12
2.3.7	Quaternion Arrays.....	2-13
3.0	AERODYNAMIC COEFFICIENT TABLES.....	3-1
4.0	EXECUTION JOB STREAM.....	4-1
5.0	GENERALIZED ESCAPE SYSTEM SIMULATION (GESS) PROGRAM ANNOTATED CODE LISTINGS.....	5-1
	Program GESS.....	5-2
	Subroutine ACSEP.....	5-11
	Subroutine AERFMOA.....	5-13
	Subroutine AERFMSA.....	5-17
	Subroutine AERFMSO.....	5-23
	Subroutine AEROIN.....	5-29
	Subroutine AIRCRAFT.....	5-31
	Subroutine ATMOS.....	5-34
	Subroutine CATAFM.....	5-36
	Subroutine CHUINIT.....	5-40
	Subroutine CHUTES.....	5-43
	Subroutine CHUTFM.....	5-44
	Subroutine CLEAR.....	5-49

TABLE OF CONTENTS - (Cont'd)

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
	Subroutine DARTFM.....	5-60
	Subroutine DRICALC.....	5-64
	Subroutine DROGUE1.....	5-67
	Subroutine DROGUE2.....	5-71
	Subroutine DYNAMCG.....	5-76
	Subroutine EVENT.....	5-81
	Subroutine HEADER.....	5-85
	Subroutine IDIRMTX.....	5-88
	Subroutine INIQUA.....	5-89
	Subroutine INITMS.....	5-91
	Subroutine INITRAJ.....	5-94
	Subroutine INIVECT.....	5-98
	Subroutine INIVRBL.....	5-104
	Subroutine INPUT.....	5-113
	Subroutine INTEG.....	5-134
	Subroutine INTEGII.....	5-136
	Subroutine INTEGIN.....	5-137
	Subroutine INTEGMD.....	5-140
	Subroutine INTEGSI.....	5-141
	Subroutine INTLZ.....	5-142
	Subroutine INTRP.....	5-143
	Subroutine LADDATE.....	5-146
	Subroutine MATRIX.....	5-147
	Subroutine MATUPD.....	5-149
	Subroutine OCCALON.....	5-150
	Subroutine PCHUTFT.....	5-154
	Subroutine PLOTBIN.....	5-159
	Subroutine PLOTWAC.....	5-165
	Subroutine QUAT.....	5-171
	Subroutine RAILFM.....	5-173
	Subroutine RECOV.....	5-178
	Subroutine REINTEG.....	5-183
	Subroutine REINZRO.....	5-184
	Subroutine REPORTS.....	5-185
	Subroutine REPRT1.....	5-188
	Subroutine REPRT2.....	5-207
	Subroutine REPRT3.....	5-209
	Subroutine REPRT4.....	5-211
	Subroutine REPRT5.....	5-213
	Subroutine REPRT6.....	5-215
	Subroutine REPRT7.....	5-217
	Subroutine REPRT8.....	5-219
	Subroutine REPRT9.....	5-222
	Subroutine REPRT10.....	5-224
	Subroutine REPRT11.....	5-226
	Subroutine REPRT12.....	5-229
	Subroutine REPRT13.....	5-232
	Subroutine REPRT19.....	5-234
	Subroutine REPRT20.....	5-236

TABLE OF CONTENTS - (Cont'd)

<u>SECTION</u>	<u>TITLE</u>	<u>PAGE</u>
	Subroutine REPRT21.....	5-238
	Subroutine REPRT22.....	5-241
	Subroutine REPRT23.....	5-243
	Subroutine REPRT24.....	5-245
	Subroutine REPRT25.....	5-248
	Subroutine REPRT26.....	5-251
	Subroutine REPRT27.....	5-254
	Subroutine REPRT28.....	5-255
	Subroutine REPRT29.....	5-257
	Subroutine REPRT30.....	5-259
	Subroutine REPRT31.....	5-261
	Subroutine REPTDRI.....	5-263
	Subroutine RESTART.....	5-264
	Subroutine RKTFM.....	5-265
	Subroutine ROTATE.....	5-269
	Subroutine RUNGE.....	5-271
	Subroutine SEATAKB.....	5-273
	Subroutine SEATOCC.....	5-277
	Subroutine SEPINIT.....	5-284
	Subroutine SLUGCON.....	5-288
	Subroutine THRUST.....	5-293
	Subroutine TMUPDAT.....	5-295
	Subroutine TUBEND.....	5-298
	Subroutine UPDVECT.....	5-300
	Subroutine VERTSK.....	5-303
	Function ZARCTAN.....	5-308
	Subroutine ZLININT.....	5-309
6.0	REFERENCES.....	6-1

APPENDICES

A	AERODYNAMIC COEFFICIENT TABLES (ACT) PROGRAM ANNOTATED CODE LISTINGS

LIST OF FIGURES

<u>FIGURE</u>	<u>TITLE</u>	<u>PAGE</u>
2-1	Coordinate Systems.....	2-2
4-1	GESS Execution Job Stream for CDC KRONOS/OS.....	4-3

LIST OF TABLES

<u>TABLE</u>	<u>TITLE</u>	<u>PAGE</u>
2-1	Integration Array Values.....	2-6
2-2	Aircraft Trajectory Array Description.....	2-7
2-3	Occupant Alone Trajectory Array Description.	2-8
2-4	Seat Alone Trajectory Array Description.....	2-9
2-5	Seat/Occupant Trajectory Array Description..	2-10
2-6	Parachute Trajectory Array Description.....	2-11
2-7	Thrust Vector Control Array Description.....	2-12
2-8	Quaternion Array Descriptions.....	2-13

1.0 INTRODUCTION

Designing ejection seat systems enabling the safe escape of crewmembers from high-speed military aircraft is a formidable task. As an adjunct to the Navy's on-going escape systems program of system engineering, testing, and incident review, a mathematical model has been formulated and developed to simulate the operation of any ejection seat-based escape system under most realistic conditions from any aircraft. This Programmer's Guide lists the annotated code of the Generalized Escape System Simulation (GESS) program, and is Volume II of two volumes of GESS documentation. A complete description of the history, theory, preparation, and use of GESS is provided in Volume I, the *GESS User's Guide*.

NADC-84068-60

(THIS PAGE INTENTIONALLY LEFT BLANK)

2.0 CONVENTIONS

2.1 Coordinate Systems

Trajectories generated by the Generalized Escape System Simulation (GESS) program are calculated with respect to specific, "right hand rule," coordinate systems coinciding with center of gravities or other specified reference points of the various system elements. Each of the program coordinate systems is defined by six (6) degrees of freedom:

A. Linear:

1. X - forward positive displacement
2. Y - left positive displacement
3. Z - upward positive displacement

B. Angular:

4. Yaw (R) - leftward positive rotation
5. Pitch (Q) - nose downward positive rotation
6. Roll (P) - right wing downward positive rotation

These coordinate systems, illustrated in Figure 2-1, are defined as follows:

• Earth-Fixed Coordinate System (EFCS)

The EFCS is a 3-axis orthogonal coordinate system with origin at a fixed point located on or near the surface of the earth. Since all simulated distances are relatively small compared to the earth's radius, the errors associated with neglecting the curvature of the earth's surface are considered negligible. The movements and rotations of all simulated system elements can be described with respect to (wrt) this fixed coordinate system.

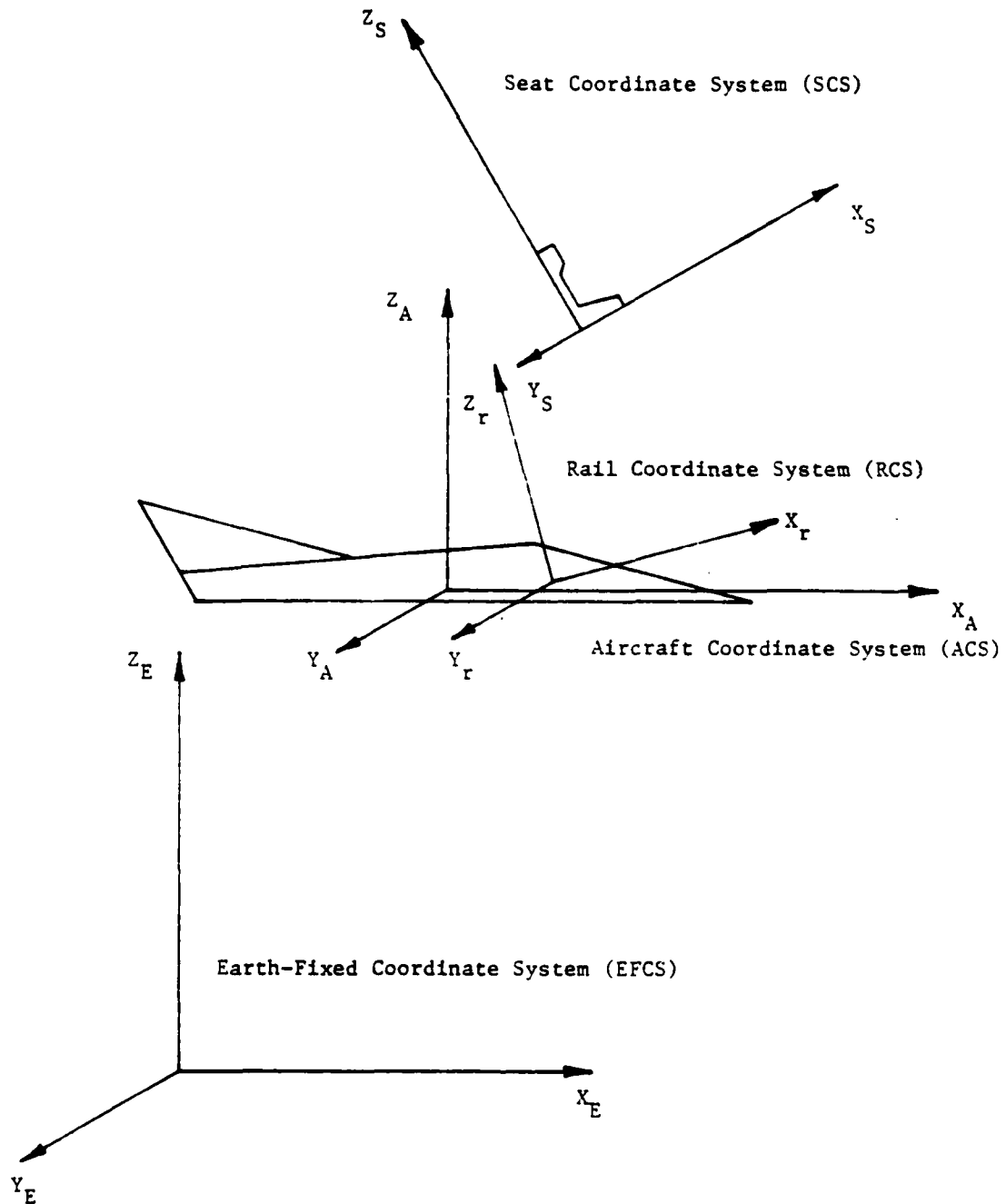


Figure 2-1. Coordinate Systems

- Aircraft Coordinate System (ACS)

The ACS is a 3-axis orthogonal coordinate system with origin at a fixed point, usually the aircraft's center of gravity. All linear and angular movements of the aircraft are determined by establishing the position of this coordinate system wrt the EFCS. The position of all other system elements, prior to and, optionally, after the separation of the seat from the aircraft, is determined wrt to the ACS.

- Rail Coordinate System (RCS)

The RCS is a 3-axis orthogonal coordinate system which is defined and fixed wrt the ACS. Its origin coincides with the mid-point between the rails of the lateral (Y) axis defined by the two lowest rail points in the ACS, and its vertical (Z) axis is parallel to the rails. The rail system serves to restrict the motion of the seat to a specific path as it moves out of the aircraft cockpit.

- Seat Coordinate System (SCS)

The SCS is a 3-axis orthogonal coordinate system with origin usually corresponding to the initial location of the RCS origin wrt the seat. The center of gravity (CG) of either the seat/occupant (S/O) combination or the seat/alone (S/A) is referenced wrt the SCS, allowing the trajectories of these system elements to be tracked wrt either the ACS or the EFCS. Points of application of the various forces acting on the S/O or S/A are also referenced wrt the SCS, allowing the calculation of moments and rotation around the respective CGs.

- Occupant Alone Coordinate System (OACS)

The OACS is a 3-axis orthogonal coordinate system with origin at the center of gravity of the seat occupant after the occupant separates from the seat. The OACS is currently defined only by the three linear degrees of freedom because the disjointed non-rigidity of the occupant alone makes any angular determination difficult, if not irrelevant. Provisions have been made for the future incorporation of OACS angular degrees of freedom if such information is necessary at a later date.

- Thrust Vector Control Coordinate System (TVCCS)

The TVCCS is a 3-axis orthogonal coordinate system with origin at the common intersection of the rocket line of thrust and the centerlines of its gimbals. This coordinate system establishes the orientation of the rocket line of thrust, which is modified dynamically as part of the vertical-seeking maneuver.

Coordinate systems and vector directions should be carefully considered when preparing simulation inputs. All output reports are referenced accordingly.

2.2 Variable Identification

Several variable naming conventions are used to identify variables used in the program. Refer to the program annotation for specific variable descriptions. These naming conventions are summarized below.

SA	- seat alone
OA	- occupant alone
SO	- seat/occupant
NPTS	- number of points
RK	- rocket

WGHT	- weight
X	- X axis
Y	- Y axis
Z	- Z axis
P	- roll rotation
Q	- pitch rotation
R	- yaw rotation
VEL	- velocity
PORO	- porosity
REC	- recovery parachute
DRO/DR	- drogue parachute
DRT	- DART
POS	- position
IGN	- ignition
CAT	- catapult
TVC	- thrust vector control
CG	- center of gravity

The variable names used in the program have been selected to facilitate the programmer's understanding. There is limited correlation between these variable names and the variable names used in the program formulation.

2.3 Integration Arrays

The large number of equations and saved values associated with many of the trajectory integrations in GESS resulted in the development of a highly structured integration array logic. This logic maintains the current value of each integration equation, in addition to all intermediate or previous equation values for both the Runge-Kutta and predictor-corrector methods.

Thus, the initial four time steps can be integrated by the Runge-Kutta routine, and subsequent integrations performed by the predictor-corrector routine.

The first value of each array indicates the number of equations involved in the integration. The remaining array values are then ordered, in accordance with the number of equations, as described in Table 2-1.

Table 2-1. Integration Array Values

<u>Array Location</u>	<u>Description</u>
1	number of equations (N)
2 -- (1+N)	results of integrations - equation values
(2+N) -- (1+2N)	equation derivatives
(2+2N) -- (1+3N)	current Runge-Kutta K-values
(2+3N) -- (1+4N)	intermediate Runge-Kutta K-value summations
(2+4N) -- (1+5N)	intermediate equation values
(2+5N) -- (1+6N)	equation values at (time- Δt)
(2+6N) -- (1+7N)	equation values at (time- $2\Delta t$)
(2+7N) -- (1+8N)	equation values at (time- $3\Delta t$)
(2+8N) -- (1+9N)	equation values at (time- $4\Delta t$)
(2+9N) -- (1+10N)	equation derivatives at (time- Δt)
(2+10N) -- (1+11N)	equation derivatives at (time- $2\Delta t$)
(2+11N) -- (1+12N)	equation derivatives at (time- $3\Delta t$)
(2+12N) -- (1+13N)	predictor values at current time
(2+13N) -- (1+14N)	predictor values at (time+ Δt)
(2+14N) -- (1+15N)	corrector values at current time
(2+15N) -- (1+16N)	corrector values at (time+ Δt)

2.3.1 Aircraft Trajectory Array. The aircraft trajectory array, TRAJAC, is based upon the 12 primary trajectory variables described in Table 2-2.

Table 2-2. Aircraft Trajectory Array Description

<u>Array Location</u>	<u>Description</u>
TRAJAC(1)	number of aircraft trajectory equations = 12 if tracking aircraft = 0 otherwise
TRAJAC(2)	x position of aircraft in EFCS
TRAJAC(3)	y position of aircraft in EFCS
TRAJAC(4)	z position of aircraft in EFCS
TRAJAC(5)	x velocity of aircraft in ACS
TRAJAC(6)	y velocity of aircraft in ACS
TRAJAC(7)	z velocity of aircraft in ACS
TRAJAC(8)	p (roll) angular position of aircraft †
TRAJAC(9)	q (pitch) angular position of aircraft †
TRAJAC(10)	r (yaw) angular position of aircraft †
TRAJAC(11)	p (roll) angular velocity of aircraft in ACS
TRAJAC(12)	q (pitch) angular velocity of aircraft in ACS
TRAJAC(13)	r (yaw) angular velocity of aircraft in ACS
TRAJAC(14) through TRAJAC(193)	derivatives and other equation values associated with the above equations as detailed in Table 2-1 and with N=12.

† - integration result; not used for describing angular position.

2.3.2 Occupant Alone Trajectory Array. The occupant alone trajectory array, TRAJOA, is based upon the 12 primary trajectory variables described in Table 2-3.

Table 2-3. Occupant Alone Trajectory Array Description

<u>Array Location</u>	<u>Description</u>
TRAJOA(1)	number of occupant alone trajectory equations = 0 before S/O separation = 12 after S/O separation
TRAJOA(2)	x position of occupant alone in EFCS
TRAJOA(3)	y position of occupant alone in EFCS
TRAJOA(4)	z position of occupant alone in EFCS
TRAJOA(5)	x velocity of occupant alone in ACS
TRAJOA(6)	y velocity of occupant alone in ACS
TRAJOA(7)	z velocity of occupant alone in ACS
TRAJOA(8)*	p (roll) angular position of occupant alone
TRAJOA(9)*	q (pitch) angular position of occupant alone
TRAJOA(10)*	r (yaw) angular position of occupant alone
TRAJOA(11)*	p (roll) velocity of occupant alone in OACS
TRAJOA(12)*	q (pitch) velocity of occupant alone in OACS
TRAJOA(13)*	r (yaw) velocity of occupant alone in OACS
TRAJOA(14) through TRAJOA(193)	derivatives and other equation values associated with the above equations as detailed in Table 2-1 and with N=12.

* - All angular positions and velocities and their derivatives are currently permanently set = 0; array space has been maintained to facilitate future incorporation of O/A angular tracking.

2.3.3 Seat Alone Trajectory Array. The seat alone trajectory array, TRAJSA, is based upon the 12 primary trajectory variables described in Table 2-4.

Table 2-4. Seat Alone Trajectory Array Description

<u>Array Location</u>	<u>Description</u>
TRAJSA(1)	number of seat alone trajectory equations = 0 before S/O separation = 12 after S/O separation
TRAJSA(2)	x position of seat alone in EFCS
TRAJSA(3)	y position of seat alone in EFCS
TRAJSA(4)	z position of seat alone in EFCS
TRAJSA(5)	x velocity of seat alone in SCS
TRAJSA(6)	y velocity of seat alone in SCS
TRAJSA(7)	z velocity of seat alone in SCS
TRAJSA(8)	p (roll) angular position of seat alone †
TRAJSA(9)	q (pitch) angular position of seat alone †
TRAJSA(10)	r (yaw) angular position of seat alone †
TRAJSA(11)	p (roll) angular velocity of seat alone in SCS
TRAJSA(12)	q (pitch) angular velocity of seat alone in SCS
TRAJSA(13)	r (yaw) angular velocity of seat alone in SCS
TRAJSA(14) through TRAJSA(193)	derivatives and other equation values associated with the above equations as detailed in Table 2-1 and with N=12.

† - integration result, not used for describing angular position.

2.3.4 Seat/Occupant Trajectory Array. The seat/occupant trajectory array, TRAJSO, is based upon the 12 primary trajectory variables described in Table 2-5.

Table 2-5. Seat/Occupant Trajectory Array Description

<u>Array Location</u>	<u>Description</u>
TRAJSO(1)	number of seat/occupant trajectory equations = 12 before S/O separation = 0 after S/O separation
TRAJSO(2)	x position of seat/occupant in EFCS
TRAJSO(3)	y position of seat/occupant in EFCS
TRAJSO(4)	z position of seat/occupant in EFCS
TRAJSO(5)	x velocity of seat/occupant in SCS
TRAJSO(6)	y velocity of seat/occupant in SCS
TRAJSO(7)	z velocity of seat/occupant in SCS
TRAJSO(8)	p (roll) angular position of seat/occupant †
TRAJSO(9)	q (pitch) angular position of seat/occupant †
TRAJSO(10)	r (yaw) angular position of seat/occupant †
TRAJSO(11)	p (roll) angular velocity of seat/occupant in SCS
TRAJSO(12)	q (pitch) angular velocity of seat/occupant in SCS
TRAJSO(13)	r (yaw) angular velocity of seat/occupant in SCS
TRAJSO(14) through TRAJSO(193)	derivatives and other equation values associated with the above equations as detailed in Table 2-1 and with N=12.

† = integration result, not used for describing angular position.

2.3.5 Parachute Trajectory Array. The parachute array, TRAJCH, is based upon 6 primary trajectory variables for each of the 3 parachutes incorporated into the GESS model, as described in Table 2-6.

Table 2-6. Parachute Trajectory Array Description

<u>Array Location</u>	<u>Description</u>
TRAJCH(1,i*)	number of parachute i* trajectory equations = 6 when applicable = 0 otherwise
TRAJCH(2,i*)	x position of parachute i* in EFCS
TRAJCH(3,i*)	y position of parachute i* in EFCS
TRAJCH(4,i*)	z position of parachute i* in EFCS
TRAJCH(5,i*)	x velocity of parachute i* in EFCS
TRAJCH(6,i*)	y velocity of parachute i* in EFCS
TRAJCH(7,i*)	z velocity of parachute i* in EFCS
TRAJCH(8,i*) through TRAJCH(97,i*)	derivatives and other equation values associated with the above equations as detailed in Table 2-1 and with N = 6.

* - i = 1 for first drogue parachute or slug/container

= 2 for second drogue parachute

= 3 for recovery parachute

2.3.6 Thrust Vector Control Array. The thrust vector control array, TVCEQS, is based upon the 14 primary control variables described in Table 2-7.

Table 2-7. Thrust Vector Control Array Description

<u>Array Location</u>	<u>Description</u>
TVCEQS(1)	number of thrust vector control (TVC) equations = 14 during TVC = 0 otherwise
TVCEQS(2)	x_1^*
TVCEQS(3)	x_2^*
TVCEQS(4)	x_3^*
TVCEQS(5)	x_4^*
TVCEQS(6)	x_5^*
TVCEQS(7)	x_6^*
TVCEQS(8)	x_7^*
TVCEQS(9)	x_8^*
TVCEQS(10)	x_9^*
TVCEQS(11)	x_{10}^*
TVCEQS(12)	x_{11}^*
TVCEQS(13)	x_{12}^*
TVCEQS(14)	x_{13}^*
TVCEQS(15)	x_{14}^*
TVCEQS(16) through TVCEQS(255)	derivatives and other equation values associated with the above equations as detailed in Table 2-1 and with $N = 14$.

* - for details see User's Manual, Section 3.5.3, and Subroutine VERTSK,
page 5-303. Note that $x_1 = \text{TVCVALS}(i)$ and $\dot{x}_1 = \text{TVCDERV}(i)$.

2.3.7 Quaternion Arrays. The 4 quaternion arrays, QUATAC, QUATOA, QUATSA, and QUATSO, are based upon the 4 primary quaternion array variables described in Table 2-8.

Table 2-8. Quaternion Array Descriptions

<u>Array Location</u>	<u>Description</u>
QUATxx*(1)	number of xx* quaternion equations = 4 if xx* trajectory is being tracked = 0 otherwise
QUATxx*(2)	λ_{0xx} (See User's Manual, Eq. 3.a)
QUATxx*(3)	λ_{1xx} (See User's Manual, Eq. 3.b)
QUATxx*(4)	λ_{2xx} (See User's Manual, Eq. 3.c)
QUATxx*(5)	λ_{3xx} (See User's Manual, Eq. 3.d)
QUATxx*(6) through QUATxx*(65)	derivatives and other equations associated with the above equations as detailed in Table 2-1 and with N = 4.

* - xx = AC for aircraft trajectory
 = OA for occupant alone trajectory
 = SA for seat alone trajectory
 = SO for seat/occupant trajectory

NADC-84068-60

(THIS PAGE INTENTIONALLY LEFT BLANK)

3.0 AERODYNAMIC COEFFICIENT TABLES

During the execution of the GESS program, equally spaced aerodynamic coefficients are required for computing the aerodynamic forces acting upon the seat/occupant system. These coefficients are usually derived through wind tunnel experiments using the subject ejection seat^{(1)*}. The forces and moments data produced by these tests are used as inputs for the Aerodynamic Coefficient Table (ACT) program⁽²⁾, which creates the necessary coefficient tables as functions of the seat orientation and velocity. The ACT program provides an efficient method for creating, modifying, and storing these aerodynamic coefficient tables on random access files for retrieval by GESS during simulation execution. The ACT program was closely modeled after the RFWTHR program⁽³⁾.

The annotated code listings of the ACT program are provided in Appendix A. These listings have been extracted from Reference 2 because of the importance of having proper aerodynamic coefficient tables for the successful execution of GESS. Detailed instructions for using the ACT program are given in the Volume I *GESS User's Guide*.

* All references may be found in Section 6.0.

NADC-84068-60

(THIS PAGE INTENTIONALLY LEFT BLANK)

4.0 EXECUTION JOB STREAM

The GESS program is executed by submitting to the computer a series of operating system commands known as the "job stream". The job stream can be entered line-by-line interactively. However, it is usually more convenient to prepare a job stream, or "submit file", in advance of execution and "batch" submit this file to the computer for execution. The interactive commands to batch submit a job stream are:

GET,filename

SUBMIT,filename

Another print site can be specified if the main computer center printer is not desired.

SUBMIT,filename,EI=print site identifier

If no printed output is desired:

SUBMIT,filename,N

Figure 4-1 presents a recommended job stream for GESS execution using the CDC KRONOS operating system. This job stream provides for:

- . the loading of the prepared input file (GESSI)
- . the loading of the ACT program-generated aerodynamic coefficient tables (AERO4)
- . the loading and execution of the pre-compiled program binary code (GESSB)
- . the optional creation or replacement of program-generated plotting data files (GESST and GESSZ)

- . the creation of a single, inclusive, indirect access output file containing a mirror copy of the original input file and all requested program-generated reports (GESSO)
- . A "day file" record of the submitted job execution (GESSDAY)
- . An error recovery logic that saves the day file and any prior output, in the eventuality of an execution error

The recommended job stream is intended for execution runs generating a nominal number of output reports and/or simulating a relatively few seconds of simulated trajectory. Should a large number of reports and/or a long simulation period be desired, the output (and, possibly, the plotting files) should be created either as a direct access file, or as several indirect files. Refer to Reference 4 for details on direct file creation on the KRONOS system.

```

/JOB
GESS(CB200000,T75)
ACCOUNT(XXXXXX,YYYYYY)
ASSIGN,MS,OUTPUT
GET(TAPE1=KGESI)
GET(TAPE2=AERO4)
GET(GESSB)
MAP(OFF)
GESSB.
REPLACE(TAPE42=KGEST)
REPLACE(TAPE41=KGESZ)
GOTO,1.
EXIT.
1,REWIND,*.
SKIPF(DAYFILE)
COPYSBF(TAPE1,OUTPUT)
COPYBF(TAPE5,OUTPUT)
COPYBF(TAPE6,OUTPUT)
COPYBF(TAPE7,OUTPUT)
.
.
.
COPYBF(TAPE40,OUTPUT)
REWIND,OUTPUT.
REPLACE(OUTPUT=KGESO)
GOTO,2.
EXIT.
2,DAYFILE,GESSDAY.
REPLACE,GESSDAY.
EXIT.
/EOR
/EOI

```

Figure 4-1. GESS Execution Job Stream for CDC KRONOS/OS

NADC-84068-60

(THIS PAGE INTENTIONALLY LEFT BLANK)

NADC-84068-60

5.0 GENERALIZED ESCAPE SYSTEM SIMULATION

(GESS) PROGRAM

ANNOTATED CODE LISTINGS

5-2

5-3

```

115 C.....
C SECTION 13 COMMON BLOCK
C.....
COMMON /DYNCGIN / IDYNGC , WY , WYV
+ , XSLACK , SXP , SXN
+ , CV , SY , CZ , ZSLACK
+ , SZP , ZBOT , SZN1 , SZN2
C.....
C EVENT MESSAGES COMMON BLOCK
C.....
COMMON /EVMS / IEVMS(3,38) , ISPMES(4,6) , ISPECL(6)
C.....
C OCCUPANT ALONE FORCES COMMON BLOCK
C.....
COMMON /FORCEA / FXCHDA(3) , FYCHDA(3) , FZCHDA(3) ,
+ FXAEDA , FYAEDA , FZAEDA
C.....
C SEAT ALONE FORCES COMMON BLOCK
C.....
COMMON /FORCEA / FXAESA , FYAESA , FZAESA
C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
COMMON /FORCEO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+ FXTUBSO , FYTUBSO , FZTUBSO
+ FXLSO(6) , FYLSO(6) , FZLSO(6)
+ FXRKS(6) , FYRKS(6) , FZRKS(6)
+ FXCHSO(3) , FYCHSO(3) , FZCHSO(3)
+ FXAESD , FYAESD , FZAESD
+ FXORTSO , FYORTSO , FZORTSO
C.....
145 C SECTION 4 COMMON BLOCK
C.....
COMMON /TAIRCRT / TEMP , PRESSUR , ZACVEL , XPOS , YPOS
+ ZPOS , XTAIL , YTAIL , ZTAIL , YAW
+ PITCH , ROLL , RVEL , QVEL , PVEL
+ WINDX , WINDY , WINDZ , XACVEL , CKPITH
+ DENSITY , NPTSAAT , AAT(4,50) , NPTSLAT , LAT(4,50) ,
+ IACSFLG
C.....
155 C SECTION 9 COMMON BLOCK
C.....
COMMON /ICATPLT / INCAT , CATLNT(2) , CATSTK(2),TCI (2) ,
+ XPOSAP(2) , YPOSAP(2) , ZPOSAP(2),NPTSCI(2) ,
+ CATHRST(2,25,2) , ITUBEND , KTUBE , CTUBE
+ PTUBE , MUTUBE , EXTNGT , ICATOU
+ KTUBE , MUTUBE
C.....
160 C SECTION 1 COMMON BLOCK
C.....
COMMON /ICONTRI / TSTART , TSIOP , ESTOP , IRESTRT , IUNITS
+ ISEATIR , ISOSEP , IPLOT , IDRIFLG
+ IPHASE1 , IPHASE2 , IPHASE3
+ ESTOP
C.....
170 C SECTION 11 COMMON BLOCK
C.....

```

```

COMMON /IDARTIN / IDART , DRTFRCE , DRTSTRT , DRTSTOP
+ XDRTAP(2), XDRTAP(2), XDRTAP(2),
+ XDRTCP(2), XDRTCP(2), XDRTCP(2)
C.....
C SECTION 3 COMMON BLOCK
C.....
COMMON /IDELTAT / DTPHAS1, DTPHAS2, DTPHAS3
C.....
C INFO04 DATA (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C.....
COMMON /INFO04 / NCXS(12) , NCYS(12) , NCZS(12)
+ DLTC(3,12) , ENDPIC(6,12) , IAECSQ(12)
C.....
C SECTION 7 COMMON BLOCK
C.....
COMMON /IRAIL / RAILNTH , RAILANG , ISTRL , NSLBKS
+ KXSB , KYSB , MUSB , YKTOR
+ XPOSRR , YPOSRR , ZPOSRR
+ XPOSRL , YPOSRL , ZPOSRL
+ XPOSSB(6), YPOSSB(6), ZPOSSB(6)
REAL
+ KXSB , KYSB , MUSB
C.....
C SECTION 2 COMMON BLOCK
C.....
COMMON /IREPORT / IREPTS(31) , PRTRFQ,P11,P12,P13
INTEGER
+ PRTRFQ,P11,P12,P13
C.....
C RECALCULATED ROCKET THRUST TABLE COMMON BLOCK
C.....
COMMON /IRKTOUT / RKTOUT(2,25,6)
C.....
C SECTION 10 COMMON BLOCK
C.....
COMMON /IROCKET / INRKT , RKDELY(6), RKNTS(6), IROKOUT ,
+ RKIGN(6) , RKWGT(6), RKBURN(6), TSTAR(6) ,
+ XPOSRK(6), YPOSRK(6), ZPOSRK(6),
+ RKALPH(6), RKBETA(6), RKGAMA(6), RKTHRST(2,25,6)
INTEGER
+ RKNTS
C.....
C SECTION 6 COMMON BLOCK
C.....
COMMON /ISEATOC / IPCNTL , XCGSO , YCGSO , ZCGSO , IXXSO ,
+ IYXSO , IZXSO , IYVSO , IYZSO , IZZSO ,
+ AREASO , AREADA , WGHIOAB , WGHIOAA ,
+ IXXOA , IYXOA , IZXOA , IYVOA , IYZOA ,
+ C150 , C250 , C350 , C450 ,
+ C10A , C20A , C30A , C40A
REAL
+ IXXSO , IYXSO , IZXSO , IYVSO , IYZSO ,
+ IZZSO , IXXOA , IYXOA , IZXOA , IYVOA ,
+ IYZOA , IZZOA
C.....
C SECTION 5 COMMON BLOCK
C.....
COMMON /ISETALN / XPOSSRP , YPOSSRP , ZPOSSRP , XCGSA , YCGSA ,
+ ZCGSA , IXXSA , IYXSA , IZXSA , IYVSA ,
+ IYZSA , IZZSA , PHISA , PSISA , THESA

```

5-6

```

290 +XSSDAP(2),YSSDAP(2),ZSSDAP(2),XESDAP ,YESDAP ,ZESDAP ,
+XSRCSAC ,YSRCSAC ,ZSRCSAC ,XSSDAP ,YESDAP ,ZSSDAP ,
+XRSOSB ,YRSOSB ,ZRSOSB ,XRRSBOI ,YRRSBOI ,ZRRSBOI ,
+XRRSB ,YRRSB ,ZRRSB ,XSSOCHI(3),YSSOCHI(3),ZSSOCHI(3),
+XAACSO ,YAACSO ,ZAACSO ,XASDAP ,YASDAP ,ZASDAP ,
+XRSOAC ,YRSOAC ,ZRSOAC ,XSCPAP(2),YSCPAP(2),ZSCPAP(2)
C .....
C SECTION 14 COMMON BLOCK
C .....
295 COMMON /PARCHUT / IRECOV ,TROPLOY ,RECOVLL ,RECOVLT ,
+ RECOVDP ,RECOVDR ,RECOVDR ,RECOVDR ,RECOVDR ,
+ YRECAB ,YRECAB ,YRECAB ,YRECAB ,YRECAB ,
+ NPTSRLS ,NPTSRLS ,NPTSRLS ,NPTSRLS ,NPTSRLS ,
+ NPTSRTF ,NPTSRTF ,NPTSRTF ,NPTSRTF ,NPTSRTF ,
+ IDROGUE ,IDROGUE ,IDROGUE ,IDROGUE ,IDROGUE ,
+ POROSD2 ,POROSD2 ,POROSD2 ,POROSD2 ,POROSD2 ,
+ NPTDFT2 ,NPTDFT2 ,NPTDFT2 ,NPTDFT2 ,NPTDFT2 ,
+ NPTDFT1 ,NPTDFT1 ,NPTDFT1 ,NPTDFT1 ,NPTDFT1 ,
+ NPTSOLS ,NPTSOLS ,NPTSOLS ,NPTSOLS ,NPTSOLS ,
+ DISPLAY ,DISPLAY ,DISPLAY ,DISPLAY ,DISPLAY ,
+ DROGPD1 ,DROGPD1 ,DROGPD1 ,DROGPD1 ,DROGPD1 ,
+ DROVELY ,DROVELY ,DROVELY ,DROVELY ,DROVELY ,
+ YDROGAP ,YDROGAP ,YDROGAP ,YDROGAP ,YDROGAP ,
+ CHALT2 ,CHALT2 ,CHALT2 ,CHALT2 ,CHALT2 ,
+ AREADC ,AREADC ,AREADC ,AREADC ,AREADC ,
+ TFP3 ,TFP3 ,TFP3 ,TFP3 ,TFP3 ,
+ CDC ,CDC ,CDC ,CDC ,CDC ,
C .....
C RAIL VARIABLES COMMON BLOCK
C .....
315 COMMON /RAILVRB / FXR ,FYR ,FZR ,XDISP ,YDISP
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
320 COMMON /RKUTTA / TIME ,TIMES , DELTAT , TRAJSD(193) ,
+ TRAJSA(193) , TRAJSA(193) , TRAJSA(193) , TRAJSD(193) ,
+ TRAJAC(193) , TRAJAC(193) , TRAJAC(193) , TRAJAC(193) ,
+ QUATSA(65) , QUATSA(65) , QUATSA(65) , QUATSA(65) ,
+ INTSTP ,INTSTP ,INTSTP ,INTSTP ,INTSTP ,
+ IPOINTS ,IPOINTS ,IPOINTS ,IPOINTS ,IPOINTS ,
+ IKX ,IKX ,IKX ,IKX ,IKX ,
+ IYIX ,IYIX ,IYIX ,IYIX ,IYIX ,
+ IY13X ,IY13X ,IY13X ,IY13X ,IY13X ,
+ IYPR12X ,IYPR12X ,IYPR12X ,IYPR12X ,IYPR12X ,
+ IY11X ,IY11X ,IY11X ,IY11X ,IY11X ,
+ IY11X ,IY11X ,IY11X ,IY11X ,IY11X ,
C .....
325 IYPRX ,IYPRX ,IYPRX ,IYPRX ,IYPRX ,
+ IKPASSX ,IKPASSX ,IKPASSX ,IKPASSX ,IKPASSX ,
+ IY12X ,IY12X ,IY12X ,IY12X ,IY12X ,
+ IYPR11X ,IYPR11X ,IYPR11X ,IYPR11X ,IYPR11X ,
+ IY11X ,IY11X ,IY11X ,IY11X ,IY11X ,
+ IY11X ,IY11X ,IY11X ,IY11X ,IY11X ,
C .....
330 IY11X ,IY11X ,IY11X ,IY11X ,IY11X ,
+ IY11X ,IY11X ,IY11X ,IY11X ,IY11X ,
C .....
C TOF PLOTTING FILE COMMON BLOCK
C .....
335 COMMON /TTLES / SENSAM(40,6) , TEXT1(6) , TEXT2(6) ,
+ TEXT3(3) , TEXT3(3) , TEXT3(3) , TEXT3(3) ,
+ NCHANFR(2) ,NCHANFR(2) ,NCHANFR(2) ,NCHANFR(2) ,
+ RECORD(35) ,RECORD(35) ,RECORD(35) ,RECORD(35) ,
+ TPOINT ,TPOINT ,TPOINT ,TPOINT ,TPOINT ,
+ PLTIME(2) ,PLTIME(2) ,PLTIME(2) ,PLTIME(2) ,
+ TEXT1 ,TEXT1 ,TEXT1 ,TEXT1 ,TEXT1 ,
+ BAUD ,BAUD ,BAUD ,BAUD ,BAUD ,
C .....
340 WORDLEN ,WORDLEN ,WORDLEN ,WORDLEN ,WORDLEN ,
C .....
C TORQUE OCCUPANT ATONE COMMON BLOCK
C .....

```

```

345 C .....
      COMMON /TORQOA / TLCHOA(3) , IMCHOA(3) , TNCIOA(3) ,
      + TLAEOA , TMAEOA , TNAEOA
C .....
C TORQUE SEAT ALONE COMMON BLOCK
C .....
350 C .....
      COMMON /TORQSA / TLAESA , TMAESA , TNAESA
C .....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C .....
      COMMON /TORQSD / TICASD(2) , TMCASD(2) , TNCASD(2) ,
      + TLTUBSD , TMTUBSD , TINTUBSD ,
      + TLSLSO(6) , TMSLSO(6) , TNSLSO(6) ,
      + TLRKSO(6) , TMRKSO(6) , TNRKSO(6) ,
      + TLCHSO(3) , IMCHSO(3) , TNCCHSO(3) ,
      + TLAESO , TMAESO , TNAESO ,
      + TLDRTSO , TMDRTSO , TNDRTSO
C .....
C THRUST VECTOR CONTROL VARIABLES COMMON BLOCK
C .....
      COMMON /TVCVRB / ITVCFLG , CMPVAL , D3(3) , RKTCMD(3) ,
      + C29 , C30 , C31 , C32 ,
      + DTH(3,2) , ANGR(3)
C .....
C DAMPING COEFFICIENT COMMON BLOCK
C .....
365 C .....
      COMMON /DAMPNG / DMPGF2 , DMPGF3 , DMPGC
C .....
C AERODYNAMICS INFORMATION COMMON BLOCK
C .....
      COMMON /AEROCFS / QALPH , QABETA , QAVEL , QAMACH ,
      + SAALPH , SABETA , SAVEL , SAMACH ,
      + SOALPH , SOBETA , SOVEL , SOMACH ,
      + CXOA , CYOA , CZOA , CLOA , CMOA , CNOA ,
      + CXSA , CYSA , CZSA , CLSA , CMSA , CNSA ,
      + CXSO , CYSO , CZSO , CLSO , CMSO , CNSO
C .....
C PLOT FILE VARIABLES COMMON BLOCK
C .....
      COMMON /PLOT / XACC(3) , YACC(3) , ZACC(3) , ACCR(3) ,
      + PVL(3) , QVL(3) , RVL(3) , VELR(3) ,
      + FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3) ,
      + RXACC(3) , RYACC(3) , RZACC(3) , RLACR(3) ,
      + RXVEL(3) , RYVEL(3) , RZVEL(3) , RLVR(3) ,
      + RXPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3) ,
      + RPVEL(2) , RQVEL(2) , RRVEL(2) ,
      + RPPDS(2) , ROPDS(2) , RRPDS(2)
C .....
C .....
      INTEGER DOWRIAT
      DIMENSION INDEX(53)
C .....
C INITIALIZE PROGRAM CONSTANTS
C .....
      MAXEVNT = 38
      MAXLINE = 45
      MAXREPT = 31
      BIAS = 4

```

```

400 C      CALL OPENMS(2,INDEX,53,0)
      READ (1,10) DOWHAT
      10 FORMAT (B410)
      IF (DOWHAT.EQ. 4HSTOP OR DOWHAT.EQ. 10H ) GO TO 9000
      IF (DOWHAT.NE. 5HSTART) GO TO 9000
405 C *****
      C DO INITIALIZATION
      C *****
      CALL INTLZ
      IF (TERRFLG.NE. 0) GO TO 800
      PRTRQ = PI2
410 C *****
      C ONLY CHECK FOR EVENTS AT THE END OF A SUCCESSFUL INTEGRATION
      C *****
      100 CONTINUE
      TIMES = TIME + 5.E-10
      IF (INTSTP.EQ. 0) GO TO 200
      C *****
      C CHECK FOR REQUESTED STOP TIME OR STOP EVENT
      C *****
      IF (TSTOP.NE. 0 AND TIME.GT. TSTOP) GO TO 300
      IF (ESTOP.NE. 0 AND TEVENTS(ESTOP).NE. 0) GO TO 300
420 C *****
      C CHECK FOR IMPACT OF SEAT/OCCUPANT OR OCCUPANT ALONE
      C *****
      DO 110 I=29,30
      IF (IEVENTS(I).NE. 0) GO TO 300
      110 CONTINUE
430 C *****
      C CHECK FOR IMPACT OF AIRCRAFT BEFORE RAIL OR CATAPULT SEPARATION
      C *****
      IF (IEVENTS(32).NE. 0 AND IEVENTS(37).EQ. 0) GO TO 300
435 C *****
      C IF STOP TIME OR STOP EVENT IS NOT REACHED, JUST CONTINUE
      C *****
      200 CONTINUE
      CALL PLOTBJN
      CALL PLOTWAC
      CALL DYNAMCG
      IF (TERRFLG.NE. 0) GO TO 800
440 C *****
      CALL DRICALC
      IF (TERRFLG.NE. 0) GO TO 900
      CALL UPDVECT
      CALL TMUPDAT
      IF (TERRFLG.NE. 0) GO TO 900
445 C *****
      201 CONTINUE
      CALL AIRCRT
      IF (TERRFLG.NE. 0) GO TO 900
      CALL SEATOCC
      IF (TERRFLG.NE. 0) GO TO 900
450 C *****
      CALL ACSEP
      CALL SEPINIT
      IF (IREIN.EQ. 0) GO TO 202
      CALL REINTEG
      GO TO 201
455 C *****
      202 CONTINUE

```

PROGRAM GESS 74/74 OPT=1

```

460      CALL DCCALQN
         IF (IERRFLG .NE. 0) GO TO 900
         CALL SEATLN
         IF (IERRFLG .NE. 0) GO TO 900
         CALL SLUGCON
         IF (IERRFLG .NE. 0) GO TO 900
         CALL QUAT
         IF (IERRFLG .NE. 0) GO TO 900
         CALL REPORTS
         IF (IERRFLG .NE. 0) GO TO 900
         CALL INTEG
         IF (IERRFLG .NE. 0) GO TO 900
         GO TO 100
300      CONTINUE
         GO TO 9000
C *****
C GET HERE IF A FATAL ERROR WAS ENCOUNTERED
C *****
475      900 WRITE (5,999)
         999 FORMAT (50X,'RUN ABORTED*')
         GO TO 9000
C *****
C GET HERE IF NO START CARD FOUND
C *****
480      1000 CONTINUE
         1010 WRITE (5,1010)
         1010 FORMAT(1H1,'50X','NO START CARD - RUN ABORTED*')
         9000 CONTINUE
         STOP
         END
485

```



```

1 SUBROUTINE ACSEP
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - DETERMINES SIMULATION PHASE STATUS AT TIME OF SEAT/
C OCCUPANT SEPARATION FROM AIRCRAFT
5 C METHOD - DETERMINES WHEN RAIL SEPARATION AND CATAPULT
C SEPARATION (FOR 0-2 CATAPULTS) HAVE OCCURRED AND SETS
C FLAG AND TIME IN EVENTS ARRAY AND TIMES ARRAY,
C RESPECTIVELY. SINCE THIS BEGINS PHASE 2 OF THE
C TRAJECTORY SIMULATION, ACSEP SETS A FLAG (IREIN) TO
C RESTART THE INTEGRATION PROCESS WITH THE TIME STEP
C DELTAT = DTPHASE2. IF THE USER DOES NOT WANT THE
C AIRCRAFT TRAJECTORY, THE AIRCRAFT EQUATIONS ARE
C TURNED OFF.
15 C COMMUNICATIONS -
C CALLED BY: GESS
C CALLS:
C NONE
C NON-COMMON VARIABLES DEFINED:
C INDX - USED AS INDEX FOR THE NUMBER OF CATAPULTS
C POTENTIAL ERROR CONDITIONS:
C NONE
25 C .....
C SECTION 9 COMMON BLOCK
C .....
C COMMON /CATPLT / INCAT : CATLNT(2), CATSTK(2), ICI (2),
C + XPOSAP(2) : YPOSAP(2), ZPOSAP(2), NPISCT(2),
C + CATHRST(2,25,2), ITUBEND : KTUBE : CTUBE
C + PTUBE : MUTUBE : EXTLNGT : ICATOUT
C REAL
C + KTUBE : MUTUBE
C .....
35 C SECTION 1 COMMON BLOCK
C .....
C COMMON /ICONTROL / TSTART, TSTOP, ESTOP, IRESTRT, IUNITS,
C + ISEATTR, ISOSEP, IPLOT, IDIRFLG,
C + IPHASE1, IPHASE2, IPHASE3
C .....
40 C INTEGER
C COMMON /IREPORT / IREPTS(31), PRTRQ, PI1, PI2, PI3
C INTEGER
C .....
45 C SECTION 3 COMMON BLOCK
C .....
C COMMON /IDELTAT / DTPHASE1, DTPHASE2, DTPHASE3
C .....
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C COMMON /MISC / IPAGECT(31), LINECT(31), IPRTCNT(31),
C + MAXREPT : MAXEVRT
C + IEVLNE : LU
C + IDATE : HEADFLG : HEADVEL
C + HEADSR : HEADYAW : HEADPIT
C + HEADROL : HEADWGT : BIAS
C + REPTYPE(9,31), PRTLNGT(2), PRTWGHT(2)
C + IHEADER(24), IEVENTS(38), TIMES(38)
C + IMVDC : PRTEMP( 2)

```

5-12

```

1 SUBROUTINE AERFMOA
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - COMPUTES AERODYNAMIC FORCES AND MOMENTS ACTING ON
C THE OCCUPANT ALONE FOLLOWING SEAT/OCCUPANT
C SEPARATION.
C METHOD - THE AERODYNAMIC COEFFICIENTS USED TO COMPUTE THE
C FORCES AND MOMENTS ARE FOUND BY INTERPOLATING IN
C THE AERODYNAMIC COEFFICIENTS TABLES WHICH WERE READ
C IN AND STORED BY SUBROUTINE AEROIN DURING PROGRAM
C INITIALIZATION.
C COMMUNICATIONS:
C CALLED BY: OCCALON
C CALLS:
C INTRP
C NON-COMMON VARIABLES DEFINED:
C XVEL -
C YVEL - COMPONENTS OF OCCUPANT ALONE VELOCITY MINUS WIND
C ZVEL - VELOCITY
C LEVEL - RESULTANT VELOCITY OF OCCUPANT ALONE, MINUS WIND
C VELOCITY
C ALPHAS - ANGLE OF ATTACK IN RADIAN
C ALPHA - ANGLE OF ATTACK IN DEGREES
C OABETA - ANGLE OF SIDESLIP
C CXOA -
C CYOA -
C CZOA -
C CLDA - AERODYNAMIC COEFFICIENTS
C CMOA -
C CNOA -
C AERFORC - TOTAL AERODYNAMIC FORCE ACTING ON THE OCCUPANT ALONE
C POTENTIAL ERROR CONDITIONS:
C IF OALPH OR OABETA IS OUTSIDE THE AERODYNAMIC
C COEFFICIENTS TABLE LIMITS, A MESSAGE IS PRINTED
C AND THE RUN IS TERMINATED.
C.....
C COEFFICIENTS (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C.....
C COMMON /COEF / COEF(700,6)
C.....
C CONSTANTS COMMON BLOCK
C.....
C COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C DENSITY COMMON BLOCK
C.....
C COMMON /DENSITY / IATMOS , OLDALT(3) , RHO5
C + PRESALT(3) , DTEMP
C + TEMPS
C + VZWIND , VYWIND , VZWIND
C.....
C OCCUPANT ALONE FORCES COMMON BLOCK
C.....
C COMMON /FORCEOA / FXCHOA(3) , FYCHOA(3) , FZCHOA(3)
C + FXAEOA , FYAEOA , FZAEOA

```

```

C.....
C SECTION 4 COMMON BLOCK
C.....
COMMON /IAIRCRT / TEMP , PRESSUR, ZACVEL , XPOS , YPOS ,
+ ZPDS , XTAIL , YTAIL , ZTAIL , YAW ,
+ PITCH , ROLL , QVEL , PVEL ,
+ WINDX , WINDY , WINDZ , XACVEL , CKPITH ,
+ DENSITY, NPTSAAT, AAT(4.50), NPTSLAT,LAT(4.50),
+ IACSFGL
C.....
C.....
C INFO04 DATA (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C.....
COMMON /INFO04 / NCXS(12) , NCYS(12) , NCZS(12) ,
+ DLTC(3,12) , ENPC(6,12) , IAERCSO(12)
C.....
C SECTION 6 COMMON BLOCK
C.....
COMMON /ISEATOC / IPCNTL , XCGSD , YCGSD , ZCGSD , IXXSO ,
+ IYXSO , IYZSO , IYYSO , IYZSO , IZZSO ,
+ AREASO , AREAOA , WGHDOAB , WGHDOAA ,
+ IXXOA , IXYOA , IYZOA , IYXOA , IYXOA , IYZOA ,
+ IZZOA , XCGOA , YCGOA , ZCGOA , SOSEP ,
+ IXXSO , IYXSO , IYZSO , IYXSO , IYZSO ,
+ IZZSO , IXXOA , IXYOA , IYZOA , IYXOA ,
+ IYZOA , IZZOA
C.....
C MATRIX COMMON BLOCK
C.....
COMMON /MATRIX / DCMRAE(3,3) , DCMRA(3,3) , DCMRA(3,3) ,
+ DCMSE(3,3) , DCMTE(3,3) , DCMTE(3,3) ,
+ DCMRAE(3,3) , DCMRAE(3,3) , DCMRA(3,3) ,
+ DCMDOUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRONT(31) ,
+ MAXLINE , MAXREPT , MAXEVT ,
+ IEVLNE , IERRFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADRDL , HEADYAW , HEADPIT ,
+ REPTYPE(5,31) , HEADWGT , BIAS ,
+ IHEADER(24) , IEVENTS(38) , PRTWGHT(2) ,
+ IMVOC , TIMES(38) , PRIEMP( 2) ,
+ PRTMAS(2) , PRTINOX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+ REPTYPE , BIAS , PRTLNGT ,
+ PRTWGHT , PRTEMP , PRINDX
C.....
C MOMARMS COMMON BLOCK
C.....
COMMON /MOMARMS /
+ REFLNSO , REFLNOA , REFLNSA , URX(6) , URY(6) , URZ(6) ,
+ XSSOCA(12),YSSOCA(12),ZSSOCA(2),XSSORR(6),YSSORR(6),ZSSORR(6),
+ XSSORRE ,YSSORRE ,ZSSORRE ,XSSOLRE ,YSSOLRE ,ZSSOLRE ,

```

```

115 +XSSOMRE ,YSSOMRE ,ZSSOMRE ,XSSOBOT ,YSSOBOT ,ZSSOBOT ,
+XSSOSB(6),YSSOSB(6),ZSSOSB(6),XRRCSAC ,YRRCSAC ,ZRRCSAC ,
+XSSCSAC ,YSSCSAC ,ZSSCSAC ,XSSOSRP ,YSSOSRP ,ZSSOSRP ,
+
+XSSASRP ,YSSASRP ,ZSSASRP ,XRRDAP(2),YRRDAP(2),ZRRDAP(2),
+XRRSBO(6),YRRSBO(6),ZRRSBO(6),XSSOCP(2),YSSOCP(2),ZSSOCP(2),
+XSSDAP(2),YSSDAP(2),ZSSDAP(2),XESDAC ,YESDAC ,ZESDAC ,
+XSRCSAC ,YSRCSAC ,ZSRCSAC ,XSSDAC ,YSSDAC ,ZSSDAC ,
+XRSOSB ,YRSOSB ,ZRSOSB ,XRRSBO ,YRRSBO ,ZRRSBO ,
+XRRSB ,YRRSB ,ZRRSB ,XSSOCH(3),YSSOCH(3),ZSSOCH(3),
+XAACSD ,YAACSD ,ZAACSD ,XASDAC ,YASDAC ,ZASDAC ,
+XRSOAC ,YRSOAC ,ZRSOAC ,XSCPAP(2),YSCPAP(2),ZSCPAP(2),
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
130 COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193) ,
+ TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3) ,
+ TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
+ QUATSA(65) , QUATDA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IPOINTS , IYX , IYPRX ,
+ IKX , IKSUMX , IKPASSX ,
+ IYIX , IYIIX , IYI2X ,
+ IYIX , IYPRIX , IYPRIIX ,
+ IYPRIX , IYIIX , IYI1IX ,
+ ICYIX , ICYIIX , IREIN ,
C .....
C TORQUE OCCUPANT ALONE COMMON BLOCK
C .....
145 COMMON /TORQOA / TLCHOA(3) , TMCHOA(3) , TNCHOA(3) ,
+ TLAEOA , TMAEOA , TNAEOA ,
C .....
C AERODYNAMICS INFORMATION COMMON BLOCK
C .....
150 COMMON /AEROCS / OALPH , OABETA , OAVEL , OAMACH ,
+ SAALPH , SABETA , SAVEL , SAMACH ,
+ SOALPH , SOBETA , SOVEL , SOMACH ,
+ CXOA , CYOA , CZOA , CLOA , CMOA , CNOA ,
+ CXSA , CYSA , CZSA , CLSA , CMSA , CNSA ,
+ CXSO , CYSO , CZSO , CLSO , CMSO , CNSO ,
C .....
C .....
C COMPUTE RESULTANT VELOCITY IN OACS (SAME AS EFCS)
C .....
160 XVEL = TRAJOA(5) - VXWIND
YVEL = TRAJOA(6) - VYWIND
ZVEL = TRAJOA(7) - VZWIND
C .....
C OAVEL = SORT(XVEL * XVEL + YVEL * YVEL + ZVEL * ZVEL)
C .....
C COMPUTE ANGULAR VARIABLES
C .....
170 SIN2 = ZVEL/OAVEL
OALPH = ASIN(SIN2)
COS2 = COS(OALPH)
OABETA = ZARCTAN(YVEL,XVEL)

```

```

C.....
C COMPUTE AERODYNAMIC COEFFICIENTS
C.....
C.....
C USE EFFECTIVE DRAG AREA PARAMETER TO INCORPORATE CONSTANT DRAG
C.....
      CXOA = -SIGN((COS2*COS(DABETA)), TRAJOA(5))
      CYOA = -SIGN((COS2*SIN(DABETA)), TRAJOA(6))
      CZOA = -SIGN((SIN2), TRAJOA(7))
C.....
C EXTRACT AERODYNAMIC INFORMATION
C.....
      DAMACH = DAVEL / (49.0212 * TEMPS ** 0.5)
C.....
C COMPUTE FORCES FOR OCCUPANT ALONE IN OACS
C.....
      AERFORC = 5 * RHOS * AREAOA * DAVEL * DAVEL
      FXAEOA = AERFORC * CXOA
      FYAEOA = AERFORC * CYOA
      FZAEOA = AERFORC * CZOA
      GOTO 500
C
C
C
C 100 IERRFLG = 1
C
C 500 CONTINUE
      RETURN
      END

```

```

1  SUBROUTINE AERFMSA
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - COMPUTES THE AERODYNAMIC FORCES AND MOMENTS ACTING
C ON THE SEAT AFTER SEAT/OCCUPANT SEPARATION
C METHOD - THE AERODYNAMIC COEFFICIENTS USED TO COMPUTE THE
C FORCES AND MOMENTS ARE FOUND BY INTERPOLATING IN
C THE AERODYNAMIC COEFFICIENTS TABLES WHICH WERE
C READ IN AND STORED BY SUBROUTINE AEROTN DURING
C PROGRAM INITIALIZATION.
C COMMUNICATIONS -
C CALLED BY: SEATLN
C CALLS:
C INTRP, ROTATE, ZARCTAN
C NON-COMMON VARIABLES DEFINED:
C VXWIND - COMPONENTS OF WIND VELOCITY IN SEAT COORDINATE SYSTEM
C VYWIND -
C VZWIND -
C XVEL -
C YVEL - COMPONENTS OF SEAT ALONE VELOCITY MINUS WIND VELOCITY
C ZVEL -
C LEVEL - RESULTANT VELOCITY OF SEAT ALONE MINUS WIND VELOCITY
C ALPHAS - ANGLE OF ATTACK IN RADIAN
C ALPHA - ANGLE OF ATTACK IN DEGREES
C SABETA - ANGLE OF SIDESLIP IN DEGREES
C VSOUND - VELOCITY OF SOUND IN FEET/SECOND
C MACHS - MACH NUMBER
C CXSA -
C CYSA -
C CZSA -
C CLSA - AERODYNAMIC COEFFICIENTS
C CMSA -
C CNSA -
C AERFORC - TOTAL AERODYNAMIC FORCE ACTING ON THE SEAT ALONE
C POTENTIAL ERROR CONDITIONS:
C IF SAALPH, SABETA, OR SAMACH IS OUTSIDE THE AERODYNAMIC
C COEFFICIENTS TABLE LIMITS, A MESSAGE IS OUTPUT, AND
C THE RUN IS TERMINATED.
C.....
C COEFFICIENTS (USED IN SUBROUTINE AEROTN) COMMON BLOCK
C.....
C COMMON /COEF / COEF(700,6)
C.....
C CONSTANT'S COMMON BLOCK
C.....
C COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C.....
C DENSITY COMMON BLOCK
C.....
C COMMON /DENSITY / IATMOS , OLDALT(3) , RHOS
C , PRESALT(3) , DTEMP

```

```

+      TEMPS
+      VWIND      VYVIND      VZWIND
C.....
60 C SEAT ALONE FORCES COMMON BLOCK
C.....
COMMON /FORCESA / FXAESA      FYAESA      FZAESA
C.....
65 C SECTION 4 COMMON BLOCK
C.....
COMMON /IAIRCRT / TEMP      PRESSUR      ZACVEL      XPOS      YPOS
+      ZPOS      XTAIL      YTAIL      ZTATT      YAW
+      PITCH      ROLL      RVEL      QVEL      PVEL
+      WINDX      WINDY      WINDZ      XACVEL      CKPITH
+      DENSITY      NPTSAAT      AAT(4.50)      NPTSLAT      LAT(4.50)
+      IACSFLG
C.....
70 C.....
75 C INFO04 DATA (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C.....
COMMON /INFO04 / NCXS(12)      NCYS(12)      NCZS(12)
+      DLTC(3,12)      ENDC(6,12)      IAERCSO(12)
C.....
80 C SECTION 1 COMMON BLOCK
C.....
COMMON /ICONTROL / TSTART      TSTOP      ESTOP      IRESTART      IUNITS
+      ISEATTIR      ISOSEP      IPLOT      IDRIFLG
+      IPHASE1      IPHASE2      IPHASE3
+      ESTOP
C.....
85 C.....
C SECTION 5 COMMON BLOCK
C.....
COMMON /ISETALN / XPOSSRP      YPOSSRP      ZPOSSRP      XCGSA      YCGSA
+      ZCGSA      IXXSA      IYXSA      IZXA      IYXSA      IYXSA
+      IYXSA      IZXA      PHISA      PSISA      THESA
+      AREASA      HGTSA      WGTSA      XPOSBOT      YPOSBOT
+      ZPOSBOT      XPOSSCS      YPOSSCS      ZPOSSCS
+      IXXSA      IYXSA      IZXA      IYXSA      IYXSA
+      IZXA
C.....
95 C MATRIX COMMON BLOCK
C.....
COMMON /MATRIX / DCMAB(3,3)      DCMRA(3,3)      DCMXA(3,3)
+      DCMSE(3,3)      DCMTS(3,3)      DCMTE(3,3)
+      DCMXAE(3,3)      DCMXAE(3,3)      DCMXSR(3,3)
+      DCMXDM(3,3)
C.....
100 C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31)      LINECT(31)      IPRTCT(31)
+      MAXLINE      MAXREPT      MAXEVT
+      TEVLIN      TERRFLG      LU
+      IDATE      HEADALT      HEADVEL
+      HEADSR      HEADYAW      HEADPIT
+      HEADROL      HEADWGT      BIAS
+      REPTYR(5,31)      PRTINGT(2)      PRTWGT(2)
+      THEADER(24)      TEVENTS(38)      TIMES(38)
+      IMVDC      PRTEMP( 2)
+      PRIMASS(2)      PRINDX      PKZVEL

```



```

115      +      ZVECT(3)      , XYZ(3)      , SAVTIME
      +      XACCEL(3)      , ZACCEL(3)
      +      REPTYPE      , BIAS      , PRTLNIGT
      +      PRTEMP      , PRTEMP      , PRINDX
C.....
C MOMARMS COMMON BLOCK
C.....
COMMON /MOMARMS /
+REFLNSO ,REFLNOA ,REFLNSA ,URX(6) ,URY(6) ,URZ(6)
+XSSOCA(2) ,YSSOCA(2) ,ZSSOCA(2) ,XSSORK(6) ,YSSORK(6) ,ZSSORK(6)
+XSSORRE ,YSSORRE ,ZSSORRE ,XSSOLRE ,YSSOLRE ,ZSSOLRE
+XSSOMRE ,YSSOMRE ,ZSSOMRE ,XSSOBOT ,YSSOBOT ,ZSSOBOT
+XSSOSB(6) ,YSSOSB(6) ,ZSSOSB(6) ,XSSCSAC ,YSSCSAC ,ZSSCSAC
+XSSCSR ,YSSCSR ,ZSSCSR
+XSSASRP ,YSSASRP ,ZSSASRP ,XRRDAP(2) ,YRRDAP(2) ,ZRRDAP(2)
+XRRSBO(6) ,YRRSBO(6) ,ZRRSBO(6) ,XSSOCP(2) ,YSSOCP(2) ,ZSSOCP(2)
+XSSDAP(2) ,YSSDAP(2) ,ZSSDAP(2) ,XESOAC ,YESOAC ,ZESOAC
+XSRCSAC ,YSRCSAC ,ZSRCSAC ,XSSOAC ,YSSOAC ,ZSSOAC
+XRSOSB ,YRSOSB ,ZRSOSB ,XRRSBO ,YRRSBO ,ZRRSBO
+XRRSB ,YRRSB ,ZRRSB ,XSSOCH(3) ,YSSOCH(3) ,ZSSOCH(3)
+XAACSO ,YAACSO ,ZAACSO ,XASOAC ,YASOAC ,ZASOAC
+XRSOAC ,YRSOAC ,ZRSOAC ,XSCPAP(2) ,YSCPAP(2) ,ZSCPAP(2)
C.....
C INTEGRATION ROUTINE COMMON BLOCK
C.....
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193)
+ TRAJSA(193) , TRAJDA(193) , TRAJCH(97.3)
+ TRAJAC(193) , TVCEQS(225) , QUATSO(65)
+ QUATSA(65) , QUATOA(65) , QUATAC(65)
+ IN1STP , IPCPASS , IRKPASS , IRKPRX
+ IPOINTS , IYX , IYPRX , IKSUMX
+ IKX , IYI1X , IYI2X , IYI3X
+ IYI1X , IYPR1X , IYPR1X , IYI1X
+ IYPR12X , IYI1X , IYI1X , IYI1X
+ IYI1X , IYI1X , IYI1X , IYI1X
C.....
C TORQUE SEAT ALONE COMMON BLOCK
C.....
COMMON /TOROSA / TLASA , TMAESA , TNAESA
C.....
C AERODYNAMICS INFORMATION COMMON BLOCK
C.....
COMMON /AERUCFS / OALPH , OABETA , OAVEL , OAMACH
+ SAALPH , SABETA , SAVEL , SAMACH
+ SOALPH , SOBETA , SOVEL , SOMACH
+ CXOA , CYOA , CZOA , CLOA , CMOA , CNOA
+ CXSA , CYSA , CZSA , CLSA , CMSA , CNSA
+ CXSO , CYSO , CZSO , CLSO , CMSO , CNSO
C.....
C DAMPING COEFFICIENT COMMON BLOCK
C.....
COMMON /DAMPING / DMPGF2 , DMPGF3 , DMPGC
C.....

```

```

175      DIMENSION COF(6)
      REAL LACCEL
      C .....
      C COMPUTE RESULTANT VELOCITY IN SACS
      C .....
      C
      CALL ROTATE(WINDX,VXWIND,ZVECT(1),DCMSAE,O)
      C
      XVEL = TRAJSA(5) - VXWIND
      YVEL = TRAJSA(6) - VYWIND
      ZVEL = TRAJSA(7) - VZWIND
      C
      SVEL = SORT(XVEL * XVEL + YVEL * YVEL + ZVEL * ZVEL)
      C
      C .....
      C COMPUTE ANGLE OF ATTACK (SAALPH), ANGLE OF SIDESLIP (SABETA),
      C AND SAMACH NUMBER (SAMACH)
      C .....
      C
      SAALPH = -ZARCTAN(ZVEL,XVEL)
      IF(SAALPH.LT.O.O) SAALPH=SAALPH*6.2831852
      SABETA = O.O
      IF(SVEL.NE.O.O) SABETA = ASIN(YVEL / SVEL)
      SAALPH = SAALPH * RADDEG
      SABETA=SABETA*RADDEG
      C
      VSDUND = 49.0212 * (TEMPS ** 0.5)
      IF(TUNITS.EQ.O) VSDUND = VSDUND/3.28
      SAMACH = SVEL / VSDUND
      IF(SAMACH.LE.1.2) GOTO 10
      AVEL = SORT(TRAJSA(11) * TRAJSA(11) + TRAJSA(12) * TRAJSA(12) +
      +TRAJSA(13) * TRAJSA(13))
      LACCEL = SORT(TRAJSA(17) * TRAJSA(17) + TRAJSA(18) * TRAJSA(18) +
      +TRAJSA(19) * TRAJSA(19))
      AACCEL = SORT(TRAJSA(23) * TRAJSA(23) + TRAJSA(24) * TRAJSA(24) +
      +TRAJSA(25) * TRAJSA(25))
      WRITE(5,40) TIME , SAMACH, SVEL , AVEL, LACCEL, AACCEL
      SAMACH = 1.2
      C
      C .....
      C COMPUTE AERODYNAMIC COEFFICIENTS
      C .....
      C
      10 CONTINUE
      DO 20 I = 2,6,2
      CALL INTRP(COE(1,1),NCXS(1),NCYS(1),DLTC(1,1),
      +ENDPC(1,1),2,SAALPH,ABS(SABETA),SAMACH,COF(1),IFAIL)
      IF(IFAIL.NE.O) GOTO 50
      IF(SABETA.LT.O.O) COF(1) = -COF(1)
      20 CONTINUE
      C
      DO 30 I = 1,5,2
      CALL INTRP(COE(1,1),NCXS(1),NCYS(1),DLTC(1,1),
      +ENDPC(1,1),2,SAALPH,ABS(SABETA),SAMACH,COF(1),IFAIL)
      IF(IFAIL.NE.O) GOTO 50
      30 CONTINUE

```

5-21

NADC-84068-60

21

PAGE

53

83/11/07 09.41

FIN 4 6:428

OPT:1

74/74

SUBROUTINE AERFMSA

RETURN
END

```

1 SUBROUTINE AERFMSD
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - COMPUTES AERODYNAMIC FORCES AND MOMENTS ACTING ON
C THE SEAT/OCCUPANT SYSTEM PRIOR TO SEAT/OCCUPANT
C SEPARATION.
C METHOD - THE AERODYNAMIC COEFFICIENTS USED TO COMPUTE THE
C FORCES AND MOMENTS ARE FOUND BY INTERPOLATING IN
C THE AERODYNAMIC COEFFICIENTS TABLES WHICH WERE
C READ IN AND STORED BY SUBROUTINE AEROIN DURING
C PROGRAM INITIALIZATION. AERODYNAMIC FORCES AND
C MOMENTS ARE ADJUSTED TO REFLECT PARTIAL EXPOSURE
C OF THE SEAT/OCCUPANT COMBINATION AS IT ENTERS
C THE AIRSTREAM.
C COMMUNICATIONS -
C CALLED BY: SEATDCC
C CALLS:
C INTRP, ROTATE, ZARCTAN
C NON COMMON VARIABLES DEFINED:
C VXWIND - COMPONENTS OF WIND VELOCITY IN SEAT COORDINATE
C VYWIND - SYSTEM
C VZWIND -
C XVEL -
C YVEL - COMPONENTS OF SEAT/OCCUPANT VELOCITY MINUS
C ZVEL - WIND VELOCITY
C LEVEL - RESULTANT VELOCITY OF SEAT/OCCUPANT MINUS
C WIND VELOCITY
C ALPHA - ANGLE OF ATTACK IN RADIAN
C ALPHA - ANGLE OF ATTACK IN DEGREES
C SOBETA - ANGLE OF SIDESLIP
C VSOUND - VELOCITY OF SOUND IN FEET/SECOND
C MACHS - MACH NUMBER
C EXPOSED - FRACTIONAL PORTION OF THE SEAT/OCCUPANT COMBINATION
C EXPOSED TO THE AIRSTREAM
C Z - Z POSITION OF THE SEAT/OCCUPANT C.G. ADJUSTED TO
C REFLECT PARTIAL EXPOSURE OF THE SEAT/OCCUPANT
C AERFORC - TOTAL AERODYNAMIC FORCE ACTING ON THE SEAT/OCCUPANT
C COMBINATION WHEN FULLY EXPOSED TO THE AIRSTREAM
C CXSO -
C CYSO -
C CZSO -
C CLSO - AERODYNAMIC COEFFICIENTS
C CMSO -
C CNSO -
C POTENTIAL ERROR CONDITIONS:
C IF SOALPH, SOBETA, OR SOMACH IS OUTSIDE AERODYNAMIC
C COEFFICIENTS TABLE LIMITS, A MESSAGE IS OUTPUT, AND
C THE RUN IS TERMINATED
C.....
C COEFFICIENTS (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C.....
COMMON /COEF / COEF(700,6)

```

```

C.....
C CONSTANT'S COMMON BLOCK
C.....
COMMON /CONSTANT / GRAVITY , RADDEG , DEGRAD , PI
C.....
C DENSITY COMMON BLOCK
C.....
COMMON /DENSITY / IATMOS , OLDALT(3) , RHOS
+
+ TEMPS
+
+ VFWIND , VVWIND , VZWIND
C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
COMMON /FORCES / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+
+ FXTUBSO , FYTUBSO , FZTUBSO ,
+
+ FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+
+ FXRKS(6) , FYRKS(6) , FZRKS(6) ,
+
+ FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+
+ FXAESO , FYAESO , FZAESO ,
+
+ FXDRTSO , FYDRTSO , FZDRTSO
C.....
C SECTION 4 COMMON BLOCK
C.....
COMMON /IAIRCRT / TEMP , PRESSUR, ZACVEL , XPOS , YPOS ,
+
+ ZPOS , XTAIL , YTAIL , ZTAIL , YAW ,
+
+ PITCH , ROLL , QVEL , PVEL ,
+
+ WINDX , WINDY , WINDZ , XACVEL , CKPITH ,
+
+ DENSITY , NPTSAAT , AAT(4,50) , NPTSLAT,LAT(4,50) ,
+
+ IACSFLG
C.....
C SECTION 1 COMMON BLOCK
C.....
COMMON /ICONTROL / TSTART , TSTOP , ESTOP , IRESTR, IUNITS ,
+
+ ISEATTR, ISESEP , IPLOT , IDIRFLG ,
+
+ IPHASE1, IPHASE2, IPHASE3
+
+ ESTOP
C.....
C INFO04 DATA (USED IN SUBROUTINE AERGIN) COMMON BLOCK
C.....
COMMON /INFO04 / NCXS(12) , NCYS(12) , NCZS(12)
+
+ DLTC(3,12) , ENOPC(8,12) , IAECSO(12)
C.....
C SECTION 6 COMMON BLOCK
C.....
COMMON /ISEATOC / IPCNIL , XCGSO , YCGSO , ZCGSO , IXXSO
+
+ IYISO , IXZSO , IYYSO , IYZSO , IZZSO ,
+
+ AREASO , AREADA , WGHTOAB, WGHTOAA ,
+
+ IXXOA , IXVOA , IXZOA , IYVOA , IYZOA ,
+
+ IZZOA , XCGOA , YCGOA , ZCGOA , SOSEP
+
+ REAL
+
+ IXXSO , IYXSO , IXZSO , IYXZO , IYZSO ,
+
+ IZZSO , IXXOA , IXVOA , IXZOA , IYVOA ,
+
+ IZZOA , IZZOA
C.....
C SECTION 5 COMMON BLOCK
C.....
COMMON /ISETALN / XPOSSRP , YPOSSRP , ZPOSSRP , XCGSA , YCGSA

```

```

115      + ZGSA , IXSA , IXSA , IXSA , IXSA , IXSA , IXSA ,
      + IYSA , IZSA , PHISA , PSISA , PSISA , THESA ,
      + AREASA , HGHTSA , WHTSA , XPOSBOT , YPOSBOT ,
      + ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS
      + IXSA , IXSA , IXSA , IXSA , IXSA , IXSA ,
      + IZSA
120      C .....
      C MATRIX COMMON BLOCK
      C .....
      COMMON /MATRIX / DCMAR(3,3) , DCMRA(3,3) , DCMRA(3,3) ,
      + DCMSE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,
      + DCMSE(3,3) , DCMOAE(3,3) , DCMSE(3,3) ,
      + DCMOUM(3,3)
125      C .....
      C MISCELLANEOUS DATA COMMON BLOCK
      C .....
      COMMON /MISC / IPAGECT(31) , LINCT(31) , IPRCNT(31) ,
      + MAXLINE , MAXREPT , MAXEVT ,
      + IEVLIN , TERFLG , LU ,
      + IDATE , HEADAL , HEADVEL ,
      + HEADSR , HEADYAW , HEADPIT ,
      + HEADROL , HEADWGT , BIAS ,
      + REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
      + IHEADER(24) , IEVENTS(38) , TIMES(38) ,
      + IMVOC , PRTEMP( 2) ,
      + PRTMAS(2) , PRTINDX , PKZVEL ,
      + ZVECT(3) , XYZ(3) , SAVTIME ,
      + XACCEL(3) , YACCEL(3) ,
      + REPTYPE , BIAS , PRTLNGT ,
      + PRWGT , PRTEMP , PRTMAS , PRTINDX
140      C .....
      C MOMARMS COMMON BLOCK
      C .....
      COMMON /MOMARMS /
      + REFNSO , REFNSA , URX(6) , URY(6) , URZ(6) ,
      + XSOCA(2) , YSOCA(2) , ZSOCA(2) , XSORR(6) , YSORR(6) , ZSORR(6) ,
      + XSORRE , YSORRE , ZSORRE , XSOILRE , YSOILRE , ZSOILRE ,
      + XSSOMRE , YSSOMRE , ZSSOMRE , XSSOBOT , YSSOBOT , ZSSOBOT ,
      + XSSOSB(6) , YSSOSB(6) , ZSSOSB(6) , XRRCSAC , YRRCSAC , ZRRCSAC ,
      + XSSCSAC , YSSCSAC , ZSSCSAC , XSSOSRP , YSSOSRP , ZSSOSRP ,
      + XSSASRP , YSSASRP , ZSSASRP , XRRDAP(2) , YRRDAP(2) , ZRRDAP(2) ,
      + XRRSBO(6) , YRRSBO(6) , ZRRSBO(6) , XSSOCP(2) , YSSOCP(2) , ZSSOCP(2) ,
      + XSSDAP(2) , YSSDAP(2) , ZSSDAP(2) , XESDAP , YESDAP , ZESDAP ,
      + XSRCSAC , YSRCSAC , ZSRCSAC , XSSOAC , YSSOAC , ZSSOAC ,
      + XRSOSB , YRSOSB , ZRSOSB , XRSOBOT , YRSOBOT , ZRSOBOT ,
      + XRRSB , YRRSB , ZRRSB , XSSOCH(3) , YSSOCH(3) , ZSSOCH(3) ,
      + XAACSO , YAACSO , ZAACSO , XASOAC , YASOAC , ZASOAC ,
      + XRSOAC , YRSOAC , ZRSOAC , XSCAP(2) , YSCAP(2) , ZSCAP(2)
165      C .....
      C INTEGRATION ROUTINE COMMON BLOCK
      C .....
      COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAUSO(193) ,
      + TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3) ,
      + TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
      + QUATSA(65) , QUATOA(65) , QUATAC(65)

```

5-26


```

230      VSOUND = 49.0212 * (TEMPS ** 0.5)
      IF (IUNITS .EQ. 0) VSOUND = VSOUND/3.28
      SOMACH = SOVEL / VSOUND
      IF (SOMACH .LE. 1.2) GOTO 10
      AVEL = SORT(TRAJSD(11) * TRAJSD(11) + TRAJSD(12) * TRAJSD(12) +
      + TRAJSD(13) * TRAJSD(13))
      LACCEL = SORT(TRAJSD(17) * TRAJSD(17) + TRAJSD(18) * TRAJSD(18) +
      + TRAJSD(19) * TRAJSD(19))
      AACCEL = SORT(TRAJSD(23) * TRAJSD(23) + TRAJSD(24) * TRAJSD(24) +
      + TRAJSD(25) * TRAJSD(25))
      WRITE(5,40) TIME, SOMACH, SOVEL, AVEL, LACCEL, AACCEL
      SOMACH = 1.2
      C.....
      C COMPUTE AERODYNAMIC COEFFICIENTS
      C.....
      10 CONTINUE
      DO 20 I=2,6,2
      CALL INTRP(COEF(1,I),NCXS(I),NCYS(I),NCZS(I),DLTC(1,I),
      +ENDPC(1,I),2,SOALPH,ABS(SOBETA),SOMACH,COF(1),IFAIL)
      IF (IFAIL .NE. 0) GOTO 50
      IF (SOBETA .LT. 0.0) COF(1) = -COF(1)
      20 CONTINUE
      C
      DO 30 I=1,5,2
      CALL INTRP(COEF(1,I),NCXS(I),NCYS(I),NCZS(I),DLTC(1,I),
      +ENDPC(1,I),2,SOALPH,ABS(SOBETA),SOMACH,COF(1),IFAIL)
      IF (IFAIL .NE. 0) GOTO 50
      30 CONTINUE
      C
      CXSO = COF(1)
      CYSO = -COF(2)
      CZSO = -COF(3)
      CLSO = COF(4)
      CMSO = -COF(5)
      CNSO = -COF(6)
      C.....
      C COMPUTE FORCES AND MOMENTS FOR SEAT/OCCUPANT
      C.....
      EXPOSED = 0.0
      IF ((ABS(ZRRSBO) + HGHTSA) .GT. CKPITHT) EXPOSED =
      + (ABS(ZRRSBO) + HGHTSA - CKPITHT) / HGHTSA
      IF (ABS(ZRRSBO) .GE. CKPITHT) EXPOSED = 1.0
      IF (IEVENTS(5) .NE. 0) EXPOSED = 1.0
      Z = ZSSOSRP + (ZARMP - (EXPOSED * ZARMP))
      AERFORC = (.5 * RHOS * AREASO * SOVEL * SOVEL) * EXPOSED
      FXAESO = AERFORC * CXSO
      FYAESO = AERFORC * CYSO
      FZAESO = AERFORC * CZSO
      C
      TLAESO = AERFORC * (YSSOSRP * CZSO - Z * CYSO +
      +REFINSO * CLSO)
      TMAESO = AERFORC * (Z * CXSO - XSSOSRP * CZSO +
      +REFINSO * CMSO)
      TNAESO = AERFORC * (XSSOSRP * CYSO - YSSOSRP * CXSO +
      +REFINSO * CNSO)
      C.....
      C.....

```

```

C COMPUTE SEAT OCCUPANT AERODYNAMIC DAMPING EFFECTS
C
C THIS ROUTINE APPROXIMATES THE EFFECTS OF AERODYNAMIC DAMPING, WHICH
C ARE NOT PROVIDED BY OTHER MEANS
C.....
C
290      DO 35 I=11,13
          TRAJSO(I) = TRAJSO(I) + DMPGF2
          35 CONTINUE
          GOTO 500

C
295      40 FORMAT(1X, //72(1H*))/, 4X, "WARNING(SUBROUTINE AERFMSO)*** ", /.
          +4X, "CALCULATED MACH NUMBER IS GREATER THAN ALLOWED", /.
          +5X, "MACH NUMBER OF 1.2 WILL BE USED INSTEAD", /.
          +5X, "TIME = ", F10.4, 2X, "MACH NUMBER = ", F10.4, 2X,
          + "LINEAR VELOCITY = ", F10.4, 2X, "ANGULAR VELOCITY = ", F10.4,
          +/, 5X, "LINEAR ACCELERATION = ", F10.4, 2X, "ANGULAR ACCELERATION = ",
          + F10.4)
          50 IF(1IFAIL-2)55,60,65
          55 WRITE(5,70) SOALPH
          GOTO 100
          60 WRITE(5,75) SOBETA
          GOTO 100
          65 WRITE(5,80) SOMACH
          GOTO 100
          70 FORMAT(1X, //72(1H*))/, 4X, "FATAL ERROR(SUBROUTINE AERFMSO)*** ",
          +/, "ALPHA = ", F10.4, " IS OUTSIDE TABLE LIMITS", /.72(1H*))
          75 FORMAT(1X, //72(1H*))/, 4X, "FATAL ERROR(SUBROUTINE AERFMSO)*** ",
          +/, "BETA = ", F10.4, " IS OUTSIDE TABLE LIMITS", /.72(1H*))
          80 FORMAT(1X, //72(1H*))/, 4X, "FATAL ERROR(SUBROUTINE AERFMSO)*** ",
          +/, "SOMACH = ", F10.4, " IS OUTSIDE TABLE LIMITS", /.72(1H*))
          100 CONTINUE
          IERRFLG = 1
          500 CONTINUE
          RETURN
          END
320

```

```

1  SUBROUTINE AEROIN
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - INPUTS THE AERODYNAMIC COEFFICIENTS USED TO CALCULATE
C THE AERODYNAMIC FORCES AND MOMENTS ACTING ON THE
C SEAT/OCCUPANT BEFORE SEAT/OCCUPANT SEPARATION, AND ON
C THE OCCUPANT ALONE AND THE SEAT ALONE AFTER
C SEAT/OCCUPANT SEPARATION.
C METHOD - READS COEFFICIENTS FROM RANDOM ACCESS FILE AEROIN
C COMMUNICATIONS -
C CALLED BY: INTLZ
C CALLS: READMS
C NON-COMMON VARIABLES DEFINED:
C ISEQNO
C NDLTC
C NENDPC
C FIDENT
C POTENTIAL ERROR CONDITIONS:
C NONE
C .....
C COEFFICIENTS (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C .....
C COMMON /COEF / COEF(700,6)
C .....
C INFO04 DATA (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C .....
C COMMON /INFO04 / NCXS(12) ; NCZS(12) ; NDLTC(3,12) ; ENDPC(6,12) ; IAERCSQ(12)
C .....
C DIMENSION INFO2(30,10), INFO3(20,13)
C REAL INFO2, INFO3
C DO 100 J=1,6
C IF (IAERCSQ(J) .GT. 20) GO TO 10
C NDLTC = 3
C NENDPC = 6
C ISEQ2=IAERCSQ(J)
C CALL READMS(2, INFO3, 260, 51)
C FIDENT = INFO3(1,1)
C NCXS(J)=INFO3(1,2)+.5
C NCZS(J)=INFO3(1,3)+.5
C NCZS(J)=INFO3(1,4)+.5
C DLTC(1,J)=INFO3(1,5)
C DLTC(2,J)=INFO3(1,6)
C DLTC(3,J)=INFO3(1,7)
C ENDPC(1,J)=INFO3(1,8)
C ENDPC(2,J)=INFO3(1,9)
C ENDPC(3,J)=INFO3(1,10)
C ENDPC(4,J)=INFO3(1,11)
C ENDPC(5,J)=INFO3(1,12)
C ENDPC(6,J)=INFO3(1,13)
C GO TO 20
10 ISEQ2=IAERCSQ(J)-20
C NDLTC = 2
C NENDPC = 4

```

PAGE 29

83/11/07. 09.41.53

FTN 4.6+428

SUBROUTINE AERDIN 74/74 OPT=1

```

60      CALL READMS(2,INFO2,300,52)
        FIDENT =INFO2(1SEQ2,1)
        NCXS(J)=INFO2(1SEQ2,2)+.5
        NCYS(J)=INFO2(1SEQ2,3)+.5
        NCZS(J)=1
        DLTC(1,J)=INFO2(1SEQ2,4)
        DLTC(2,J)=INFO2(1SEQ2,5)
        ENDPC(1,J)=INFO2(1SEQ2,6)
        ENDPC(2,J)=INFO2(1SEQ2,8)
        ENDPC(3,J)=INFO2(1SEQ2,7)
        ENDPC(4,J)=INFO2(1SEQ2,9)
20      CONTINUE
70      NPTS=NCXS(J)+NCYS(J)+NCZS(J)
        CALL READMS(2,CDEF(1,J),NPTS,IAERCSO(J))
100     CONTINUE
        RETURN
        END

```

```

1 SUBROUTINE AIRCRAFT
C.....
C DESCRIPTION - LEVEL 2
C FUNCTION - COMPUTES AIRCRAFT TRAJECTORY PARAMETERS
C METHOD - COMPUTES AIRCRAFT ACCELERATIONS AND VELOCITIES.
C LINEAR AND ANGULAR ACCELERATIONS ARE INTERPOLATED
C FROM A TABLE OF INPUT VALUES.
C LINEAR AND ANGULAR VELOCITY DERIVATIVES ARE SET TO
C THOSE COMPUTED FROM THE PREVIOUS INTEGRATION STEP.
C COMMUNICATIONS:
C CALLED BY: GESS
C CALLS:
C ZLININT, ROTATE
C NON-COMMON VARIABLES DEFINED: NONE
C POTENTIAL ERROR CONDITIONS: NONE
C.....
C SECTION 4 COMMON BLOCK
C.....
COMMON /IAIRCRT / TEMP, PRESSUR, ZACVEL, XPOS, YPOS,
+ ZPOS, XTAIL, YTAIL, ZTAIL, YAW,
+ PITCH, ROLL, RVEL, QVEL, PVEL,
+ WINDX, WINDY, WINDZ, XACVEL, CKPITH,
+ DENSITY, NPTSAAT, AAT(4.50), NPTSLAT, LAT(4.50),
+ IACSFGL
C.....
C SECTION 1 COMMON BLOCK
C.....
COMMON /ICONTROL / TSTART, TSTOP, ESTOP, IRESTRY, IUNITS,
+ ISEATTR, ISOSEP, IPLOT, IDIRFLG,
+ IPHASE1, IPHASE2, IPHASE3
C.....
C INTEGER
C.....
C MATRIX COMMON BLOCK
C.....
COMMON /MATRIX / DCMAC(3,3), DCMRA(3,3), DCMXA(3,3),
+ DCMSE(3,3), DCMYS(3,3), DCMTE(3,3),
+ DCMOAE(3,3), DCMOAE(3,3), DCMOAE(3,3),
+ DCMOUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31), LINECT(31), IPRCNT(31),
+ MAXREPT, MAXEVNT,
+ TERFLG, LU,
+ IDATE, HEADALT, HEADVEL,
+ HEADSR, HEADYAW, HEADPIT,
+ HEADROL, HEADWTG, BIAS,
+ REPTYPE(5,31), PRTLNGT(2), PRTWGHT(2),
+ IHEADER(24), TEVENTS(38),
+ IMVOC, PRTEMP( 2),
+ PRIMASS(2), PRINDX,
+ ZVECT(3), XYZ(3),
+ XACCEL(3), YACCEL(3),
+ REPTYPE, BIAS,
C.....
C INTEGER
C.....
PRTWGHT
+

```

```

60      + PRTEM          . PRIMASS          . PRINDX
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
    COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAUJOA(193) , TRAUSO(193) ,
+     TRAJSA(193) , TRAJAC(193) , TVCEQS(225) , TRAJCH(97,3) ,
+     QUATSA(65) , QUATOAI(65) , QUATSD(65) ,
+     INTSTP , IPCPASS , IRKPASS ,
+     IPOINTS , IYX , IYPRX ,
+     IKX , IKSUMX , IKPASSX ,
+     IYIX , IYIIX ,
+     IYIX , IYPRIX ,
+     IYRI2X , IPYIX ,
+     ICYIX , ICYIIX , IREIN
C .....

75      IF(IEVENTS(32) .GE. 1) GOTO 500
        IF(INTSTP.EQ.O)GOTO 5
C .....
C PROGRAM ALLOWS SIMULATION OF AIRCRAFT SLIDING
C AIRCRAFT TRACING CEASES UPON GROUND IMPACT, OR A CHANGE IN
C DIRECTION. HENCE, A VERY SMALL ALTITUDE MAY BE SPECIFIED
C IN CONJUNCTION WITH A DECELERATION TABLE.
C .....
        IF(TRAJAC(4).GE.O.O)GOTO 5
        IF(ABS(TRAJAC(14)) .GE. O.5) GOTO 5
        IF(ABS(TRAJAC(15)) .GE. O.5) GOTO 5
        IEVENTS(32)=1
        TIMES(32)=TIME
        TRAJAC(1) = O.O
        GOTO 500
    5 CONTINUE
C .....
C SET UP ACCELERATIONS IN ACS
C .....
C IACSFLG: MEANING:
C   0 CONSTANT AIRCRAFT LINEAR ACCELERATION IN ACS;
C       NO ANGULAR MOTION
C   1 CONSTANT AIRCRAFT LINEAR ACCELERATION IN ACS;
C       WITH ANGULAR MOTION
C   2 VARIABLE AIRCRAFT LINEAR ACCELERATION IN EFCS;
C       NO ANGULAR MOTION
C   3 VARIABLE AIRCRAFT LINEAR ACCELERATION IN EFCS;
C       WITH ANGULAR MOTION
C .....
        IF(NPTSLAT .EQ. O) GOTO 30
        IF(NPTS LAT GT. 1) GOTO 10
        IF(IACSFLG LT. 3) GOTO 20
        TRAJAC(17) = LAT(2,1)
        TRAJAC(18) = LAT(3,1)
        TRAJAC(19) = LAT(4,1)
        GOTO 20
    10 CONTINUE
        CALL ZLININT(TIME ,LAT,NPTS LAT,50,TRAJAC(17),4)
    20 CONTINUE
        IF(IACSFLG .GE. 2) CALL ROTATE(TRAJAC(17),TRAJAC(17)+ZVECT(1)).

```

```

115      + DCMAC,0)
      C
      30 CONTINUE
      IF(NPTSAAT .GT. 1) CALL ZLININT(TIME,AAT,NPTSAAT,50,TRAJAC(23),4)
      C
      120      C.....
      C COMPUTE VELOCITY COMPONENTS IN EFCS
      C.....
      C
      40 CONTINUE
      CALL ROTATE(TRAJAC(5),TRAJAC(14),ZVECT(1),DCMAE,1)
      TRAJAC(20) = TRAJAC(11)
      TRAJAC(21) = TRAJAC(12)
      TRAJAC(22) = TRAJAC(13)
      500 CONTINUE
      RETURN
      END
130

```

83/11/07 09 41.53

FTN 4 6+428

74/74 OPT=1

```

1 SUBROUTINE ATMOS(ALTITUD,OLDALTT,PRESAL)
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - UPDATES AMBIENT CONDITIONS
C METHOD - ATMOS IS CALLED AT EVERY TIME STEP TO CALCULATE
C TO THE SEAT/OCCUPANT, OCCUPANT ALONE, AND SEAT ALONE.
C THE CURRENT VALUE FOR ATMOSPHERIC DENSITY, RELEVANT
C THE CHANGE IN TRAJECTORY HEIGHT IS USED TO COMPUTE
C THE CURRENT PRESSURE ALTITUDE, STANDARD DAY
C TEMPERATURE, AMBIENT TEMPERATURE, AND STANDARD DAY
C PRESSURE, WHICH ARE THEN USED TO CALCULATE THE
C ATMOSPHERIC DENSITY.
C COMMUNICATIONS -
C CALLED BY:
C SEATDCC, OCCALN, SEATALN
C CALLS:
C NONE
C NON-COMMON VARIABLES DEFINED:
C ALTITUD - CURRENT ALTITUDE OF THE SEAT/OCCUPANT, OCCUPANT
C ALONE, OR SEAT ALONE
C DHT - CHANGE IN TRAJECTORY HEIGHT SINCE THE LAST CALL BY
C THE RELEVANT CALLING SUBROUTINE
C OLDALTT - LATEST PREVIOUS ALTITUDE OF THE SEAT/OCCUPANT,
C OCCUPANT ALONE, OR SEAT ALONE
C PWR - VALUE OF A POWER
C TSTD - STANDARD DAY TEMPERATURE IN DEGREES RANKINE AT
C CURRENT PRESSURE ALTITUDE
C POTENTIAL ERROR CONDITIONS:
C DIVISION BY 0 (ZERO) - RHDS = 0.0023769 * (PSTD/1013.25)**
C (518.688/TEMPS) - A MESSAGE IS PRINTED AND THE RUN IS
C TERMINATED
C.....
C SECTION 1 COMMON BLOCK
C.....
COMMON /ICONTROL / TSTART, TSTOP, ESTOP, IRESTR, IUNITS,
+ ISEATT, ISOSEP, IPLOT, IDIRFLG,
+ IPHASE1, IPHASE2, IPHASE3
+ ISTOP
C.....
C INTEGER
C.....
C DENSITY COMMON BLOCK
C.....
COMMON /DENSITY / IATMOS, OLDALTT(3), RHOS,
+ TEMPS,
+ VFWIND, VFWIND, VFWIND, VFWIND
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECI(31), LINECT(31), IPRICNT(31),
+ MAXLINE, MAXREPT, MAXEVT,
+ TEVLIN, TERFLG, LU,
+ IDATE, HEADALT, HEADVEL,
+ HEADSR, HEADYAW, HEADPIT,
+ HEADROL, HEADWTG, BIAS,
+ REPTYPE(5,31), PRITNGT(2), PRITWTG(2),
+ IHEADER(24), IEVENTS(38), TIMES(38)

```


5-35

```

1 SUBROUTINE CATAFM
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - COMPUTES THE CATAPULT FORCES AND MOMENTS ACTING ON
C THE SEAT/OCCUPANT DURING THE CATAPULT STAGE OF THE
C EJECTION.
C METHOD - THRUST FOR UP TO TWO CATAPULTS IS CONSIDERED
C BETWEEN THE TIME OF CATAPULT IGNITION AND THE TIME
C WHEN THE SEAT BOTTOM CLEARS THE END OF THE CATAPULT
C STROKE. THRUST VECTORS FROM THE INPUT TABLES ARE
C CONVERTED TO THE SEAT COORDINATE SYSTEM, AND CATAPULT
C FORCES AND MOMENTS ARE CALCULATED.
C COMMUNICATIONS -
C CALLED BY: SEATOCC
C CALLS:
C ZLININT
C NON-COMMON VARIABLES DEFINED:
C TSTAR(1) - TIME THAT HAS ELAPSED SINCE CATAPULT(1) IGNITION
C CTC(1) - THRUST VALUES FROM THE CATAPULT(1) INPUT THRUST TABLE
C POTENTIAL ERROR CONDITIONS - NONE
C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
COMMON /FORCES / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+ FXTUBSO , FYTUBSO , FZTUBSO ,
+ FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+ FXRKS(6) , FYRKS(6) , FZRKS(6) ,
+ FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+ FXAESO , FYAESO , FZAESO ,
+ FXDRISO , FYDRISO , FZDRISO
C.....
C SECTION 9 COMMON BLOCK
C.....
COMMON /ICATPLT / INCAT , CATLNT(2) , CATSIK(2) , TCI (2) ,
+ XPOSAP(2) , YPOSAP(2) , ZPOSAP(2) , NPTSCT(2) ,
+ CATHRST(2,25,2) , ITUBEND , KTUBE , CTUBE ,
+ PTUBE , MUTUBE , EXTNGT , ICATOUT
C.....
C REAL
C.....
C SECTION 1 COMMON BLOCK
C.....
COMMON /ICONTRI / TSTART , TSTOP , ESTOP , IRESTRT , IUNITS ,
+ ISEATR , ISOSEP , IPLOT , IDRI , LG ,
+ IPHASE1 , IPHASE2 , IPHASE3
C.....
C INTEGER
C.....
C SECTION 3 COMMON BLOCK
C.....
COMMON /IDELTAT / DIPHAS1 , DIPHAS2 , DIPHAS3
C.....
C SECTION 2 COMMON BLOCK
C.....
COMMON /IREPORT / IREPTS(31) , PRIFRQ , P11 , P12 , P13
C.....
C INTEGER
C.....
PRIFRQ , P11 , P12 , P13
C.....

```


SUBROUTINE CATAPM 74/74 OPT=1 FTN 4.61428 83/11/07 09.41 53 PAGE 37

```

115      +      TLCHSO(3) , TMCHSO(3) , TNCCHSO(3) ,
      +      TLAESO , TMAESO , TNAESO ,
      +      TLDRTSO , TMORTSO , TNDRTSO
C.....
120      C
      C DIMENSION TSTAR(2)
C.....
C RETURN IF NO CATAPULTS
C.....
125      IF (INCAT .EQ. ICATOUT) GO TO 5000
      DO 500 I = 1, INCAT
C.....
C CHECK FOR CATAPULT IGNITION
C.....
130      IF (EVENTS(I+2) .GE. 1) GO TO 500
      IF (EVENTS(I) .GE. 1) GO TO 100
      IF (INTSTP .EQ. 0) GO TO 500
      IF (TIMES .LT. TCI(I)) GO TO 500
C.....
C CATAPULT IGNITION
C.....
135      TIMES(I) = TIME
      IEVENTS(I) = 1
100 CONTINUE
      IF (IPHASE2 .GT. 0) GO TO 110
      L = IFIX(1.E6 * TIMES)
      K = IFIX(1.E6 * (DTPHAS1+5.E-10))
      IF (MOD(L,K) .NE. 0) GO TO 150
      IPHASE2 = 1
      DELTAT = DTPHAS1
110 CONTINUE
      IF (IPHASE2 .GT. 1) GO TO 150
      L = IFIX(1.E6 * TIMES)
      K = IFIX(1.E6 * (DTPHAS1+PI*1+5.E-10))
      IF (MOD(L,K) .NE. 0) GO TO 150
      PRIFRQ = PI1
      DO 125 J = 1, 31
      IPRCNT(J) = PRIFRQ - 1
125 CONTINUE
      IPHASE2 = 2
150 CONTINUE
C.....
C CHECK FOR CATAPULT SEPARATION
C.....
160      IF (ZRRSBOI .LT. CATSTK(I) .OR. (INTSTP .EQ. 0)) GO TO 200
C.....
C CATAPULT SEPARATION
C.....
      TIMES(I+2) = TIME
      IEVENTS(I+2) = 1
      ICATOUT = ICATOUT + 1
      FYCASO(I) = FYCASO(I) + FZCASO(I) + 0.0
      TLCASO(I) = TMCASO(I) + TNCASO(I) + 0.0
      GO TO 900
170      C.....
      C COMPUTE CATAPULT THRUST ON SEAT/OCCUPANT COMBINATION

```

```

C.....
200 CONTINUE
  TSTAR(1) = TIME - TCI(1)
  CALL ZLININT(TSTAR(1),CATHRST(1,1,1),NPISCT(1),25,CTC,2)
C.....
C TRANSFORM THRUST VECTORS TO SCS
C.....
  FZCASO(1) = CTC
C.....
C COMPUTE CATAPULT MOMENTS
C.....
  TLCASO(1) = YSSOCA(1) + FZCASO(1) - ZSSOCA(1) + FYCASO(1)
  TMCASO(1) = ZSSOCA(1) + FXCASO(1) - XSSOCA(1) + FZCASO(1)
  TNCASO(1) = XSSOCA(1) + FYCASO(1) - YSSOCA(1) + FXCASO(1)
  500 CONTINUE
C
  5000 CONTINUE
  RETURN
  END

```

```

1      SUBROUTINE CHUINIT
C.....
C DESCRIPTION - LEVEL 4
C FUNCTION - CALCULATES THE INITIAL VELOCITY AND POSITION COMPONENTS*
C OF THE DROGUE SLUG/CONTAINER
C METHOD - AT DROGUE PROJECTION, INITIAL VALUES FOR POSITION
C COORDINATES AND VELOCITY COMPONENTS OF THE DROGUE SLUG/
C CONTAINER ARE COMPUTED IN THE EFCS
C COMMUNICATIONS -
C CALLED BY:
C DROGUE1
C DROGUE2
C CALLS:
C ROTATE
C NON-COMMON VARIABLES DEFINED - NONE
C POTENTIAL ERROR CONDITIONS - NONE
C.....
C MATRIX COMMON BLOCK
C.....
COMMON /MATRIX / DCMAR(3,3), DCMRA(3,3), DCMRA(3,3), DCMRA(3,3),
+ DCMSE(3,3), DCMTS(3,3), DCMTE(3,3),
+ DCMSE(3,3), DCMTE(3,3), DCMSE(3,3), DCMTE(3,3),
+ DCMDSUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31), LINECT(31), IPRCNT(31),
+ MAXLINE, MAXREPT, LU,
+ IEVLNE, IERRFLG, LU,
+ IDATE, HEADALT, HEADVEL,
+ HEADSR, HEADYAW, HEADPIT,
+ HEADROL, HEADWGT, BIAS,
+ REPTYPE(5,31), PRTLNCT(2), PRTWGT(2),
+ IHEADER(24), IEVENTS(38),
+ IMVBO,
+ PRIMASS(2), PRTINDX, PKZVEL,
+ ZVECT(3), XYZ(4), SAVTIME,
+ XACCEL(3), YACCEL(3), ZACCEL(3),
+ REPTYPE, BIAS, PRTLNCT,
+ PRTWGT, PRTINDX,
+ PRTWGT, PRTINDX,
+ PRTWGT, PRTINDX
C.....
C MOMARMS COMMON BLOCK
C.....
COMMON /MOMARMS /
+ REFLNSO, REFLNOA, REFLNSA, URX(6), URY(6), URZ(6),
+ XSSOCA(2), YSSOCA(2), ZSSOCA(2), XSSORK(6), YSSORK(6), ZSSORK(6),
+ XSSORRE, YSSORRE, ZSSORRE, XSSOLRE, YSSOLRE, ZSSOLRE,
+ XSSOMRE, YSSOMRE, ZSSOMRE, XSSOBOT, YSSOBOT, ZSSOBOT,
+ XSSOHB(6), YSSOHB(6), ZSSOHB(6), XRRCSAC, YRRCSAC, ZRRCSAC,
+ XSSCSAC, YSSCSAC, ZSSCSAC, XSSOSRP, YSSOSRP, ZSSOSRP,
+ XSSASRP, YSSASRP, ZSSASRP, XRRDAP(2), YRRDAP(2), ZRRDAP(2),
+ XRRSBO(6), YRRSBO(6), ZRRSBO(6), XSSDCP(2), YSSDCP(2), ZSSDCP(2),
+ XSSDAP(2), YSSDAP(2), ZSSDAP(2), XESDAC, YESDAC, ZESDAC,
+ XSRCSAC, YSRCSAC, ZSRCSAC, XSSOAC, YSSOAC, ZSSOAC

```

NADC-84068-60

[illegible]

```

115 C.....
C TRANSFORM INITIAL VELOCITY COMPONENTS TO EFCS
C.....
C
CALL ROTATE(TRAJCH(5,1),TRAJCH(5,1),ZVECT(1),DCMSE,1)
GOTO 20

120 C
10 CONTINUE
TRAJCH(5,1) = TRAJSO(14)
TRAJCH(6,1) = TRAJSO(15)
TRAJCH(7,1) = TRAJSO(16)
TVEL = SORT(TRAJSO(14) + TRAJSO(14) + TRAJSO(15) + TRAJSO(15) +
+ TRAJSO(16) + TRAJSO(16))
CALL ZLININT(TVEL,DROGLS,NPTSOLS,25,TDROGLS,2)
C.....
C TRANSFORM VECTOR FROM S/O CG TO DROGUE ATTACHMENT POINT, FROM
C SCS TO EFCS
C.....
C
20 CONTINUE
XYZ(1) = XSSOCH(1)
XYZ(2) = YSSOCH(1)
XYZ(3) = ZSSOCH(1)
CALL ROTATE(XYZ(1),XYZ(1),ZVECT(1),DCMSE,1)
C.....
C COMPUTE INITIAL POSITION COORDINATES IN EFCS
C.....
C
TRAJCH(2,1) = TRAJSO(2) + XYZ(1)
TRAJCH(3,1) = TRAJSO(3) + XYZ(2)
TRAJCH(4,1) = TRAJSO(4) + XYZ(3)
C
RETURN
END

```



```

1 SUBROUTINE CHUTES
C.....
C DESCRIPTION - LEVEL 2
C FUNCTION - CONTROLS THE SEQUENCING OF CALLS TO THE RECOVERY
C CHUTE AND DROGUE CHUTE ROUTINES
C COMMUNICATIONS -
C CALLED BY:
C OCCALON, SEATOCC
C CALLS:
C DROGUE1
C DROGUE2
C RECOV
C NON-COMMON VARIABLES DEFINED - NONE
C POTENTIAL ERROR CONDITIONS - NONE
C.....
C SECTION 14 COMMON BLOCK
C.....
COMMON /PARCHUT / IRECOV, RECDRAG, XRECAP, NPTSRLS,
+ NPTSRTFT, IDROGUE, POROSD2, NPTDFT2, NPTDFT1, NPTSOLS,
+ DISPLY, DROGPD1, DROVELY, YDROGAP, CHALT2, AREADC,
+ TFP2, CDDC
+ TRDPLOY, RECOVLL, RECOVPD, POROSR, YRECAP, ZRECAP,
+ RECOVLS(2,25), IFTRCV, RECOVFT(2,25), SEPRCE,
+ DRDRAG2, DRDRO2, VELCON, DROGFT2(2,25), IFTRD01,
+ DROGFT1(2,25), IDROGLS, DRDGLL, DRDROG1, DROVELX,
+ DROGPD1, DROVELZ, ZDROGAP, CHALT1, TDELAY, TFP1,
+ TIDROGLS, RECOVD1(2,25), NPTSRTD1
C
C IF (IRECOV .GT. 0) CALL RECOV
C IF (IDROGUE .EQ. 1) CALL DROGUE1
C IF (IDROGUE .EQ. 2) CALL DROGUE2
C
C RETURN
C END

```

83/11/07 09.41.53

FIN 4 6+428

74/74 OPT=1

```

1  SUBROUTINE CHUTEFM(ICHUTE,DP)
2  C.....
3  C DESCRIPTION - LEVEL 4
4  C FUNCTION - CALCULATES PARACHUTE FORCES AND MOMENTS
5  C METHOD - THIS ROUTINE COMPUTES TOTAL PARACHUTE FORCE USING
6  C INPUT VALUES FOR PARACHUTE DIAMETER AND DRAG
7  C COEFFICIENT, AND CALCULATED VALUES FOR TOTAL VELOCITY
8  C AND CHUTE PROJECTED AREA. THE CHUTE IS THEN ALIGNED
9  C WITH THE WIND TO COMPUTE COMPONENT FORCES AND MOMENTS.
10 C ACCELERATIONS AND VELOCITIES OF THE CHUTE ARE SET
11 C EQUAL TO THOSE OF THE SEAT/OCCUPANT, OR SEAT ALONE
12 C FOLLOWING SEAT/OCCUPANT SEPARATION.
13 C COMMUNICATIONS -
14 C CALLED BY:
15 C DROGUE1
16 C DROGUE2
17 C RECOV
18 C
19 C CALLS:
20 C ROTATE, ZARCTAN
21 C NON-COMMON VARIABLES DEFINED -
22 C DIAFRAC - FRACTION OF THE CHUTE DIAMETER THAT IS EXTENDED
23 C AREACHU - FRACTION OF THE CHUTE AREA THAT IS EXPOSED
24 C CD - DRAG COEFFICIENT
25 C R - TOTAL VELOCITY OF THE CHUTE
26 C PFORCE - TOTAL PARACHUTE FORCE
27 C ANG1 - ANGLES USED IN ALIGNING THE CHUTE WITH THE WIND
28 C ANG2 -
29 C
30 C POTENTIAL ERROR CONDITIONS - NONE
31 C.....
32 C
33 C CONSTANTS COMMON BLOCK
34 C.....
35 C COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
36 C DENSITY COMMON BLOCK
37 C.....
38 C COMMON /DENSITY / IATMOS , OLDALT(3) , RHOS
39 C + PRESALT(3) , DTEMP ,
40 C + VWIND , VYWIND , VZWIND
41 C
42 C OCCUPANT ALONE FORCES COMMON BLOCK
43 C.....
44 C COMMON /FORCEA / FXCHOA(3) , FYCHOA(3) , FZCHOA(3) ,
45 C + FXAEOA , FYAEOA , FZAEOA
46 C
47 C SEAT/OCCUPANT FORCES COMMON BLOCK
48 C.....
49 C COMMON /FORCEO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
50 C + FXTUBSO , FYTUBSO , FZTUBSO
51 C + FXSLSO(6) , FYSLSO(6) , FZSLSO(6) ,
52 C + FXRKS(6) , FYRKS(6) , FZRKS(6) ,
53 C + FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
54 C + FXAESO , FYAESO , FZAESO ,
55 C + FXDRTSO , FYDRTSO , FZDRTSO
56 C.....

```

60	C MATRIX COMMON BLOCK C COMMON /MATRIX / DCMAE(3,3) . DCMRA(3,3) . DCMSA(3,3) . + DCMSE(3,3) . DCMTS(3,3) . DCMTE(3,3) . DCMSAE(3,3) . DCMOAE(3,3) . DCMRSR(3,3) . + DCDUM(3,3) C C MISCELLANEOUS DATA COMMON BLOCK C COMMON /MSC / IPAGECT(31) . LINECT(31) . IPRCNT(31) . + MAXLINE . MAXREPT . MAXEVNT . + IEVLNE . TERRFLG . LU . + IDATE . HEADALT . HEADVEL . + HEADSR . HEADYAW . HEADPIT . + HEADROL . HEADWGT . BIAS . + REPTYE(5,31) . PRTLNCT(2) . PRTWGHT(2) . + IHEADER(24) . IEVENTS(38) . TIMES(38) . + IMVDC . PRTEMP(2) . + PRIMASS(2) . PRINDX . PKZVEL . + ZVECT(3) . XYZ(3) . SAVTIME . + XACCEL(3) . YACCEL(3) . + INTEGER . REPTYE . ZACCEL(3) . + PRTWGHT . BIAS . PRTLNCT . + PRTEMP . PRIMASS . PRINDX . C C MOMARMS COMMON BLOCK C COMMON /MOMARMS / +REFLNSO . REFLENDA . REFLESA . URX(6) . URY(6) . URZ(6) . +XSSOCA(2) . YSSOCA(2) . ZSSOCA(2) . XSSORK(6) . YSSORK(6) . ZSSORK(6) . +XSSORRE . YSSORRE . ZSSORRE . XSSOLRE . YSSOLRE . ZSSOLRE . +XSSOMRE . YSSOMRE . ZSSOMRE . XSSOBOT . YSSOBOT . ZSSOBOT . +XSSOSB(6) . YSSOSB(6) . ZSSOSB(6) . XRCSAC . YRCSAC . ZRCSAC . +XSSCSAC . YSSCSAC . ZSSCSAC . XSSOSRP . YSSOSRP . ZSSOSRP . +XSSASRP . YSSASRP . ZSSASRP . XRRDAP(2) . VRRDAP(2) . ZRRDAP(2) . +XRRSBO(6) . YRRSBO(6) . ZRRSBO(6) . XSSOCP(2) . YSSOCP(2) . ZSSOCP(2) . +XSSDAP(2) . YSSDAP(2) . ZSSDAP(2) . XESOAC . YESOAC . ZESOAC . +XSRCAC . YSRCAC . ZSRCAC . XSSOAC . YSSOAC . ZSSOAC . +XRSDSB . YRSDSB . ZRSDSB . XRSBOT . YRSBOT . ZRSBOT . +XRRSB . YRRSB . ZRRSB . XSSOCH(3) . YSSOCH(3) . ZSSOCH(3) . +XAACSBO . YAACSBO . ZAACSBO . XASOAC . YASOAC . ZASOAC . +XAACSBO . YAACSBO . ZAACSBO . XASOAC . YASOAC . ZASOAC . +XRSOAC . YRSOAC . ZRSOAC . XSCCAP(2) . YSCCAP(2) . ZSCCAP(2) . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2,25) . SEPRFCE . + IDROQUE . DRDRAG2 . DRGPD2 . + PORUSD2 . VELCON . IFDR02 . + NPTDF12 . DRGGFT2(2,25) . IFIDR01 . + NPTDF11 . DRGGFT1(2,25) . IDRGLS . + NPTDLS . DRQGLS(2,25) . TDDPLOY . + DISPLY . DRQGLL . DRDGL1 . C C SECTION 14 COMMON BLOCK C COMMON /PARACHUT / IRECOV . TRDPLOY . RECOVLL . + RECDRAG . RECOVERPD . POROSR . + XRECAP . YRECAP . ZRECAP . + NPTSRLS . RECVLS(2,25) . IFTRCV . + NPTSRTT . RECVFT(2

5-46

```

20 IF(IEVENTS(26) .EQ. 0) DIAFRAC = DIAFRAC + ((TIME-TIMES(25))/1FP3)
   CD = RECDRAG

```

```

175 C
25 CONTINUE

```

```

C AREACHU = .25 * PI * DIAFRAC * DIAFRAC

```

```

180 C IF(IEVENTS(28) .GE. 1) GOTO 50
   XVEL = TRAUJO(14) - WINDX
   YVEL = TRAUJO(15) - WINDY
   ZVEL = TRAUJO(16) - WINDZ
   GOTO 75

```

```

185 C 50 CONTINUE
   XVEL = TRAUJOA(5) - WINDX
   YVEL = TRAUJOA(6) - WINDY
   ZVEL = TRAUJOA(7) - WINDZ
75 CONTINUE

```

```

190 C R = SQRT(XVEL * XVEL + YVEL * YVEL + ZVEL * ZVEL)
   IF(R .EQ. 0.0) GOTO 100

```

```

C SIN2 = ZVEL/R
  COS2 = COS(ASIN(SIN2))
  BETA = ZARCTAN(YVEL,XVEL)

```

```

195 C PFORCE = .5 * RHOS * R * R * CD * AREACHU
   IF(IEVENTS(28) .GE. 1) GOTO 80

```

```

200 C .....
C CALCULATE CHUTE FORCES, ROTATE FORCE ARRAY, AND ASSIGN FORCE PARAMETERS *
C .....
   XYZ(1) = -SIGN((ABS(PFORCE * COS2 * COS(BETA))), XVEL)
   XYZ(2) = -SIGN((ABS(PFORCE * COS2 * SIN(BETA))), YVEL)
   XYZ(3) = -SIGN((ABS(PFORCE * SIN2)), ZVEL)

```

```

C CALL ROTATE(XYZ(1),XYZ(1),ZVECT(1),DCMSE,0)

```

```

210 C FXCHSO(1CHUTE) = XYZ(1)
   FYCHSO(1CHUTE) = XYZ(2)
   FZCHSO(1CHUTE) = XYZ(3)

```

```

C TLCHSO(1CHUTE) = YSSOCH(1CHUTE) * FZCHSO(1CHUTE) - ZSSOCH(1CHUTE)
  + FYCHSO(1CHUTE)
  TMCHSO(1CHUTE) = ZSSOCH(1CHUTE) * FXCHSO(1CHUTE) - XSSOCH(1CHUTE)
  + FZCHSO(1CHUTE)
  TNCHSO(1CHUTE) = XSSOCH(1CHUTE) * FYCHSO(1CHUTE) - YSSOCH(1CHUTE)
  + FXCHSO(1CHUTE)

```

```

220 C .....
C SET VELOCITIES OF CHUTE EQUAL TO THOSE OF S/O
C .....

```

```

225 C TRAJCH(5,1CHUTE) = TRAUJO(14)
   TRAJCH(6,1CHUTE) = TRAUJO(15)
   TRAJCH(7,1CHUTE) = TRAUJO(16)
   GOTO 500
80 CONTINUE

```

```

230 C.....
C CALCULATE CHUTE FORCES, ROTATE FORCE ARRAY, AND ASSIGN FORCE PARAMETERS *
C.....
      FXCHOA(ICHUTE) = -SIGN((ABS(PFORCE* $\cos^2(\text{BETA})$ )).XVEL)
      FYCHOA(ICHUTE) = -SIGN((ABS(PFORCE* $\cos^2(\text{BETA})$ )).YVEL)
      FZCHOA(ICHUTE) = -SIGN((ABS(PFORCE* $\sin^2$ )).ZVEL)

235 C
C      TLCHOA(ICHUTE) = YSSOCH(ICHUTE) * FZCHOA(ICHUTE) -
C      + ZSSOCH(ICHUTE) * FYCHOA(ICHUTE)
C      TMCHOA(ICHUTE) = ZSSOCH(ICHUTE) * FXCHOA(ICHUTE) -
C      + XSSOCH(ICHUTE) * FZCHOA(ICHUTE)
C      TNCHOA(ICHUTE) = XSSOCH(ICHUTE) * FYCHOA(ICHUTE) -
C      + YSSOCH(ICHUTE) * FXCHOA(ICHUTE)
C
C.....
245 C SET VELOCITIES OF CHUTE EQUAL TO THOSE OF THE OCCUPANT ALONE *
C.....
C
      TRAJCH(5,ICHUTE) = TRAJOA(5)
      TRAJCH(6,ICHUTE) = TRAJOA(6)
      TRAJCH(7,ICHUTE) = TRAJOA(7)
      GOTO 500

C
100 CONTINUE
      WRITE(5,110)
110 FORMAT(2X, //72(1H*))/4X, "FATAL ERROR(SUBROUTINE CHUTFM)*** ", /,
      + "R EQUAL TO ZERO RESULTS IN DIVISION BY ZERO", /,72(1H*)
      IERRFLG = 1

C
500 CONTINUE
      RETURN
      END

```

```

1  SUBROUTINE CLEAR
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - CLEARS MEMORY (SETS TO 0) BEFORE EACH RUN, AS PART
C OF THE INITIALIZATION PROCESS
C COMMUNICATIONS -
C CALLED BY:
C INTLZ
C CALLS:
C NONE
C NON-COMMON VARIABLES DEFINED
C TO BE DEFINED
C POTENTIAL ERROR CONDITIONS:
C TO BE DEFINED
C .....
C COEFFICIENTS (USED IN SUBROUTINE AEROIN) COMMON BLOCK
C .....
COMMON /COEF / COEF(700,6)
EQUIVALENCE (VCOEF(1), COEF(1,1))
DIMENSION VCOEF(4200)
C .....
C CONSTANTS COMMON BLOCK
C .....
COMMON /CONST / GRAVITY, RADDEG, DEGRAD, PI
EQUIVALENCE (VCONST(1), GRAVITY)
DIMENSION VCONST(4)
C .....
C DENSITY COMMON BLOCK
C .....
COMMON /DENSITY / IATMOS, OLDALT(3), RHOS
+ PRESALT(3), DTEMP,
+ TEMPS,
+ VZWIND, VYVWIND, VZWIND
EQUIVALENCE (VDENS(1), OLDALT(1))
DIMENSION VDENS(12)
C .....
C SECTION 13 COMMON BLOCK
C .....
COMMON /DYNCGIN / IDYNGC, WY, WYV, SXN
+ CX, XSLACK, SXP, SY, CZ, ZSLACK
+ CY, ZBOT, SZN1, SZN2
+ SZP
EQUIVALENCE (VDYN(1), W/)
DIMENSION VDYN(14)
C .....
C DYNAMIC CG VARIABLES COMMON BLOCK
C .....
COMMON /DYNCGVB / CGVAL(6), CGDERV(6),
+ XCGDAO, YCGDAO, ZCGDAO
+ DXCG, DYCG, DZCG
EQUIVALENCE (VDYNV(1), XCGDAO)
DIMENSION VDYNV(6)
C .....
C DYNAMIC RESPONSE INDEX VARIABLES COMMON BLOCK
C .....
COMMON /DRIVB / DRIVAL(2), DRIDERV(2), ACCEL1

```

5-50


```

175      + XPOSRK(6), YPOSRK(6), ZPOSRK(6),
      + RKALPH(6), RKBETA(6), RKGAMA(6), RKTHIRST(2,25,6)
      + INTEGER
      + RKNPTS
      + EQUIVALENCE (LIROCK , INRKT)
      + EQUIVALENCE (KIROCK(1) , RKNPTS(1))
      + EQUIVALENCE (KIROCK(1) , RKDELY(1))
      + EQUIVALENCE (VIROCK(1) , RKIGN(1))
      + DIMENSION KIROCK(6) , VIROCK(354) , RIROCK(6)
      + C*****
180      C SECTION 6 COMMON BLOCK
      C*****
      + COMMON /ISEATOC / IPCNTL , XCGSO , YCGSO , ZCGSO , IXXSO ,
      + IYXSO , IYZSO , IYZSO , IYZSO , IZZSO ,
      + AREASO , ARFADA , WHTOAB , WHTOAA ,
      + IXXOA , IXYOA , IXZOA , IYXOA , IYXOA , IYZOA ,
      + IZZOA , XCGOA , YCGOA , ZCGOA , SOSEP ,
      + REAL
      + IXXSO , IXYSO , IXZSO , IYXSO , IYXSO , IYZSO ,
      + IZZSO , IXXOA , IXYOA , IXZOA , IYXOA , IYXOA ,
      + IYZOA , IZZOA , XCGSO
      + EQUIVALENCE (VISEATO(1) , XCGSO)
      + DIMENSION VISEATO(24)
      + C*****
195      C SECTION 5 COMMON BLOCK
      C*****
      + COMMON /ISETALN / XPOSSRP , YPOSSRP , ZPOSSRP , XCGSA , YCGSA ,
      + ZCGSA , IXXSA , IYXSA , IXZSA , IYXSA , IYXSA ,
      + IYZSA , IZZSA , PHISA , PSISA , THESA ,
      + AREASA , HIGHTSA , WHTSAB , XPOSBOT , YPOSBOT ,
      + ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS ,
      + REAL
      + IXXSA , IYXSA , IXZSA , IYXSA , IYXSA ,
      + IZZSA
      + EQUIVALENCE (VISEATA(1) , XPOSSRP)
      + DIMENSION VISEATA(24)
      + C*****
205      C SECTION 12 COMMON BLOCK
      C*****
      + COMMON /ITVCIN / ITVC , MPH1 , MPST , MTHE ,
      + ROLLR , PITCHRL , SMPLRAT , TVCDLAY ,
      + RKANG
      + REAL
      + MPH1 , MPST , MTHE
      + EQUIVALENCE (VITVC(1) , MPH1)
      + DIMENSION VITVC(8)
      + C*****
215      C MASSES COMMON BLOCK
      C*****
      + COMMON /MASSES / MASSOA1 , MASSOA2 , MASSSO , MASSO ,
      + MASSA , MASSRK(6) , MASSDC , MASSO ,
      + REAL
      + MASSOA1 , MASSOA2 , MASSO , MASSO ,
      + MASSA , MASSRK , MASSDC , MASSDC
      + EQUIVALENCE (VMAS(1) , MASSOA1)
      + DIMENSION VMAS(11)
      + C*****
220      C MATRIX COMMON BLOCK
      C*****
      + COMMON /MATRIX / DCMAE(3,3) , DCMA(3,3) , DCMSA(3,3) ,
      + DCNSE(3,3) , DCMTS(3,3) , DCMT(3,3) ,
      + DCNSAE(3,3) , DCNOAE(3,3) , DCMSR(3,3) ,
      +

```

```

230      + EQUIVALENCE (VMATR(1), DCMAT(1,1))
      DIMENSION VMATR(90)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
      COMMON /MISC / IPAGECT(31), LINECT(31), MAXREPT, MAXEVNT,
      + MAXLINE, LEVLINE, IERRFLG, LU,
      + IDATE, HEADAL, HEADVEL, HEADPT,
      + HEADSR, HEADYAW, HEADWGT, BIAS,
      + REPTYPE(5,31), PRTLNGT(2), PRTWGT(2),
      + IHEADER(24), IEVENTS(38), TIMES(38),
      + IMVDC, PRTIMP(2),
      + PRIMASS(2), PRTINDX, PKZVEL,
      + ZVECT(3), XYZ(3), SAVTIME,
      + XACCEL(3), YACCEL(3),
      + INTEGER REPTYPE, BIAS, PRTLNGT,
      + PRTWGT, PRTIMP, PRIMASS, PRTINDX
235      EQUIVALENCE (KMISC(1), IPAGECT(1))
      EQUIVALENCE (VMISC(1), HEADAL)
      EQUIVALENCE (VMISC(2), IHEADER(1))
      EQUIVALENCE (VMISC(3), PKZVEL)
      EQUIVALENCE (VMISC(4), LEVLINE)
      DIMENSION KMISC(93), VMISC(7), KMISC2(101), VMISC2(17)
240      + KMISC3(4)
C.....
C MOMARMS COMMON BLOCK
C.....
      COMMON /MOMARMS /
      +REFLNSO,REFLNSA,REFLNSA,URY(6),URY(6),
      +XSSOCA(2),YSSOCA(2),ZSSOCA(2),XSSORK(6),YSSORK(6),ZSSORK(6),
      +XSSORRE,YSSORRE,ZSSORRE,XSSOLRE,YSSOLRE,ZSSOLRE,
      +XSSOMRE,YSSOMRE,ZSSOMRE,XSSOBOT,YSSOBOT,ZSSOBOT,
      +XSSOSB(6),YSSOSB(6),ZSSOSB(6),XSSCSAC,YSSCSAC,ZSSCSAC,
      +XSSCSR,YSRCSR,ZSSCSR,XSSSRP,YSRSP,ZSSSRP,
      +XSSASRP,YSASRP,ZSSASRP,XRRDAP(2),YRRDAP(2),ZRRDAP(2),
      +XRRSBO(6),YRRSBO(6),ZRRSBO(6),XSSDCP(2),YSSDCP(2),ZSSDCP(2),
      +XSSDAP(2),YSSDAP(2),ZSSDAP(2),XSSOAC,YESOAC,ZESOAC,
      +XSRCSAC,YSRCSAC,ZSRCSAC,XSSOAC,YSOAC,ZSOAC,
      +XRSOSB,YRSOSB,ZRSOSB,XRRSBO,YRRSBO,ZRRSBO,
      +XRRSB,YRRSB,ZRRSB,XSSOCH(3),YSSOCH(3),ZSSOCH(3),
      +XAACSO,YAACSO,ZAACSO,XASOAC,YASOAC,ZASOAC,
      +XRSOAC,YRSOAC,ZRSOAC,XSCAP(2),YSCAP(2),ZSCAP(2)
      EQUIVALENCE (VMOMARM(1), REFLNSO)
      DIMENSION VMOMARM(166)
245      +
C.....
C SECTION 14 COMMON BLOCK
C.....
      COMMON /PARCHIT / IRECOV,
      + RECDRAG,
      + XRECAP,
      + NPTSRLS,
      + NPTSRTF
250      + TROPLOY,
      + RECOVPL,
      + POROSR,
      + ZRECAP,
      + RECOVLS(2,25),
      + IFTRECV,
      + RECOVFT(2,25),
      + SEPRCE
255      +
260      +
265      +
270      +
275      +
280      +
285      +

```

```

+ TORQUE      . DRORAG2      . OROGP02
+ POROSD2     . VELCON       . IFIDR02
+ NPDTF12     . DROGFT2(2,25) . IFTDR01
+ NPTDF11     . DROGF11(2,25) . IDROGLS
+ NPTSOLS     . DROGLS(2,25) . TDDPL0Y
+ DISPLAY     . DROQLL        . DRORAG1
+ DROGPD1     . POROSD1      . DROVELX
+ DROVELY     . DROVEI2      . XDROGAP
+ YDROGAP     . ZDROGAP      . CHALT1
+ CHALT2      . GLIMIT       . TOELAY
+ AREADC      . WGHTRC       . TFP1
+ TFP2        . TFP3         . TDROGLS
+ CDDC        . NPTSRDT     . RECOVDT(2,25)
+ EQUIVALENCE (VPAR(1), TRDPL0Y)
EQUIVALENCE (VPAR2(1), RECVLS(1,1))
EQUIVALENCE (VPAR3(1), RECVFT(1,1))
EQUIVALENCE (VPAR4(1), DRORAG2)
EQUIVALENCE (VPAR5(1), DROGFT2(1,1))
EQUIVALENCE (VPAR6(1), DROGFT1(1,1))
EQUIVALENCE (VPAR7(1), DROGLS(1,1))
EQUIVALENCE (VPAR8(1), RECOVDT(1,1))
DIMENSION VPAR(8), VPAR2(50), VPAR3(51),
+ VPAR4(4), VPAR5(50), VPAR6(50),
VPAR7(73), VPAR8(50)
C .....
C RAIL VARIABLES COMMON BLOCK
C .....
COMMON /RAILVRB / FXR , FYR , FZR , XDISP , YDISP
EQUIVALENCE VRail(1), FXR)
DIMENSION VRail(5)
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
COMMON /RKUTTA / TIME , TIME5 , DELTAT , TRAJSO(193) ,
+ TRAJSA(193) , TRAUJA(193) , TRAJCH(97,3) ,
TRAJAC(193) , TVEQSI(225) , QUATSO(65) ,
+ QUATSA(65) , QUATOAI(66) , QUATAC(66) ,
INTSIP , IPCPASS , IRKPASS ,
+ IPOINIS , IYX , IYPRX , IYPASX ,
IKK , IKSUMX , IYI2X , IYI2X ,
+ IYIX , IYI3X , IYPRIX , IYPRIX ,
IYPR12X , IYPIX , IPVI1X , IREIN ,
+ ICYIX , ICY1X ,
EQUIVALENCE (VRKUT(1), TIME)
EQUIVALENCE (KRKUT(1), INTSIP)
DIMENSION VRKUT(1551), KRKUT(21)
C .....
C TORQUE OCCUPANT ALONE COMMON BLOCK
C .....
COMMON /TORQOA / TLCHOA(3), TMCHOA(3), TNCHOA(3),
+ TLAEOA , TMAEOA , TNAEOA
EQUIVALENCE (VTORQOA(1), TLCHOA(1))
DIMENSION VTORQOA(12)
C .....
C TORQUE SEAT ALONE COMMON BLOCK
C .....

```

```

COMMON /TOROSA / TLAESA , TMAESA , TNAESA
EQUIVALENCE (VTOROSA(1) , TLAESA)
DIMENSION VTOROSA(3)
C*****
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C*****
COMMON /TORQSO / TLRQSO(2) , TMCASO(2) , TNCASO(2) ,
+ TLTUBSO , TMTUBSO , TNSLSO(6) , TINTUBSO ,
+ TLRKSO(6) , TMRKSO(6) , TNRKSO(6) ,
+ TLRHSD(3) , TMRHSD(3) , TNRHSD(3) ,
+ TLAESO , TMAESO , TNAESO ,
+ TLDRTSO , TMDRTSO , TNDRTSO
EQUIVALENCE (VTORQSO(1) , TLRQSO(1))
DIMENSION VTORQSO(60)
C*****
C THRUST VECTOR CONTROL VARIABLES COMMON BLOCK
C*****
COMMON /TVCVRB / ITVCF(1) , CMPVAL , D3(3) , RKTOMNO(3),
+ C29 , C30 , C31 , C32 ,
+ DTH(3,2) , ANGR(3)
EQUIVALENCE (VTVCF(1) , CMPVAL)
DIMENSION VTVCF(20)
C*****
C TOF PLOTTING FILE COMMON BLOCK
C*****
COMMON /TTILES / SENSNAM(40,6) , TEXT1(6) , TEXT2(6) ,
+ TEXT3(3) , BAUD , WORDLEN ,
+ NCHANFR(2) , NSENSOR(2) , IDUMMY(40) ,
+ RECORD(35) , TIMINT ,
+ TPOINT , PLTIME(2) , NHEADER(2) ,
+ INTEGER , TEXT1 , TEXT2 , TEXT3 ,
+ BAUD , WORDLEN
C*****
C LDOOPS TO SET ALL VARIABLES TO 0 (ZERO)
C*****
DO 10 I=1,4200
VCOEF(I) = 0.0
10 CONTINUE
C
DO 20 I=1,4
VCONST(I) = 0.0
20 CONTINUE
C
IATMOS = 0
DO 30 I=1,12
VDENS(I) = 0.0
30 CONTINUE
C
IDYNCG = 0
DO 35 I=1,2
DRIVERV(I) = 0.0
35 CONTINUE
DO 40 I=1,14
VDYN(I) = 0.0
40 CONTINUE

```

```

400      DO 45 I=1,6
         VDYNV(I) = 0.0
         CGVAL(I) = 0.0
         CGDERV(I) = 0.0
         VDRI(I) = 0.0
405      45 CONTINUE
         C
         DO 50 I=1,12
            VFOA(I) = 0.0
410      50 CONTINUE
         C
         DO 60 I=1,3
            VFSA(I) = 0.0
415      60 CONTINUE
         C
         DO 70 I=1,60
            VFSD(I) = 0.0
420      70 CONTINUE
         C
            NPTSAAT = 0
            NPTSLAT = 0
            IACFLG = 0
            DO 80 I=1,200
               VAAT(I) = 0.0
               VLAT(I) = 0.0
               IF (I .GT. 21) GOTO 80
425      80 CONTINUE
            VIAIR(I) = 0.0
         C
            INCAT = 0
            ITUBEND = 0
            DO 90 I=1,100
               VICAT2(I) = 0.0
               IF (I .GT. 12) GOTO 90
               VICAT(I) = 0.0
               IF (I .GT. 5) GOTO 90
               VICAT3(I) = 0.0
               IF (I .GT. 2) GOTO 90
430      90 CONTINUE
            NPTSC1(I) = 0
         C
            TSTART = 0.0
            TSTOP = 0.0
            DO 100 I=1,7
               KITCONT(I) = 0
435      100 CONTINUE
         C
            IDART = 0
            DO 110 I=1,15
               VIDART(I) = 0.0
440      110 CONTINUE
         C
            DO 120 I=1,3
               VIDELT(I) = 0.0
445      120 CONTINUE
         C
            DO 130 I=1,108

```

```

460      VINFO(1) = 0.0
      IF(I .GT. 36) GOTO 130
      KINFO(1) = 0
      IF(I .GT. 12) GOTO 130
      IAERCSQ(I) = 0
      130 CONTINUE
      C
465      RAILNTH = 0.0
      RAILANG = 0.0
      ISIRL = 0
      NSLBKS = 0
      DO 140 I=1,28
      VIRAIL(I) = 0.0
      140 CONTINUE
      C
470      DO 150 I=1,31
      KIREP(I) = 0
      150 CONTINUE
      C
475      DO 160 I=1,300
      VIRKT(I) = 0.0
      160 CONTINUE
      C
480      DO 170 I=1,354
      VIROCK(I) = 0.0
      IF(I .GT. 13) GOTO 170
      KIROCK(I) = 0
      170 CONTINUE
      C
485      IPCNTL = 0
      DO 180 I=1,24
      VISEATO(I) = 0.0
      180 CONTINUE
      C
490      DO 190 I=1,24
      VISEATA(I) = 0.0
      190 CONTINUE
      C
495      ITVC = 0
      DO 200 I=1,8
      VITVC(I) = 0.0
      200 CONTINUE
      C
500      DO 210 I=1,11
      VMASS(I) = 0.0
      210 CONTINUE
      C
505      DO 220 I=1,90
      VMATR(I) = 0.0
      220 CONTINUE
      C
510      PRINTX = 0
      IPHASE1 = IPHASE2 = IPHASE3 = 0
      DO 230 I=1,101
      KMISC2(I) = 0
      IF(I .GT. 93) GOTO 230
      KMISC(I) = 0

```

```

515      IF(1.GT.17) GOTO 230
        VMISC2(I) = 0.0
        IF(1.GT.7) GOTO 230
        VMISC(I) = 0.0
        IF(1.GT.4) GOTO 230
        VMISC3(I) = 0
230      CONTINUE
        C
240      DO 240 I=1,166
        VMOMARM(I) = 0.0
240      CONTINUE
        C
245      DO 245 I=1,2
        PLTIME(I) = 0.0
        NHEADER(I) = 0
245      CONTINUE
        ICATOUT = IROKOUT = 0
        IRECOV = 0
        NPTSRLS = 0
        IFIRECV = 0
        NPTSRT = 0
        IDROQUE = 0
        IFDRO2 = 0
        NPTDFT2 = 0
        IFDRO1 = 0
        NPTDFT1 = 0
        IDROGLS = 0
        NPTSOLS = 0
        DO 250 I=1,73
        VPAR7(I) = 0.0
        IF(1.GT.51) GOTO 250
        VPAR3(I) = 0.0
        VPAR8(I) = 0.0
        IF(1.GT.50) GOTO 250
        VPAR2(I) = 0.0
        VPAR5(I) = 0.0
        VPAR6(I) = 0.0
        IF(1.GT.8) GOTO 250
        VPAR(I) = 0.0
        IF(1.GT.4) GOTO 250
        VPAR4(I) = 0.0
250      CONTINUE
        C
260      DO 260 I=1,5
        VRAIL(I) = 0.0
260      CONTINUE
        C
270      DO 270 I=1,1551
        VRKUT(I) = 0.0
        IF(1.GT.21) GOTO 270
        KRKUT(I) = 0
270      CONTINUE
        C
280      DO 280 I=1,12
        VTORQA(I) = 0.0
280      CONTINUE
        C
570

```


SUBROUTINE CLEAR 74/74 OPT=1 FTN 4.6+428 83/11/07 09.41.53 PAGE 58

DO 290 I=1,3
VTORQA(I) = 0.0
290 CONTINUE

C

575

DO 300 I=1,160
VTORSO(I) = 0.0
300 CONTINUE

C

580

ITVCFLG = 0
DO 310 I=1,20
VTVCF(I) = 0.0
310 CONTINUE

C

585

RETURN
END

```

1  SUBROUTINE DARTFM
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - COMPUTES THE FORCES AND MOMENTS OF THE DART LINES
C
5  C METHOD - ON THE SEAT/OCCUPANT SYSTEM
C VECTORS FROM THE LEFT AND RIGHT DART CONFLUENCE
C POINTS, TO THE LEFT AND RIGHT DART ATTACHMENT
C POINTS, RESPECTIVELY, ARE COMPUTED AND USED TO
C CALCULATE THE CURRENT LENGTH OF THE DART LINES.
C IF THE LENGTH OF EITHER LINE IS GREATER THAN THE
C DART START INPUT DISTANCE, BUT LESS THAN THE DART
C STOP INPUT DISTANCE, THEN THE FORCES AND MOMENTS
C FOR THAT DART LINE ARE CALCULATED.
C
15 C COMMUNICATIONS:
C CALLED BY: SEATOC
C
C CALLS:
C ROTATE
C NON-COMMON VARIABLES DEFINED:
C LDI(1) - LENGTH OF LEFT AND RIGHT DART LINES, RESPECTIVELY
C FDI(1) - TOTAL FORCE OF THE LEFT AND RIGHT DART LINES,
C RESPECTIVELY
C
20 C FXD(1) -
C FYD(1) - COMPONENT FORCES OF LEFT AND RIGHT DART LINES,
C FZD(1) - RESPECTIVELY
C
C TLDART(1) -
C TMDART(1) - COMPONENTS OF THE LEFT AND RIGHT DART LINE MOMENTS,
C TNDART(1) - RESPECTIVELY
C
30 C POTENTIAL ERROR CONDITIONS:
C DIVISION BY ZERO WHEN LRDL OR LLDL EQUAL ZERO -
C A MESSAGE IS PRINTED AND THE RUN IS TERMINATED
C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
C COMMON /FORCES / FXCASO(2) , FZCASO(2) ,
+ FYTUBSO , FZTUBSO
+ FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+ FXRKS(6) , FYRKS(6) , FZRKS(6) ,
+ FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+ FXAESO , FYAESO , FZAESO
+ FXORTSO , FYORTSO , FZORTSO
C.....
45 C SECTION 11 COMMON BLOCK
C.....
C COMMON /DARTIN / IDART , DRTFCE , DRTSTRT , DRTSTOP ,
+ XDRTAP(2) , YDRTAP(2) , ZDRTAP(2) ,
+ XDRTCP(2) , YDRTCP(2) , ZDRTCP(2)
C.....
C MATRIX COMMON BLOCK
C.....
C COMMON /MATRIX / DCMRA(3,3) , DCMRAI(3,3) , DCMRAJ(3,3) ,
+ DCMSE(3,3) , DCMST(3,3) , DCMTE(3,3) ,
+ DCMSEAE(3,3) , DCMSTAE(3,3) , DCMSTRI(3,3) ,

```



```

115      + TMSLSO(6) : TMSLSO(6) : TMSLSO(6) :
      + TMRKSO(6) : TMRKSO(6) : TMRKSO(6) :
      + TLMHSD(3) : TLMHSD(3) : TLMHSD(3) :
      + TLAESO : TMAESO : TNAESO :
      + TLDRTSO : TMDRTSO : TNDRTSO :
120      C
      C REAL LDL
      C
      C IF IDART .EQ. 0) GO TO 500
125      C
      C .....
      C TRANSFORM VECTOR IN RCS FROM ORIGIN OF RCS TO A/C C.G., TO SCS
      C .....
130      C
      C CALL ROTATE(XRCSAC,XSRCAC,ZVECT(1),DCMSR,0)
      C .....
      C TRANSFORM VECTOR IN EFCS FROM S/O C.G. TO A/C C.G., TO SCS
      C .....
135      C
      C CALL ROTATE(XESOAC,XSSOAC,ZVECT(1),DCMSE,0)
      C .....
      C TRANSFORM VECTOR IN RCS FROM ORIGIN OF RCS TO RIGHT AND LEFT
      C DART ATTACH POINTS, TO SCS
      C .....
140      C
      C FZDRTSO = FZDRTSO + FZDRTSO = 0.0
      C TLDRTSO = TMDRTSO + TMDRTSO = 0.0
145      C
      C DO 100 I=1,2
      C IF (IEVENTS(34+I) .GE. 1) GO TO 100
      C
      C XYZ(1) = XRRDAP(1)
      C XYZ(2) = YRRDAP(1)
      C XYZ(3) = ZRRDAP(1)
150      C
      C CALL ROTATE(XYZ(1),XYZ(1),ZVECT(1),DCMSR,0)
      C
155      C
      C XSSOAP(1) = XYZ(1)
      C YSSOAP(1) = XYZ(2)
      C ZSSOAP(1) = XYZ(3)
160      C
      C .....
      C COMPUTE VECTOR FROM RIGHT AND LEFT CONFLUENCE POINTS, TO RIGHT
      C AND LEFT ATTACHMENT POINTS, RESPECTIVELY
      C .....
165      C
      C XSCPAP(1) = XSSOAC - XSSOCP(1) - XSRCSAC + XSSDAP(1)
      C YSCPAP(1) = YSSOAC - YSSOCP(1) - YSRCSAC + YSSDAP(1)
      C ZSCPAP(1) = ZSSOAC - ZSSOCP(1) - ZSRCSAC + ZSSDAP(1)
170      C
      C .....
      C COMPUTE LENGTH OF DART LINES
      C .....

```

```

175      LDL = SQRT(XSCPAP(1) * XSCPAP(1) + YSCPAP(1) * YSCPAP(1) +
      *      ZSCPAP(1) * ZSCPAP(1))
      IF (LDL EQ. 0.0) GOTO 200
      IF (EVENTS(32+1) .GE. 1) GOTO 30
      IF (INTSTP .EQ. 0) GO TO 100
      IF (LDL .LT. DRTSTRT) GOTO 100
      EVENTS(32+1) = 1
      TIMES(32+1) = TIME
      GOTO 50
180      C
      30 CONTINUE
      IF (INTSTP .EQ. 0) GOTO 50
      IF (LDL .LE. DRTSTOP) GOTO 50
      EVENTS(34+1) = 1
      TIMES(34+1) = TIME
      IF (EVENTS(35) .EQ. EVENTS(36)) IDART = 0
      GOTO 100
190      C
      C *****
      C CALCULATE DART LINE FORCES AND MOMENTS
      C *****
      50 CONTINUE
      FDL = DRTFORCE * .5/LDL
      FXD = FDL * XSCPAP(1)
      FYD = FDL * YSCPAP(1)
      FZD = FDL * ZSCPAP(1)
      C
      FXORTSO = FXORTSO + FXD
      FYDRTSO = FYDRTSO + FYD
      FZDRTSO = FZDRTSO + FZD
      C
      TLDRTSO = TLDRTSO + (YSSOCP(1) * FZD - ZSSOCP(1) * FYD)
      TMDRTSO = TMDRTSO + (ZSSOCP(1) * FXD - XSSOCP(1) * FZD)
      TNDRTSO = TNDRTSO + (XSSOCP(1) * FYD - YSSOCP(1) * FXD)
      C
      100 CONTINUE
      C
      GOTO 500
210      C
      200 WRITE(5,250)
      250 FORMAT(1X, //72(1H*)//.4X, "FATAL ERROR(SUBROUTINE DARTFM)*** ",
      * /, "RIGHT OR LEFT DART LINE CALCULATED TO BE EQUAL TO ZERO", /,
      * "RESULTS IN DIVISION BY ZERO", /,72(1H*))
      C
      IERRFLG = 1
      C
      500 CONTINUE
      RETURN
      END
220

```

```

1  SUBROUTINE DRICALC
C .....
C DESCRIPTION - LEVEL 2
C FUNCTION COMPUTES THE DYNAMIC RESPONSE INDEX
C METHOD USING THE CURRENT TIME (TIME) AND LAST PREVIOUS
C VALUE (TPREV) FOR TIME, AND THE CURRENT AND LAST
C PREVIOUS SEAT ACCELERATIONS PARALLEL TO THE SEAT
C BACK, THE ROUTINE INTEGRATES FROM TPREV TO TIME,
C THE INTEGRATION STEP SIZE BEING DELTAT
C COMMUNICATIONS
C CALLED BY:
C GESS
C CALLS:
C REPTDRI, RUNGE
C NON COMMON VARIABLES DEFINED
C ACCEL1 - SEAT/OCCUPANT OR OCCUPANT ALONE Z-ACCELERATION
C ACINTRP - INTERPOLATED VALUE FOR Z-ACCELERATION
C POTENTIAL ERROR CONDITIONS - NONE
C .....
20 C .....
C DYNAMIC RESPONSE INDEX VARIABLES COMMON BLOCK
C .....
25 COMMON /DRIVB / DRVAL(2), DRIDERV(2), ACCEL1
+ DRI, DRIMAX, TMAX
+ ZACCMAX, DRICON
C .....
C SECTION 1 COMMON BLOCK
C .....
30 COMMON /ICONTROL / TSTART, TSTOP, ESTOP, IRESTRT, IUNITS
+ ISEATTR, ISOSEP, IPLOT, IDRIFLG,
+ IPHASE1, IPHASE2, IPHASE3
+ ISTOP
C .....
C MISCELLANEOUS DATA COMMON BLOCK
C .....
35 COMMON /MISC / IPAGEC(131), LINECT(131), IPRCTN(131)
+ MAXLINE, MAXREPT, MAXEVNI
+ TEVLIN, TERFLG, LU
+ IDATE, HEADALT, HEADDEL
+ HEADSR, HEADYAW, HEADPTT
+ HEADRCI, HEADWGT, BIAS
+ REPTYPE(5,31), PRTLNGT(2), PRTWGHT(2)
+ THEADER(24), TEVENTS(38), PRTEMP(2)
+ IMVOC
+ PRIMASS(2), PRINDX, PKZVFI
+ ZVECT(3), XYZ(3), SAVTIME
+ XACCEL(3), YACCEL(3), ZACCEL(3)
+ REPTYPE, BIAS, PRTLNGT
+ PRWGT, PRIMASS, PRINDX
+ PRTEMP
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
40 COMMON /RKUTTA / TIME, TIME5, DELTAT, TRAJSO(193)
+ TRAJSA(193), TRAJOA(193), TRAJCH(97,3)
+ TRAJAC(193), TVCE05(225), QUAT50(65)
+
45
50
55

```

AD-A148 363

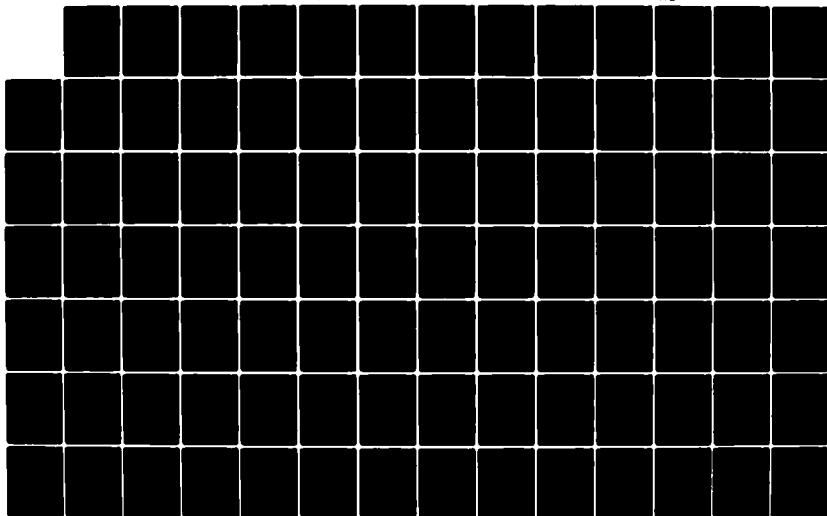
A GENERALIZED ESCAPE SYSTEM SIMULATION (GESS) COMPUTER
PROGRAM VOLUME 2 G. (U) KETRON INC WARMINSTER PA
L A D'AULERIO ET AL. APR 84 NADC-84068-60

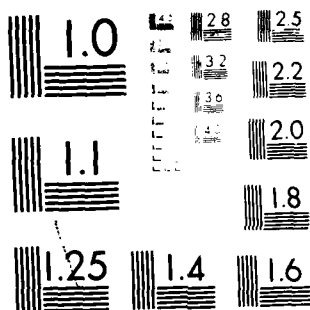
2/4

UNCLASSIFIED N62269-81-C-0206

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A


```

60      + QUATSA(65)      QUATOA(68)      QUATAC(65)
      + INSTP      IPCPASS      IRKPASS
      + IPOINTS      IYX      IYPRX
      + IKX      IKSUMX      IKPASSX
      + IYIX      IYIIX      IYI2X
      + IYI3X      IYPRIX      IYPRI1X
      + IYPRI2X      IYVIX      IYV1IX
      + ICVIX      ICVIN      IREIN
      C
      IF(IDRIFLG.EQ.O) GOTO 999
      IF(INTSTP.EQ.O) GOTO 999
      IF(TIME.LE.O.O) GO TO 900
      C *****
      C STOP DRI CALCULATIONS IF SEAT SEPARATES FROM RAILS
      C *****
      IF(IEVENTS(37).EQ.O) GOTO 10
      CALL REPTDRI
      IDRIFLG=O
      GOTO 999
      10 CONTINUE
      DT= TIME - TPREV
      DO 30 IP=1,4
80      C *****
      C INTERPOLATE Z-ACCELERATION BETWEEN TPREV AND TIME KEEPING IN MIND
      C THAT TPREV IS BEING INCREMENTED BY SUBROUTINE RUNGE
      C *****
      C
      ACCEL= ZACCEL(1) - (ZACCEL(1) - ACCEL1) * (TIME - IPREV)/DT
      C *****
      C CALCULATE DERIVATIVES
      C *****
      C DRIVAL(1) = DISPLACEMENT
      C DRIVAL(2) = VELOCITY
      C DRIDERV(1) = DERIVATIVE OF DRIVAL(1)
      C DRIDERV(2) = DERIVATIVE OF DRIVAL(2)
      C *****
      C DRIDERV(1) = DRIVAL(2)
      C DRIDERV(2) = ACCEL - 23.7 * DRIVAL(2) - 2798.41 * DRIVAL(1)
      C *****
      C
      C CALL THE FOURTH ORDER RUNGE-KUTTA FUNCTION
      C *****
      C
      CALL RUNGE(2,DRIVAL,DRIDERV,IPREV,DT,IP)
      30 CONTINUE
      C
      C *****
      C DISPLACEMENT MUST BE MULTIPLIED BY A CONSTANT TO OBTAIN THE
      C DIMENSIONLESS DRI VALUE (DRICON = 2798.41/GRAVITY)
      C *****
      C
      DRI= DRICON * DRIVAL(1)
      C *****
      C
      C CHECK FOR MAXIMUM VALUE
      C *****

```

```
115      C      IF(DRI .LE. DRIMAX) GOTO 900
          DRIMAX = DRI
          TMAX = TIME
          C
120      900 CONTINUE
          ACCEL1=ZACCEL(1)
          TPREV= TIME
          999 CONTINUE
          RETURN
125      END
```

```

1 SUBROUTINE DROGUE 1
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - CONTROLS SEQUENCING OF SINGLE & DUPLEX DROGUE SYSTEMS
C METHOD - THIS ROUTINE CHECKS FOR DROGUE PROJECTION, LINE
C STRETCH, AND FULL INFLATION, THEN COMPUTES THE CHUTE'S
C POSITION AND VELOCITY
C COMMUNICATIONS -
C CALLED BY: CHUTES
C CALLS:
C CHUTFM
C CHJINIT
C PCHUTFT
C ROTATE
C ZARCTAN
C ZLININT
C NON-COMMON VARIABLES DEFINED -
C TVEL - TOTAL VELOCITY OF THE DROGUE SYSTEM
C XLENGTH - COMPONENTS OF THE LENGTH OF THE EXTENDED DROGUE LINES
C YLENGTH - PRIOR TO LINE STRETCH
C ZLENGTH - TOTAL LENGTH OF THE EXTENDED DROGUE LINES PRIOR TO
C LINE STRETCH
C XDTS -
C YDIS - COMPONENTS OF THE VECTOR FROM THE DROGUE ATTACHMENT
C POINT TO THE DROGUE CHUTE
C POTENTIAL ERROR CONDITIONS - NONE
C .....
C MATRIX COMMON BLOCK
C .....
C COMMON /MATRIX / DCMAR(3,3) , DCMRA(3,3) , DCMSE(3,3) , DCMSE(3,3) ,
+ DCMSE(3,3) , DCMTE(3,3) , DCMTE(3,3) ,
+ DCMSE(3,3) , DCMSE(3,3) , DCMSE(3,3) ,
+ DCMSE(3,3)
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31)
+ MAXLINE , MAXREPT , MAXEVT ,
+ IEVLNE , TERRFLG , LU
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWTG , BIAS ,
+ REPTYPE(5,31) , PRTLNGT(2) , PRTLNGT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ PRTHASS(2) , PRTHASS(2) , PRTHASS(2) ,
+ ZVECT(3) , XYZ(3) , PKZVEL ,
+ XACCEL(3) , YACCEL(3) , SAVTIME ,
+ REPTYPE , BIAS , ZACCEL(3) ,
+ PRTLNGT , PRTLNGT , PRTLNGT ,
+ PRTHASS , PRTHASS , PRTHASS ,
C .....
C MOMARMS COMMON BLOCK
C .....

```


5-69

SUB. JUTINE DROGUE 1 74/74 OPT=1 FTN 4.6+428 83/11/07. 09.41.53 PAGE 69

```

C.....
C COMPUTE DROGUE CHUTE FORCE
C.....
175 C 200 CONTINUE
    CALL CHUTFM(1,DROGPD1)
C
C
180 C.....
C COMPUTE DROGUE CHUTE POSITION
C.....
C
    XYZ(1) = XSSOCH(1)
    XYZ(2) = VSSOCH(1)
    XYZ(3) = ZSSOCH(1)
C
    CALL ROTATE(XYZ(1),XYZ(1),ZVECT(1),DCMSE,1)
C
    R = SORT((TRAJSD(14)*TRAJSD(14) + TRAJSD(15)*TRAJSD(15) +
    + TRAJSD(16)*TRAJSD(16))
C
    IF(R .EQ. 0.0) GOTO 400
    SIN2 = TRAJSD(16)/R
    COS2 = COS(ASIN(SIN2))
    BETA = ZARCTAN(TRAJSD(15),TRAJSD(14))
C
    XD1S = -SIGN((DROGLL+COS2*COS(BETA)),TRAJSD(14))
    YD1S = -SIGN((DROGLL+COS2*SIN(BETA)),TRAJSD(15))
    ZD1S = -SIGN((DROGLL+SIN2),TRAJSD(16))
C
C
    TRAJCH(2,1) = XYZ(1) + XD1S + TRAJSD(2)
    TRAJCH(3,1) = XYZ(2) + YD1S + TRAJSD(3)
    TRAJCH(4,1) = XYZ(3) + ZD1S + TRAJSD(4)
    GOTO 500
C
400 CONTINUE
    WRITE(5,410)
410 FORMAT(2X,/,72(1H+)/.4X,"FATAL ERROR(SUBROUTINE DROGUE1)*** "
    + "R EQUAL TO ZERO RESULTS IN DIVISION BY ZERO"/.72(1H+))
    IERRFLG = 1
C
500 CONTINUE
    RETURN
    END
215

```

```

1  SUBROUTINE DROGUE2
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - CONTROLS SEQUENCING FOR THE VELCON DROGUE
C METHOD - THIS ROUTINE CHECKS FOR PROJECTION, LINE STRETCH,
C AND FULL INFLATION OF BOTH THE LARGE AND SMALL CHUTES,
C THEN COMPUTES THE CHUTES' POSITION AND VELOCITIES
C COMMUNICATIONS -
C CALLED BY:
C CHUTES
C CALLS:
C CHUTEM
C CHUNIT
C PCHUTFT
C ROTATE
C ROTATE
C ZLININT
C NON-COMMON VARIABLES DEFINED -
C TVEL - TOTAL VELOCITY OF THE DROGUE SYSTEM
C XLENGTH -
C YLENGTH - COMPONENTS OF THE LENGTH OF THE EXTENDED DROGUE LINES
C ZLENGTH - PRIOR TO LINE STRETCH
C TLENGTH - TOTAL LENGTH OF THE EXTENDED DROGUE LINES PRIOR
C TO LINE STRETCH
C XD1S -
C YD1S - COMPONENTS OF THE VECTOR FROM THE DROGUE ATTACHMENT
C ZD1S - POINT TO THE DROGUE CHUTE
C POTENTIAL ERROR CONDITIONS - NONE
C.....
C MATRIX COMMON BLOCK
C.....
C COMMON /MATRIX / DCMRAE(3,3), DCMRA(3,3), DCMSE(3,3), DCMSE(3,3),
C + DCMSE(3,3), DCMSE(3,3), DCMSE(3,3), DCMSE(3,3),
C + DCMSE(3,3), DCMSE(3,3), DCMSE(3,3), DCMSE(3,3),
C + DCMSE(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
C COMMON /MISC / IPAGECT(31), LINECT(31), IPRTCNT(31),
C + MAXLINE, MAXREPT, MAXEVNT,
C + IEVLIN, TERRFLG, LU,
C + IDATE, HEADALT, HEADVEL,
C + HEADRDL, HEADYAW, HEADPIT,
C + REPTYPE(5,31), HEADWGT, BIAS,
C + IHEADER(24), PRTLNGT(2), PRTWGHT(2),
C + PRTMAS(2), IEVENTS(38), TIMES(38),
C + ZVECT(3), IMVDC, PRTEMP( 2),
C + XACCEL(3), PKZVEL,
C + REPTYPE, XYZ(3), SAVTIME,
C + PRTWGHT, YACCEL(3), ZACCEL(3),
C + PRTEMP, BIAS, PRTLNGT,
C + PRTMAS, PRTINDX
C.....

```

```

C.....
C MOMARMS COMMON BLOCK
C.....
COMMON /MOMARMS /
+REFLNSO ,REFLNOA ,REFLNSA ,URX(6) ,URY(6) ,URZ(6) ,
+XSSOCA(2) ,YSSOCA(2) ,ZSSOCA(2) ,XSSORK(6) ,YSSORK(6) ,ZSSORK(6) ,
+XSSORRE ,YSSORRE ,ZSSORRE ,XSSOLRE ,YSSOLRE ,ZSSOLRE ,
+XSSOMRE ,YSSOMRE ,ZSSOMRE ,XSSOBOT ,YSSOBOT ,ZSSOBOT ,
+XSSOSB(6) ,YSSOSB(6) ,ZSSOSB(6) ,XSSCSAC ,YSSCSAC ,ZSSCSAC ,
+XSSCSR ,YSSCSR ,ZSSCSR ,XSSSRP ,YSSSRP ,ZSSSRP ,
+
+XSSASRP ,YSSASRP ,ZSSASRP ,XRRDAP(2) ,YRRDAP(2) ,ZRRDAP(2) ,
+XRRSBO(6) ,YRRSBO(6) ,ZRRSBO(6) ,XSSOCP(2) ,YSSOCP(2) ,ZSSOCP(2) ,
+XSSDAP(2) ,YSSDAP(2) ,ZSSDAP(2) ,XESOAC ,YESOAC ,ZESOAC ,
+XSRCSAC ,YSRCSAC ,ZSRCSAC ,XSSOAC ,YSSOAC ,ZSSOAC ,
+XRSOSB ,YRSOSB ,ZRSOSB ,XRRSBO ,YRRSBO ,ZRRSBO ,
+XRRSB ,YRRSB ,ZRRSB ,XSSOCH(3) ,YSSOCH(3) ,ZSSOCH(3) ,
+XAACSO ,YAACSO ,ZAACSO ,XASOAC ,YASOAC ,ZASOAC ,
+XRSOAC ,YRSOAC ,ZRSOAC ,XSCPAP(2) ,YSCPAP(2) ,ZSCPAP(2) ,
C.....
C SECTION 14 COMMON BLOCK
C.....
COMMON /PARCHUT / IRECOV ,
+ RECOVDP , RECOVLL ,
+ XRECAP , YRECAP , ZRECAP ,
+ NPTSRLS , RECOVLS(2,25) , IFTRCV ,
+ NPTSRT , RECOVFT(2,25) , SEPERCE ,
+ IDROGUE , DRDRAG2 , DRGPD2 ,
+ POROSD2 , VELCON , IFDR02 ,
+ NPTDFT1 , DRGFT2(2,25) , IFDR01 ,
+ NPTDFT2 , DRGFT1(2,25) , IDROGLS ,
+ NPTSLS , DRGLS(2,25) , TDPLDY ,
+ DISPLY , DRGLL , DRDRAG1 ,
+ DRGPD1 , POROSD1 , DROVELX ,
+ DROVELY , DROVELZ , XDRGAP ,
+ YDRGAP , ZDRGAP , CHALT1 ,
+ CHALT2 , GLIMIT , TDELAY ,
+ AREADC , WGTDC , TFP1 ,
+ TFP2 , TFP3 , TDRGLS ,
+ CDDC , NPTSRT , RECOVDT(2,25)
C.....
C INTEGRATION ROUTINE COMMON BLOCK
C.....
COMMON /RKUTTA / TIME , TIMES , DELTAT ,
+ TRAJSA(193) , TRAJOA(193) , TRAJSO(193) ,
+ TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
+ QUATSA(65) , QUATOA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IPOINTS , IYX , IYPRX ,
+ IKX , IKSUMX , IKPASSX ,
+ IYIX , IYIIX , IYI2X ,
+ IYIX , IYI3X , IYPRIX ,
+ IYPR12X , IYIYX , IYI1X ,
+ ICYIX , ICYIIX , IREIN
C.....
IF(IEVENTS(24) NE 0) GOTO 500
C

```



```

115 C*****
C CHECK FOR DROGUE PROJECTION
C*****
C
      IF(IEVENTS(18) .NE. 0) GOTO 100
      IF(INTSTP .EQ. 0) GOTO 500
      IF((TIME6 .LT. TDDPLY) .OR. (ABS(ZRRSROT) .LT. DISPLAY))GOTO 500
      IEVENTS(18)=IEVENTS(21) = 1
      TIMES(18)=TIMES(21) = TIME
      CALL CHJINIT
      GOTO 500

125 C
C
C*****
C CHECK FOR DROGUE CHUTE LINE STRETCH
C*****
C
      100 CONTINUE
      IF(IEVENTS(19) .NE. 0) GOTO 150
      IF(INTSTP .EQ. 0) GOTO 500

135 C
C*****
C DETERMINE HOW FAR DROGUE LINES ARE STRETCHED
C*****
C
      XYZ(1) = XSSDCH(1)
      XYZ(2) = YSSDCH(1)
      XYZ(3) = ZSSDCH(1)

      CALL ROTATE(XYZ(1),XYZ(1),ZVECT(1),DCMSE,1)

145 C
      XLENGTH = TRAJSD(2) - TRAJCH(2,1) + XYZ(1)
      YLENGTH = TRAJSD(3) - TRAJCH(3,1) + XYZ(2)
      ZLENGTH = TRAJSD(4) - TRAJCH(4,1) + XYZ(3)
      TLENGTH = SORT(XLENGTH*XLENGTH+YLENGTH*YLENGTH+ZLENGTH*ZLENGTH)

150 C
      IF(TLENGTH .LT. DROGLL) GOTO 500
      IEVENTS(19)=IEVENTS(22) = 1
      TIMES(19)=TIMES(22) = TIME
      TRAJCH(1,1) = 0.0
      TRAJCH(1,2) = 0.0

155 C
      TVEL = SQRT(TRAJSD(14)*TRAJSD(14) + TRAJSD(15)*TRAJSD(15) +
+ TRAJSD(16)*TRAJSD(16))
      DO 140 I=1,2
      IF(I .EQ. 2) GOTO 125
      IF(IFTDR01 .EQ. 0) GOTO 110
      CALL ZLININT(TVEL,DROGFT1,NPTDFT1,25,TFP1,2)
      GOTO 140
110 CONTINUE
      CALL PCHUITF(TVEL,DROGPD1,DRDRAG1,POROSD1,TFP1)
      GOTO 140
125 CONTINUE
      IF(IFTDR02 .EQ. 0) GOTO 130
      CALL ZLININT(TVEL,DROGFT2,NPTDFT2,25,TFP2,2)
      GOTO 140
130 CONTINUE

```

```

      CALL PCHUTFT(TVEL,DROGPD2,DRDRAG2,POROSD2,TFP2)
140 CONTINUE
      GOTO 500
C.....
C CHECK FOR LARGE DROGUE CHUTE FULL INFLATION
C.....
150 CONTINUE
      IF(IEVENTS(20).NE.0) GOTO 200
      IF(INTSTP.EQ.0) GOTO 300
      IF(TIMES.LT.(TIMES(19)+TFP1)) GOTO 200
      IEVENTS(20) = 1
      TIMES(20) = TIME
C.....
C CHECK FOR SMALL DROGUE CHUTE FULL INFLATION
C.....
200 CONTINUE
      IF(IEVENTS(23).EQ.0) GOTO 225
      IF(IMVDC.NE.1) GOTO 300
      GOTO 350
C.....
225 CONTINUE
      IF(INTSTP.EQ.1) GOTO 250
      GOTO 300
C.....
250 CONTINUE
      IF(TIMES.LT.(TIMES(22)+TFP2)) GOTO 300
      IEVENTS(23) = 1
      TIMES(23) = TIME
C.....
C COMPUTE CURRENT TOTAL VELOCITY
C.....
      TVEL = SORT(TRAUSD(14)*TRAUSD(14)+TRAUSD(15)*TRAUSD(15)+
      + TRAUSD(16)*TRAUSD(16))
C.....
      IF(TVEL.LE.VELCON) GOTO 300
      IMVDC = 1
      GOTO 350
C.....
C COMPUTE FORCES DUE TO LARGE CHUTE
C.....
300 CONTINUE
      CALL CHUTFM(1,DROGPD1)
C.....
C COMPUTE FORCES DUE TO SMALL CHUTE
C.....
350 CONTINUE
      CALL CHUTFM(2,DROGPD2)

```

5-75

```

1  SUBROUTINE DYNAMCG
C .....
C DESCRIPTION - LEVEL 2
C FUNCTION - SOLVES THE EQUATIONS THAT COMPUTE THE DYNAMIC C.G.
C MOVEMENT
C METHOD - ROTATE, RUNGE
C COMMUNICATIONS -
C CALLED BY:
C GESS
C CALLS:
C TO BE DEFINED
C NON-COMMON VARIABLES DEFINED:
C TO BE DEFINED
C POTENTIAL ERROR CONDITIONS:
C TO BE DEFINED
C .....
C CONSTANTS COMMON BLOCK
C .....
C COMMON /CONSTANT / GRAVITY , RADDEG , DEGRAD , PI
C .....
C SECTION 13 COMMON BLOCK
C .....
C COMMON /DYNCGIN / IDYNGC , WY , WYI , SXN ,
+ CX , XSLACK , SXP ,
+ CY , SY , CZ , ZSLACK ,
+ SZP , ZBOT , SZN1 , SZN2
C .....
C DYNAMIC CG VARIABLES COMMON BLOCK
C .....
C COMMON /DYNCGVB / GVAL(6) , CGDERV(6) ,
+ XCGDAO , YCGDAO , ZCGDAO ,
+ DXCG , DYCG , DZCG
C .....
C SECTION 1 COMMON BLOCK
C .....
C COMMON /ICONTROL / TSTART , TSTOP , ESTOP , IRESTRT , IUNITS ,
+ ISEATTR , ISOSEP , IPLOT , IDRIFLG ,
+ IPHASE1 , IPHASE2 , IPHASE3
C .....
C INTEGER
C .....
C SECTION 6 COMMON BLOCK
C .....
C COMMON /ISEATOC / IPCNTL , XCGSO , YCGSO , ZCGSO , IXXSO ,
+ IYXSO , IYZSO , IYYSO , IZZSO ,
+ AREASO , AREAOA , WHTOAB , WHTOAA ,
+ IXXOA , IYXOA , IYZOA , IYYOA , IZZOA ,
+ IZSOA , XCGOA , YCGOA , ZCGOA , SOSEP ,
+ IXXSU , IYXSU , IYZSU , IYYSU , IZZSU ,
+ IXXOA , IYXOA , IYZOA , IYYOA ,
+ IZZOA , IZZOA
C .....
C SECTION 5 COMMON BLOCK
C .....
C COMMON /ISETALN / XPOSSRP , YPOSSRP , ZPOSSRP , XCGSA , YCGSA ,
+ ZCGSA , IXXSA , IYXSA , IYZSA , IYYSA ,
+ IZZSA , IZZSA , PHISA , PSISA , IHESA ,
+

```

```

+
+
+ REAL
+
60
+
+ AREA5A , HGH5TA , WGH5TA , XPOSBOT , YPOSBOT ,
+ ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS
+ IX5A , IY5A , IZ5A , IXZ5A , IY5A , IZ5A ,
+ IZ5A
C*****
C MASSES COMMON BLOCK
C*****
+
+ COMMON /MASSES / MASSOA1 , MASSOA2 , MASSO , MASSO
+ MASSA , MASSRK(6) , MASSDC
+ MASSOA1 , MASSOA2 , MASSO , MASSO
+ MASSA , MASSRK , MASSDC
C*****
C MATRIX COMMON BLOCK
C*****
+
+ COMMON /MATRIX / DCMAB(3,3) , DCMRA(3,3) , DCM5A(3,3) ,
+ DCM5E(3,3) , DCM5S(3,3) , DCM5E(3,3) ,
+ DCM5AE(3,3) , DCM5AE(3,3) , DCM5R(3,3) ,
+ DCM5UM(3,3)
C*****
C MISCELLANEOUS DATA COMMON BLOCK
C*****
+
+ COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRICNT(31) ,
+ MAXLINE , MAXEVT ,
+ IEVLNE , IERRFLG , LU
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS
+ REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
+ IHEADER(24) , IEVENTS(38) ,
+ IMVDC ,
+ PRIMASS(2) , PRTINDX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+ REPTYPE , BIAS , PRTLNCT ,
+ PRTWGT , PRTMP , PRTMASS , PRTINDX
C*****
C MOMARMS COMMON BLOCK
C*****
+
+ COMMON /MOMARMS /
+ REFLNSO , REFLNOA , REFLNSA , URX(6) , URY(6) , URZ(6) ,
+ X5SOCA(2) , Y5SOCA(2) , Z5SOCA(2) , X5SORK(6) , Y5SORK(6) , Z5SORK(6) ,
+ X5SORRE , Y5SORRE , Z5SORRE , X5SOLRE , Y5SOLRE , Z5SOLRE ,
+ X5SOMRE , Y5SOMRE , Z5SOMRE , X5SOBOT , Y5SOBOT , Z5SOBOT ,
+ X5SO5B(6) , Y5SO5B(6) , Z5SO5B(6) , X5RCSAC , Y5RCSAC , Z5RCSAC ,
+ X5SCSAC , Y5SCSAC , Z5SCSAC , X5SOSRP , Y5SOSRP , Z5SOSRP ,
+ X5SASRP , Y5SASRP , Z5SASRP , X5RDAP(2) , Y5RDAP(2) , Z5RDAP(2) ,
+ X5RSBO(6) , Y5RSBO(6) , Z5RSBO(6) , X5SOCP(2) , Y5SOCP(2) , Z5SOCP(2) ,
+ X5SDAP(2) , Y5SDAP(2) , Z5SDAP(2) , X5SOAC , Y5SOAC , Z5SOAC ,
+ X5RCSAC , Y5RCSAC , Z5RCSAC , X5SOAC , Y5SOAC , Z5SOAC ,
+ X5RSOB , Y5RSOB , Z5RSOB , X5RSBOT , Y5RSBOT , Z5RSBOT ,
+ X5RSB , Y5RSB , Z5RSB , X5SOCH(3) , Y5SOCH(3) , Z5SOCH(3) ,
+ X5ACSO , Y5ACSO , Z5ACSO , X5SOAC , Y5SOAC , Z5SOAC ,
+ X5RSOAC , Y5RSOAC , Z5RSOAC , X5CPAP(2) , Y5CPAP(2) , Z5CPAP(2)
C *****
C INTEGRATION ROUTINE COMMON BLOCK

```

```

115 C .....
      COMMON /RKUTTA / TIME, TIMES, DELTAT,
      + TRAJSA(193), TRAJDA(193), TRAJSO(193),
      + TRAJAC(193), TVCEOS(225), TRAJCH(97.3),
      + QUATSA(65), QUATOA(65), QUATSO(65),
      + INTSTP, IPCPASS, QUATAC(65),
      + IPOINTS, IVX, IRKPASS,
      + IKX, IKSUMX, IVPRX,
      + IVIX, IKPASSX, IVI2X,
      + IVI3X, IVPRI1X,
      + IVPRI2X, IPY1X,
      + ICY1X, ICY2X, ICY3X,
      + IREIN
      C .....

120 C .....

125 C .....

130 C .....
      IF(INTSTP.EQ.0) RETURN
      IF(IDYNGC.EQ.0) RETURN
      IF(TIME.LE.0.0) GOTO 250
      IF(TEVENTS(28).EQ.0) GOTO 50
      IDYNGC=0
      CONTINUE

135 C .....
      DT= TIME - TPREV

140 C .....
      DO 100 IP=1,4
      X=CGVAL(1)
      FX= WX = 0.0
      IF(X.LT.0.0) FX= SXN*X
      IF(X.GT.XSLACK) FX= SXP*(X-XSLACK)
      WXX= FX/MASSDA1
      IF(X.NE.0.0) WX = SORT(ABS(WXX/X))

145 C .....
      Z=CGVAL(5)
      FZ= WZ = 0.0
      IF(Z.LT.ZBOT) FZ= SZN2*(Z-ZBOT) + SZN1*ZBOT
      IF(Z.LT.0.0) .AND. (Z.GT.ZBOT)) FZ= SZN1*Z
      IF(Z.GT.ZSLACK) FZ= SZP*(Z-ZSLACK)
      WZZ= FZ/MASSDA1
      IF(Z.NE.0.0) WZ = SORT(ABS(WZZ/Z))

150 C .....
      TFRAC= (TIME - TPREV)/DT

155 C .....
      C .....
      C INTERPOLATE LINEAR ACCELERATIONS BETWEEN TPREV AND TIME:
      C KEEP IN MIND THAT TPREV IS BEING INCREMENTED BY SUBR RUNGE.
      C THE EQUATIONS ALSO TAKE INTO ACCOUNT THE FACT THAT THE
      C ACCELERATIONS MUST BE REVERSED, SINCE WE WANT THE
      C ACCELERATION OF THE OCCUPANT WRT THE SEAT.
      C .....
      C .....
      XACC = TFRAC * (XACCEL(1) - XACCL1) - XACCEL(1)
      YACC = TFRAC * (YACCEL(1) - YACCL1) - YACCEL(1)
      ZACC = TFRAC * (ZACCEL(1) - ZACCL1) - ZACCEL(1)
      C .....
      C .....
      C COMPUTE DIFFERENTIAL EQUATIONS
      C .....
      C .....
170 C .....

```

```

CGDERV(1)= CGVAL(2)
CGDERV(2)= XACC - 2.0*CX*WX*CGVAL(2) - WXX
C
175 CGDERV(3)= CGVAL(4)
CGDERV(4)= YACC - 2.0*CY*WY*CGVAL(4) - WYY*CGVAL(3)
C
180 CGDERV(5)= CGVAL(6)
CGDERV(6)= ZACC - 2.0*CZ*WZ*CGVAL(6) - WZZ
C
CALL RUNGE(6,CGVAL,CGDERV,TPREV,DT,IP)
100 CONTINUE
C *****
C UPDATE POSITION OF OCCUPANT ALONE CG
C *****
XCGOA = XCGOAO + CGVAL(1)
YCGOA = YCGOAO + CGVAL(3)
ZCGOA = ZCGOAO + CGVAL(5)
C
C *****
C UPDATE SEAT OCCUPANT CG
C *****
XCGNEW = (MASSOA1*XCGOA + MASSA*XCOSA)/MASSSO
YCGNEW = (MASSOA1*YCGOA + MASSA*YCOSA)/MASSSO
ZCGNEW = (MASSOA1*ZCGOA + MASSA*ZCOSA)/MASSSO
C
DXCG = XCGNEW-XCGSO
DYCG = YCGNEW-YCGSO
DZCG = ZCGNEW-ZCGSO
C
XCGSO = XCGNEW
YCGSO = YCGNEW
ZCGSO = ZCGNEW
C *****
C UPDATE MOMENT ARMS
C *****
DO 210 I=1,2
XSSOCA(1) = XSSOCA(1) - DXCG
YSSOCA(1) = YSSOCA(1) - DYCG
ZSSOCA(1) = ZSSOCA(1) - DZCG
C
XSSOCP(1) = XSSOCP(1) - DXCG
YSSOCP(1) = YSSOCP(1) - DYCG
ZSSOCP(1) = ZSSOCP(1) - DZCG
210 CONTINUE
C
DO 220 I=1,6
XSSORK(1) = XSSORK(1) - DXCG
YSSORK(1) = YSSORK(1) - DYCG
ZSSORK(1) = ZSSORK(1) - DZCG
C
XSSOSB(1) = XSSOSB(1) - DXCG
YSSOSB(1) = YSSOSB(1) - DYCG
ZSSOSB(1) = ZSSOSB(1) - DZCG
220 CONTINUE
C
DO 230 I=1,3
XSSOCH(1) = XSSOCH(1) - DXCG

```

SUBROUTINE DYNAMCG 74/74 OPT=1

```

230      YSSOCH(1) = YSSOCH(1) - DYCG
      ZSSOCH(1) = ZSSOCH(1) - DZCG
      230 CONTINUE
      XSSOBT = XSSOBT - DXCG
      YSSOBT = YSSOBT - DYCG
      ZSSOBT = ZSSOBT - DZCG
      C
      XSSOSRP = XSSOSRP - DXCG
      YSSOSRP = YSSOSRP - DYCG
      ZSSOSRP = ZSSOSRP - DZCG
      C
      240      CALL ROTATE (DXCG,DXCG,ZVECT,DCMSE,1)
      TRAUSO(2) = TRAUSO(2) - DXCG
      TRAUSO(3) = TRAUSO(3) - DYCG
      TRAUSO(4) = TRAUSO(4) - DZCG
      250      CONTINUE
      TPREV = TIME
      XACCI = XACCEL(1)
      YACCI = YACCEL(1)
      ZACCI = ZACCEL(1)
      RETURN
      END
250

```



```

1 SUBROUTINE EVENT
2 .....
3 C DESCRIPTION - LEVEL 3
4 .....
5 C FUNCTION - TO RECORD THE STATUS OF VARIOUS ESCAPE SEQUENCE EVENTS
6 .....
7 C COMMUNICATIONS -
8 .....
9 C CALLED BY: REPORTS
10 .....
11 C CALLS: NONE
12 .....
13 C .....
14 .....
15 C EVENT MESSAGES COMMON BLOCK
16 .....
17 C .....
18 C .....
19 .....
20 C COMMON /EVMS / IEVMS(3,38) , ISPMES(4,8) , ISPECL(8)
21 .....
22 C SECTION 2 COMMON BLOCK
23 .....
24 .....
25 C .....
26 C .....
27 C .....
28 C .....
29 C .....
30 C .....
31 C .....
32 C .....
33 C .....
34 C .....
35 C .....
36 C .....
37 C .....
38 C .....
39 C .....
40 C .....
41 C .....
42 C .....
43 C .....
44 C .....
45 C .....
46 C .....
47 C .....
48 C .....
49 C .....
50 C .....
51 C .....
52 C .....
53 C .....
54 C .....
55 C .....
56 C .....
57 C .....
58 C .....
59 C .....
60 C .....
61 C .....
62 C .....
63 C .....
64 C .....
65 C .....
66 C .....
67 C .....
68 C .....
69 C .....
70 C .....
71 C .....
72 C .....
73 C .....
74 C .....
75 C .....
76 C .....
77 C .....
78 C .....
79 C .....
80 C .....
81 C .....
82 C .....
83 C .....
84 C .....
85 C .....
86 C .....
87 C .....
88 C .....
89 C .....
90 C .....
91 C .....
92 C .....
93 C .....
94 C .....
95 C .....
96 C .....
97 C .....
98 C .....
99 C .....
100 C .....

```

83/11/07. 09.41.53

FTN 4.6+428

74/74 OPT=1

SUBROUTINE EVENT

```

60      DATA (1SPMES(1, 4), 1-1, 4) / 10HPERIGEE OF , 10H TRAJECTOR , /
      + DATA (1EVMS(1, 1), 1-1, 3) / 10HCATAPULT 1 , 10H IGNITION , /
      + DATA (1EVMS(1, 2), 1-1, 3) / 10HCATAPULT 2 , 10H IGNITION , /
      + DATA (1EVMS(1, 3), 1-1, 3) / 10HCATAPULT 1 , 10H SEPARATIO , /
      + DATA (1EVMS(1, 4), 1-1, 3) / 10HCATAPULT 2 , 10H SEPARATIO , /
      + DATA (1EVMS(1, 5), 1-1, 3) / 10HRAIL SEPAR , 10HATION , /
      + DATA (1EVMS(1, 6), 1-1, 3) / 10HROCKET 1 I , 10HIGNITION , /
      + DATA (1EVMS(1, 7), 1-1, 3) / 10HROCKET 2 I , 10HIGNITION , /
      + DATA (1EVMS(1, 8), 1-1, 3) / 10HROCKET 3 I , 10HIGNITION , /
      + DATA (1EVMS(1, 9), 1-1, 3) / 10HROCKET 4 I , 10HIGNITION , /
      + DATA (1EVMS(1, 10), 1-1, 3) / 10HROCKET 5 I , 10HIGNITION , /
      + DATA (1EVMS(1, 11), 1-1, 3) / 10HROCKET 6 I , 10HIGNITION , /
      + DATA (1EVMS(1, 12), 1-1, 3) / 10HROCKET 1 B , 10HURNOUT , /
      + DATA (1EVMS(1, 13), 1-1, 3) / 10HROCKET 2 B , 10HURNOUT , /
      + DATA (1EVMS(1, 14), 1-1, 3) / 10HROCKET 3 B , 10HURNOUT , /
      + DATA (1EVMS(1, 15), 1-1, 3) / 10HROCKET 4 B , 10HURNOUT , /
      + DATA (1EVMS(1, 16), 1-1, 3) / 10HROCKET 5 B , 10HURNOUT , /
      + DATA (1EVMS(1, 17), 1-1, 3) / 10HROCKET 6 B , 10HURNOUT , /
      + DATA (1EVMS(1, 18), 1-1, 3) / 10HROGUE GUN , 10H/CONTAIN , /
      + DATA (1EVMS(1, 19), 1-1, 3) / 10HDEPLOYMENT , 10HTE LINE ST , /
      + DATA (1EVMS(1, 20), 1-1, 3) / 10HROGUE CHU , 10HTE FULL IN , /
      + DATA (1EVMS(1, 21), 1-1, 3) / 10HVELCON DRD , 10HQUE SYSTEM , /
      + DATA (1EVMS(1, 22), 1-1, 3) / 10H , 10H , /
      + DATA (1EVMS(1, 23), 1-1, 3) / 10HSMALL CHUT , 10HTE FULL IN , /
      + DATA (1EVMS(1, 24), 1-1, 3) / 10HRECOVERY C , 10HUTE DEPLO , /
      + DATA (1EVMS(1, 25), 1-1, 3) / 10HRECOVERY C , 10HUTE LINE , /
      + DATA (1EVMS(1, 26), 1-1, 3) / 10HRECOVERY C , 10HUTE FULL , /
      + DATA (1EVMS(1, 27), 1-1, 3) / 10HPEAK TRAJE , 10HCTORY , /
      + DATA (1EVMS(1, 28), 1-1, 3) / 10HSEAT/OCCUP , 10HANT SEPARA , /

```

```

115 + DATA (IEVMES(1,29),I=1,3) /IOHSEAT/OCCUP , IOHANT IMPACT /
+ DATA (IEVMES(1,30),I=1,3) /IOHOCUPPANT A , IOHLONE IMPAC /
+ DATA (IEVMES(1,31),I=1,3) /IOHSEAT ALONE , IOH IMPACT /
+ DATA (IEVMES(1,32),I=1,3) /IOHAIRCRAFT 1 , IOHIMPACT /
+ DATA (IEVMES(1,33),I=1,3) /IOHDART START ,IOH RIGHT LIN. /
+ DATA (IEVMES(1,34),I=1,3) /IOHDART START ,IOH LEFT LINE /
+ DATA (IEVMES(1,35),I=1,3) /IOHDART STOP ,IOHRIGHT LINE /
+ DATA (IEVMES(1,36),I=1,3) /IOHDART STOP ,IOHLEFT LINE /
+ DATA (IEVMES(1,37),I=1,3) /IOHSEAT SEPAR ,IOHATION FROM /
+ DATA (IEVMES(1,38),I=1,3) /IOHSEAT FIRST ,IOH MOTION /
+
120 +
125 +
130 +
135 +
C.....
C ARRAY IEVENTS IS DIMENSIONED SUCH THAT EVERY EVENT THAT COULD HAPPEN *
C HAS A CORRESPONDING POSITION IN THE ARRAY. WHEN THE EVENT OCCURS. *
C THAT POSITION IS SET TO 1 AND A CORRESPONDING EVENT MESSAGE (FOUND *
C IN ARRAY IEVMES) IS PRINTED ON EVERY REPORT REQUESTED. *
C ALSO, AFTER A PARTICULAR EVENT HAS OCCURRED THE CORRESPONDING *
C POSITION IN ARRAY IEVENT IS ALSO SET TO TWO (2) TO INDICATE THAT THE *
C EVENT HAD OCCURRED AND THAT THE MESSAGE HAS BEEN PRINTED. LINE *
C COUNTS ARE KEPT (IN ARRAY LINECT) TO PREVENT PAGE RUN OFF. *
C.....
100 CONTINUE
IEVLINE=0
DO 500 I=1,MAXEVT
IF(IEVENTS(I) .NE. 1) GO TO 500
IEVENTS(I) = 2
IEVLINE = 1
DO 250 K=2, MAXREPT
IF(IREPTS(K) .EQ. 0) GO TO 250
LU = K + BIAS
WRITE(LU,7000) TIMES(I), (IEVMES(J,I),J=1,3)
LINECT(K) = LINECT(K) + 1
250 CONTINUE
500 CONTINUE
C.....
C THERE ARE SIX SPECIAL CASES WHICH ARE TESTED FOR BELOW. *
C IF A SPECIAL CONDITION IS FOUND THEN A FLAG IS SET (IN ARRAY ISPECL) *
C AND USED IN THE FOLLOWING DO LOOPS TO DETERMINE WHICH SPECIAL EVENT *
C MESSAGE IS TO BE PRINTED. THEN THE SPECIAL MESSAGES (FOUND IN ARRAY *
C IEVMES) ARE PRINTED ON EVERY REPORT THAT HAS BEEN REQUESTED *
C.....
IF (IEVENTS(32) .NE. 0 .AND. IEVENTS(37) .EQ. 0) ISPECL(1)=1
IF (IEVENTS(29) .NE. 0 .AND. IEVENTS(26) .EQ. 0) ISPECL(2)=1
IF (IEVENTS(30) .NE. 0 .AND. IEVENTS(26) .EQ. 0) ISPECL(3)=1
DO 800 I=1, 8
IF(ISPECL(I) .NE. 1) GO TO 800
DO 750 K=2, MAXREPT

```

PAGE 83

83/11/07. 09.41.53

FTN 4.6+428

SUBROUTINE EVENT 74/74 OPT=1

```

175 IF(I REPTS(K) .EQ. 0) GO TO 750
    LU = K + BIAS
    WRITE(LU,7500) TIME, (ISPMES(J,I),J=1,4)
    LINECT(K) = LINECT(K) + 1
    ISPECL(I) = 0
    750 CONTINUE
    800 CONTINUE
    RETURN
180 C.....
    C WRITE FORMAT STATEMENTS
    C.....
    7000 FORMAT(1X,F9.4,".....",1X,3(A10))
    C.....
    C WRITE FORMAT STATEMENTS FOR SPECIAL EVENT MESSAGES
    C.....
    7500 FORMAT(1X,F9.4,".....",1X,4(A10))
    END
185

```

```

1  SUBROUTINE HEADER
C.....
C  VERSION KGESSAB - DATED 7 NOVEMBER 1983
C.....
5  C  DESCRIPTION - LEVEL 3
C  FUNCTION - TO IDENTIFY SIMULATION RUNS ON REPORT PAGES
C.....
10 C  COMMUNICATIONS -
C  CALLED BY: ALL REPORTS
C  CALLS: NONE
C.....
15 C  SECTION 4 COMMON BLOCK
C.....
C  COMMON /IAIRCRT / TEMP , PRESSUR, ZACVEL , XPOS , YPOS ,
+ ZPOS , XTAIL , YTAIL , ZTAIL , YAW ,
+ PITCH , ROLL , RVEL , QVEL , PVEL ,
+ WINDX , WINDY , WINDZ , XACVEL , CKPITH ,
+ DENSITY , NPTSAAT , AAT(4.50) , NPTSLAT , LAT(4.50) ,
+ IACSFGL
C.....
20 C.....
C  SECTION 1 COMMON BLOCK
C.....
C  COMMON /ICONTROL / TSTART , TSTOP , ESTOP , IRESTRT , IUNITS ,
+ ISEATTR , ISOSEP , IPLOT , IDRIFLG ,
+ IPHASE1 , IPHASE2 , IPHASE3
+ INTEGER
+ ESTOP
C.....
30 C  SECTION 2 COMMON BLOCK
C.....
C  COMMON /IREPORT / IREPTS(31) , PRIFRQ,P11,P12,P13
+ INTEGER
C.....
35 C  SECTION 6 COMMON BLOCK
C.....
C  COMMON /ISEATOC / IPCNTL , XCGSO , YCGSO , ZCGSO , IXXSO ,
+ IXYSO , IYZSO , IYYSO , IYZSO , IZZSO ,
+ AREASO , AREAOA , WHTOAB , WHTOAA ,
+ IXXOA , IXYOA , IYZOA , IYYSOA , IYZOA ,
+ IZZOA , XCGOA , YCGOA , ZCGOA , ISOSEP ,
+ REAL
+ IXXSO , IXYSO , IYZSO , IYYSO , IYZSO ,
+ IZZSO , IXXOA , IXYOA , IYZOA , IYYSOA ,
+ IYZOA , IZZOA
C.....
45 C  MISCELLANEOUS DATA COMMON BLOCK
C.....
C  COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRTCT(31) ,
+ MAXLINE , MAXREPT , MAXEVRT ,
+ TEVLIN , TERRFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS ,
+ REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
+ IHEADER(24) , TEVENTS(38) , TIMES(38) ,
+ IMVDC , PRTEMP( 2) ,
+ PRIMASS(2) , PRTINDX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME

```

```

60      + INTEGER      XACCEL(3)      YACCEL(3)      ZACCEL(3)
      + REPTYPE      , BIAS      , PRTLNLT
      + PRTWGT      , PRTEMP      , PRTMAS      , PRTINDX
C.....
C IF IUNITS = 0 THEN METRIC VALUES ARE TO BE PRINTED
C IF IUNITS = 1 THEN ENGLISH VALUES ARE TO BE PRINTED
C THEREFORE CALCULATE AN INDEX INTO ARRAYS PRTLNLT AND PRTWGT
C USED FOR PRINTING THE PROPER UNITS OF MEASUREMENTS
C.....
      PRTINDX = IUNITS + 1
C.....
C LOGICAL UNIT NUMBERS FOR OUTPUT MUST BE GREATER THAN OR EQUAL TO
C FIVE (5) AND LESS THAN OR EQUAL TO THE MAX NUMBER OF REPORTS +
C A BIAS OF FOUR (4) . NOTE: REPORT1 IS WRITTEN TO LU 5,
C REPORT2 IS WRITTEN TO LU 6 AND SO ON
C.....
      IF((LU.LT. 5) OR (LU.GT. (MAXREPT + 4))) GO TO 9900
C.....
C WRITE HEADER INFO.
C.....
      K=LU-BIAS
      IPAGECT(K)=IPAGECT(K)+1
      LINECT(K) = 0
      WRITE (LU,7001) IDATE,IPAGECT(K)
      WRITE(LU,7002) (IHEADER(I),I=1,16)
      WRITE(LU,7003) (IHEADER(I),I=17,24)
      WRITE(LU,7004) HEADALT, PRTLNLT(PRTINDX), HEADSR, PRTLNLT(PRTINDX)
      + PRTLNLT(PRTINDX), HEADDEL, PRTLNLT(PRTINDX)
      WRITE(LU,7005) HEADYAW, HEADPIT, HEADROL
      WRITE(LU,7006) HEADWGT, PRTWGT(PRTINDX)
      WRITE(LU,8000)
C.....
C CALCULATE THE INDEX INTO ARRAY REPTYPE AND PRINT THE REPORT TYPE
C NOTE: EACH REPORT HAS A REPORT TYPE IN ARRAY REPTYPE
C I E. REPORT6 USES REPTYPE(1,6) THRU REPTYPE(5,6)
C AND REPORT8 USES LU (LOGICAL UNIT) 10 FOR OUTPUT
C.....
      WRITE(LU,7007) K, (REPTYPE(I,K),I=1,5)
      GO TO 9999
C.....
C WRITE AN ERROR MESSAGE OUT TO REPORT1
C AND SET ERROR FLAG (IERRFLG) TO ONE(1)
C TO INDICATE TO THE CALLING ROUTINE THAT A FATAL ERROR HAS OCCURRED
C.....
      9900 CONTINUE
      WRITE(5,9001) LU
      IERRFLG = 1
C.....
C RETURN TO THE CALLING ROUTINE
C.....
      9999 CONTINUE
      RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....
      7001 FORMAT(1H1,12X,"DATE: ",A10,90X,"PAGE ",I3)

```

```

115 7002 FORMAT(2X,"RUN DESCRIPTION: ",8(A10),/,19X,8(A10))
7003 FORMAT(1X,"SEAT DESCRIPTION: ",8(A10))
7004 FORMAT(2X,"TEST CONDITIONS: ",12X,"ALTITUDE: ",
+ F10.2,1X,A2,12X,"X VELOCITY: ",F10.2,1X,A2,"/SEC",6X,
+ Z VELOCITY: ",F10.2,1X,A2,"/SEC")
120 7005 FORMAT(19X,"YAW: ",F10.2,1X,"DEG ",13X,"PITCH: ",
+ F10.2,1X,"DEG ",12X,"ROLL: ",F10.2,1X,"DEG")
7006 FORMAT(7X,"TOTAL EJECTED WEIGHT: ",F10.2,1X,A4)
7007 FORMAT(6X,"REPORT NO.",13," ",7(A10))
8000 FORMAT(5X,"COORDINATE CONVENTIONS: +X= FORWARD +Y= LEFT +Z= UP",
+ " +YAW= TURN LEFT +PITCH= NOSE DOWN +ROLL= RIGHT WING DOWN")
125 C.....
C FORMATS FOR FATAL ERROR MESSAGES
C.....
9001 FORMAT(1X,72(1H*))/,4X,"FATAL ERROR(SUBROUTINE HEADER)*** ",/,
+ 72(1H*))
130 END

```

```

1      SUBROUTINE IDIRMTX (DCM,PHI,PSI,THETA)
C .....
C DESCRIPTION - LEVEL 4
C FUNCTION - FUNCTIONAL SUBROUTINE TO SET UP INITIAL TRANSFORMATION
C .....
C MATRICES
5      METHOD - USES INPUT MODIFIED EULER ANGLES
C COMMUNICATIONS -
C CALLED BY: INITMS
C .....
C CALLS:
10     NONE
C NON-COMMON VARIABLES DEFINED -
C DCM - COMPUTED 3X3 MATRIX (RETURNED IN CALL)
C PHI - ROTATION ABOUT THE X-AXIS IN RADIAN (PASSED IN CALL)
15     PSI - ROTATION ABOUT THE Y-AXIS IN RADIAN (PASSED IN CALL)
C THETA - ROTATION ABOUT THE Z-AXIS IN RADIAN (PASSED IN CALL)
C SINPHI )
C COSPHI )
C SINPSI ) - INTERMEDIATE SINE AND COSINE VALUES
20     COSPSI )
C SIN THE )
C COSTHE )
C POTENTIAL ERROR CONDITIONS -
25     NONE
C .....
C DIMENSION DCM(3,3)
C SINPHI = SIN(PHI)
C COSPHI = COS(PHI)
C SINPSI = SIN(PSI)
C COSPSI = COS(PSI)
C SIN THE = SIN(THETA)
C COSTHE = COS(THETA)
C DCM(1,1) = COSPSI * COSTHE + SINPHI * SINPSI * COSTHE
35     DCM(2,1) = -SIN THE * COSPHI + COSPHI * SINPSI * COSTHE
C DCM(3,1) = SIN THE * SINPHI + COSPHI * SINPSI * COSTHE
C DCM(1,2) = COSPSI * SIN THE
C DCM(2,2) = COSTHE * COSPHI + SINPHI * SINPSI * SIN THE
C DCM(3,2) = -SINPHI * COSTHE + COSPHI * SINPSI * SIN THE
40     DCM(1,3) = -SINPSI
C DCM(2,3) = COSPSI * SINPHI
C DCM(3,3) = COSPSI * COSPHI
C RETURN
C END

```


NADC-84068-60

```

1      SUBROUTINE INIQUA (DCM,Q)
C .....
C DESCRIPTION - LEVEL 4
C FUNCTION - COMPUTES INITIAL NORMALIZED QUATERNIONS
C METHOD - USES DIRECTION COSINE MATRIX PASSED BY CALLING ROUTINE
C - SEE "FACTORED COMPLEX QUATERNIONS", FAAC REPORT NO.
C FR 3689/2557, PAGES 23-28
C COMMUNICATIONS -
C CALLED BY: INITMS
C CALLS:
C NONE
C NON-COMMON VARIABLES DEFINED:
C Q - EQUAL TO QUATERN(1) PASSED IN SUBROUTINE CALL
C DCM - EQUAL TO DIRECTION COSINE MATRIX PASSED IN SUBR. CALL
C POTENTIAL ERROR CONDITIONS:
C NONE
C .....
C DIMENSION Q(4), S(4), DCM(3,3), A(3,3)
C L = 0
C
C DO 10 I=1,3
C DO 10 J=1,3
C A(I,J)= DCM(I,J)
C 10 CONTINUE
C .....
C FIND QUATERNIONS
C .....
C 25 CONTINUE
C S(1) = 0.5 * SORT(ABS(1.0 + A(1,1) + A(2,2) + A(3,3)))
C IF (S(1) .GT. 0.25) GOTO 100
C .....
C FIND COLUMN CONTAINING MAXIMUM DIAGONAL ELEMENT
C .....
C L = 1
C IF ((A(2,2) .GT. A(1,1)) .AND. (A(2,2) .GT. A(3,3))) L = 2
C IF ((A(3,3) .GT. A(1,1)) .AND. (A(3,3) .GT. A(2,2))) L = 3
C .....
C REVERSE SIGN IN THE OTHER TWO COLUMNS
C .....
C DO 75 J = 1,3
C IF (J .EQ. L) GOTO 75
C DO 50 I = 1,3
C A(I,J) = -A(I,J)
C 50 CONTINUE
C 75 CONTINUE
C GOTO 25
C
C 100 CONTINUE
C FS = 4.0 * S(1)
C
C S(2) = (A(2,3) - A(3,2))/FS
C S(3) = (A(3,1) - A(1,3))/FS
C S(4) = (A(1,2) - A(2,1))/FS
C

```

SUBROUTINE INIQUA 74/74 OPT=1

IF (L .GT. 0) GOTO 150

C

60

Q(1) = S(1)
 Q(2) = S(2)
 Q(3) = S(3)
 Q(4) = S(4)
 RETURN

65

C 150 CONTINUE

IF (L .GT. 1) GOTO 200

Q(1) = S(2)
 Q(2) = -S(1)
 Q(3) = S(4)
 Q(4) = -S(3)
 RETURN

70

C

200 CONTINUE

IF (L .GT. 2) GOTO 250

Q(1) = S(3)
 Q(2) = -S(4)
 Q(3) = -S(1)
 Q(4) = S(2)
 RETURN

80

C

250 CONTINUE

Q(1) = S(4)
 Q(2) = S(3)
 Q(3) = -S(2)
 Q(4) = -S(1)

85

C

RETURN
 END

```

1 SUBROUTINE INITMS
C .....
C DESCRIPTION - INITIALIZES THE TRANSFORMATION MATRICES, TVC DIRECTION
C FUNCTION - COSINE ARRAY AND THE QUATERNION ARRAYS.
C .....
5 C METHOD - CALLS FUNCTION SUBROUTINES IDIRMTX AND MATRIX WITH
C APPROPRIATE PARAMETERS TO SET UP THE FOLLOWING TRANS-
C FORMATION MATRICES:
C .....
10 C DCMAR - EARTH-FIXED C.S. TO AIRCRAFT C.S.
C DCMRA - AIRCRAFT C.S. TO RAIL C.S.
C DCMRA - AIRCRAFT C.S. TO SEAT C.S.
C DCMRA - RAIL C.S. TO SEAT C.S.
C DCMSE - EARTH-FIXED C.S. TO SEAT C.S.
C DCMST - SEAT C.S. TO TVC C.S.
C DCMET - EARTH-FIXED C.S. TO TVC C.S.
C .....
15 C CALLS FUNCTION SUBROUTINE INIQUA WITH APPROPRIATE
C PARAMETERS TO SET UP THE QUATERNION ARRAYS USED TO
C UPDATE THE SEAT/OCCUPANT AND AIRCRAFT TRAJECTORIES.
C .....
C COMMUNICATIONS -
C CALLED BY: INTLZ
C .....
C CALLS: IDIRMTX
C INIQUA
C MATRIX
C .....
25 C NON-COMMO VARIABLES DEFINED -
C .....
C PSISR - ROTATION ABOUT THE Y AXIS FROM RAIL TO SEAT C.S.
C .....
C THETA )
C PHI ) EULER ANGLES FOR CALL TO INITIALIZE QUATERNIONS
C PSI )
C .....
C POTENTIAL ERROR CONDITIONS -
C NONE
C .....
35 C .....
C SECTION 4 COMMON BLOCK
C .....
C COMMON /IAIRCRT / TEMP , PRESSUR, ZACVEL , XPOS , YPOS ,
+ ZPOS , XTAIL , YTAIL , ZTAIL , YAW ,
+ PITCH , ROLL , RVEL , QVEL , PVEL ,
+ WINDX , WINDY , WINDZ , XACVEL , CKPITH,
+ DENSITY, NPISAAT, AAT(4.50), NPISLAT,LAT(4.50),
+ JACSFLG
C .....
45 C .....
C SECTION 7 COMMON BLOCK
C .....
C COMMON /IRAIL / RAILNTH , RAILANG , ISTRL , NSLBKS ,
+ KXSB , KYSB , YPOSRR , ZPOSRR ,
+ XPOSRL , YPOSRL , ZPOSRL ,
+ XPOSSB(6) , YPOSSB(6) , ZPOSSB(6) ,
+ REAL KXSB , KYSB , MUSB
C .....
55 C .....
C SECTION 5 COMMON BLOCK
C .....
C COMMON /ISETAIN / XPOSSRP, YPOSSRP, ZPOSSRP, XCGSA , YCGSA ,

```

```

+ ZCGSA , IXRSA , IXYSA , IKZSA , IYISA ,  

+ IYZSA , IZZSA , PHISA , PSISA , THESA ,  

+ AREASA , HIGHTSA , WGHISA , XPOSBOT , YPOSBOT ,  

+ ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS  

+ IXRSA , IXYSA , IKZSA , IYISA , IYZSA ,  

+ IZZSA  

C .....  

C SECTION 12 COMMON BLOCK  

C .....  

COMMON /ITVCIN / ITVC , MPHI , MPSI , MTHE ,  

ROLLRL , PITCHRL , SMPLRAT , TVCDLAY ,  

RKANG  

+ REAL  

+ MPHI , MPSI , MTHE  

C .....  

C MATRIX COMMON BLOCK  

C .....  

COMMON /MATRIX / DCMAE(3,3) , DCMRA(3,3) , DCMSE(3,3) ,  

DCMSE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,  

DCMSAE(3,3) , DCMOAE(3,3) , DCMSR(3,3) ,  

DCMDUM(3,3)  

C .....  

C INTEGRATION ROUTINE COMMON BLOCK  

C .....  

COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSOI(193) ,  

TRAJSA(193) , TRAUJA(193) , TRAJCH(97,3) ,  

TRAJAC(193) , TVEQOS(225) , QUATOAI(65) ,  

QUATOA(65) , QUATAC(65) ,  

INTSTP , IPCPASS , IRKPASS ,  

IPOINTS , IYX , IYPRX ,  

IKX , IKSUMX , IKPASSX ,  

IYIX , IYIIX , IYI2X ,  

IYI3X , IYPRIIX , IYPRIIX ,  

IYPRI2X , IPYIX , IPYIIX ,  

ICYIX , ICYIIX , IREIN  

+  

C .....  

C THRUST VECTOR CONTROL VARIABLES COMMON BLOCK  

C .....  

COMMON/TUCVRB / ITVCFGL , CMPVAL , D3(3) , RKTGMND(3) ,  

C29 , C30 , C31 , C32 ,  

DTH(3,2) , ANGR(3)  

+  

C .....  

C SET UP TRANSFORMATION MATRICES  

C .....  

CALL IDIRMTX(DCMAE,ROLL,PITCH,YAW)  

CALL IDIRMTX(DCMRA,O.O,RAILANG,O.O)  

CALL IDIRMTX(DCMSA,PHISA,PSISA,THESA)  

PSISR = PSISA - RAILANG  

CALL IDIRMTX(DCMSR,O.O,PSISR,O.O)  

CALL MATRIX(DCMSA,DCMAE,DCMSE,1)  

IF (ITVC EQ. O) GO TO 200  

CALL IDIRMTX(DCMTS,MPSI,MPSI,MTHE)  

CALL MATRIX(DCMIS,DCMSE,DCMTE,1)  

DO 100 I=1,3  

D3(I) =  

+DCMTE(1,3)/(SORT(DCMTE(I,1)+DCMTE(I,1)*DCMTE(I,2)+DCMTE(I,2)+  

+DCMTE(I,3)*DCMTE(I,3)))  

100 CONTINUE

```

```

115 C .....
C SET UP QUATERNIONS
C .....
200 CONTINUE
    CALL INIQUA(DCMSE,QUATSO(2))
    QUATSO(1) = 4.
    CALL INIQUA(DCMSE,QUATAC(2))
    QUATAC(1) = 4.
    RETURN
    END
120

```

```

1      SUBROUTINE INITRAJ
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - SETS UP INITIAL TRAJECTORY VALUES (LINEAR POSITIONS AND
C VELOCITIES AND ANGULAR POSITIONS AND VELOCITIES)
C METHOD - USES INPUT CONDITIONS AND CONVERTS TO THE PROPER
C COORDINATE SYSTEMS
C COMMUNICATIONS -
C CALLED BY:
C INTLZ
C CALLS:
C ROTATE, ZLININT
C NON-COMMON VARIABLES DEFINED -
C NONE
C POTENTIAL ERROR CONDITIONS :
C NONE
C .....
C MATRIX COMMON BLOCK
C .....
C COMMON /MATRIX / DCMAR(3,3), DCMRA(3,3), DCMSE(3,3), DCMTE(3,3),
+ DCMSE(3,3), DCMTE(3,3), DCMSE(3,3), DCMTE(3,3),
+ DCMSE(3,3), DCMTE(3,3), DCMSE(3,3), DCMTE(3,3),
+ DCMSE(3,3), DCMTE(3,3), DCMSE(3,3), DCMTE(3,3),
C .....
25  C .....
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C COMMON /MISC / IPAGCT(31), LINECT(31), IPRCNT(31),
+ MAXREPT, MAXEVT,
+ TERRFLG, LU,
+ HEADALT, HEADVEL,
+ HEADPIT, HEADWT,
+ PRTLNGT(2), PRTWGT(2),
+ REPTYPE(5,31), PRTLNGT(2), PRTWGT(2),
+ IHEADER(24), IEVENTS(38),
+ INVDC, PRTMP( 2),
+ PRIMASS(2), PRINDX, PKZVEL,
+ ZVECT(3), XYZ(3), SAVTIME,
+ XACCEL(3), YACCEL(3), ZACCEL(3),
+ REPTYPE, BIAS, PRTLNGT,
+ PRTWGT, PRTEMP, PRIMASS, PRINDX
C .....
45  C SECTION 4 COMMON BLOCK
C .....
C COMMON /TAIRCRT / TEMP, PRESSUR, ZACVEL, XPOS, YPOS,
+ ZPOS, XTAL, YTAIL, ZTAIL, YAW,
+ PITCH, ROLL, RVEL, QVEL, PVEL,
+ WINDX, WINDY, WINDZ, XACVEL, CKPITH,
+ DENSITY, NPTSAAT, AAT(4,50), NPTSLAT, LAT(4,50),
+ IACSEFLG
C .....
55  C MOMARMS COMMON BLOCK
C .....
C COMMON /MOMARMS /
+ REFLNSO, REFLNOA, REFLNSA, URX(6), URY(6), URZ(6)

```

```

+XSSOCA(2),YSSOCA(2),ZSSOCA(2),XSSORK(6),YSSORK(6),ZSSORK(6),
+XSSORRE,YSSORRE,ZSSORRE,XSSOLRE,YSSOLRE,ZSSOLRE,
+XSSOMRE,YSSOMRE,ZSSOMRE,XSSOBOT,YSSOBOT,ZSSOBOT,
+XSSOSB(6),YSSOSB(6),ZSSOSB(6),XRRCSAC,YRRCSAC,ZRRCSAC,
+XSSCSAC,YSSCSAC,ZSSCSAC,XSSOSRP,YSSOSRP,ZSSOSRP,
+XSSASRP,YSSASRP,ZSSASRP,XRRDAP(2),YRRDAP(2),ZRRDAP(2),
+XRRSBO(6),YRRSBO(6),ZRRSBO(6),XSSOCP(2),YSSOCP(2),ZSSOCP(2),
+XSSDAP(2),YSSDAP(2),ZSSDAP(2),XESOAC,YESOAC,ZESOAC,
+XSRCSAC,YSRCSAC,ZSRCSAC,XSSOAC,YSSOAC,ZSSOAC,
+XRSOSB,YRSOSB,ZRSOSB,XRRSBO,YRRSBO,ZRRSBO,ZRRSBO(3),
+XRRSB,YRRSB,ZRRSB,XSSOCH(3),YSSOCH(3),ZSSOCH(3),
+XAACSD,YAACSD,ZAACSD,XASOAC,YASOAC,ZASOAC,
+XRSOAC,YRSOAC,ZRSOAC,XSCPAP(2),YSCPAP(2),ZSCPAP(2),
C *****
C INTEGRATION ROUTINE COMMON BLOCK
C *****
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193) ,
+ TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3) ,
+ TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
+ QUATOA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IPOINITS , IYX , IYPRX , IYKSUMX ,
+ IKX , IKSUMX , IKPASSX ,
+ IYIX , IYIIX , IYI2X ,
+ IYI3X , IYPRIIX , IYI2X ,
+ IYPR12X , IYVIX , IYI1X ,
+ ICYIX , ICYIIX , ICYIIX , IREIN ,
C *****
C SECTION 14 COMMON BLOCK
C *****
COMMON /PARCHUT / IRECOV , TRDPLOY , RECOVLL ,
+ RECDRAG , RECOVPD , POROSR ,
+ XRECAP , YRECAP , ZRECAP ,
+ NPTSRLS , RECOVLS(2,25) , IFTRFCV ,
+ NPTSRT , RECOVFT(2,25) , SEPFCE ,
+ IDROGUE , DRDRAG2 , DRGPD2 ,
+ POROSD2 , VELCON , IFTRD02 ,
+ NPTDFT2 , DRGFT2(2,25) , IFTRD01 ,
+ NPTDFT1 , DRGFT1(2,25) , IDROGLS ,
+ NPTSOLS , DRGOLS(2,25) , TDDPLOY ,
+ DISPLAY , DRGOLL , DRDRAG1 ,
+ DRGPD1 , POROSD1 , DROVELX ,
+ DROVELY , DROVELZ , XDROGAP ,
+ YDROGAP , ZDROGAP , CHALT1 ,
+ CHALT2 , GLIMIT , TDELAY ,
+ AREADC , WGHOTC , TFP1 ,
+ TFP2 , TFP3 , TOROGLS ,
+ CDDC , NPTSRTD , RECOVDI(2,25) ,
C *****
C SET UP INITIAL AIRCRAFT TRAJECTORY VARIABLES
C *****
TRAJAC(1)=12,
TRAJAC(2) = XPOS
TRAJAC(3) = YPOS
TRAJAC(4) = ZPOS

```

```

115      TRAJAC(14) = XACVEL
      TRAJAC(15) = O.O
      TRAJAC(16) = ZACVEL
      CALL ROTATE(TRAJAC(14),TRAJAC(5),ZVECT(1),DCMAE,O)
120      TRAJAC(8) = ROLL
      TRAJAC(9) = PITCH
      TRAJAC(10) = YAW
      TRAJAC(11) = PVEL
      TRAJAC(12) = QVEL
      TRAJAC(13) = RVEL

125      C .....
      C SETUP AIRCRAFT ACCELERATIONS ACCORDING TO COORDINATE SYSTEM
      C AND NUMBER OF ACCELERATION POINTS
      C .....
      C IACSFILG = MEANING:
      C   0      CONSTANT AIRCRAFT LINEAR ACCELERATION IN ACS:
      C           NO ANGULAR MOTION
      C   1      CONSTANT AIRCRAFT LINEAR ACCELERATION IN ACS:
      C           WITH ANGULAR MOTION
      C   2      VARIABLE AIRCRAFT LINEAR ACCELERATION IN EFCS:
      C           NO ANGULAR MOTION
      C   3      VARIABLE AIRCRAFT LINEAR ACCELERATION IN EFCS:
      C           WITH ANGULAR MOTION
      C .....

130      TRAJAC(23) = AAT(2,1)
      TRAJAC(24) = AAT(3,1)
      TRAJAC(25) = AAT(4,1)
      TRAJAC(17) = LAT(2,1)
      TRAJAC(18) = LAT(3,1)
      TRAJAC(19) = LAT(4,1)
145      IF(NPTS LAT NE 1) GO TO 100
      IF((NPTS LAT NE O) OR (PVEL NE O.O) OR (QVEL NE O.O)
      * OR (RVEL NE O.O)) IACSEFLG = IACSEFLG + 1
      IF(IACSEFLG NE 2) GO TO 100
      CALL ROTATE(TRAJAC(17),TRAJAC(17),ZVECT(1),DCMAE,O)
      IACSEFLG = O
150      100 CONTINUE
      C .....
      C SET UP INITIAL SEAT/OCCUPANT TRAJECTORY VARIABLES
      C .....
      TRAJSO(1)=12
      TRAJSO(2)=XPOS-XESDAG
      TRAJSO(3)=YPOS-YESDAG
      TRAJSO(4)=ZPOS-ZESDAG
      TRAJSO(5) = TRAJAC(5) + OVEL*ZAACSO-RVEL*YAACSO
      TRAJSO(6) = TRAJAC(6) + RVEL*XAACSO-PVEL*ZAACSO
      TRAJSO(7) = TRAJAC(7) + PVEL*YAACSO-QVEL*XAACSO
      CALL ROTATE (TRAJSO(5),TRAJSO(5),ZVECT,DCMSA,O)
      CALL ROTATE (TRAJAC(11),TRAJSO(11),ZVECT,DCMSA,O)
165      C .....
      C DETERMINE RECOVERY CHUTE DEPLOYMENT TIME BASED ON TABLE DATA FOR
      C DEPLOYMENT TIME VS TOTAL S/O VELOCITY
      C .....
      IF(NPTS RDT EQ 1) GO TO 200
      XVEL = TRAJAC(14) - WINDX
      YVEL = TRAJAC(15) - WINDY
      ZVEL = TRAJAC(16) - WINDZ
170

```


NADC-84068-60

SUBROUTINE INITRAJ 74/74 OPT=1 FTN 4.6+428 83/11/07. 09.41.53 PAGE 96

TVEL = SORT(XVEL*XVEL + YVEL*YVEL + ZVEL*ZVEL)
CALL ZLININT(TVEL,RECOVDI,NPTSROD,25,TRDPLOY,2)
200 CONTINUE
RETJRN
END

175

```

1  SUBROUTINE INIVECT
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - INITIALIZES ALL PROGRAM VECTORS
C METHOD - INITIALIZES ALL MOMENT ARMS, AERODYNAMIC REFERENCE
C LENGTHS, AND ROCKET THRUST LINE UNIT VECTORS
C COMMUNICATIONS -
C CALLED BY:
C INTLZ
C CALLS:
C ROTATE
C NON-COMMON VARIABLES DEFINED -
C NONE
C POTENTIAL ERROR CONDITIONS -
C NONE
C .....
C CONSTANTS COMMON BLOCK
C .....
C COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C .....
C SECTION 4 COMMON BLOCK
C .....
C COMMON /TAIRCRT / TEMP , PRESSUR, ZACVEL , XPOS , YPOS ,
+ ZPOS , XTAIL , YTAIL , ZTAIL , YAW ,
+ PITCH , ROLL , RVEL , QVEL , PVEL ,
+ WINDX , WINDY , WINDZ , XACVEL , CKPITH ,
+ DENSITY, NPTSAAT, AAT(4.50), NPTSLAT,LAT(4.50),
+ IACSEFLG
C .....
C SECTION 9 COMMON BLOCK
C .....
C COMMON /ICATPLT / INCAT , CATLNT(2), CATSTK(2), ICI (2),
+ XPOSAP(2) , YPOSAP(2), ZPOSAP(2), NPTSCT(2),
+ CATHRST(2.25,2), ITUBEND , KTUBE , CTUBE ,
+ PTUBE , MUTUBE , EXTLNGT , ICATOUT
+ REAL
+ KTUBE , MUTUBE
C .....
C SECTION 11 COMMON BLOCK
C .....
C COMMON /IDARTIN / IDART , DRTFCE , DRTSTRT , DRTSTOP ,
+ XDRTAP(2), YDRTAP(2), ZDRTAP(2),
+ XDRTCP(2), YDRTCP(2), ZDRTCP(2)
C .....
C SECTION 7 COMMON BLOCK
C .....
C COMMON /TRAIL / RAILNTH , RAILANG , ISTRL , NSLBKS ,
+ KXSB , KYSB , ZPOSRR , ZPOSRR , YKTOR ,
+ XPOSRR , YPOSRR , ZPOSRR ,
+ XPOSRL , YPOSRL , ZPOSRL ,
+ XPOSBB(6), YPOSBB(6), ZPOSBB(6)
+ REAL
+ KXSB , KYSB , MUSB
C .....
C SECTION 10 COMMON BLOCK
C .....
C COMMON /IROCKET / INRKT , RKDELY(6), RKNPTS(6), IROKOUT ,
+ RKIGN(6) , RKWGT(6), RKBURN(6), TSTAR(6) ,
+

```

```

60      + XPOSRK(6), YPOSRK(6), ZPOSRK(6),
      + RKALPH(6), RKBETA(6), RKGAMA(6), RKTHRS(2,25,6)
      + RKNPTS
C.....
C SECTION 6 COMMON BLOCK
C.....
COMMON /ISEATOC / IPCNTL, XCGSO, YCGSO, ZCGSO, IAXSO,
+ IXYSO, IYZSO, IYYSO, IZZSO,
+ AREASO, AREADA, WGHTOAB, WGHTOAA,
+ IXXOA, IXVOA, IYVOA, IYZOA,
+ IZZOA, XCGOA, YCGOA, ZCGOA, SOSEP,
+ REAL
+ IXYSO, IYZSO, IYYSO, IZZSO,
+ IXXOA, IXVOA, IYVOA, IZZOA
C.....
C SECTION 5 COMMON BLOCK
C.....
COMMON /ISETALN / XPOSSRP, YPOSSRP, ZPOSSRP, XCGSA, YCGSA,
+ ZCGSA, IXXSA, IYXSA, IYZSA, IYXSA,
+ IYZSA, IZZSA, PHISA, PSISA, THESA,
+ AREASA, HGHTSA, WGHYSA, XPOSBOT, YPOSBOT,
+ ZPOSBOT, XPOSSCS, YPOSSCS, ZPOSSCS,
+ REAL
+ IXXSA, IYXSA, IYZSA, IYXSA, IYZSA,
+ IZZSA
C.....
C MATRIX COMMON BLOCK
C.....
COMMON /MATRIX / DCMAC(3,3), DCMRA(3,3), DCMXA(3,3),
+ DCMSE(3,3), DCMTS(3,3), DCMTE(3,3),
+ DCMOAE(3,3), DCMOAE(3,3), DCMOAE(3,3),
+ DCMOUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31), LINECT(31), IPRTCT(31),
+ MAXLINE, MAXREPT, MAXEVT,
+ IEVLINE, IERRFLG, LU,
+ IDATE, HEADALT, HEADVEL,
+ HEADSR, HEADWT, HEADPIT,
+ HEADROL, HEADWGT, BIAS,
+ REPTYPE(5,31), PRTLNCT(2), PRTWGT(2),
+ IHEADER(24), IEVENTS(38), TIMES(38),
+ PRIMASS(2), PRINDX, PKZVEL,
+ ZVECT(3), XYZ(3), SAVTIME,
+ XACCEL(3), YACCEL(3), ZACCEL(3),
+ REPTYPE, BIAS, PRTLNCT,
+ PRTWGT, PRIMASS, PRINDX
+ INTEGER
+ PRTWGT, PRIMASS, PRINDX
C.....
C MOMARMS COMMON BLOCK
C.....
COMMON /MOMARMS /
+ REFLNSO, REFLNSA, URX(6), URY(6), URZ(6),
+ XSSOCA(2), YSSOCA(2), ZSSOCA(2), XSSORK(6), YSSORK(6), ZSSORK(6),
+ XSSORRE, YSSORRE, ZSSORRE, XSSOLRE, YSSOLRE, ZSSOLRE,
+ XSSOMRE, YSSOMRE, ZSSOMRE, XSSOBOT, YSSOBOT, ZSSOBOT

```

```

115 +XSSOSB(6),YSSOSB(6),ZSSOSB(6),XRRCSAC ,YRRCSAC ,ZRRCSAC ,
+XSSCSAC ,YSSCSAC ,ZSSCSAC ,XSSSRP ,YSSSRP ,ZSSSRP ,
+
+XSSASRP ,YSSASRP ,ZSSASRP ,XRRDAP(2),YRRDAP(2),ZRRDAP(2),
+XRRSBO(6),YRRSBO(6),ZRRSBO(6),XSSOCP(2),YSSOCP(2),ZSSOCP(2),
+XSSDAP(2),YSSDAP(2),ZSSDAP(2),XESDAC ,YESDAC ,ZESDAC ,
+XRRCSAC ,YRRCSAC ,ZRRCSAC ,XSSOAC ,YSSOAC ,ZSSOAC ,
+XRRSBO ,YRRSBO ,ZRRSBO ,XRRSBO ,YRRSBO ,ZRRSBO ,
+XRRSB ,YRRSB ,ZRRSB ,XSSOCH(3),YSSOCH(3),ZSSOCH(3),
+XACSO ,YACSO ,ZACSO ,XASOAC ,YASOAC ,ZASOAC ,
+XRSOAC ,YRSOAC ,ZRSOAC ,XSCAP(2),YSCAP(2),ZSCAP(2)
125 C*****
C SECTION 14 COMMON BLOCK
C*****
130 COMMON /PARCHUT / IRECOV ,
+ RECDRAG ,
+ XRECAP ,
+ NPTSRLS ,
+ NPTSRLT ,
+ IDROGUE ,
+ POROSD2 ,
+ NPTDFT2 ,
+ NPTDFT1 ,
+ NPTDLS ,
+ DISPLOY ,
+ DROGPD1 ,
+ DROVELY ,
+ YDROGAP ,
+ CHALT2 ,
+ AREADC ,
+ TFP2 ,
+ CDDC ,
+
+ TRDPLAY ,
+ RECOVLD ,
+ POROSR ,
+ YRECAP ,
+ RECOVLS(2,25) ,
+ RECOVLT(2,25) ,
+ SEPRCE ,
+ DRDAG2 ,
+ DROGPD2 ,
+ VELCON ,
+ IFDRO2 ,
+ DROGFT2(2,25) ,
+ IFDRO1 ,
+ DROGFT1(2,25) ,
+ IDROGLS ,
+ DROGLS(2,25) ,
+ TDDPLAY ,
+ DRDAG1 ,
+ POROSD1 ,
+ DROVELX ,
+ XDRGAP ,
+ CHALT1 ,
+ TDELAY ,
+ TFP1 ,
+ TDRGLS ,
+ RECOVDT(2,25)
145 C*****
C SET UP AERODYNAMIC REFERENCE LENGTHS
C*****
150 C
REFLNSO = SQRT(4.0*AREASO/P1)
REFLNOA = SQRT(4.0*AREAOA/P1)
REFLNSA = SQRT(4.0*AREASA/P1)
155 C*****
C SET UP ROCKET THRUST LINE UNIT VECTOR PROJECTED ON SCS FOR EACH ROCKET
C*****
C
IF (INRKT EQ 0) GO TO 200
DO 100 I=1,INRKT
URX(I) = COS(RKALPH(I))
URY(I) = COS(RKBETA(I))
URZ(I) = COS(RKGAMA(I))
100 CONTINUE
160 C*****
C SET UP MOMENT ARMS (VECTORS FROM SEAT/OCCUPANT CG TO VARIOUS
C ATTACHMENT POINTS)
C*****
170 C*****
C SET UP VECTORS FROM SEAT/OCCUPANT CG TO EACH CATAPULT ATTACHMENT
C POINT IN SCS

```

```

C .....
200 CONTINUE
  IF (INCAT .EQ. 0) GO TO 210
  DO 201 I=1, INCAT
    XSSOCA(1)=XPOSAP(1)-XCGSO
    YSSOCA(1)=YPOSAP(1)-YCGSO
    ZSSOCA(1)=ZPOSAP(1)-ZCGSO
  201 CONTINUE
  C .....
  C SET UP VECTORS FROM SEAT/OCCUPANT CG TO EACH ROCKET ATTACHMENT POINT
  C IN SC5
  C .....
  210 CONTINUE
    IF (INRKT .EQ. 0) GO TO 220
    DO 211 I=1, INRKT
      XSSORK(1) = XPOSRK(1) - XCGSO
      YSSORK(1) = YPOSRK(1) - YCGSO
      ZSSORK(1) = ZPOSRK(1) - ZCGSO
    211 CONTINUE
    C .....
    C SET UP VECTOR FROM SEAT/OCCUPANT CG TO RIGHT RAIL END
    C .....
  220 CONTINUE
    C .....
    C SET UP VECTOR FROM SEAT/OCCUPANT CG TO LEFT RAIL END
    C .....
    C .....
    C .....
    C SET UP VECTOR FROM SEAT/OCCUPANT CG TO ORIGIN OF RCS
    C .....
    CALL ROTATE (ZVECT, XSSOMRE, XPOSSCS, DCM5R, 0)
    XSSOMRE=XSSOMRE-XCGSO
    YSSOMRE=YSSOMRE-YCGSO
    ZSSOMRE=ZSSOMRE-ZCGSO
    C .....
    C SET UP VECTOR FROM SEAT/OCCUPANT CG TO SEAT BOTTOM
    C .....
    CALL ROTATE(XPOSBOT, XSSOBOT, XPOSSCS, DCM5R, 0)
    XSSOBOT = XSSOBOT - XCGSO
    YSSOBOT = YSSOBOT - YCGSO
    ZSSOBOT = ZSSOBOT - ZCGSO
    C .....
    C SET UP VECTOR FROM SEAT/OCCUPANT CG TO EACH SLIDER BLOCK (SLIPPER)
    C .....
    IF (NSLBKS .EQ. 0) GO TO 222
    DO 221 I=1, NSLBKS
      XSSOSB(1) = XPOSSB(1) - XCGSO
      YSSOSB(1) = YPOSSB(1) - YCGSO
      ZSSOSB(1) = ZPOSSB(1) - ZCGSO
    221 CONTINUE
    GO TO 230
  222
  C .....
  C NSLBKS=0 IS A SPECIAL CASE ('CONTINUOUS' SLIDER BLOCK)
  C ASSUME INITIALLY THAT THERE IS A SLIDER BLOCK AT THE TOP OF THE SEAT
  C AND AT THE SEAT BOTTOM
  C .....

```

SUBROUTINE	INJECT	74/74	OPT=1	FIN 4	6+428	83/11/07	09.41	53	PAGE	101

```

C SET UP VECTOR FROM SEAT/OCCUPANT CG TO EACH
C SUCH THAT XSSOSB(1)=VECTOR FROM (RIGHT) CG TO SEAT BOTTOM
C      XSSOSB(2)=VECTOR FROM CG TO SEAT TOP (RIGHT)
C      XSSOSB(3)=VECTOR FROM CG TO SEAT BOTTOM (LEFT)
C      XSSOSB(4)=VECTOR FROM CG TO SEAT TOP (LEFT)
C .....
222 CONTINUE
XSSOSB(1) = XSSOBOT
YSSOSB(1) = YSSORRE
ZSSOSB(1) = ZSSOBOT
XSSOSB(2) = XSSOBOT
YSSOSB(2) = YSSORRE
ZSSOSB(2) = ZSSOBOT+HGTSA
XSSOSB(3) = XSSOBOT
YSSOSB(3) = YSSOLRE
ZSSOSB(3) = ZSSOBOT
XSSOSB(4) = XSSOBOT
YSSOSB(4) = YSSOLRE
ZSSOSB(4) = ZSSOBOT+HGTSA
230 CONTINUE
C .....
C SET UP INITIAL POSITION OF EACH SLIPPER (SLIDER BLOCK) IN RCS
C .....
N=NSLBKS
IF (NSLBKS.EQ.0) N=4
DO 231 I=1,N
XYZ(1)=XSSOSB(1)
XYZ(2)=YSSOSB(1)
XYZ(3)=ZSSOSB(1)
CALL ROTATE (XYZ(1),XYZ(1),XSSOMRE,DCMSR,1)
XRRSBO(1)=XYZ(1)
YRRSBO(1)=XYZ(2)
ZRRSBO(1)=XYZ(3)
231 CONTINUE
C .....
C SET UP VECTOR FROM SEAT/OCCUPANT CG TO PARACHUTE ATTACHMENT POINTS
C .....
IF (IDROGUE.EQ.0) GO TO 241
XSSOCH(1)=XDROGAP-XCGSO
YSSOCH(1)=YDROGAP-YCGSO
ZSSOCH(1)=ZDROGAP-ZCGSO
241 CONTINUE
IF (IRECOV.EQ.0) GO TO 250
XSSOCH(3)=XRECAP-XCGSO
YSSOCH(3)=YRECAP-YCGSO
ZSSOCH(3)=ZRECAP-ZCGSO
250 CONTINUE
C .....
C SET UP AERODYNAMIC MOMENT ARMS
C FOR FULLY EXPOSED SEAT/OCCUPANT      = VECTOR FROM CG TO AEROSRP
C FOR PARTIALLY EXPOSED SEAT/OCC      SET UP VECTOR FROM SEAT/OCC AEROSRP
C                                         TO TOP OF SEAT
C                                         FOR SEAT ALONE. = VECTOR FROM SEAT CG TO AEROSRP
C .....
XSSOSRP = XPOS5SRP - XCGSO
YSSOSRP = YPOS5SRP - YCGSO
ZSSOSRP = ZPOS5SRP - ZCGSO
285

```

```

C      XSSASRP = XPOSSRP - XCGSA
      YSSASRP = YPOSSRP - YCGSA
      ZSSASRP = ZPOSSRP - ZCGSA

C      ZARME = ZSSOBOT-ZSSOSRP+HGHTSA
C      .....
C      SET UP VECTOR FROM ORIGIN OF RCS TO A/C C.G. IN RCS
C      .....

295      XRRCSAC = XPOSSRE
      YRRCSAC = (YPOSSRE+YPOSIRE)/2.
      ZRRCSAC = ZPOSSRE
      CALL ROTATE(ZVECT,XRRCSAC,YRRCSAC,DCMRA,O)
C      .....
C      SET UP VECTOR FROM SOCG TO ACCG IN EFGS
C      .....

300      CALL ROTATE (XRRCSAC,XSSCSAC,XPOSSCS,DCMSR,O)
      CALL ROTATE (XSSCSAC,XESDAC,XCGSD,DCMSE,I)
C      .....
C      SET UP VECTOR FROM ACCG TO SOCG IN ACS
C      .....

305      CALL ROTATE (ZVECT,XAACSD,XESDAC,DCMAE,O)
C      .....
C      SET UP VECTOR FROM SOCG TO ACCG IN SCS
C      .....

310      CALL ROTATE (XESDAC,XSSDAC,ZVECT,DCMSE,O)
C      .....
C      IF THERE IS A DART SYSTEM,SET UP DART VECTORS
C      .....
C      IF (IDART.EQ. O) GO TO 9000
C      .....
C      SET UP VECTOR FROM ORIGIN OF RAIL C.G. TO DART ATT. PIS IN RCS
C      .....

315      DO 260 I=1,2
      XYZ(1)=XDRTAP(I)
      XYZ(2)=VDRTAP(I)
      XYZ(3)=ZDRTAP(I)
      CALL ROTATE (XYZ(1),XYZ(1),ZVECT,DCMSR,I)
      XRRDAP(I) = XYZ(1)+XPOSSCS
      YRRDAP(I) = XYZ(2)+YPOSSCS
      ZRRDAP(I) = XYZ(3)+ZPOSSCS
      260 CONTINUE
C      .....
C      SET UP VECTOR FROM SEAT/OCC C.G. TO DART CONFLUENCE PIS. IN SCS
C      .....

330      DO 261 I = 1,2
      XSOCPI(1)=XDRTCP(I)-XCGSD
      YSSOCPI(1)=VDRTCP(I)-YCGSD
      ZSSOCPI(1)=ZDRTCP(I)-ZCGSD

335      261 CONTINUE
      9000 RETURN
      END

```

5-104


```

60      +      DENSITY, NPTSAAT, AAT(4.50), NPTSLAT,LAT(4.50),
      +      IACSFLG
      C.....
      C SECTION 9 COMMON BLOCK
      C.....
      COMMON /ICATPLT / INCAT      , CATLNT(2), CATSTK(2),TCI      (2),
      +      XPOSAP(2)      , YPOSAP(2), ZPOSAP(2),NPTSTC(2),
      +      CATHRST(2,25,2), ITUBEND , KTUBE      ,CTUBE      ,
      +      PTUBE      , MUTUBE      , EXTLNGT , ICATOUT
      REAL
      KUTUBE
      C.....
      C SECTION 1 COMMON BLOCK
      C.....
      COMMON /ICNTRL / ISTART, ISTOP , ESTOP , IRESTRT, IUNITS ,
      +      ISEATTR, ISOSEP , IPLOT , IDRIFLG,
      +      IPHASE1, IPHASE2, IPHASE3
      INTEGER
      ESTOP
      C.....
      C SECTION 3 COMMON BLOCK
      C.....
      COMMON /IDELTAT / DTPHASE1, DTPHASE2, DTPHASE3
      C.....
      C SECTION 7 COMMON BLOCK
      C.....
      COMMON /IRAIL / RAILNTH , RAILANG , ISTRL , NSLBKS ,
      +      KXSB , KYSB , MUSB , YKTOR ,
      +      XPOSRR , YPOSRR , ZPOSRR ,
      +      XPOSRL , YPOSRL , ZPOSRL ,
      +      XPOSSB(6), YPOSSB(6), ZPOSSB(6)
      REAL
      KXSB , KYSB , MUSB
      C.....
      C SECTION 2 COMMON BLOCK
      C.....
      COMMON /IREPORT / IREPTS(31) , PRIFRQ,P11,P12,P13
      INTEGER
      PRIFRQ,P11,P12,P13
      C.....
      C SECTION 10 COMMON BLOCK
      C.....
      COMMON /IROCKET / INRKT , RKDELY(6), RKNPTS(6), IROKOUT ,
      +      RKIGN(6) , RKWGT(6), RKBURN(6), TSTAR(6) ,
      +      XPOSRK(6), YPOSRK(6), ZPOSRK(6),
      +      RKALPH(6), RKBETA(6), RKGAMA(6), RKTHRST(2,25,6)
      INTEGER
      RKNPTS
      C.....
      C INFO04 DATA (USED IN SUBROUTINE AEROIN) COMMON BLOCK
      C.....
      COMMON /INFO04 / NCXS(12) , NCYS(12) , NCZS(12)
      +      DLTC(3,12) , ENDP(6,12) , IAECSQ(12)
      C.....
      C SECTION 6 COMMON BLOCK
      C.....
      COMMON /ISEATOC / IPCNLT , XCGSO , YCGSO , ZCGSO , IXXSO ,
      +      IXYSO , IYZSO , IYVSO , IYZSO , IZZSO ,
      +      AREASO , AREAOA , WGTTOAB, WGTTOAA ,
      +      IXXOA , IXYOA , IYZOA , IYVOA , IYZOA ,
      +      IZZOA , XCGOA , YCGOA , ZCGOA , SOSEP ,
      +      IXXSO , IXYSO , IYZSO , IYVSO , IYZSO ,
      REAL

```

```

115      +      IZZSO , IXXOA , IXYOA , IXZOA , IYYOA ,
      +      IYZOA , IZZOA
C.....
C SECTION 5 COMMON BLOCK
C.....
      COMMON /ISETALN / XPOSSRP , ZPOSSRP , XCGSA , YCGSA ,
      +      ZCGSA , IXXSA , IXYSA , IXZSA , IYYSA ,
      +      IYZSA , IZZSA , PHISA , PSISA , THESA ,
      +      AREASA , HGHTSA , WGHISA , XPOSBOT , YPOSBOT ,
      +      ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS
      REAL
      +      IXXSA , IXYSA , IXZSA , IYYSA , IYZSA ,
      +      IZZSA
C.....
C SECTION 12 COMMON BLOCK
C.....
      COMMON /ITVCIN / ITVC , MPHI , MPSI , MIHE ,
      +      ROLLRL , PITCHRL , SMLPRAT , TVCDLAY ,
      +      RKANG , MPHI , MPSI , MIHE
      REAL
      +      MPHI , MPSI , MIHE
C.....
C MASSES COMMON BLOCK
C.....
      COMMON /MASSES / MASSOA1 , MASSOA2 , MASSOD , MASSO ,
      +      MASSA , MASSRK(6) , MASSDC ,
      +      MASSOA1 , MASSOA2 , MASSO , MASSO ,
      +      MASSA , MASSRK , MASSDC
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
      COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRICNT(31) ,
      +      MAXLINE , MAXREPT , MAXEVT ,
      +      IEVLINE , IERRFLG , LU ,
      +      IDATE , HEADALT , HEADVEL ,
      +      HEADSR , HEADYAW , HEADPTI ,
      +      HEADROL , HEADWGT , BIAS ,
      +      REPTYPE(5,31) , PRTLNGT(2) , PRTLNGT(2) ,
      +      IHEADER(24) , IEVENTS(38) , TIMES(38) ,
      +      IMVDC , PRTEMP( 2) ,
      +      PRTMAS(2) , PRTINDX , PKZVEL ,
      +      ZVECT(3) , XYZ(3) , SAVTIME ,
      +      XACCEL(3) , YACCEL(3) ,
      +      REPTYPE , BIAS , PRTLNGT ,
      +      PRTWGHT , PRTEMP , PRTMAS , PRTINDX
      INTEGER
      +      PRTWGHT , PRTEMP , PRTMAS , PRTINDX
C.....
C SECTION 14 COMMON BLOCK
C.....
      COMMON /PARCHUT / IRECOV , RECOVLL ,
      +      RECDRAG , RECOVPD , POROSR ,
      +      XRECAP , YRECAP , ZRECAP ,
      +      NPTSRLS , RECOVLS(2,25) , IFTRECV ,
      +      NPTSRTI , RECOVFT(2,25) , SEPRCE ,
      +      IDROGUE , DRDGRAG2 , DRGPD02 ,
      +      POROSD2 , VELCON , IFTDR02 ,
      +      NPTDFT2 , DROGFT2(2,25) , IFTDR01 ,
      +      NPTDFT1 , DROGFT1(2,25) , IDROGLS ,
      +      NPTSLS , DROGLS(2,25) , TODPLOY

```

SUBROUTINE INIVRBL

```

+ + DISPLAY , DROGLL , DRDRAG1 ,
+ + DROGP01 , DROVSO1 ,
+ + DROVELY , DROVELX , XDRGAP ,
+ + YDROGAP , ZDROGAP , CHALT1 ,
+ + CHALT2 , GLIMIT , TDELAY ,
+ + AREADC , WGHITDC , TFP1 ,
+ + TFP2 , TFP3 , TDROGLS ,
+ + CDCC , NPTSRTD , RECOVDI(2,25) ,
C *****
C C INTEGRATION ROUTINE COMMON BLOCK
C *****
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193) , TRAJCH(97,3) ,
+ TRAJDA(193) , TRAJQS(225) , QUATOA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IPOINTS , IX , IVPRX , IKSUMX , IKPASSX ,
+ IX , IYIX , IYIX , IYIX ,
+ IYIX , IYIX , IYIX , IYIX ,
+ IPRI2X , IPRI1X , IPRI1X ,
+ ICYIX , ICYIX , ICYIX , IREIN ,
C *****
C C THRUST VECTOR CONTROL VARIABLES COMMON BLOCK
C *****
COMMON/TVCVB / TVCFGL , CMPVAL , D3(3) , RKTCMND(3) ,
+ C29 , C30 , C31 , C32 ,
+ DTH(3,2) , ANGR(3) ,
C *****
C C DAMPING COEFFICIENT COMMON BLOCK
C *****
COMMON/DAMPING / DMPCF2 , DMPCF3 , DMPCG
C *****
C C INITIALIZE REPORT HEADINGS
C *****
DATA (RETYPE(I), I=1,5) /
+ IOHINPUT VALI , IOHDATION , IOH ,
+ IOH , IOH ,
DATA (RETYPE(I), I=1,5) /
+ IOHSEAT/OCCUP , IOHANT LINEAR , IOH TIME HIST ,
+ IOHDRY , IOH ,
DATA (RETYPE(I), I=1,5) /
+ IOHSEAT/OCCUP , IOHANT ANGULA , IOHR TIME HIS ,
+ IOHTORY , IOH ,
DATA (RETYPE(I), I=1,5) /
+ IOHOCCUPANT A , IOHLONE LINEA , IOHR TIME HIS ,
+ IOHTORY , IOH ,
DATA (RETYPE(I), I=1,5) /
+ IOHSEAT ALONE , IOH LINEAR TI , IOHME HISTORY ,
+ IOH , IOH ,
DATA (RETYPE(I), I=1,5) /
+ IOHSEAT ALONE , IOH ANGULAR I , IOHIME HISTOR ,
+ IOHY , IOH ,
DATA (RETYPE(I), I=1,5) /
+ IOHSEAT/OCCUP , IOHANT LINEAR , IOH TIME HIST ,
+ IOHDRY WRT AT , IOHCRAFT ,
DATA (RETYPE(I), I=1,5) /

```

```

230      + IOHSEAT/OCCUP , IOHANT ANGULA , IOHR TIME HIS , /
      + IOHTORY WRT A , IOHIRCRAFT
      DATA (REPTYPE(1, 9), I=1, 5) /
      + IOHOCUPANT A , IOHLONE LINEA , IOHR TIME HIS , /
      + IOHTORY WRT A , IOHIRCRAFT
      DATA (REPTYPE(1, 10), I=1, 5) /
      + IOHSEAT ALONE , IOH LINEAR TI , IOHME HISTORY , /
      + IOH WRT AIRCR , IOHAFT
      DATA (REPTYPE(1, 11), I=1, 5) /
      + IOHSEAT ALONE , IOH ANGULAR T , IOHME HISTOR , /
      + IOHY WRT AIRC , IOHRAFT
      DATA (REPTYPE(1, 12), I=1, 5) /
      + IOHCATAPULT F , IOHORCES, MOM , IOHENIS , /
      + IOH , IOH
      DATA (REPTYPE(1, 13), I=1, 5) /
      + IOHROCKET 1 F , IOHORCES, MOM , IOHENIS , /
      + IOH , IOH
      DATA (REPTYPE(1, 14), I=1, 5) /
      + IOHROCKET 2 F , IOHORCES, MOM , IOHENIS , /
      + IOH , IOH
      DATA (REPTYPE(1, 15), I=1, 5) /
      + IOHROCKET 3 F , IOHORCES, MOM , IOHENIS , /
      + IOH , IOH
      DATA (REPTYPE(1, 16), I=1, 5) /
      + IOHROCKET 4 F , IOHORCES, MOM , IOHENIS , /
      + IOH , IOH
      DATA (REPTYPE(1, 17), I=1, 5) /
      + IOHROCKET 5 F , IOHORCES, MOM , IOHENIS , /
      + IOH , IOH
      DATA (REPTYPE(1, 18), I=1, 5) /
      + IOHROCKET 6 F , IOHORCES, MOM , IOHENIS , /
      + IOH , IOH
      DATA (REPTYPE(1, 19), I=1, 5) /
      + IOHDART FORCE , IOHS, MOMENTS , IOH
      + IOH , IOH
      DATA (REPTYPE(1, 20), I=1, 5) /
      + IOHDROGUE FOR , IOHCS, WOMEN , IOHTS , /
      + IOH , IOH
      DATA (REPTYPE(1, 21), I=1, 5) /
      + IOHPARACHUTE , IOHFORCES, MO , IOHMENTS , /
      + IOH , IOH
      DATA (REPTYPE(1, 22), I=1, 5) /
      + IOHTVC MICROP , IOHROCESSOR D , IOHATA , /
      + IOH , IOH
      DATA (REPTYPE(1, 23), I=1, 5) /
      + IOHRAIL FORCE , IOHS, MOMENTS , IOH
      + IOH , IOH
      DATA (REPTYPE(1, 24), I=1, 5) /
      + IOHSEAT/OCCUP , IOHANT AERODY , IOHYNAMIC FORC , /
      + IOHES, MOMENT , IOHS
      DATA (REPTYPE(1, 25), I=1, 5) /
      + IOHOCUPANT A , IOHLONE AEROD , IOHYNAMIC FOR , /
      + IOHCS, WOMEN , IOHTS
      DATA (REPTYPE(1, 26), I=1, 5) /
      + IOHSEAT ALONE , IOH AERODYNAM , IOHIC FORCES , /
      + IOH MOMENTS , IOH
      DATA (REPTYPE(1, 27), I=1, 5) /

```

```

290      + IOHOCUPANT A , IOHLONE ANGUL , IOHAR TIME HI ,
      + IOHISTORY , IOH
      DATA (REPTYPE(1,28),I=1,5) /
      + IOHAIRCRAFT L , IOHINEAR TIME , IOH HISTORY ,
      + IOH
      DATA (REPTYPE(1,29),I=1,5) /
      + IOHAIRCRAFT A , IOHNGULAR TIM , IOHE HISTORY ,
      + IOH
      DATA (REPTYPE(1,30),I=1,5) /
      + IOHAERODYNAMI , IOHC COEFFICI , IOHENTS TIME ,
      + IOHISTORY , IOH
      DATA (REPTYPE(1,31),I=1,5) /
      + IOHDYNAMIC CE , IOHINTER OF GR , IOHAVITY TIME ,
      + IOH HISTORY , IOH
      DATA PRTLNGT /2HM , 2HFT /
      DATA PRTWGT /2HNT , 2HLB /
      DATA PRTEMP /IOHCENTIGRADE , IOHFARENHEIT /
      DATA PRTMAS /4HKG , 4HSLUG /
      C .....
      C CALCULATE DAMPING COEFFICIENTS
      C .....
      DMPGF2 = EXP (-10.*DMPGC*DTPHAS2)
      DMPGF3 = EXP (-10.*DMPGC*DTPHAS3)
      C .....
      C INITIALIZE CONSTANTS
      C .....
      C29=1.4
      C30=0.2
      C31=1.4
      C32=0.2
      GRAVITY=9.8066
      IF (IUNITS.EQ. 1) GRAVITY=32.1745
      RADDEG=57.295779513082
      DRICON = 2798.41 / GRAVITY
      DRIMAX = -1.0 E 98
      DEGRAD=0.01745329251994
      PI=3.1415926536
      C .....
      C INITIALIZE HEADING VARIABLES
      C .....
      CALL LADDATE(IDATE)
      HEADALT = ZPOS
      HEADVEL = XACVEL
      HEADSR = ZACVEL
      HEADYAW = YAW
      HEADPIT = PITCH
      HEADROL = ROLL
      HEADWGT = WGTOTAB + WGTISA
      C .....
      C INITIALIZE AERODYNAMIC COEFFICIENT TABLE SEQUENCE NUMBERS
      C .....
      DO 101 I=1,6
      IAERCSQ(I) = 1
      IAERCSQ(I+6) = I+20
      101 CONTINUE
      C .....
      C INITIALIZE FLAGS AND PRINT VARIABLES
      C .....

```

```

C .....
345 ITVCLG = 1
    INSTP = 1
    IPOINTS = 1
    DELTAT = DPHAS2
    PRTRQ = -1
    DO 100 I=1,MAXREPT
    LINECT(I)=99
    IPRICNT(I) = P12 - 1
    100 CONTINUE
    IF(IPLST.EQ. 0) GO TO 150
    DO 125 I=7,11
    IF(IPTS(I) .NE. 0) IPLST= 2
    125 CONTINUE
    150 CONTINUE
C .....
355 C INITIALIZE DEFAULT PARAMETERS FOR DYNAMIC C.G. AND TUBE BENDING IF
    C REQUESTED BY USER.
C .....
    IF (IDYNGC .NE. 1) GO TO 200
    CX=0.30
    CY=0.30
    CZ=0.30
    IF(IUNITS.EQ. 0) GO TO 175
    SXP=18000.0
    SXN=18750.0
    SY = 12000.0
    SZP=18000.0
    SZN1=12000.0
    SZN2=25500.0
    XSLACK = 0.083
    ZSLACK = 0.083
    ZBOT = -0.083
    375 GO TO 200
    175 CONTINUE
    SXP = 13275.0
    SXN = 13500.0
    SY = 8850.0
    SZP = 13275.0
    SZN1 = 8850.0
    SZN2 = 18825
    XSLACK = 0.0253
    ZSLACK = 0.0253
    ZBOT = -0.0253
    200 CONTINUE
    IF (ITUBEND .NE. 1) GO TO 300
    KTUBE = 20000.0
    CTUBE = 100.0
    MTUBE = 0.028
    YKTOR = 261.7801
    PTUBE = 0.25
    390
C .....
395 C INITIALIZE MASSES
C .....
    300 CONTINUE
    MASSO = MASSO + WGHISA + WGHIOAB
    MASSA = WGHISA

```

```

400 MASSOA1=WGHTOAB
    MASSOA2=WGHTOAA
    MASSDC=WGHTDC
    IF (INRKT.EQ.0) GO TO 320
    DO 310 I=1,INRKT
    MASSRK(I) = RKWGT(I)
310 CONTINUE
320 CONTINUE
    IF ( IUNITS.EQ.0) GO TO 400
    MASSSO = MASSO / GRAVITY
    MASSSA = MASSA / GRAVITY
    MASSOA1= MASSOA1/GRAVITY
    MASSOA2= MASSOA2/GRAVITY
    IF (INRKT.EQ.0) GO TO 400
    DO 330 I=1,INRKT
    MASSRK(I) = MASSRK(I)/GRAVITY
330 CONTINUE
C *****
C INITIALIZE OCCUPANT ALONE CG FOR DYNAMIC CG CALCULATIONS
C *****
400 CONTINUE
    WY= SY/MASSOA1
    WY= SORT(WY)
    XCGOA = XCGOAO = (MASSSO*XCGSO - MASSSA*XCGSA)/MASSOA1
    YCGOA = YCGOAO = (MASSSO*YCGSO - MASSSA*YCGSA)/MASSOA1
    ZCGOA = ZCGOAO = (MASSSO*ZCGSO - MASSSA*ZCGSA)/MASSOA1
C *****
C COMPUTE INITIAL ATMOSPHERIC CONDITIONS
C *****
    RHOS=DENSITY
    TEMPS = 459.688+TEMP
    IF ( IUNITS.EQ.0) TEMPS = 459.688 + (32.+(9./5.)*TEMP)
    IF (DENSITY.NE.0.0) GO TO 600
    LATMOS = 1
    OLDALT(1) = ZPOS
    PRESALT(1) = (1.-(PRESSUR/1013.25)**0.19)/.000006875
    IF (PRESALT(1).GT.36089.0) GO TO 510
    TSTD = 518.688 - (0.003566 * PRESALT(1))
    GO TO 520
510 CONTINUE
    TSTD = 390.0
    PRESALT(1) = -(ALOG((PRESSUR/1013.25)/(0.2234)))*20787.0+36089.0
620 CONTINUE
    DTEMP = TEMPS - TSTD
C *****
C CONVERT SECTION 4 INPUT VARIABLES
C *****
600 CONTINUE
    YAW=YAW*DEGRAD
    PITCH=PITCH*DEGRAD
    ROLL=ROLL*DEGRAD
    RVEL=RVEL*DEGRAD
    QVEL=QVEL*DEGRAD
    PVEL=PVEL*DEGRAD
    IF (INPTSAT.EQ.0) GO TO 820
    DO 610 I=1,NPTSAT
    AAT(2,I)=AAT(2,I)*DEGRAD

```

```

      AAT(3,1)=AAT(3,1)*DEGRAD
      AAT(4,1)=AAT(4,1)*DEGRAD
      610 CONTINUE
      C .....
      C CONVERT SECTION 5 INPUTS
      C .....
      620 CONTINUE
      PHISA=PHISA*DEGRAD
      PSISA=PSISA*DEGRAD
      THESA=THESA*DEGRAD
      C .....
      C CONVERT SECTION 7 INPUT VARIABLES
      C .....
      RAILANG=RAILANG*DEGRAD
      C .....
      C CONVERT SECTION 10 INPUT VARIABLES
      C .....
      IF (INRKT.EQ. 0) GO TO 640
      DO 630 I=1,INRKT
      RKALPH(I)=RKALPH(I)*DEGRAD
      RKBETA(I)=RKBETA(I)*DEGRAD
      RKGAMA(I)=RKGAMA(I)*DEGRAD
      TSTAR(I) = -1.0
      630 CONTINUE
      C .....
      C CONVERT SECTION 12 INPUT VARIABLES
      C .....
      640 CONTINUE
      IF (ITVC.EQ. 0) GO TO 650
      MPHI=MPHI*DEGRAD
      MPSI=MPSI*DEGRAD
      MTHE=MTHE*DEGRAD
      ROLLRL=ROLLRL*DEGRAD
      PITCHRL=PITCHRL*DEGRAD
      SMPLRAT=SMPLRAT*DEGRAD
      RKANG=RKANG*DEGRAD
      C .....
      C CONVERT SECTION 14 INPUT VARIABLES
      C .....
      650 CONTINUE
      GLIMIT = GLIMIT * GRAVITY
      8000 RETURN
      END

```



```

1      SUBROUTINE INPUT
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - CONTROLS READING OF THE INPUT FILE
C METHOD - TO BE DEFINED
C COMMUNICATIONS -
C CALLED BY: INTLZ
C
C CALLS:
C TO BE DEFINED
C NON-COMMON VARIABLES DEFINED:
C TO BE DEFINED
C POTENTIAL ERROR CONDITIONS:
C TO BE DEFINED
C .....
C SECTION 13 COMMON BLOCK
C .....
COMMON /DYNGCN / IDYNGC , WY , WYV , SXN ,
+ XSLACK , SXP , SY , CZ , ZSLACK ,
+ SZP , ZBOT , SZN1 , SZN2
C .....
C SECTION 4 COMMON BLOCK
C .....
COMMON /IAIRCRT / TEMP , PRESSUR , ZACVEL , XPOS , YPOS ,
+ ZPOS , XTAIL , VTAIL , ZTAIL , VAW ,
+ PITCH , ROLL , RVEL , QVEL , PVEL ,
+ WINDX , WINDY , WINDZ , XACVEL , CKPIIHT ,
+ DENSITY , NPTSAAT , AAT(4.50) , NPTS LAT(4.50) ,
+ IACFLG
C .....
C SECTION 9 COMMON BLOCK
C .....
COMMON /ICATPLT / INCAT , CATLNT(2) , CATSK(2) , TCI (2) ,
+ XPOSAP(2) , YPOSAP(2) , ZPOSAP(2) , NPTSCT(2) ,
+ CATHRST(2.25.2) , ITUBEND , KTUBE , CTUBE ,
+ PTUBE , MUTUBE , EXTLNGT , ICATOUT ,
+ REAL , KTUBE , MUTUBE
C .....
C SECTION 1 COMMON BLOCK
C .....
COMMON /ICONTRL / ISTART , TSTOP , ESTOP , IRESTRT , IUNITS ,
+ ISEATTR , ISOSEP , IPLOT , IDRIFLG ,
+ IPHASE1 , IPHASE2 , IPHASE3 ,
+ INTEGER , ESTOP
C .....
C SECTION 11 COMMON BLOCK
C .....
COMMON /IDARTIN / IDART , ORIFRCE , DRISRT , DRISTOP ,
+ XDRTAP(2) , YDRTAP(2) , ZDRTAP(2) ,
+ XDRTCP(2) , YDRTCP(2) , ZDRTCP(2)
C .....
C SECTION 3 COMMON BLOCK
C .....
COMMON /IDELTAT / DIPHAS1 , DIPHAS2 , DIPHAS3
C .....

```

```

C SECTION 7 COMMON BLOCK
C.....
COMMON /IRAIL / RAILNTH / RAILANG / ISTRL / NSLBKS /
+ KXSB / KYSB / ZPOSRR / ZPOSRE / YKTOR /
+ XPSLRE / YPSLRE / ZPSLRE /
+ XPOSSB(6) / YPOSSB(6) / ZPOSSB(6) /
+ REAL / KXSB / KYSB / MUSB /
C.....
C SECTION 2 COMMON BLOCK
C.....
COMMON /IREPORT / IREPTS(31) / PRTRQ,P11,P12,P13
+ INTEGER / PRTRQ,P11,P12,P13
C.....
C RECALCULATED ROCKET THRUST TABLE COMMON BLOCK
C.....
COMMON /IRKTOUT / RKTOUT(2,25,6)
C.....
C SECTION 10 COMMON BLOCK
C.....
COMMON /IROCKET / INRKT / RKDELY(6) / RKNPSTS(6) / IROKOUT /
+ RKIGN(6) / RKWGT(6) / RKBURN(6) / TSTAR(6) /
+ XPOSRK(6) / YPOSRK(6) / ZPOSRK(6) /
+ RKALPH(6) / RKBETA(6) / RKGAMA(6) / RKTHRST(2,25,6)
+ INTEGER / RKNPSTS
C.....
C SECTION 6 COMMON BLOCK
C.....
COMMON /ISEATDC / IPCNTL / XCGSO / YCGSO / ZCGSO / IXXSO /
+ IYXSO / IYZSO / IYXSO / IYZSO / IZZSO /
+ AREASO / AREAOA / WGHIDAB / WGHIDAA /
+ IXXOA / IYXOA / IZXA / IYXOA / IYXOA / IYZOA /
+ IZZOA / XCGOA / YCGOA / ZCGOA / SOSEP /
+ C1SO / C2SO / C3SO / C4SO /
+ C1OA / C2OA / C3OA / C4OA /
+ REAL / IXXSO / IYXSO / IYZSO / IYZSO /
+ IZZSO / IYXOA / IYXOA / IYXOA / IYXOA /
+ IYXOA / IZZOA / IZZOA
C.....
C SECTION 5 COMMON BLOCK
C.....
COMMON /ISETALN / XPOSSRP / YPOSSRP / ZPOSSRP / XCGSA / YCGSA /
+ ZCGSA / IXXSA / IYXSA / IYXSA / IYXSA /
+ IYXSA / IZZSA / PHISA / PHISA / PHISA /
+ AREASA / HGTISA / WGHISA / XPOSBOT / YPOSBOT /
+ ZPOSBOT / XPOSSCS / YPOSSCS / ZPOSSCS /
+ C1SA / C2SA / C3SA / C4SA /
+ REAL / IXXSA / IYXSA / IYXSA / IYXSA /
+ IYXSA / IYXSA / IYXSA / IYXSA /
+ IZZSA / IZZSA
C.....
C SECTION 12 COMMON BLOCK
C.....
COMMON /ITVCIN / ITVC / MPHI / MPST / MTHE /
+ ROLLRL / PITCHRL / SIMPLRAT / TVCDLAY /
+ RKANG / MPHI / MPST / MTHE
+ REAL / MPHI / MPST / MTHE
C.....

```

```

115 C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+ MAXREPT , MAXEVT ,
+ IERFLG , LU ,
+ HEADALT , HEADVEL ,
+ HEADYAW , HEADPIT ,
+ HEADWGT , BIAS ,
+ REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ IMVDC , PRTINDX ,
+ PRTINDX , XYZ(3) , PKZVEL ,
+ XACCEL(3) , YACCEL(3) , SAVTIME ,
+ REPTYPE , BIAS , ZACCEL(3) ,
+ PRTWGT , PRTLNCT ,
+ PRTINDX , PRTINDX ,
C.....
C SECTION 14 COMMON BLOCK
C.....
COMMON /PARCUT / IRECOV , TROPLOY , RECOVLL ,
+ RECOVPO , POROSR ,
+ XRECAP , YRECAP , ZRECAP ,
+ NPISRLS , RECOVLS(2,25) , IFTREC ,
+ NPISRFT , RECOVFT(2,25) , SEPFRCE ,
+ IDROGUE , DRDRAG2 , DRGPD2 ,
+ POROSD2 , VELCON , IFTDR02 ,
+ NPIDFT2 , DRGFT2(2,25) , IFTDR01 ,
+ NPIDFT1 , DRGFT1(2,25) , IDROGLS ,
+ NPISDLs , DRGGLL , TDDPLOY ,
+ DISPLY , DRGSD1 , DROVELX ,
+ DROGPD1 , DROVELZ , XDROGAP ,
+ DROVELY , ZDROGAP , CHALT1 ,
+ YDROGAP , GLIMIT , TDELAY ,
+ CHALT2 , WHTDC , TFP1 ,
+ AREADC , TFP2 , IDROGLS ,
+ TFP3 , RECOVDT(2,25) ,
+ CDDC , NPISRDT ,
C.....
C DAMPING COEFFICIENT COMMON BLOCK
C.....
COMMON /DAMPING / DMPGF2 , DMPGF3 , DMPGC
C.....
C
DIMENSION INPDATA( 5), IVARBL( 5), WARNING( 4), FATLERR( 4)
EQUIVALENCE (INPDATA(1),IRESTR)
DATA IVARBL /7HIRESTR, 7HIUNITS ,
+ 7HISEATR, 7HISOSEP ,
+ 7HIPLOT /
DATA WARNING/10HWARNING(SU,10HROUTINE 1,10HINPUT)*****
+ 10H /
DATA FATLERR/10HFATAL ERRO.10HR(SUBROUTI,10HNE INPUT)*.
+ 10H*** /
IERFLG = 0
IWRNFG = 0
LU = 5
C.....
C READ HEADER DATA
C.....

```

```

C .....
  READ(1,5001) (IHEADER(1),I=1,8)
  READ(1,5001) (IHEADER(1),I=9,16)
  READ(1,5001) (IHEADER(1),I=17,24)
C .....
C READ SECTION 1 DATA AND EDIT
C .....
  READ(1,5011) TSTART, TSTOP, ESTOP, IRESTR
  READ(1,5012) IUNITS, ISEATTR, ISOSEP, IPLOT, IDRIFLG
  IF(TSTART .LT. 0) GO TO 9000
  IF(TSTOP .LT. 0) GO TO 9005
  IF(TSTOP .NE. 0) GO TO 5
  IF(ESTOP .LT. 1) OR (ESTOP .GT. 37)) GO TO 9010
  GO TO 10
5 CONTINUE
  IF(ESTOP .NE. 0) GO TO 9015
10 CONTINUE
C .....
C READ SECTION 2 DATA
C .....
  READ(1,5012) (IREPTS(1),I=1,31)
  READ(1,5025) PI1,PI2,PI3
  IF((PI1.LT.0).OR.(PI2.LT.0).OR.(PI3.LT.0)) GO TO 9012
  DO 20 I=1, 3
    IF(INPDATA(I) .EQ. 0 .OR. INPDATA(I) .EQ. 1) GO TO 20
    IERROR = 1
  GO TO 9020
20 CONTINUE
  IF(ISOSEP.GE.0.AND.ISOSEP.LE.2) GOTO 21
  IERROR=1
  GO TO 9017
21 IF(IPLOT GE. 0 .AND. IPLOT.LE. 4) GOTO 22
  IERROR = 1
  GO TO 9022
22 CONTINUE
  IF(ISTART .GT. 0) AND (TSTOP .NE. 0)
  + AND (TSTART .GE. TSTOP)) GO TO 9025
  IF(ISOSEP .NE. 0) GO TO 30
  IF(ESTOP .EQ. 28) GO TO 9027
  IF(ISEATTR .NE. 0) GO TO 8000
25 CONTINUE
  IF(IREPTS( 4) .NE. 0) GO TO 8005
  IF(IREPTS( 5) .NE. 0) GO TO 8008
  IF(IREPTS( 6) .NE. 0) GO TO 8005
  IF(IREPTS( 9) .NE. 0) GO TO 8005
  IF(IREPTS(10) .NE. 0) GO TO 8005
  IF(IREPTS(11) .NE. 0) GO TO 8005
  IF(IREPTS(25) .NE. 0) GO TO 8005
  IF(IREPTS(26) .NE. 0) GO TO 8008
30 CONTINUE
  IF(ISEATTR .NE. 0) GO TO 35
  IF(IREPTS( 5) .NE. 0) GO TO 8010
  IF(IREPTS( 6) .NE. 0) GO TO 8010
  IF(IREPTS(10) .NE. 0) GO TO 8010
  IF(IREPTS(11) .NE. 0) GO TO 8010
  IF(IREPTS(26) .NE. 0) GO TO 8010
35 CONTINUE

```

```

230 C.....
C READ SECTION 3 DATA AND EDIT
C.....
      READ(1,5031) DTPHAS1, DTPHAS2, DTPHAS3
      IF (DTPHAS1 .LE. 0.0) GO TO 9030
      IF (DTPHAS2 .LE. 0.0) GO TO 9035
      IF ((DTPHAS3 .LE. 0.0) .AND. (ISOSEP .GT. 0)) GO TO 9040
C.....
235 C READ SECTION 4 DATA AND EDIT
C.....
      READ(1,5031) TEMP, PRESSUR, DENSITY
      READ(1,5031) XPOS, YPOS, ZPOS
      READ(1,5031) YAW, PITCH, ROLL
      READ(1,5031) RVEL, QVEL, PVEL
      READ(1,5031) WINDX, WINDY, WINDZ
      READ(1,5031) XACVEL, ZACVEL, CKPITHI
      READ(1,5025) NPTSLAT
      IF (NPTSLAT .LT. 0) OR (NPTSLAT .GT. 50)) GO TO 9045
      IF (NPTSLAT .EQ. 0) GO TO 220
C.....
240 C READ VALUES INTO ARRAY AAT BASED ON THE VALUE IN NPTSLAT
C I.E. IF NPTSLAT = 10 THEN 10 SETS OF 4 VALUES ARE READ IN.
C.....
      READ(1,5070) ((AAT(I,J), I=1,4), J=1,NPTSLAT)
220 CONTINUE
      READ(1,5025) NPTSLAT
      IF (NPTSLAT .LT. 0) IACSFGL = 2
      NPTSLAT = IABS(NPTSLAT)
      IF (NPTSLAT .GT. 50) GO TO 9050
      IF (NPTSLAT .EQ. 0) GO TO 260
C.....
245 C READ VALUES INTO ARRAY LAT BASED ON THE VALUE IN NPTSLAT
C I.E. IF NPTSLAT = 5 THEN 5 SETS OF 4 VALUES ARE READ IN.
C.....
      READ(1,5070) ((LAT(I,J), I=1,4), J=1,NPTSLAT)
260 CONTINUE
      IF (XACVEL .LT. 0.0) GO TO 9070
C.....
255 C READ SECTION 5 DATA AND EDIT
C.....
      READ(1,5031) XPOSSRP, YPOSSRP, ZPOSSRP
      READ(1,5031) XCGSA, YCGSA, ZCGSA
      READ(1,5031) IXSA, IYSA, IZSA
      READ(1,5031) JYSA, JZSA, IZSA
      READ(1,5031) PHISA, PSISA, THESA
      READ(1,5031) AREASA, HGTISA, WGTISA
      READ(1,5031) XPOSSBOT, YPOSSBOT, ZPOSSBOT
      IF ((IXSA .EQ. 0) .OR. (IYSA .EQ. 0) .OR. (IZSA .EQ. 0)) GO TO 9100
      IF (WGTISA .LE. 0.0) GO TO 9075
      C1SA = IXSA + IZSA + IXSA + IYSA
      C2SA = IXSA + IZSA + IXSA + IZSA
      C3SA = IXSA + IYSA + IXSA + IZSA
      C4SA = IYSA + C2SA + IXSA + C1SA + IYSA + C3SA
      IF (C2SA .EQ. 0) GO TO 9102
      IF (C4SA .EQ. 0) GO TO 9104
C.....
265 C.....
270 C.....
275 C.....
280 C.....
285 C.....

```

```

C READ SECTION 6 DATA AND EDIT
C.....
290 READ(1,5031) XCGSO , YCGSO , ZCGSO
    READ(1,5031) IXXSO , IYXSO , IZXSO
    READ(1,5031) IYXSO , IYZSO , IZZSO
    READ(1,5050) AREASO
    C READ(1,5031) IXXOA , IYXOA , IZXOA
    C READ(1,5031) IYXOA , IYZOA , IZZOA
    READ(1,5031) AREAOA , WGHTOAB , WGHTOAA
    READ(1,5031) SOSEP , DMPGC
    IF(WGHTOAB .LE. 0.0) GO TO 9085
    IF(WGHTOAA .LE. 0.0) GO TO 9090
    IF((IXXSO.EQ.O).OR.(IYXSO.EQ.O).OR.(IZZSO.EQ.O)) GO TO 9110
    IF((IXXOA.EQ.O).OR.(IYXOA.EQ.O).OR.(IZZOA.EQ.O)) GO TO 9115
    C150=IXXSO + IZZSO + IZXSO + IYZSO
    C250=IXXSO + IZZSO + IZXSO + IYZSO
    C350=IXXSO + IYZSO + IZXSO + IYZSO
    C450= IYXSO + C250 - IYXSO + C150 - IYZSO + C350
    IF(C250.EQ.O.O) GO TO 9117
    IF(C450.EQ.O.O) GO TO 9119
    C C10A=IXXOA + IZZOA + IZXOA + IYZOA
    C C20A=IXXOA + IZZOA - IZXOA + IZXOA
    C C30A=IXXOA + IYZOA + IYXOA + IZXOA
    C C40A= IYXOA + C20A - IYXOA + C10A - IYZOA + C30A
    C IF(C20A.EQ.O.O) GO TO 9121
    C IF(C40A.EQ.O.O) GO TO 9123
    C IF(DMPGC .LT. 0.0) GOTO 8255
    270 CONTINUE
C.....
315 C READ SECTION 7 DATA AND EDIT
C.....
    READ(1,5050) RAILNTH, RAILANG
    READ(1,5013) ISTRL , NSLBKS
    READ(1,5065) KXSB , KYSB , MUSB
    READ(1,5046) YKTOR
    READ(1,5031) XPOSRE , YPOSRE , ZPOSRE
    READ(1,5031) XPOSRE , YPOSRE , ZPOSRE
    IF (NSLBKS .EQ. 0) GO TO 290
    DO 280 I=1,NSLBKS
    READ(1,5031) XPOSSB(I), YPOSSB(I), ZPOSSB(I)
    280 CONTINUE
    IF((ISTRL .NE. 0) .AND. (ISTRL .NE. 1)) GO TO 9125
    IF((NSLBKS .LT. 0) .OR. (NSLBKS .GT. 6)) GO TO 9127
C.....
330 C READ SECTION 8 DATA AND EDIT
C.....
    290 CONTINUE
C.....
335 C READ SECTION 9 DATA AND EDIT
C.....
    READ(1,5025) INCAT
    IF((INCAT .EQ. 0) GO TO 650
    IF((INCAT .NE. 1) .AND. (INCAT .NE. 2)) GO TO 9400
    DO 620 I=1, INCAT
    ISAVE = I
    READ(1,5031) CATINT(I) , CATSTR(I) , TCI(I)
    READ(1,5031) XPOSAP(I) , YPOSAP(I) , ZPOSAP(I)

```

```

345 READ(1,5025) NPTSTCT(I)
      INDX = NPTSTCT(I)
      IF((INDX.LT. 2) .OR. (INDX.GT. 25)) GO TO 9410
      C.....
      C READ VALUES INTO ARRAY CATHRST BASED ON THE VALUE IN NPTSTCT.
      C I.E. IF I = 1 AND NPTSTCT = 3 THEN 3 SETS OF 2 VALUES ARE READ
      C      IN FOR CATAPULT #1.
      C.....
350 READ(1,5070) (CATHRST(1,J,1),CATHRST(2,J,1),J=1,INDX)
      620 CONTINUE
      IF((1.GT. ESTOP) .OR. (ESTOP.GT. 4)) GO TO 630
      IF (ESTOP.EQ. 2 .AND. INCAT.LT. 2) GO TO 9422
      IF (ESTOP.EQ. 4 .AND. INCAT.LT. 2) GO TO 9422
      630 CONTINUE
      READ(1,5025) ITUBEND
      IF((ITUBEND.EQ. 0) .OR. (ITUBEND.EQ. 1)) GOTO 660
      IF( ITUBEND.NE. 2)
      READ(1,5031) KTUBE , CTUBE , PTUBE
      READ(1,5050) MUTUBE , EXTLNGT
      GO TO 660
      650 CONTINUE
      IF((1.LE. ESTOP) .AND. (ESTOP.LE. 4)) GO TO 9425
      IF(IREPTS(12).NE. 0) GO TO 8230
      660 CONTINUE
      C.....
      C READ SECTION 10 DATA AND EDIT
      C.....
370 READ(1,5025) INRKT
      IF(INRKT.EQ. 0) GO TO 700
      IF((INRKT.LT. 1) .OR. (INRKT.GT. 6)) GO TO 9430
      DO 680 I=1,INRKT
      ISAVE = I
      READ(1,5070) RKIGN(1) , RKWGT(1), RKURN(1), RKDELV(1)
      READ(1,5031) XPOSRK(1) , YPOSRK(1), ZPOSRK(1)
      READ(1,5031) RKALPH(1) , RKBETA(1), RKGAMA(1)
      READ(1,5013) RKNPTS(1)
      INDX = RKNPTS(1)
      IF((INDX.LT. 2) .OR. (INDX.GT. 25)) GO TO 9450
      C.....
      C READ VALUES INTO ARRAY RKTHRST BASED ON THE VALUE IN RKNPTS.
      C I.E. IF INRKT = 5 AND RKNPTS = 20 THEN 20 SETS OF 2 VALUES ARE
      C      READ IN FOR ROCKETS #1 THRU #5.
      C.....
385 READ(1,5070) (RKTHRST(1,J,1),RKTHRST(2,J,1),J=1,INDX)
      680 CONTINUE
      680 CALL THRUST
      700 CONTINUE
      IF (INRKT.EQ. 6 .OR. (ESTOP.NE. 0 .AND.
      + (6.GT. ESTOP .OR. ESTOP.GT. 17))) GO TO 710
      INDX = INRKT+1
      DO 701 I=1,INDX
      IF ((ESTOP.GT. INDX+4 .AND. ESTOP.LT. 12) .OR.
      + (ESTOP.GT. INDX+10 .AND. ESTOP.LT. 18)) GO TO 9452
      701 CONTINUE
      INDX=INRKT+13
      DO 702 I=INDX,18
      IF (IREPTS(1).NE. 0) GO TO 8235

```

```

400      702 CONTINUE
      710 CONTINUE
      C.....
      C READ SECTION 13 DATA AND EDIT
      READ (1,5025) IDROGUE
      IF (IDROGUE.EQ. 0) GOTO 975
      IF ((IDROGUE.LT. 0) .OR. (IDROGUE.GT. 2)) GOTO 9590
      READ (1,5031) TODPLOY, DISPLAY, DROGLL
      IF (TODPLOY.LT. 0) GOTO 9680
      IF (DISPLAY.LT. 0) GOTO 9690
      IF (DROGLL.LE. 0) GOTO 9700
      READ (1,5031) DRDRAG1, DROGPD1, POROSD1
      IF (DRDRAG1.LT. 0) GOTO 9705
      IF (DROGPD1.LE. 0) GOTO 9710
      IF (POROSD1.LT. 0) GOTO 9715
      READ (1,5031) AREADC, WHTDC, CDC
      IF (AREADC.LT. 0) GOTO 9740
      IF (WHTDC.LT. 0) GOTO 9745
      IF (CDC.LT. 0) GOTO 9750
      READ (1,5031) XDROGAP, YDROGAP, ZDROGAP
      READ (1,5025) IDROGLS
      IF (IDROGLS.EQ. 1) GOTO 930
      IF (IDROGLS.NE. 0) GOTO 9660
      C.....
      C IF IDROGLS = 0, USE SLUG TO STRETCH LINES
      C.....
      READ (1,5031) DROVELX, DROVELY, DROVELZ
      GOTO 935
      C.....
      C IF IDROGLS = 1, USE TABLES TO STRETCH LINES
      C.....
      930 CONTINUE
      READ (1,5025) NPTSOLS
      IF ((NPTSOLS.LT. 2) .OR. (NPTSOLS.GT. 25)) GOTO 9670
      C.....
      C READ TABLE DATA INTO DROGLS
      C.....
      READ (1,5070) (DROGLS(1,I), DROGLS(2,I), I=1,NPTSOLS)
      935 CONTINUE
      READ (1,5025) IFTDRO1
      IF (IFTDRO1.EQ. 0) GOTO 940
      IF (IFTDRO1.NE. 1) GOTO 9640
      READ (1,5025) NPTDFT1
      IF ((NPTDFT1.LT. 2) .OR. (NPTDFT1.GT. 25)) GOTO 9650
      C.....
      C READ DROGFT1
      C.....
      READ (1,5070) (DROGFT1(1,I), DROGFT1(2,I), I=1,NPTDFT1)
      940 CONTINUE
      IF (IDROGUE.NE. 3) GOTO 975
      C.....
      C READ VELCON DROGUE DATA
      C.....
      READ (1,5046) VELCON
      IF (VELCON.LT. 0) GOTO 9620
      READ (1,5031) DRDRAG2, DROGPD2, POROSD2
      IF (DRDRAG2.LT. 0) GOTO 9600

```



```

460 IF (DROGPD2 .LE. 0.0) GOTO 9610
    IF (POROSD2 .LT. 0.0) GOTO 9615
    READ (1,5025) IFTRD2
    IF (IFTRD2 .EQ. 0) GOTO 975
    IF (IFTRD2 .NE. 1) GOTO 9630
    READ (1,5025) NPTDFT2
    IF (NPTDFT2 .LT. 2) .OR. (NPTDFT2 .GT. 25)) GOTO 9635
    C .....
    C READ IN DROGFT2
    C .....
    READ (1,5070) (DROGFT2(1,1),DROGFT2(2,1), I=1,NPTDFT2)
    975 CONTINUE
    IF ((IDROGUE .EQ. 0) .AND. (IREPIS(20) .NE. 0)) GO TO 8250
    980 CONTINUE
    C .....
    C READ SECTION 14 DATA AND EDIT
    C .....
    READ(1,5025) IRECOV
    IF (IRECOV .EQ. 0) GO TO 927
    IF ((IRECOV .LT. 0) .OR. (IRECOV .GT. 2)) GO TO 9510
    READ(1,5050) RECOVLL , RECDRAG
    IF (RECOVLL .LT. 0.0) GO TO 9530
    IF (RECDRAG .LT. 0.0) GO TO 9540
    READ(1,5050) RECOVPD , POROSR
    IF (RECOVPD .LT. 0.0) GO TO 9550
    IF (POROSR .LT. 0.0) GO TO 9555
    READ(1,5031) XRECAP , YRECAP , ZRECAP
    READ(1,5031) CHALT1 , CHALT2 , GLIMIT
    IF (CHALT1 .LT. 0.0) GO TO 9720
    IF (CHALT2 .LT. 0.0) GO TO 9725
    READ(1,5046) TDELAY
    IF (TDELAY .LT. 0.0) GO TO 9730
    READ(1,5025) NPTSRDT
    IF (NPTSRDT .EQ. 1) GOTO 860
    IF ((NPTSRDT .LT. 2) .OR. (NPTSRDT .GT. 25)) GO TO 9565
    C .....
    C READ VALUES INTO ARRAY RECOVDI BASED ON THE VALUE IN NPTSRDT.
    C I.E. IF NPTSRDT = 20 THEN 20 SETS OF 2 VALUES ARE READ IN.
    C .....
    READ(1,5070) (RECOVDI(1,1),RECOVDI(2,1),I=1,NPTSRDT)
    GOTO 875
    860 CONTINUE
    READ (1,5046) TRDPLOY
    IF (TRDPLOY .LT. 0.0) GOTO 9520
    875 CONTINUE
    READ(1,5025) NPTSRLS
    IF ((NPTSRLS .LT. 2) .OR. (NPTSRLS .GT. 25)) GO TO 9560
    C .....
    C READ VALUES INTO ARRAY RECOVLS BASED ON THE VALUE IN NPTSRLS.
    C I.E. IF NPTSRLS = 20 THEN 20 SETS OF 2 VALUES ARE READ IN.
    C .....
    READ(1,5070) (RECOVLS(1,1),RECOVLS(2,1),I=1,NPTSRLS)
    READ(1,5025) IFTRCV
    IF (IFTRCV .EQ. 0) GO TO 925
    IF (IFTRCV .NE. 1) GO TO 9570
    READ(1,5025) NPTSRET
    IF ((NPTSRET .LT. 2) .OR. (NPTSRET .GT. 25)) GO TO 9580

```

```

515 C.....
C READ VALUES INTO ARRAY RECOVFI BASED ON THE VALUE IN NPTSRT.
C I.E. IF NPTSRT = 7 THEN 7 SETS OF 2 VALUES ARE READ IN.
C.....
      READ(1,5070) (RECOVFI(1,1),RECOVFI(2,1),I=1,NPTSRT)
925 CONTINUE
      IF((IRECOV.EQ.0).AND.(IREPTS(21).NE.0)) GO TO 8245
927 CONTINUE
C.....
C READ SECTION 11 DATA AND EDIT
C.....
      READ(1,5025) IDART
      IF(IDART.EQ.0) GO TO 760
      IF(IDART.NE.1) GO TO 9460
      READ(1,5031) DRTFCE, DRTSTRT, DRTSTOP
      IF(DRTSTOP.LE.DRTSTRT) GO TO 9470
C.....
C XDRTAP(1) = XDRTAPR ..... XDRTCP(1) = XDRTCPR
C.....
      DO 730 I=1,2
      READ(1,5031) XDRTAP(1), YDRTAP(1), ZDRTAP(1)
      READ(1,5031) XDRTCP(1), YDRTCP(1), ZDRTCP(1)
730 CONTINUE
750 CONTINUE
C.....
C READ SECTION 12 DATA AND EDIT
C.....
      READ(1,5025) ITVC
      IF(ITVC.EQ.0) GO TO 770
      IF(ITVC.NE.1) GO TO 9480
      READ(1,5031) MPH1, MPST, MTHE
      READ(1,5031) ROLLRL, PITCHRL, SMPLRAT
      READ(1,5050) TVCOLAY, RKANG
      IF(TVCOLAY.LT.0.0) GO TO 9490
      GO TO 800
770 IF (IREPTS(22).NE.0) GO TO 8240
800 CONTINUE
C.....
      READ(1,5025) IDYNCG
      IF((IDYNCG.EQ.0).OR.(IDYNCG.EQ.1)) GO TO 850
      IF(IDYNCG.NE.2)
      READ(1,5031) CX, XSLACK, SXP
      READ(1,5031) SXN, CY, SY
      READ(1,5031) CZ, ZSLACK, SZP
      READ(1,5031) ZBOT, SZN1, SZN2
850 CONTINUE
      GO TO 9999
560 C.....
C PROCESS WARNING MESSAGES
C.....
      8000 ISEATR = 0
      IF(IWRNFLG.NE.0)GOTO 90
      WRITE(LU,7000)
      IWRNFLG = 1
      90 WRITE(LU,8001)(WARNING(K),K=1,4)
      GO TO 25
570 8005 IREPTS( 4) = 0

```

```

575      IREPTS( 5) = 0
        IREPTS( 6) = 0
        IREPTS( 9) = 0
        IREPTS(10) = 0
        IREPTS(11) = 0
        IF(IWRNFLG.NE.O)GOTO 91
        WRITE(LU,7000)
        IWRNFLG = 1
        91 WRITE(LU,8006)WARNING
        GO TO 30
580      8010 IREPTS( 5) = 0
        IREPTS( 6) = 0
        IREPTS(10) = 0
        IREPTS(11) = 0
        IF(IWRNFLG.NE.O) GOTO 93
        WRITE(LU,7000)
        IWRNFLG = 1
        93 WRITE(LU,8011)
        GOTO 35
590      8230 IREPTS(12) = 0
        IF(IWRNFLG.NE.O)GOTO 95
        WRITE(LU,7000)
        IWRNFLG = 1
        95 WRITE(LU,8231)WARNING
        GO TO 860
595      8235 CONTINUE
        DO 8237 I = INDX, 18
        IREPTS(I) = 0
        8237 CONTINUE
        600      IF(IWRNFLG.NE.O)GOTO 96
        WRITE(LU,7000)
        IWRNFLG = 1
        96 WRITE(LU,8239)WARNING, INRKT
        GO TO 710
        605      8240 IREPTS(22) = 0
        IF(IWRNFLG.NE.O)GOTO 97
        WRITE(LU,7000)
        IWRNFLG = 1
        97 WRITE(LU,8241)WARNING
        GO TO 800
        610      8245 IREPTS(21) = 0
        IF(IWRNFLG.NE.O)GOTO 98
        WRITE(LU,7000)
        IWRNFLG = 1
        98 WRITE(LU,8246)WARNING
        GO TO 927
        615      8250 IREPTS(20) = 0
        IF(IWRNFLG.NE.O)GOTO 99
        WRITE(LU,7000)
        IWRNFLG = 1
        99 WRITE(LU,8251)WARNING
        GO TO 980
        620      8255 DMPGC = 0.0
        IF(IWRNFLG.NE.O)GOTO 908
        WRITE(LU,7000)
        IWRNFLG = 1
        908 WRITE(LU,8256)WARNING

```

```

630      GO TO 270
        C.....
        C PROCESS FATAL ERROR MESSAGES
        C.....
        9000 WRITE(LU,9001) FATALERR,TSTART
            GO TO 9900
        9005 WRITE(LU,9006) FATALERR,TSTOP
            GO TO 9900
        9010 WRITE(LU,9011) FATALERR,ESTOP
            GO TO 9900
        9015 WRITE(LU,9016) FATALERR,ESTOP
            GO TO 9900
        9012 WRITE(LU,9013) FATALERR,PI1,PI2,PI3
            GO TO 9900
        9017 WRITE(LU,9018) FATALERR,ISOSEP
            GO TO 9900
        9020 WRITE(LU,9021) FATALERR,IVARBL(IERROR),
            * INPDATA(IERROR)
            GO TO 9900
        9022 WRITE(LU,9023) FATALERR,IPILOT
            GO TO 9900
        9025 WRITE(LU,9026) FATALERR,TSTART , TSTOP
            GO TO 9900
        9027 WRITE(LU,9028) FATALERR,ESTOP
            GO TO 9900
        9030 WRITE(LU,9031) FATALERR,DIPHAS1
            GO TO 9900
        9035 WRITE(LU,9036) FATALERR,DIPHAS2
            GO TO 9900
        9040 WRITE(LU,9041) FATALERR,DIPHAS3,ISOSEP
            GO TO 9900
        9045 WRITE(LU,9046) FATALERR,NPTSAAT
            GO TO 9900
        9050 WRITE(LU,9051) FATALERR,NPTSLAT
            GO TO 9900
        9070 WRITE(LU,9071) FATALERR,XACVEL
            GO TO 9900
        9075 WRITE(LU,9076) FATALERR,WGHTSA
            GO TO 9900
        9085 WRITE(LU,9086) FATALERR,WGHTOAB
            GO TO 9900
        9090 WRITE(LU,9081) FATALERR
            GO TO 9900
        9100 WRITE(LU,9101) FATALERR
            GO TO 9900
        9102 WRITE(LU,9103) FATALERR
            GO TO 9900
        9104 WRITE(LU,9105) FATALERR
            GO TO 9900
        9110 WRITE(LU,9111) FATALERR
            GO TO 9900
        9115 WRITE(LU,9116) FATALERR
            GO TO 9900
        9117 WRITE(LU,9118) FATALERR
            GO TO 9900
        9119 WRITE(LU,9120) FATALERR
            GO TO 9900

```

```

685 C9121 WRITE(LU,9122) FATLERR
      C GO TO 9900
C9123 WRITE(LU,9124)
      C GO TO 9900
      9125 WRITE(LU,9126) FATLERR,1STRL
      GO TO 9900
      9127 WRITE(LU,9128) FATLERR,NSLBKS
      GO TO 9900
      9400 WRITE(LU,9401) FATLERR,INCAT
      GO TO 9900
      9410 WRITE(LU,9411) FATLERR,NPTSCT(1SAVE)
      GO TO 9800
      9420 WRITE(LU,9421) FATLERR,ITUBEND
      GO TO 9900
      9422 WRITE(LU,9423) FATLERR,ESTOP, INCAT
      GO TO 9900
      9425 WRITE(LU,9426) FATLERR,ESTOP
      GO TO 9800
      9430 WRITE(LU,9431) FATLERR,INRKT
      GO TO 9900
      9450 WRITE(LU,9451) FATLERR,RKNPTS(1SAVE)
      GO TO 9900
      9452 WRITE(LU,9453) FATLERR,ESTOP, INRKT
      GO TO 9800
      9480 WRITE(LU,9481) FATLERR,IDART
      GO TO 9900
      9470 WRITE(LU,9471) FATLERR,DRTSTOP, DRTSTRT
      GO TO 9900
      9480 WRITE(LU,9481) FATLERR,ITVC
      GO TO 9900
      9480 WRITE(LU,9491) FATLERR,TVCDELAY
      GO TO 9900
      9500 WRITE(LU,9501) FATLERR,IDYNCG
      GO TO 9900
      9510 WRITE(LU,9511) FATLERR,IRECOV
      GO TO 9800
      9520 WRITE(LU,9521) FATLERR,ITROPLOY
      GO TO 9900
      9530 WRITE(LU,9531) FATLERR,RECOVLL
      GO TO 9900
      9540 WRITE(LU,9541) FATLERR,RECOVRAG
      GO TO 9900
      9550 WRITE(LU,9551) FATLERR,RECOVPD
      GO TO 9900
      9555 WRITE(LU,9556) FATLERR,POROSR
      GO TO 9900
      9560 WRITE(LU,9561) FATLERR,NPTSRLS
      GO TO 9900
      9565 WRITE(LU,9566) FATLERR,NPTSRTD
      GO TO 9900
      9570 WRITE(LU,9571) FATLERR,IFTRCV
      GO TO 9900
      9580 WRITE(LU,9581) FATLERR,NPTSRTFT
      GO TO 9800
      9590 WRITE(LU,9591) FATLERR,IDROGUE
      GO TO 9900
      9600 WRITE(LU,9601) FATLERR,DRDRAG2

```

```

GO TO 9900
9610 WRITE(LU,9611) FATLERR,DROGPD2
GO TO 9900
745 9615 WRITE(LU,9616) FATLERR,POROSD2
GO TO 9900
9620 WRITE(LU,9621) FATLERR,VELCON
GO TO 9900
750 9630 WRITE(LU,9631) FATLERR,IFTDRO2
GO TO 9900
9635 WRITE(LU,9636) FATLERR,NPTDFT2
GO TO 9900
9640 WRITE(LU,9641) FATLERR,IFTDRO1
GO TO 9900
755 9650 WRITE(LU,9651) FATLERR,NPTDFT1
GO TO 9900
9660 WRITE(LU,9661) FATLERR,IDROGLS
GO TO 9900
760 9670 WRITE(LU,9671) FATLERR,NPTSDLS
GO TO 9900
9680 WRITE(LU,9681) FATLERR,TDOPLOY
GO TO 9900
9690 WRITE(LU,9691) FATLERR,DISPLOY
GO TO 9900
765 9700 WRITE(LU,9701) FATLERR,DROGLL
GO TO 9900
9705 WRITE(LU,9706) FATLERR,DRDRAG1
GO TO 9900
770 9710 WRITE(LU,9711) FATLERR,DROGPD1
GO TO 9900
9715 WRITE(LU,9716) FATLERR,POROSD1
GO TO 9900
775 9720 WRITE(LU,9721) FATLERR,CHALT1
GO TO 9900
9725 WRITE(LU,9726) FATLERR,CHALT2
GO TO 9900
9730 WRITE(LU,9731) FATLERR,TDelay
GO TO 9900
780 9740 WRITE(LU,9741) FATLERR,AREADC
GO TO 9900
9745 WRITE(LU,9746) FATLERR,WGHTDC
GO TO 9900
9750 WRITE(LU,9751) FATLERR,CDDC
C.....
C SET ERROR FLAG (IERFLG) TO ONE(1) TO INDICATE TO THE CALLING
C ROUTINE THAT A FATAL ERROR HAS OCCURRED
C.....
785 9900 CONTINUE
WRITE(LU,9980)
STOP "ERROR IN INPUT"
C.....
C RETURN TO CALLING PROGRAM
C.....
790 9999 CONTINUE
RETURN
C.....
C READ FORMAT STATEMENTS
C.....
795

```

```

800      5001 FORMAT(8(A10))
      5011 FORMAT(2(10X,F10.4),18X,12,19X,11)
      5012 FORMAT(4(19X,11))
      5013 FORMAT(2(17X,13))
      5025 FORMAT(3(17X,13))
      5031 FORMAT(3(10X,F10.4))
      5046 FORMAT(10X,F10.4)
      5050 FORMAT(2(10X,F10.4))
      5065 FORMAT(2(10X,F10.6),10X,F10.4)
      5070 FORMAT(4(10X,F10.4))
      C*****
      C WRITE FORMAT STATEMENTS
      C*****
      7000 FORMAT(1H1,48X,35(1H*),/,
      + 49X,"-WARNING MESSAGES AND FATAL ERRORS",/,
      + 49X,"-REPORT
      + 49X,35(1H*),//
      C*****
      C FORMATS FOR WARNING MESSAGES
      C*****
      8001 FORMAT(5X,4A10,
      + "1SEATTR HAS BEEN RESET TO ZERO")
      8006 FORMAT(5X,4A10,
      + "REPORT FLAGS FOR REPORTS 4,5,6,9,10,11,25 AND ",
      + "26 HAVE BEEN RESET TO ZERO")
      8011 FORMAT(5X,4A10,
      + "REPORT FLAGS FOR REPORTS 5,6,10,11 AND 26 ",
      + "HAVE BEEN RESET TO ZERO")
      8231 FORMAT(5X,4A10,
      + "REPORT FLAG FOR REPORT 12 HAS BEEN RESET TO ZERO")
      8239 FORMAT(5X,4A10,
      + "ROCKET REPORT FLAGS FOR ROCKET NUMBER(S) GREATER THAN ",12,1X,
      + "WERE RESET TO ZERO")
      8241 FORMAT(5X,4A10,
      + "REPORT 22 FLAG HAS BEEN RESET TO ZERO")
      8246 FORMAT(5X,4A10,
      + "REPORT FLAG FOR REPORT 21 HAS BEEN RESET TO ZERO")
      8251 FORMAT(5X,4A10,
      + "REPORT FLAG FOR REPORT 20 HAS BEEN RESET TO ZERO")
      8256 FORMAT(5X,4A10,
      + "DMPGC HAS BEEN RESET TO ZERO")
      C*****
      C FORMATS FOR FATAL ERROR MESSAGES
      C*****
      9001 FORMAT(1X,//72(1H*),/4X,4A10,/1X,
      + "TSTART ENTERED AS ",F10.4,/1X,
      + 72(1H*))
      9006 FORMAT(1X,//72(1H*),/4X,4A10,/1X,
      + "TSTOP ENTERED AS ",F10.4,/1X,
      + 72(1H*))
      9011 FORMAT(1X,//72(1H*),/4X,4A10,/1X,
      + "ESTOP ENTERED AS ",12,/1X,
      + "BUT SHOULD BE A NUMBER IN THE RANGE 1 THRU 28",/1X,
      + 72(1H*))
      9016 FORMAT(1X,//72(1H*),/4X,4A10,/1X,
      + "ESTOP ENTERED AS ",12,/1X,
      + "BUT SHOULD BE ZERO WHEN TSTOP IS NOT ZERO",/1X,

```

SUBROUTINE INPUT 74/74 OPT=1 FTN 4 6428 83/11/07. 09.41.53 PAGE 127

```

      +72(1H*)
9013 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+PRINT FREQUENCIES ENTERED AS ".314./, 1X,
+*BUT MUST BE GREATER THAN OR EQUAL TO ZERO"/, 1X,
+72(1H*))
9018 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+ISOSEP ENTERED AS ".12./, 1X,
+* AND NOT ZERO(O).ONE(1), OR TWO(2)"/, 1X,
+72(1H*))
9021 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*7. WAS ENTERED AS ". 13./, 1X,
+* AND NOT A ONE(1) OR ZERO(O)"/, 1X,
+72(1H*))
9023 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*IPLT ENTERED AS ". 12./, 1X,
+*BUT SHOULD BE A NUMBER IN THE RANGE 1 THRU 4"/, 1X,
+72(1H*))
9026 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*TSTART ENTERED AS ".F10.4./, 21X,
+*TSTOP ENTERED AS ".F10.4./, 21X,
+*IF TSTART > O AND TSTOP NOT ZERO THEN TSTART"/, 21X,
+*MUST BE < TSTOP"/, 1X,
+72(1H*))
9028 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*ESTOP ENTERED AS ". 12./, 1X,
+*BUT CANNOT BE AN EVENT STOP REPRESENTING A SEAT/OCCUPANT "/, 1X,
+*SEPARATION WHEN ISOSEP EQUALS ZERO(O)"/, 1X,
+72(1H*))
9031 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*DTPHAS1 ENTERED AS ".F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN ZERO (O)"/, 1X,
+72(1H*))
9036 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*DTPHAS2 ENTERED AS ".F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN ZERO (O)"/, 1X,
+72(1H*))
9041 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*DTPHAS3 ENTERED AS ".F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN ZERO (O) WHEN ISOSEP EQUALS ".12./, 1X,
+72(1H*))
9046 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*NTPSAAT ENTERED AS ". 13./, 21X,
+*BUT SHOULD BE IN THE RANGE O THRU 50"/, 1X,
+72(1H*))
9051 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*NPTSLAT ENTERED AS ". 13./, 21X,
+*BUT SHOULD BE IN THE RANGE O THRU 50"/, 1X,
+72(1H*))
9071 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*XACVEL ENTERED AS ".F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(O)"/, 1X,
+72(1H*))
9076 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,
+*WGHTSA ENTERED AS ".F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN ZERO (O)"/, 1X,
+72(1H*))
9086 FORMAT(1X, //72(1H*)//4X, 4A10.//, 1X,

```



```

915  +*WGHTOAB ENTERED AS ".F10.4./,21X,
      +*BUT SHOULD BE GREATER THAN ZERO (O)"./,1X,
      +72(1H*)
9091 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*WGHTOAA ENTERED AS ".F10.4./,21X,
      +*BUT SHOULD BE GREATER THAN ZERO (O)"./,1X,
      +72(1H*))
9101 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*MOMENTS OF INERTIA ABOUT THE MAIN AXES(SEAT ALONE) MAY NOT ",
      +*BE ZERO"./,1X,
      +72(1H*))
9103 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*IXXSA+IYZSA-IXZSA+IXZSA MAY "NOT BE ZERO"./,1X,
      +72(1H*))
9105 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*IXYSA+C1SA+IYVSA+C2SA-IZYSA+C3SA MAY NOT BE ZERO"./,1X,
      +72(1H*))
9111 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*MOMENTS OF INERTIA ABOUT THE MAIN AXES(SEAT/OCCUPANT) MAY NOT ",
      +*BE ZERO"./,1X,
      +72(1H*))
9116 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*MOMENTS OF INERTIA ABOUT THE MAIN AXES(OCCUPANT ALONE) MAY NOT ",
      +*BE ZERO"./,1X,
      +72(1H*))
9118 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*IXXSO+IYZSO-IXZSO+IXZSO MAY NOT BE ZERO"./,1X,
      +72(1H*))
9120 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*IXYSO+C1SO+IYVSO+C2SO-IZYSO+C3SO MAY NOT BE ZERO"./,1X,
      +72(1H*))
C9122 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
C  +*IXXOA+IYZOA-IXZOA+IXZOA MAY NOT BE ZERO"./,1X,
C  +72(1H*))
C9124 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
C  +*IXYOA+C1OA+IYVOA+C2OA-IZYOA+C3OA MAY NOT BE ZERO"./,1X,
C  +72(1H*))
9126 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*1STRL ENTERED AS ".13./,21X,
      +*BUT SHOULD BE ZERO (O) OR ONE (1)"./,1X,
      +72(1H*))
9128 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*NSLBKS ENTERED AS ".13./,21X,
      +*BUT SHOULD BE IN THE RANGE ZERO (O) THRU SIX (6)"./,1X,
      +72(1H*))
9401 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*INCAT ENTERED AS ".13./,21X,
      +*BUT SHOULD BE IN THE RANGE 0 THRU 2"./,1X,
      +72(1H*))
9411 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*NPTSCT ENTERED AS ".13./,21X,
      +*BUT SHOULD BE IN THE RANGE 2 THRU 25"./,1X,
      +72(1H*))
9421 FORMAT(1X,/,72(1H*)/,4X,4A10./,1X,
      +*ITUBEND ENTERED AS ".13./,21X,
      +*BUT SHOULD BE ZERO (O) . ONE (1) . OR TWO (2)"./,1X,
      +72(1H*))

```

```

970      9423 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*ESTOP ENTERED AS ". 13.//.21X,
+*INCAT ENTERED AS ". 13.//.21X,
+*CATAPULT EVENT STOP CANNOT REPRESENT A CATAPULT"/.21X,
+*GREATER THAN THE NUMBER OF CATAPULTS THAT EXIST"/.1X,
+72(1H*))
975      9426 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*ESTOP ENTERED AS ". 13.//.1X,
+*REPRESENTING A CATAPULT EVENT WHEN NO CATAPULTS "
+*EXIST"/.1X,
+72(1H*))
980      9431 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*INRKT ENTERED AS ". 13.//.21X,
+*BUT SHOULD BE IN THE RANGE 0 THRU 6"/.1X,
+72(1H*))
985      9451 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*RKNPTS ENTERED AS ". 13.//.21X,
+*BUT SHOULD BE IN THE RANGE 2 THRU 25"/.1X,
+72(1H*))
990      9453 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*FSTOP ENTERED AS ". 13.//.21X,
+*INRKT ENTERED AS ". 13.//.21X,
+*ROCKET EVENT STOP CANNOT REPRESENT A ROCKET"/.21X,
+*GREATER THAN THE NUMBER OF ROCKETS THAT EXIST"/.1X,
+72(1H*))
995      9461 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*IDART ENTERED AS ". 13.//.21X,
+*BUT SHOULD BE ZERO (0) OR ONE (1)"/.1X,
+72(1H*))
1000     9471 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*DRSTOP ENTERED AS ".F10.4.//.21X,
+*DRSTRT ENTERED AS ".F10.4.//.21X,
+*BUT DRSTOP MUST BE GREATER THAN DRSTRT"/.1X,
+72(1H*))
1005     9481 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*ITVC ENTERED AS ".13.//.21X,
+*BUT SHOULD BE ZERO (0) OR ONE (1)"/.1X,
+72(1H*))
1010     9491 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*TVCOLAY ENTERED AS ".F10.4.//.21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO"/.1X,
+72(1H*))
1015     9501 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*IDYNCG ENTERED AS ".13.//.21X,
+*BUT SHOULD BE ZERO (0) , ONE (1) , OR TWO (2)"/.1X,
+72(1H*))
1020     9511 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*IRECOV ENTERED AS ". 13.//.21X,
+*BUT SHOULD BE IN THE RANGE 0 THRU 2"/.1X,
+72(1H*))
1025     9521 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*TRIPLY ENTERED AS ".F10.4.//.21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)"/.1X,
+72(1H*))
1030     9531 FORMAT(1X, //72(1H*)//.4X.4A10.//.1X,
+*RECOVLL ENTERED AS ".F10.4.//.21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)"/.1X,

```

```

+72(1H*)
9541 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+RECDRAG ENTERED AS ". F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(O)". /, 1X,
+72(1H*))
9551 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+RECOVPD ENTERED AS ". F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(O)". /, 1X,
+72(1H*))
9556 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+POROSP ENTERED AS ". F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(O)". /, 1X,
+72(1H*))
9561 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+NPTSRLS ENTERED AS ". I3./, 21X,
+*BUT SHOULD BE IN THE RANGE 2 THRU 25". /, 1X,
+72(1H*))
9566 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+NPTSROT ENTERED AS ". I3./, 21X,
+*BUT SHOULD BE IN THE RANGE 1 THRU 25". /, 1X,
+72(1H*))
9571 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+IFTRCV ENTERED AS ". I3./, 21X,
+*BUT SHOULD BE ZERO (O) OR ONE (1)". /, 1X,
+72(1H*))
9581 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+NPTSRT ENTERED AS ". I3./, 21X,
+*BUT SHOULD BE IN THE RANGE 2 THRU 25". /, 1X,
+72(1H*))
9591 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+IDROGUE ENTERED AS ". I3./, 21X,
+*BUT SHOULD BE IN THE RANGE 0 THRU 2". /, 1X,
+72(1H*))
9601 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+DRDRAG2 ENTERED AS ". F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(O)". /, 1X,
+72(1H*))
9611 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+DROGPD2 ENTERED AS ". F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN ZERO (O)". /, 1X,
+72(1H*))
9616 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+POROSD2 ENTERED AS ". F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(O)". /, 1X,
+72(1H*))
9621 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+VELCON ENTERED AS ". F10.4./, 21X,
+*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(O)". /, 1X,
+72(1H*))
9631 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+IFDRD2 ENTERED AS ". I3./, 21X,
+*BUT SHOULD BE ZERO (O) OR ONE (1)". /, 1X,
+72(1H*))
9636 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
+NPTDFT2 ENTERED AS ". I3./, 21X,
+*BUT SHOULD BE IN THE RANGE 2 THRU 25". /, 1X,
+72(1H*))

```

```

1085 9641 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "IFIDR01 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE ZERO (0) OR ONE (1)", //1X,
      *72(1H*))
1090 9651 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "NPTDFT1 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE IN THE RANGE 2 THRU 25", //1X,
      *72(1H*))
1095 9661 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "IDROGLS ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE ZERO (0) OR ONE (1)", //1X,
      *72(1H*))
1095 9671 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "NPTSDLS ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE IN THE RANGE 2 THRU 25", //1X,
      *72(1H*))
1100 9681 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "TDDPLY ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1105 9691 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "DISPLY ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1110 9701 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "DROGL1 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN ZERO (0)", //1X,
      *72(1H*))
1115 9706 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "DRDRAG1 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1115 9711 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "DROGPD1 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN ZERO (0)", //1X,
      *72(1H*))
1120 9716 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "POROSD1 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1125 9721 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "CHIAL1 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1130 9726 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "CHIAL2 ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1135 9731 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "TOELAY ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1135 9741 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,
      * "AREADC ENTERED AS ", F10.4, //21X,
      * "BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)", //1X,
      *72(1H*))
1140 9746 FORMAT(1X, //72(1H*)//.4X.4A10./, 1X,

```

```

*WGHIDC ENTERED AS ".F10.4,/,21X,
*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)"./,1X,
+72(1H*)
9751 FORMAT(1X,/,72(1H*),/,4X,4A10,/,1X,
+*CODEC ENTERED AS ".F10.4,/,21X,
*BUT SHOULD BE GREATER THAN OR EQUAL TO ZERO(0)"./,1X,
+72(1H*))
9990 FORMAT(1X,/,72(1H*),/,4X,"RUN ABORTED",/,72(1H*))
END

```

1145

5-134

SUBROUTINE INTEG 74/74 OPT-1

60 CALL INTEGIN(TRAJCH(1,1))
 CALL INTEGIN(TRAJCH(1,2))
 CALL INTEGIN(TRAJCH(1,3))
 CALL INTEGIN(TRAJAC(1))
 CALL INTEGIN(TVCEOS(1))
 CALL INTEGIN(QUATSD(1))
 CALL INTEGIN(QUATDA(1))
 CALL INTEGIN(QUATSA(1))
 CALL INTEGIN(QUATAC(1))

65 C
 C SET FLAGS FOR NEXT PASS
 C

70 400 IF (IPCASS.EQ.0) GO TO 600
 IF (IPCASS.EQ.2) GO TO 500
 IPCASS = 2
 GO TO 700

75 500 IPOINTS = IPOINTS+1
 INTSTP = 1
 IPCASS = 1
 GO TO 700

80 600 IF (IRKPASS.NE.4) GO TO 700
 IRKPASS = 0
 IF (TVCEOS(1).EQ.0) IPOINTS=IPOINTS+1
 INTSTP = 1
 700 RETURN
 END

5-136


```

1  SUBROUTINE INTEGIN(ARRAY)
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - THIS SUBROUTINE PERFORMS THE NUMERICAL INTEGRATION
C OF THE DIFFERENTIAL EQUATIONS IN THE SYSTEM.
C METHOD - THE FIRST PHASE OF PROCESSING USES THE GILLS
C VARIATION OF THE RUNGE-KUTTA METHOD TO CALCULATE
C THE FIRST FOUR POINTS. FOR THIS PHASE, FOUR PASSES
C ARE NEEDED TO CALCULATE THE FINAL Y ESTIMATE. EACH
C PASS RESULTS IN THE CALCULATION OF A K VALUE TO BE
C USED IN THE FINAL ESTIMATE. THE K VALUE IS THE
C DERIVATIVE EVALUATED AT A CERTAIN Y VALUE. SINCE
C THESE VALUES ARE NEEDED IN FUTURE PASSES, THEY ARE
C SAVED IN THE WORK ARRAY.
C THE SECOND PHASE USES THE HAMMING PREDICTOR-CORRECTOR
C METHOD TO CALCULATE THE REMAINING POINTS. TWO PASSES
C ARE NEEDED TO CALCULATE THE FINAL VALUE IN THIS PHASE.
C THE FIRST PASS EVALUATES THE PREDICTOR AND MODIFIER
C EQUATIONS. THE Y VALUE CALCULATED IN THE MODIFIER
C IS USED TO EVALUATE THE DERIVATIVE. THE SECOND PASS
C EVALUATES THE CORRECTOR AND FINAL-VALUE EQUATIONS
C TO DETERMINE THE FINAL ESTIMATE OF THE NEXT POINT.
C THE Y VALUE TO BE USED IN EVALUATING THE DERIVATIVE
C IS PLACED IN THE WORK ARRAY BY THIS ROUTINE. SINCE
C VALUES FROM PREVIOUS POINTS ARE NEEDED IN THE
C PREDICTOR-CORRECTOR PHASE, THESE VALUES ARE SAVED
C IN THE WORK ARRAY. EACH TYPE OF DATA IS GROUPED
C TOGETHER. WITHIN EACH GROUP ARE THE VALUES FOR
C EACH EQUATION BEING EVALUATED. THE FOLLOWING LIST
C SHOWS THE DATA THAT IS SAVED AND THE INDEX NAME
C ASSOCIATED WITH IT.
C DATA TYPE INDEX NAME ARRAY POS
C Y IYX 1
C Y' IYPRX 2
C K VALUE IKX 3
C K SUM IKSUMX 4
C K PASSED IKPASXX 5
C Y(I) IYIX 6
C Y(I-1) IYI1X 7
C Y(I-2) IYI2X 8
C Y(I-3) IYI3X 9
C Y'(I) IYPRIX 10
C Y'(I-1) IYPRI1X 11
C Y'(I-2) IYPRI2X 12
C P(I) IPYIX 13
C P(I+1) IPYI1X 14
C C(I) ICYIX 15
C C(I+1) ICYI1X 16
C COMMUNICATIONS -
C CALLED BY: INTEG
C CALLS: INTEGSI INTEGII INTEGMO
C NON-COMMON VARIABLES DEFINED -
C ARRAY - CALL PARAMETER USED AS COMMON WORK AREA
C INUMEO - INTEGER VARIABLE USED TO SAVE NUMBER
C OF EQUATIONS IN THE ARRAY
C .....

```

```

C POTENTIAL ERROR CONDITIONS -
C ..... NONE .....
C .....
60 C .....
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
COMMON /KUTTA / TIME , TIMES , DELTAT , TRAJSD(193) ,
+ TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3) ,
+ TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
+ QUATSA(65) , QUATOA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IYX , IYPRX ,
+ IKX , IKSUMX ,
+ IYIX , IYI2X ,
+ IYI3X , IYPRIX ,
+ IYPRI2X , IPYIX ,
+ ICYIX , ICYI1X ,
+
75 DIMENSION ARRAY(225)
IF (ARRAY(1) .EQ. 0) GO TO 9000
INUMEO = ARRAY(1)
IF (IPOINTS.GE.5) GO TO 2000
C .....
C PERFORM RUNGE-KUTTA INTEGRATION
C .....
IF (IRKPASS.EQ.1) CALL INTEGMD(ARRAY(1))
CALL INTEGSI(ARRAY(1))
DO 1000 IENUM = 1, INUMEO
CALL INTEGSI
ARRAY(IKX) = ARRAY(IYPRX)
GO TO (100,200,300,400)IRKPASS
100 ARRAY(IYX) = ARRAY(IYIX)+DELTAT*ARRAY(IKX)/2.0
ARRAY(IKSUMX) = ARRAY(IKX)
ARRAY(IKPASSX) = .2071068*DELTAT*ARRAY(IKX)
GO TO 1000
200 ARRAY(IYX) = ARRAY(IYIX)+ARRAY(IKPASSX)+
+ .29289321881345*DELTAT*ARRAY(IKX)
ARRAY(IKSUMX) = ARRAY(IKSUMX)+.58578643762690*ARRAY(IKX)
ARRAY(IKPASSX) = .7071067811865*DELTAT*ARRAY(IKX)
GO TO 1000
300 ARRAY(IYX) = ARRAY(IYIX)-ARRAY(IKPASSX)+
+ 1.7071667811865*DELTAT*ARRAY(IKX)
ARRAY(IKSUMX) = ARRAY(IKSUMX)+3.4142135623731*ARRAY(IKX)
GO TO 1000
400 ARRAY(IKSUMX) = ARRAY(IKSUMX)+ARRAY(IKX)
ARRAY(IYX) = ARRAY(IYIX)+.16666666666667*DELTAT*ARRAY(IKSUMX)
1000 CONTINUE
GO TO 9000
C .....
C PERFORM HAMMING PREDICTOR-CORRECTOR INTEGRATION
C .....
2000 IF (IPCPASS.NE.2) CALL INTEGMD(ARRAY(1))
CALL INTEGSI(ARRAY(1))
DO 3000 IENUM = 1, INUMEO
CALL INTEGSI
2100 IF (IPCPASS.EQ.2) GO TO 2200
ARRAY(IPYI1X) = ARRAY(IYI3X)+1.3333333333333*DELTAT*
+ (2.0*ARRAY(IYPRIX)-ARRAY(IYPRI1X)+2.0*ARRAY(IYPRI2X))

```

PAGE 138

83/11/07. 09.41.53

FTN 4.6+428

SUBROUTINE INTEGIN 74/74 OPT=1

```

115      ARRAY(IVX) = ARRAY(IPYIIX) - (112.0/121.0) *
      + (ARRAY(IPYIX) - ARRAY(ICYIX))
      GO TO 3000
2200  ARRAY(ICYIIX) = .125 * (9.0 * ARRAY(IVIX) - ARRAY(IYI2X) * 3.0 * DELTAT +
      + (ARRAY(IYPRX) * 2.0 * ARRAY(IYPRIX) - ARRAY(IYPRIX)))
120      ARRAY(IVX) = ARRAY(ICYIIX) * (9.0/121.0) + (ARRAY(IPYIIX) - ARRAY(ICYIIX))
3000  CONTINUE
      IF (IPCPASS.EQ.0) IPCPASS = 1
9000  RETURN
      END

```

```

1 SUBROUTINE INTEGMD(ARRAY)
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - THIS SUBROUTINE SHIFTS THE DATA BEING SAVED IN THE
C WORK ARRAY SO THAT A NEW POINT CAN BE CALCULATED.
C METHOD - THE DATA IS MOVED IN THE FOLLOWING MANNER:
C .....
C      Y(I-2) TO Y(I-3)
C      Y(I-1) TO Y(I-2)
C      Y(I) TO Y(I-1)
C      Y TO Y(I)
C      Y'(I-1) TO Y'(I-2)
C      Y'(I) TO Y'(I-1)
C      Y' TO Y'(I)
C      P(I+1) TO P(I)
C      C(I+1) TO C(I)
C .....
C COMMUNICATIONS -
C CALLED BY: INTEGSI
C CALLS: INTEGSI INTEGII
C NON-COMMON VARIABLES DEFINED -
C INUMEQ - INTEGER VARIABLE USED TO SAVE THE NUMBER OF
C EQUATIONS IN THE ARRAY.
C POTENTIAL ERROR CONDITIONS -
C .....
C NONE
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
C COMMON /RKUTTA / TIME, TIMES, DELTAT, TRAJSD(193),
C + TRAJSA(193), TRAJJA(193), TRAJQA(193), TRAJCH(97.3),
C + TRAJAC(193), TVCEQS(225), QUATSD(65),
C + QUATSA(65), QUATOA(65), QUATAC(65),
C + INTSTP, IPCPASS, IRKPASS,
C + IPOINTS, IYX, IYPRX,
C + IKX, IKSUMX, IKPASSX,
C + IYIX, IYIIX, IYI2X,
C + IYI3X, IYPRIX, IYPR1X,
C + IYPR2X, IYVIX, IYV1X,
C + IYV2X, IYV3X, IYVPR1X, IYVPR2X,
C + IYVPR3X, IYVPR4X, IYVPR5X,
C + IYVPR6X, IYVPR7X, IYVPR8X,
C + IYVPR9X, IYVPR10X, IYVPR11X,
C + IYVPR12X, IYVPR13X, IYVPR14X,
C + IYVPR15X, IYVPR16X, IYVPR17X,
C + IYVPR18X, IYVPR19X, IYVPR20X,
C + IYVPR21X, IYVPR22X, IYVPR23X,
C + IYVPR24X, IYVPR25X, IYVPR26X,
C + IYVPR27X, IYVPR28X, IYVPR29X,
C + IYVPR30X, IYVPR31X, IYVPR32X,
C + IYVPR33X, IYVPR34X, IYVPR35X,
C + IYVPR36X, IYVPR37X, IYVPR38X,
C + IYVPR39X, IYVPR40X, IYVPR41X,
C + IYVPR42X, IYVPR43X, IYVPR44X,
C + IYVPR45X, IYVPR46X, IYVPR47X,
C + IYVPR48X, IYVPR49X, IYVPR50X,
C + IYVPR51X, IYVPR52X, IYVPR53X,
C + IYVPR54X, IYVPR55X, IYVPR56X,
C + IYVPR57X, IYVPR58X, IYVPR59X,
C + IYVPR60X, IYVPR61X, IYVPR62X,
C + IYVPR63X, IYVPR64X, IYVPR65X,
C + IYVPR66X, IYVPR67X, IYVPR68X,
C + IYVPR69X, IYVPR70X, IYVPR71X,
C + IYVPR72X, IYVPR73X, IYVPR74X,
C + IYVPR75X, IYVPR76X, IYVPR77X,
C + IYVPR78X, IYVPR79X, IYVPR80X,
C + IYVPR81X, IYVPR82X, IYVPR83X,
C + IYVPR84X, IYVPR85X, IYVPR86X,
C + IYVPR87X, IYVPR88X, IYVPR89X,
C + IYVPR90X, IYVPR91X, IYVPR92X,
C + IYVPR93X, IYVPR94X, IYVPR95X,
C + IYVPR96X, IYVPR97X, IYVPR98X,
C + IYVPR99X, IYVPR100X, IYVPR101X,
C + IYVPR102X, IYVPR103X, IYVPR104X,
C + IYVPR105X, IYVPR106X, IYVPR107X,
C + IYVPR108X, IYVPR109X, IYVPR110X,
C + IYVPR111X, IYVPR112X, IYVPR113X,
C + IYVPR114X, IYVPR115X, IYVPR116X,
C + IYVPR117X, IYVPR118X, IYVPR119X,
C + IYVPR120X, IYVPR121X, IYVPR122X,
C + IYVPR123X, IYVPR124X, IYVPR125X,
C + IYVPR126X, IYVPR127X, IYVPR128X,
C + IYVPR129X, IYVPR130X, IYVPR131X,
C + IYVPR132X, IYVPR133X, IYVPR134X,
C + IYVPR135X, IYVPR136X, IYVPR137X,
C + IYVPR138X, IYVPR139X, IYVPR140X,
C + IYVPR141X, IYVPR142X, IYVPR143X,
C + IYVPR144X, IYVPR145X, IYVPR146X,
C + IYVPR147X, IYVPR148X, IYVPR149X,
C + IYVPR150X, IYVPR151X, IYVPR152X,
C + IYVPR153X, IYVPR154X, IYVPR155X,
C + IYVPR156X, IYVPR157X, IYVPR158X,
C + IYVPR159X, IYVPR160X, IYVPR161X,
C + IYVPR162X, IYVPR163X, IYVPR164X,
C + IYVPR165X, IYVPR166X, IYVPR167X,
C + IYVPR168X, IYVPR169X, IYVPR170X,
C + IYVPR171X, IYVPR172X, IYVPR173X,
C + IYVPR174X, IYVPR175X, IYVPR176X,
C + IYVPR177X, IYVPR178X, IYVPR179X,
C + IYVPR180X, IYVPR181X, IYVPR182X,
C + IYVPR183X, IYVPR184X, IYVPR185X,
C + IYVPR186X, IYVPR187X, IYVPR188X,
C + IYVPR189X, IYVPR190X, IYVPR191X,
C + IYVPR192X, IYVPR193X, IYVPR194X,
C + IYVPR195X, IYVPR196X, IYVPR197X,
C + IYVPR198X, IYVPR199X, IYVPR200X,
C + IYVPR201X, IYVPR202X, IYVPR203X,
C + IYVPR204X, IYVPR205X, IYVPR206X,
C + IYVPR207X, IYVPR208X, IYVPR209X,
C + IYVPR210X, IYVPR211X, IYVPR212X,
C + IYVPR213X, IYVPR214X, IYVPR215X,
C + IYVPR216X, IYVPR217X, IYVPR218X,
C + IYVPR219X, IYVPR220X, IYVPR221X,
C + IYVPR222X, IYVPR223X, IYVPR224X,
C + IYVPR225X, IYVPR226X, IYVPR227X,
C + IYVPR228X, IYVPR229X, IYVPR230X,
C + IYVPR231X, IYVPR232X, IYVPR233X,
C + IYVPR234X, IYVPR235X, IYVPR236X,
C + IYVPR237X, IYVPR238X, IYVPR239X,
C + IYVPR240X, IYVPR241X, IYVPR242X,
C + IYVPR243X, IYVPR244X, IYVPR245X,
C + IYVPR246X, IYVPR247X, IYVPR248X,
C + IYVPR249X, IYVPR250X, IYVPR251X,
C + IYVPR252X, IYVPR253X, IYVPR254X,
C + IYVPR255X, IYVPR256X, IYVPR257X,
C + IYVPR258X, IYVPR259X, IYVPR260X,
C + IYVPR261X, IYVPR262X, IYVPR263X,
C + IYVPR264X, IYVPR265X, IYVPR266X,
C + IYVPR267X, IYVPR268X, IYVPR269X,
C + IYVPR270X, IYVPR271X, IYVPR272X,
C + IYVPR273X, IYVPR274X, IYVPR275X,
C + IYVPR276X, IYVPR277X, IYVPR278X,
C + IYVPR279X, IYVPR280X, IYVPR281X,
C + IYVPR282X, IYVPR283X, IYVPR284X,
C + IYVPR285X, IYVPR286X, IYVPR287X,
C + IYVPR288X, IYVPR289X, IYVPR290X,
C + IYVPR291X, IYVPR292X, IYVPR293X,
C + IYVPR294X, IYVPR295X, IYVPR296X,
C + IYVPR297X, IYVPR298X, IYVPR299X,
C + IYVPR300X, IYVPR301X, IYVPR302X,
C + IYVPR303X, IYVPR304X, IYVPR305X,
C + IYVPR306X, IYVPR307X, IYVPR308X,
C + IYVPR309X, IYVPR310X, IYVPR311X,
C + IYVPR312X, IYVPR313X, IYVPR314X,
C + IYVPR315X, IYVPR316X, IYVPR317X,
C + IYVPR318X, IYVPR319X, IYVPR320X,
C + IYVPR321X, IYVPR322X, IYVPR323X,
C + IYVPR324X, IYVPR325X, IYVPR326X,
C + IYVPR327X, IYVPR328X, IYVPR329X,
C + IYVPR330X, IYVPR331X, IYVPR332X,
C + IYVPR333X, IYVPR334X, IYVPR335X,
C + IYVPR336X, IYVPR337X, IYVPR338X,
C + IYVPR339X, IYVPR340X, IYVPR341X,
C + IYVPR342X, IYVPR343X, IYVPR344X,
C + IYVPR345X, IYVPR346X, IYVPR347X,
C + IYVPR348X, IYVPR349X, IYVPR350X,
C + IYVPR351X, IYVPR352X, IYVPR353X,
C + IYVPR354X, IYVPR355X, IYVPR356X,
C + IYVPR357X, IYVPR358X, IYVPR359X,
C + IYVPR360X, IYVPR361X, IYVPR362X,
C + IYVPR363X, IYVPR364X, IYVPR365X,
C + IYVPR366X, IYVPR367X, IYVPR368X,
C + IYVPR369X, IYVPR370X, IYVPR371X,
C + IYVPR372X, IYVPR373X, IYVPR374X,
C + IYVPR375X, IYVPR376X, IYVPR377X,
C + IYVPR378X, IYVPR379X, IYVPR380X,
C + IYVPR381X, IYVPR382X, IYVPR383X,
C + IYVPR384X, IYVPR385X, IYVPR386X,
C + IYVPR387X, IYVPR388X, IYVPR389X,
C + IYVPR390X, IYVPR391X, IYVPR392X,
C + IYVPR393X, IYVPR394X, IYVPR395X,
C + IYVPR396X, IYVPR397X, IYVPR398X,
C + IYVPR399X, IYVPR400X, IYVPR401X,
C + IYVPR402X, IYVPR403X, IYVPR404X,
C + IYVPR405X, IYVPR406X, IYVPR407X,
C + IYVPR408X, IYVPR409X, IYVPR410X,
C + IYVPR411X, IYVPR412X, IYVPR413X,
C + IYVPR414X, IYVPR415X, IYVPR416X,
C + IYVPR417X, IYVPR418X, IYVPR419X,
C + IYVPR420X, IYVPR421X, IYVPR422X,
C + IYVPR423X, IYVPR424X, IYVPR425X,
C + IYVPR426X, IYVPR427X, IYVPR428X,
C + IYVPR429X, IYVPR430X, IYVPR431X,
C + IYVPR432X, IYVPR433X, IYVPR434X,
C + IYVPR435X, IYVPR436X, IYVPR437X,
C + IYVPR438X, IYVPR439X, IYVPR440X,
C + IYVPR441X, IYVPR442X, IYVPR443X,
C + IYVPR444X, IYVPR445X, IYVPR446X,
C + IYVPR447X, IYVPR448X, IYVPR449X,
C + IYVPR450X, IYVPR451X, IYVPR452X,
C + IYVPR453X, IYVPR454X, IYVPR455X,
C + IYVPR456X, IYVPR457X, IYVPR458X,
C + IYVPR459X, IYVPR460X, IYVPR461X,
C + IYVPR462X, IYVPR463X, IYVPR464X,
C + IYVPR465X, IYVPR466X, IYVPR467X,
C + IYVPR468X, IYVPR469X, IYVPR470X,
C + IYVPR471X, IYVPR472X, IYVPR473X,
C + IYVPR474X, IYVPR475X, IYVPR476X,
C + IYVPR477X, IYVPR478X, IYVPR479X,
C + IYVPR480X, IYVPR481X, IYVPR482X,
C + IYVPR483X, IYVPR484X, IYVPR485X,
C + IYVPR486X, IYVPR487X, IYVPR488X,
C + IYVPR489X, IYVPR490X, IYVPR491X,
C + IYVPR492X, IYVPR493X, IYVPR494X,
C + IYVPR495X, IYVPR496X, IYVPR497X,
C + IYVPR498X, IYVPR499X, IYVPR500X,
C + IYVPR501X, IYVPR502X, IYVPR503X,
C + IYVPR504X, IYVPR505X, IYVPR506X,
C + IYVPR507X, IYVPR508X, IYVPR509X,
C + IYVPR510X, IYVPR511X, IYVPR512X,
C + IYVPR513X, IYVPR514X, IYVPR515X,
C + IYVPR516X, IYVPR517X, IYVPR518X,
C + IYVPR519X, IYVPR520X, IYVPR521X,
C + IYVPR522X, IYVPR523X, IYVPR524X,
C + IYVPR525X, IYVPR526X, IYVPR527X,
C + IYVPR528X, IYVPR529X, IYVPR530X,
C + IYVPR531X, IYVPR532X, IYVPR533X,
C + IYVPR534X, IYVPR535X, IYVPR536X,
C + IYVPR537X, IYVPR538X, IYVPR539X,
C + IYVPR540X, IYVPR541X, IYVPR542X,
C + IYVPR543X, IYVPR544X, IYVPR545X,
C + IYVPR546X, IYVPR547X, IYVPR548X,
C + IYVPR549X, IYVPR550X, IYVPR551X,
C + IYVPR552X, IYVPR553X, IYVPR554X,
C + IYVPR555X, IYVPR556X, IYVPR557X,
C + IYVPR558X, IYVPR559X, IYVPR560X,
C + IYVPR561X, IYVPR562X, IYVPR563X,
C + IYVPR564X, IYVPR565X, IYVPR566X,
C + IYVPR567X, IYVPR568X, IYVPR569X,
C + IYVPR570X, IYVPR571X, IYVPR572X,
C + IYVPR573X, IYVPR574X, IYVPR575X,
C + IYVPR576X, IYVPR577X, IYVPR578X,
C + IYVPR579X, IYVPR580X, IYVPR581X,
C + IYVPR582X, IYVPR583X, IYVPR584X,
C + IYVPR585X, IYVPR586X, IYVPR587X,
C + IYVPR588X, IYVPR589X, IYVPR590X,
C + IYVPR591X, IYVPR592X, IYVPR593X,
C + IYVPR594X, IYVPR595X, IYVPR596X,
C + IYVPR597X, IYVPR598X, IYVPR599X,
C + IYVPR600X, IYVPR601X, IYVPR602X,
C + IYVPR603X, IYVPR604X, IYVPR605X,
C + IYVPR606X, IYVPR607X, IYVPR608X,
C + IYVPR609X, IYVPR610X, IYVPR611X,
C + IYVPR612X, IYVPR613X, IYVPR614X,
C + IYVPR615X, IYVPR616X, IYVPR617X,
C + IYVPR618X, IYVPR619X, IYVPR620X,
C + IYVPR621X, IYVPR622X, IYVPR623X,
C + IYVPR624X, IYVPR625X, IYVPR626X,
C + IYVPR627X, IYVPR628X, IYVPR629X,
C + IYVPR630X, IYVPR631X, IYVPR632X,
C + IYVPR633X, IYVPR634X, IYVPR635X,
C + IYVPR636X, IYVPR637X, IYVPR638X,
C + IYVPR639X, IYVPR640X, IYVPR641X,
C + IYVPR642X, IYVPR643X, IYVPR644X,
C + IYVPR645X, IYVPR646X, IYVPR647X,
C + IYVPR648X, IYVPR649X, IYVPR650X,
C + IYVPR651X, IYVPR652X, IYVPR653X,
C + IYVPR654X, IYVPR655X, IYVPR656X,
C + IYVPR657X, IYVPR658X, IYVPR659X,
C + IYVPR660X, IYVPR661X, IYVPR662X,
C + IYVPR663X, IYVPR664X, IYVPR665X,
C + IYVPR666X, IYVPR667X, IYVPR668X,
C + IYVPR669X, IYVPR670X, IYVPR671X,
C + IYVPR672X, IYVPR673X, IYVPR674X,
C + IYVPR675X, IYVPR676X, IYVPR677X,
C + IYVPR678X, IYVPR679X, IYVPR680X,
C + IYVPR681X, IYVPR682X, IYVPR683X,
C + IYVPR684X, IYVPR685X, IYVPR686X,
C + IYVPR687X, IYVPR688X, IYVPR689X,
C + IYVPR690X, IYVPR691X, IYVPR692X,
C + IYVPR693X, IYVPR694X, IYVPR695X,
C + IYVPR696X, IYVPR697X, IYVPR698X,
C + IYVPR699X, IYVPR700X, IYVPR701X,
C + IYVPR702X, IYVPR703X, IYVPR704X,
C + IYVPR705X, IYVPR706X, IYVPR707X,
C + IYVPR708X, IYVPR709X, IYVPR710X,
C + IYVPR711X, IYVPR712X, IYVPR713X,
C + IYVPR714X, IYVPR715X, IYVPR716X,
C + IYVPR717X, IYVPR718X, IYVPR719X,
C + IYVPR720X, IYVPR721X, IYVPR722X,
C + IYVPR723X, IYVPR724X, IYVPR725X,
C + IYVPR726X, IYVPR727X, IYVPR728X,
C + IYVPR729X, IYVPR730X, IYVPR731X,
C + IYVPR732X, IYVPR733X, IYVPR734X,
C + IYVPR735X, IYVPR736X, IYVPR737X,
C + IYVPR738X, IYVPR739X, IYVPR740X,
C + IYVPR741X, IYVPR742X, IYVPR743X,
C + IYVPR744X, IYVPR745X, IYVPR746X,
C + IYVPR747X, IYVPR748X, IYVPR749X,
C + IYVPR750X, IYVPR751X, IYVPR752X,
C + IYVPR753X, IYVPR754X, IYVPR755X,
C + IYVPR756X, IYVPR757X, IYVPR758X,
C + IYVPR759X, IYVPR760X, IYVPR761X,
C + IYVPR762X, IYVPR763X, IYVPR764X,
C + IYVPR765X, IYVPR766X, IYVPR767X,
C + IYVPR768X, IYVPR769X, IYVPR770X,
C + IYVPR771X, IYVPR772X, IYVPR773X,
C + IYVPR774X, IYVPR775X, IYVPR776X,
C + IYVPR777X, IYVPR778X, IYVPR779X,
C + IYVPR780X, IYVPR781X, IYVPR782X,
C + IYVPR783X, IYVPR784X, IYVPR785X,
C + IYVPR786X, IYVPR787X, IYVPR788X,
C + IYVPR789X, IYVPR790X, IYVPR791X,
C + IYVPR792X, IYVPR793X, IYVPR794X,
C + IYVPR795X, IYVPR796X, IYVPR797X,
C + IYVPR798X, IYVPR799X, IYVPR800X,
C + IYVPR801X, IYVPR802X, IYVPR803X,
C + IYVPR804X, IYVPR805X, IYVPR806X,
C + IYVPR807X, IYVPR808X, IYVPR809X,
C + IYVPR810X, IYVPR811X, IYVPR812X,
C + IYVPR813X, IYVPR814X, IYVPR815X,
C + IYVPR816X, IYVPR817X, IYVPR818X,
C + IYVPR819X, IYVPR820X, IYVPR821X,
C + IYVPR822X, IYVPR823X, IYVPR824X,
C + IYVPR825X, IYVPR826X, IYVPR827X,
C + IYVPR828X, IYVPR829X, IYVPR830X,
C + IYVPR831X, IYVPR832X, IYVPR833X,
C + IYVPR834X, IYVPR835X, IYVPR836X,
C + IYVPR837X, IYVPR838X, IYVPR839X,
C + IYVPR840X, IYVPR841X, IYVPR842X,
C + IYVPR843X, IYVPR844X, IYVPR845X,
C + IYVPR846X, IYVPR847X, IYVPR848X,
C + IYVPR849X, IYVPR850X, IYVPR851X,
C + IYVPR852X, IYVPR853X, IYVPR854X,
C + IYVPR855X, IYVPR856X, IYVPR857X,
C + IYVPR858X, IYVPR859X, IYVPR860X,
C + IYVPR861X, IYVPR862X, IYVPR863X,
C + IYVPR864X, IYVPR865X, IYVPR866X,
C + IYVPR867X, IYVPR868X, IYVPR869X,
C + IYVPR870X, IYVPR871X, IYVPR872X,
C + IYVPR873X, IYVPR874X, IYVPR875X,
C + IYVPR876X, IYVPR877X, IYVPR878X,
C + IYVPR879X, IYVPR880X, IYVPR881X,
C + IYVPR882X, IYVPR883X, IYVPR884X,
C + IYVPR885X, IYVPR886X, IYVPR887X,
C + IYVPR888X, IYVPR889X, IYVPR890X,
C + IYVPR891X, IYVPR892X, IYVPR893X,
C + IYVPR894X, IYVPR895X, IYVPR896X,
C + IYVPR897X, IYVPR898X, IYVPR899X,
C + IYVPR900X, IYVPR901X, IYVPR902X,
C + IYVPR903X, IYVPR904X, IYVPR905X,
C + IYVPR906X, IYVPR907X, IYVPR908X,
C + IYVPR909X, IYVPR910X, IYVPR911X,
C + IYVPR912X, IYVPR913X, IYVPR914X,
C + IYVPR915X, IYVPR916X, IYVPR917X,
C + IYVPR918X, IYVPR919X, IYVPR920X,
C + IYVPR921X, IYVPR922X, IYVPR923X,
C + IYVPR924X, IYVPR925X, IYVPR926X,
C + IYVPR927X, IYVPR928X, IYVPR929X,
C + IYVPR930X, IYVPR931X, IYVPR932X,
C + IYVPR933X, IYVPR934X, IYVPR935X,
C + IYVPR936X, IYVPR937X, IYVPR938X,
C + IYVPR939X, IYVPR940X, IYVPR941X,
C + IYVPR942X, IYVPR943X, IYVPR944X,
C + IYVPR945X, IYVPR946X, IYVPR947X,
C + IYVPR948X, IYVPR949X, IYVPR950X,
C + IYVPR951X, IYVPR952X, IYVPR953X,
C + IYVPR954X, IYVPR955X, IYVPR956X,
C + IYVPR957X, IYVPR958X, IYVPR959X,
C + IYVPR960X, IYVPR961X, IYVPR962X,
C + IYVPR963X, IYVPR964X, IYVPR965X,
C + IYVPR966X, IYVPR967X, IYVPR968X,
C + IYVPR969X, IYVPR970X, IYVPR971X,
C + IYVPR972X, IYVPR973X, IYVPR974X,
C + IYVPR975X, IYVPR976X, IYVPR977X,
C + IYVPR978X, IYVPR979X, IYVPR980X,
C + IYVPR981X, IYVPR982X, IYVPR983X,
C + IYVPR984X, IYVPR985X, IYVPR986X,
C + IYVPR987X, IYVPR988X, IYVPR989X,
C + IYVPR990X, IYVPR991X, IYVPR992X,
C + IYVPR993X, IYVPR994X, IYVPR995X,
C + IYVPR996X, IYVPR997X, IYVPR998X,
C + IYVPR999X, IYVPR1000X, IYVPR1001X,
C + IYVPR1002X, IYVPR1003X, IYVPR1004X,
C + IYVPR1005X, IYVPR1006X, IYVPR1007X,
C + IYVPR1008X, IYVPR1009X, IYVPR1010X,
C + IYVPR1011X, IYVPR1012X, IYVPR1013X,
C + IYVPR1014X, IYVPR1015X, IYVPR1016X,
C + IYVPR1017X, IYVPR1018X, IYVPR1019X,
C + IYVPR1020X, IYVPR1021X, IYVPR1022X,
C + IYVPR1023X, IYVPR1024X, IYVPR1025X,
C + IYVPR1026X, IYVPR1027X, IYVPR1028X,
C + IYVPR1029X, IYVPR1030X, IYVPR1031X,
C + IYVPR1032X, IYVPR1033X, IYVPR1034X,
C + IYVPR1035X, IYVPR1036X, IYVPR1037X,
C + IYVPR1038X, IYVPR1039X, IYVPR1040X,
C + IYVPR1041X, IYVPR1042X, IYVPR1043X,
C + IYVPR1044X, IYVPR1045X, IYVPR1046X,
C + IYVPR1047X, IYVPR1048X, IYVPR1049X,
C + IYVPR1050X, IYVPR1051X, IYVPR1052X,
C + IYVPR1053X, IYVPR1054X, IYVPR1055X,
C + IYVPR1056X, IYVPR1057X, IYVPR1058X,
C + IYVPR1059X, IYVPR1060X, IYVPR1061X,
C + IYVPR1062X, IYVPR1063X, IYVPR1064X,
C + IYVPR1065X, IYVPR1066X, IYVPR1067X,
C + IYVPR1068X, IYVPR1069X, IYVPR1070X,
C + IYVPR1071X, IYVPR1072X, IYVPR1073X,
C + IYVPR1074X, IYVPR1075X, IYVPR1076X,
C + IYVPR1077X, IYVPR1078X, IYVPR1079X,
C + IYVPR1080X, IYVPR1081X, IYVPR1082X,
C + IYVPR1083X, IYVPR1084X, IYVPR1085X,
C + IYVPR1086X, IYVPR1087X, IYVPR1088X,
C + IYVPR1089X, IYVPR1090X, IYVPR1091X,
C + IYVPR1092X, IYVPR1093X, IYVPR1094X,
C + IYVPR1095X, IYVPR1096X, IYVPR1097X,
C + IYVPR1098X, IYVPR1099X, IYVPR1100X,
C + IYVPR1101X, IYVPR1102X, IYVPR1103X,
C + IYVPR1104X, IYVPR1105X, IYVPR1106X,
C + IYVPR1107X, IYVPR1108X, IYVPR1109X,
C + IYVPR1110X, IYVPR1111X, IYVPR1112X,
C + IYVPR1113X, IYVPR1114X, IYVPR1115X,
C + IYVPR1116X, IYVPR1117X, IYVPR1118X,
C + IYVPR1119X, IYVPR1120X, IYVPR1121X,
C + IYVPR1122X, IYVPR1123X, IYVPR1124X,
C + IYVPR1125X, IYVPR1126X, IYVPR1127X,
C + IYVPR1128X, IYVPR1129X, IYVPR1130X,
C + IYVPR1131X, IYVPR1132X, IYVPR1133X,
C + IYVPR1134X, IYVPR1135X, IYVPR1136X,
C + IYVPR1137X, IYVPR1138X, IYVPR1139X,
C + IYVPR1140X, IYVPR1141X, IYVPR1142X,
C + IYVPR1143X, IYVPR1144X, IYVPR1145X,
C + IYVPR1146X, IYVPR1147X, IYVPR1148X,
C + IYVPR1149X, IYVPR1150X, IYVPR1151X,
C + IYVPR1152X, IYVPR1153X, IYVPR1154X,
C + IYVPR1155X, IYVPR1156X, IYVPR1157X,
C + IYVPR1158X, IYVPR1159X, IYVPR11
```


SUBROUTINE INTLZ 74/74 OPT=1 FIN 4.6+428 83/11/07 09.41 53 PAGE 141

```

1  SUBROUTINE INTLZ
C .....
C DESCRIPTION - LEVEL 2
C FUNCTION - CONTROLS INITIALIZATION
5  C METHOD - EXECUTES A SERIES OF CALLS TO SUBROUTINES CONTROLLING
C COMMUNICATIONS - SPECIFIC INITIALIZATION FUNCTIONS
C CALLED BY: GESS
C CALLS:
10 C AEROIN
C CLEAR
C INITMS
C INITRAJ
15 C INIVECT
C INIVRBL
C INPUT
C REPORTS
C REPT1
20 C NON-COMMON VARIABLES DEFINED:
C NONE
C POTENTIAL ERROR CONDITIONS:
C NONE
C .....
25 C CALL CLEAR
C CALL INPUT
C CALL INIVRBL
C CALL AEROIN
C CALL INITMS
30 C CALL INIVECT
C CALL INITRAJ
C CALL REPT1
C CALL REPORTS
35 C RETURN
C END

```

```

1 SUBROUTINE INTRP(QCOEFF,QFD,QSD,QTD,QDLTII,QISI,QTYP,V1,V2,V3,
1 INTVAL,IOUT)

```

```

5 C .....
C DESCRIPTION - LEVEL 3
C FUNCTION -
C THE ROUTINE IS USED WHEN INTERPOLATING WITH A
C FUNCTION OF TWO OR THREE VARIABLES.
C
C METHOD -
C THE METHOD USED IS NEWTON'S FIRST ORDER INTERPOLATION.
C IT IS NECESSARY TO LOCATE THE FOUR SEGMENT POINTS REQUIRED
C TO PERFORM THE INTERPOLATION IN THE TWO-WAY CASE. OR THE
C EIGHT POINTS REQUIRED FOR THE THREE-WAY CASE. THIS IS
C ACCOMPLISHED BY DETERMINING I, J, AND K WITHIN THE SEGMENT,
C WHERE I SATISFIES THE CONDITION  $X(I) \leq V1 \leq X(I+1)$ ; J
C SATISFIES  $Y(J) \leq V2 \leq Y(J+1)$ ; AND K SATISFIES
C  $Z(K) \leq V3 \leq Z(K+1)$ . THESE VALUES ARE THEN USED IN
C NEWTON'S FORMULA.
C
C NOTE RESTRICTIONS:
C V1, V2, AND V3 MUST APPEAR WITHIN THE BOUNDS OF THE
C TABLE.
C
C COMMUNICATIONS -
C CALLED BY : AERFMSA
C AERFMSO
C
C CALLS : NONE
C
C NON-COMMON VARIABLES DEFINED :
C QCOEFF - THE BEGINNING OF THE COEFFICIENT ARRAY.
C QFD - THE NUMBER OF X-POINTS (INTEGER).
C QSD - THE NUMBER OF Y-POINTS (INTEGER).
C QTD - THE NUMBER OF Z-POINTS (INTEGER).
C QDLTII - THE ARRAY NAME THAT CONTAINS THE
C INCREMENTS FOR THE INDEPENDENT VARIA-
C BLES (REAL).
C QDLTII(1) CONTAINS DELTA X.
C QDLTII(2) CONTAINS DELTA Y.
C QDLTII(3) CONTAINS DELTA Z.
C
C QISI - THE NAME OF THE ARRAY CONTAINING THE END
C POINTS OF THE INDEPENDENT VARIABLES (REAL).
C QISI(1) CONTAINS X(1)
C QISI(2) CONTAINS X(QFD)
C QISI(3) CONTAINS Y(1)
C QISI(4) CONTAINS Y(QFD)
C QISI(5) CONTAINS Z(1)
C QISI(6) CONTAINS Z(QFD)
C
C QTYP - TYPE OF INTERPOLATION (INTEGER).
C = 1, IF INTERPOLATION IS A FUNCTION OF TWO
C VARIABLES
C = 2, IF INTERPOLATION IS A FUNCTION OF THREE
C VARIABLES.
C
C V1 - THE COMPUTED X TABLE ENTRY (REAL).
C V2 - THE COMPUTED Y TABLE ENTRY (REAL).
C V3 - THE COMPUTED Z TABLE ENTRY (REAL).
C INTVAL - THE INTERPOLATED COEFFICIENT (REAL).
C IOUT - TABLE LIMITS EXCEEDED ERROR FLAG (REAL).

```

```

C      0 - V1, V2, V3 WITHIN TABLE LIMITS.
C      1 - V1 OUTSIDE TABLE LIMITS
C      2 - V2 OUTSIDE TABLE LIMITS
C      3 - V3 OUTSIDE TABLE LIMITS
C      C POTENTIAL ERROR CONDITIONS
C      NONE
C.....
65      DIMENSION QCOEFF(QD,QSD,QTD),QDLTII(3),QISI(6)
      INTEGER QJ,JL,QK
      INTEGER QFD,QSD,QTD
      REAL INTVAL
      IOUT = 0
      IF (V1 .GE. QISI(1) .AND. V1 .LE. QISI(2)) GOTO 100
      IOUT = 1
      GOTO 30
      100 IF (V2 .GE. QISI(3) .AND. V2 .LE. QISI(4)) GOTO 110
      IOUT = 2
      GOTO 30
      110 IF (QJTP .EQ. 1) GOTO 120
      IF (V3 .GE. QISI(5) .AND. V3 .LE. QISI(6)) GOTO 120
      IOUT = 3
      GOTO 30
      120 CONTINUE
C.....
C      DETERMINE THE INDEX FOR THE 1ST VARIABLE
C.....
      QI=1+IFIX((V1-QISI(1))/QDLTII(1))
      QR=(V1-QI*QISI(1))/QDLTII(1)
C.....
C      DETERMINE THE INDEX FOR THE 2ND VARIABLE
C.....
      QJ=1+IFIX((V2-QISI(3))/QDLTII(2))
      QS=(V2-QJ*QISI(3))/QDLTII(2)
C.....
C      TEST IF A FUNCTION OF 2 VARIABLES ONLY
C.....
      IF (QJTP .EQ. 1) GO TO 10
C.....
C.....
C      DETERMINE THE INDEX FOR THE 3RD VARIABLE
C.....
      QK=1+IFIX((V3-QISI(5))/QDLTII(3))
      QT=(V3-QK*QISI(5))/QDLTII(3)
      IF (QI .EQ. QFD .AND. QJ .EQ. QSD .AND. QK .EQ. QTD) GO TO 20
C.....
C.....
C      INTERPOLATE THE FUNCTION, F(X,Y,Z)
C.....
      INTVAL = QT*QR*QCOEFF(QI+1,QJ+1,QK+1) + QR*QS*(1-QT) +
      1 QCOEFF(QI+1,QJ+1,QK) + QS*QT*(1-QR) +
      2 QCOEFF(QI,QJ+1,QK+1) + QR*QT*(1-QS) +
      3 QCOEFF(QI+1,QJ,QK+1) + QS*(1-QR)*(1-QT) +
      4 QCOEFF(QI,QJ+1,QK) + QR*(1-QS)*(1-QT) +
      5 QCOEFF(QI+1,QJ,QK) + QT*(1-QS)*(1-QR) +
      6 QCOEFF(QI,QJ,QK+1) + (1-QR)*(1-QS)*(1-QT)

```



```

115      7      QC0EFF(QI,QJ,QK)
      GOTO 30
C.....
C      INTERPOLATE THE FUNCTION F(X,Y)
C.....
120      10 INTVAL = QR*OS*QC0EFF(QI+1,QJ+1,1) + QR*(1.-OS)*
      1      QC0EFF(QI+1,QJ,1) + OS*(1.-QR)*QC0EFF(QI,QJ+1,1) +
      2      (1.-QR)*(1.-OS)*QC0EFF(QI,QJ,1)
      GOTO 30
125      20 INTVAL=QC0EFF(QI,QJ,QK)
      30 RETURN
      END

```

```

1      SUBROUTINE LADDATE(IDATE)
C .....
C DESCRIPTION - LEVEL 4
C FUNCTION - DETERMINE PRESENT DATE IN THE FORM DD MON YY
C METHOD - OBTAIN PRESENT DATE FROM SYSTEM AND ENCODE IT
C COMMUNICATIONS -
C CALLED BY: INIVRBL
C
C CALLS:
C
C NON-COMMON VARIABLES DEFINED -
C MONTHS - ARRAY CONTAINING ENCODED MONTHS FOR PRINTOUT
C MONTH - DECODED MONTH INDEX INTO MONTHS ARRAY
C YEAR - DECODED YEAR
C DAY - DECODED DATE
C POTENTIAL ERROR CONDITIONS -
C
C .....
C INTEGER MONTHS(12), DAY, MONTH, YEAR
C DATA MONTHS/ 3HJAN, 3HFEB, 3HMAR, 3HAPR, 3HMAY, 3HJUN,
C + 3HJUL, 3HAUG, 3HSEP, 3HOCT, 3HNOV, 3HDEC /
C CALL DATE(0)
C DECODE (9,101,D) YEAR,MONTH,DAY
C IF (DAY .GE. 10) ENCODE(10,102,IDATE) DAY, MONTHS(MONTH), YEAR
C IF (DAY .LT. 10) ENCODE(10,103,IDATE) DAY, MONTHS(MONTH), YEAR
C RETURN
101 FORMAT (1X,12,1X,12,1X,12)
102 FORMAT (12,1X,A3,1X,12,1X)
103 FORMAT (11,1X,A3,1X,12,2X)
END
30

```

```

1  SUBROUTINE MATRIX(A,B,C,KODE)
C .....
C
C  KODE= 1      C= A * B
C
C  KODE= 2      C= A * (B )T
C
C  KODE= 3      C= (A ) * BT
C
C  KODE= 4      C= (A ) * (B )T
C
C  WHERE  A = TRANSPOSE OF A
C  AND
C  B = TRANSPOSE OF B
C .....
C
C  DIMENSION A(3,3),B(3,3),C(3,3)
C  GO TO(100,200,300,400) KODE
C
C 100 CONTINUE
C
C  KODE= 1      C= A * B
C
C  DO 150 I=1,3
C  DO 150 J=1,3
C  C(I,J)= 0.0
C  DO 150 K=1,3
C  C(I,J)= C(I,J) + A(I,K) * B(K,J)
C 150 CONTINUE
C  RETURN
C
C 200 CONTINUE
C
C  KODE= 2      C= A * (B )T
C
C  DO 250 I=1,3
C  DO 250 J=1,3
C  C(I,J)= 0.0
C  DO 250 K=1,3
C  C(I,J)= C(I,J) + A(I,K) * B(J,K)
C 250 CONTINUE
C  RETURN
C
C 300 CONTINUE
C
C  KODE= 3      C= (A ) * BT
C
C  KODE= 4      C= (A ) * BT

```

PAGE 147

83/11/07 09 41 53

FIN 4 6+428

SUBROUTINE MATRIX 74/74 OPT=1

```

60      DO 350 I=1,3
          DO 350 J=1,3
            C(I,J)= 0.0
          DO 350 K=1,3
            C(I,J)= C(I,J) + A(K,I) * B(K,J)
          CONTINUE
        RETURN
      350 C
      400 CONTINUE
      C
      70      CODE= 4
          C= (A ) * (B )
          T
      DO 450 I=1,3
        DO 450 J=1,3
          C(I,J)= 0.0
        DO 450 K=1,3
          C(I,J)= C(I,J) + A(K,I) * B(J,K)
        CONTINUE
      450 C
      RETURN
      END

```

```

1  SUBROUTINE MATUPD(O,DCM)
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - UPDATES A TRANSFORMATION MATRIX
C METHOD - SUBROUTINE MATUPD USES THE QUATERNIONS THAT WERE
C PREVIOUSLY UPDATED TO CALCULATE A NEW 3 X 3
C TRANSFORMATION MATRIX
C COMMUNICATIONS -
C CALLED BY:
C TMUPDAT
C CALLS:
C NONE
C NON-COMMON VARIABLES DEFINED:
C Q(1) -
C Q(2) - NORMALIZED QUATERNIONS
C Q(3) -
C Q(4) -
C Q00 -
C Q01 -
C Q02 -
C Q03 - INTERMEDIATE VALUES IN CALCULATIONS
C Q12 - TO UPDATE MATRIX
C Q13 -
C Q23 -
C DCM(3,3) - 3 X 3 TRANSFORMATION MATRIX BEING UPDATED
C
C POTENTIAL ERROR CONDITIONS: NONE
C .....
C DIMENSION DCM(3,3),Q(4)
C
C Q00= Q(1) * Q(1)
C Q01= Q(1) * Q(2)
C Q02= Q(1) * Q(3)
C Q03= Q(1) * Q(4)
C Q12= Q(2) * Q(3)
C Q13= Q(2) * Q(4)
C Q23= Q(3) * Q(4)
C DCM(1,1)= 2. * (Q00 + Q(2) * Q(2) - 1.
C DCM(2,2)= 2. * (Q00 + Q(3) * Q(3) - 1.
C DCM(3,3)= 2. * (Q00 + Q(4) * Q(4) - 1.
C DCM(1,2)= 2. * (Q12 + Q03)
C DCM(1,3)= 2. * (Q13 - Q02)
C DCM(2,1)= 2. * (Q12 - Q03)
C DCM(2,3)= 2. * (Q23 + Q01)
C DCM(3,1)= 2. * (Q13 + Q02)
C DCM(3,2)= 2. * (Q23 - Q01)
C
C RETURN
C END

```

```

1  SUBROUTINE OCCALON
C.....
C DESCRIPTION - LEVEL 2
C FUNCTION - CALCULATE OCCUPANT ALONE TRAJECTORY PARAMETERS
C METHOD - OCCUPANT ALONE EQUATIONS OF MOTION ARE COMPUTED
C FROM SEAT/OCCUPANT SEPARATION TO IMPACT.
C AERODYNAMIC AND RECOVERY CHUTE FORCES ACT ON THE
C OCCUPANT ALONE. FORCES AND MOMENTS ARE SUMMED,
C AND LINEAR ACCELERATIONS ARE CALCULATED IN THE
C EARTH FIXED COORDINATE SYSTEM.
C COMMUNICATIONS -
C CALLED BY: GESS
C CALLS:
C ATMOS, AERFMOA, CHUTES
C NON-COMMON VARIABLES DEFINED:
C OLDALT(2) - LATEST PREVIOUS ALTITUDE OF OCCUPANT ALONE
C FX - COMPONENTS OF SUMMED AERODYNAMIC AND PARACHUTE
C FZ - FORCES
C TL0A -
C TMOA - COMPONENTS OF SUMMED AERODYNAMIC AND PARACHUTE
C TMOA - MOMENTS
C POTENTIAL ERROR CONDITIONS: NONE
C.....
C CONSTANTS COMMON BLOCK
C.....
C COMMON /CONSTANT / GRAVITY , RADDEG , DEGRAD , PI
C DENSITY COMMON BLOCK
C.....
C COMMON /DENSITY / IATMOS , OLDALT(3) , RHOS
C + PRESALT(3) , DTEMP
C + TEMPS
C + VZWIND , VYVIND , VZWIND
C.....
C OCCUPANT ALONE FORCES COMMON BLOCK
C.....
C COMMON /FORCEOA / FXCHOA(3) , FYCHOA(3) , FZCHOA(3) ,
C + FXAEOA , FYAEOA , FZAEOA
C.....
C SECTION 6 COMMON BLOCK
C.....
C COMMON /ISEATOC / IPCNTL , XCGSO , YCGSO , ZCGSO , IXXSO ,
C + IYXSO , IYZSO , IYXSO , IYZSO , IZZSO ,
C + AREASO , AREAOA , WGTTOAB , WGTTOAA ,
C + IXXOA , IYXOA , IYZOA , IYZOA , IYZOA ,
C + IZZOA , XCGOA , YCGOA , ZCGOA , S0SEP ,
C + C1SO , C2SO , C3SO , C4SO ,
C + C1OA , C2OA , C3OA , C4OA ,
C + REAL
C + IXXSO , IYXSO , IYZSO , IYZSO ,
C + IZZSO , IXXOA , IYXOA , IYZOA , IYZOA ,
C.....
C MASSES COMMON BLOCK
C.....

```

```

COMMON /MASSES / MASSOA1 , MASSOA2 , MASSSO , MASSO
+
REAL MASSA , MASSRK(6) , MASSDC , MASSO
+
MASSOA1 , MASSOA2 , MASSSO , MASSO
MASSA , MASSRK , MASSDC
C *****
C MATRIX COMMON BLOCK
C *****
COMMON /MATRIX / DCMSE(3,3) , DCMRA(3,3) , DCMSE(3,3) ,
+
DCMSAE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,
+
DCMSAE(3,3) , DCMOAE(3,3) , DCMSE(3,3) ,
+
DCMDUM(3,3)
C *****
C MISCELLANEOUS DATA COMMON BLOCK
C *****
COMMON /MISC / IPAGCT(31) , LINECT(31) , IPRCNT(31)
+
MAXLINE , MAXREPT , MAXEVT
+
IEVLNE , IERRFLG , LU
+
IDATE , HEADALT , HEADVEL
+
HEADSR , HEADYAW , HEADPIT
+
HEADROL , HEADWGT , BIAS
+
REPTYPE(5,31) , PRTNGT(2) , PRTWGT(2)
+
IHEADER(24) , IEVENTS(38) , TIMES(38)
+
PRTMAS(2) , PRTINDX , PKZVEL
+
ZVECT(3) , XYZ(3) , SAVTIME
+
XACCEL(3) , YACCEL(3) , ZACCEL(3)
+
REPTYPE , BIAS , PRTNGT
+
PRTWGT , PRTEMP , PRTMAS , PRTINDX
C *****
C INTEGRATION ROUTINE COMMON BLOCK
C *****
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193)
+
TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3)
+
TRAJAC(193) , TVCEQS(225) , QUATSO(65)
+
QUATOA(65) , QUATAC(65) , IRKPASS
+
INTSTP , IPCPASS , IRKPRX
+
IPOINTS , IYX , IYPRX
+
IKX , IKSUMX , IKPASSX
+
IYIX , IYIXX , IYI2X
+
IYIGX , IYPRIX , IYPR1X
+
IYPR2X , IYPIIX , IYPI1X
+
ICYIX , ICYI1X , IREIN
C *****
C TORQUE OCCUPANT ALONE COMMON BLOCK
C *****
COMMON /TORQOA / TLCHOA(3) , TMCHOA(3) , TNCHOA(3) ,
+
TLAEOA , TMAEOA , TNAEOA
C *****
C
C
REAL IXZP,IYZO,IZZR,IXXP,IXYQ,IXYP,IXZR,IYYQ,IYZR
IF(IEVENTS(28) .EQ. 0) GOTO 500
IF(INTSTP .EQ. 0) GOTO 10
C *****
C SAVE PREVIOUS Z-VELOCITY FOR PEAK TRAJECTORY
C *****
PKZVEL = TRAJOA(16)

```

```

115 IF (TRAJDA(4) .GT. 0.0) GOTO 10
    TRAJDA(1) = 0.0
    IVENTS(30) = 1
    TIMES(30) = TIME
    GOTO 500

120 C
    C 10 CONTINUE
    C
    C .....
    C COMPUTE VELOCITY COMPONENTS IN EFCS
    C .....
    C
    TRAJDA(14) = TRAJDA(5)
    TRAJDA(15) = TRAJDA(6)
    TRAJDA(16) = TRAJDA(7)
    TRAJDA(20) = TRAJDA(11)
    TRAJDA(21) = TRAJDA(12)
    TRAJDA(22) = TRAJDA(13)
    C
    C .....
    C COMPUTE FORCES AND MOMENTS ON OCCUPANT ALONE
    C .....
    C
    CALL ATMOS(TRAJDA(4), OLDALT(2), PRESALT(2))
    OLDALT(2) = TRAJDA(4)
    CALL AERFMOA
    CALL CHUTES
    C
    C .....
    C SUM FORCES
    C .....
    C
    FX = FXAEOA + FXCHOA(3)
    FY = FYAEOA + FYCHOA(3)
    FZ = FZAEOA + FZCHOA(3)
    C
    TL0A = TLAEOA + TLCHOA(3)
    TMOA = TMAEOA + TMCHOA(3)
    TNOA = TNAEOA + TNCHOA(3)
    C
    C .....
    C COMPUTE ACCELERATIONS IN OACS
    C .....
    C POINT MASS ASSUMED - NO ANGULAR MOTION ALLOWED
    C .....
    C
    XACCEL(2) = FX/MASSOA2 - GRAVITY*DCMOAE(1,3)
    TRAJDA(17) = XACCEL(2)
    C
    + TRAJDA(13)*TRAJDA(6) - TRAJDA(12)*TRAJDA(7)
    YACCEL(2) = FY/MASSOA2 - GRAVITY*DCMOAE(2,3)
    TRAJDA(18) = YACCEL(2)
    C
    + TRAJDA(11)*TRAJDA(7) - TRAJDA(13)*TRAJDA(5)
    ZACCEL(2) = FZ/MASSOA2 - GRAVITY*DCMOAE(3,3)
    TRAJDA(19) = ZACCEL(2)
    C
    + TRAJDA(12)*TRAJDA(5) - TRAJDA(11)*TRAJDA(6)
    TRAJDA(17) = XACCEL(2) = FX/MASSOA2
    TRAJDA(18) = YACCEL(2) = FY/MASSOA2

```


TRAJDA(19) = ZACCEL(2) * FZ/MASSOA2 - GRAVITY

```

C .....
C ANGULAR MOMENTUM EQUATIONS .....
C .....
C POINT MASS ASSUMED - NO ANGULAR MOTION ALLOWED .....
C .....
C
C IXZP = IXZOA * TRAJDA(11)
C IYZO = IYZOA * TRAJDA(12)
C IZZR = IZZOA * TRAJDA(13)
C IXXP = IXXOA * TRAJDA(11)
C IXYQ = IXYOA * TRAJDA(12)
C IXYP = IXYOA * TRAJDA(11)
C IXZR = IXZOA * TRAJDA(13)
C IYQO = IYQOA * TRAJDA(12)
C IYZR = IYZOA * TRAJDA(13)
C
C Taux = TRAJDA(12) * (IXZP + IYZO - IZZR) - TRAJDA(13) *
+ (IXYP - IYQO + IYZR) + TLOA
C TauY = TRAJDA(13) * (IXZR + IXYQ - IXXP) - TRAJDA(11) *
+ (IXZP + IYZO - IZZR) + TMOA
C TauZ = TRAJDA(11) * (IXYP - IYQO + IYZR) - TRAJDA(12) *
+ (IXZR + IXYQ - IXXP) + TNOA
C
C TRAJDA(24) = ((Taux * C1OA + TauY * C2OA + TauZ * C3OA)/C4OA)
C TRAJDA(23) = ((IZZO * (Taux + IXYOA * TRAJDA(24)) +
+ IXZOA * (TauY + IYZOA * TRAJDA(24)))/C2OA)
C TRAJDA(25) = ((TauZ + IXZOA * TRAJDA(23) + IYZOA *
+ TRAJDA(24))/IZZO)
C
C
C 500 CONTINUE
C RETURN
C END

```

```

1 C SUBROUTINE PCHUTFT(V0,DP,CD,POROSTY,TFP)
C .....
5 C DESCRIPTION - LEVEL 4
C FUNCTION - PCHUTFT CALCULATES THE AREA UNDER THE CURVE
C METHOD - REPRESENTING CURRENT PARACHUTE VOLUME FOR VALUES OF
C T FROM 0.0 TO 1.0, ASSUMING SOME VALUE FOR PARACHUTE
C FILLING TIME. THE AREA IS THEN COMPARED TO ACTUAL
C PARACHUTE VOLUME AT FULL INFLATION. IF THE TWO
C VOLUMES VARY BY MORE THAN 1%, A NEW VALUE FOR
C PARACHUTE FILLING TIME IS ASSUMED, AND THE AREA UNDER
C THE NEW VOLUME CURVE IS CALCULATED. THE PROPER VALUE
C FOR PARACHUTE FILLING TIME HAS BEEN REACHED WHEN THE
C TWO VOLUMES ARE WITHIN A TOLERANCE OF 1%.
15 C COMMUNICATIONS -
C CALLED BY:
C DROGUE1
C DROGUE2
C RECOV
20 C CALLS:
C NONE
C NON-COMMON VARIABLES DEFINED -
C ICNT - COUNTER TO KEEP TRACK OF NUMBER OF ITERATIONS
C NEEDED TO FIND THE FILLING TIME
25 C GRAVITY - ACCELERATION DUE TO GRAVITY IN METRIC OR ENGLISH UNITS
C DODG - NOMINAL DIAMETER OF THE PARACHUTE
C SP - PROJECTED AREA OF THE PARACHUTE
C VMAX - ACTUAL MAXIMUM PARACHUTE VOLUME
30 C VOL - CALCULATED PARACHUTE VOLUME
C AREA - AREA UNDER THE CURVE REPRESENTING PARACHUTE VOLUME
C TFP - PARACHUTE FILLING TIME
C POTENTIAL ERROR CONDITIONS -
C DIVISION BY ZERO WHEN VMAX, A, OR AREA
C EQUALS ZERO
C .....
40 C .....
C CONSTANTS COMMON BLOCK
C .....
C COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C DENSITY COMMON BLOCK
C .....
C COMMON /DENSITY / IATMOS , OLDALT(3) , RHOS
C + PRESALT(3) , DTEMP
C + TEMPS
C + VFWIND , VFWIND , VZWIND
50 C .....
C SECTION 1 COMMON BLOCK
C .....
C COMMON /ICONTROL / TSTART , TSTOP , ESTOP , IRESTR1, IUNITS ,
C + ISEATTR, ISDSEP , IPLOT , IDRIFLG,
C + IPHASE1, IPHASE2, IPHASE3
C INTEGER
C ESTOP

```

```

C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31) , LINECT(31) , MAXREPT , IPRICT(31) ,
+ MAXEVNT ,
+ MAXLINE , TERRFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS ,
+ REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ PRTEMP( 2) ,
+ PRIMASS(2) , PRTINDX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) ,
+ REPTYPE , BIAS , ZACCEL(3) ,
+ PRTWGT , PRTLNCT ,
+ PRTEMP , PRIMASS , PRTINDX
C.....
C SECTION 6 COMMON BLOCK
C.....
COMMON /ISEATOC / IPCNTL , XCGSO , YCGSO , ZCGSO , IXXSO ,
+ IYXSO , IYZSO , IYXSO , IYZSO , IZZSO ,
+ AREASO , AREADA , WGHTOAB , WGHTOAA ,
+ IXXOA , IXYOA , IYZOA , IYXOA , IYZOA ,
+ IZZOA , XCGOA , YCGOA , ZCGOA , SOSEP ,
+ REAL , IXXSO , IYXSO , IYZSO , IYXSO , IYZSO ,
+ IZZSO , IXXOA , IXYOA , IYZOA , IYZOA ,
+ IZZOA , IZZOA
C.....
C SECTION 14 COMMON BLOCK
C.....
COMMON /PARCHUT / IRECOV , TROPLOY , RECOVLL ,
+ RECOVPR , POROSR ,
+ XRECAP , YRECAP , ZRECAP ,
+ NPTSRLS , RECOVLS(2,25) , IFTRECV ,
+ NPTSRTF , RECOVFT(2,28) , SEPRCE ,
+ IDROGUE , DRDRAG2 , DRGPD2 ,
+ POROSD2 , VELCON , IFTDR02 ,
+ NPTDFT1 , DRGFT2(2,25) , IFTDR01 ,
+ NPTDFT1 , DRGFT1(2,25) , IDROGLS ,
+ NPTSOLS , DRGOLS(2,25) , TROPLOY ,
+ DISPLAY , DRDGLL , DRDRAG1 ,
+ DRGPD1 , POROSD1 , DROVELX ,
+ DROVELY , XROGAP ,
+ YROGAP , ZROGAP ,
+ CHALT2 , GLIMIT ,
+ AREADC , TDELY ,
+ TFP2 , TFP1 ,
+ CODC , NPTSRTD ,
+ RECOVDT(2,25)
C.....

```

```

I = J = K = ICNT = 0
CODC = .5 * PI * DP
SP = .25 * PI * OP ** 2
VMAX = (PI * DP ** 3) / 12.
IF(VMAX EQ 0.0) GO TO 135
FTP1 = 1.000

```

```

115 TFP = FTP1
    A = 5 * WHTOAA * 10000 * (GRAVITY * (RHOS/.0023769) * DDDC ** 3)
    + ** (-1)
    IF (A .EQ. 0.0) GO TO 145
15 B = TFP * 120 * CD * SP * DDDC ** (-3)
C
C *****
C CALCULATE AREA UNDER THE CURVE REPRESENTING PARACHUTE VOLUME *****
C *****
20 I = 1 + 1
    PRINT *, "I, TFP"
    PRINT 21, I, TFP
21 FORMAT(2X, I3, 2X, F10.4)
    IF (I .GT. 50) GO TO 190
    TD = 0.0
    P1 = 1 + 2.2 * POROSTV * TD - TD
    P2 = (B * VO) / 253.125
    P3 = 11.25 * TD + A
    R1 = ((P1 * TD ** (1./3.)) - 2.2 * POROSTV) * TD
    R2 = P3 * ALOG(P3/A) - 11.25 * TD
    IF ((P2 * R2 + (P3/A)) .EQ. 0.0) GO TO 145
    AREA = R1 / (P2 * R2 + (P3/A))
    DO 35 K = 1, 9
    S = K
    TD = S / 10.
    P1 = 1 + 2.2 * POROSTV * TD - TD
    P3 = 11.25 * TD + A
    R1 = ((P1 * TD ** (1./3.)) - 2.2 * POROSTV) * TD
    R2 = P3 * ALOG(P3/A) - 11.25 * TD
    IF ((P2 * R2 + (P3/A)) .EQ. 0.0) GO TO 145
    IF (MOD(K, 2) .EQ. 0) GO TO 25
    AREA1 = 4 * (R1 / (P2 * R2 + P3/A))
    GO TO 30
25 AREA1 = 2 * (R1 / (P2 * R2 + P3/A))
30 AREA = AREA + AREA1
35 CONTINUE
    TD = 1.0
    P1 = 1 + 2.2 * POROSTV * TD - TD
    P3 = 11.25 * TD + A
    R1 = ((P1 * TD ** (1./3.)) - 2.2 * POROSTV) * TD
    R2 = P3 * ALOG(P3/A) - 11.25 * TD
    IF ((P2 * R2 + (P3/A)) .EQ. 0.0) GO TO 155
    AREA1 = R1 / (P2 * R2 + P3/A)
    AREA = (1./3.) * (AREA + AREA1)
C
C *****
C COMPARE CALCULATED WITH ACTUAL PARACHUTE VOLUME *****
C *****
40 IF (AREA .EQ. 0.0) GO TO 165
    VOL = TFP * VO * (DDDC ** 2) / PI * AREA
    AREA = 0.0
    IF (ABS((VMAX - VOL) / VMAX) .LE. 0.01) GO TO 200
    IF (J .EQ. 0) GO TO 45
    IF (J .EQ. 1 .AND. ((VMAX - VOL) / VMAX) .GT. 0.0

```

```

175      + .AND. ((VMAX - VOL1) / VMAX) .GT. 0.0) GO TO 70
      J = J + 1
      GO TO 75
45      J = J + 1
      VOL1 = VOL
      IF ((VMAX - VOL) / VMAX) 50,200,80
C
C DECREASE TFP
50      FTP2 = FTP1 - .5 * FTP1
      GO TO 65
C INCREASE TFP
60      FTP2 = FTP1 + .5 * FTP1
65      TFP = FTP2
      GO TO 15
70      FTP2 = FTP2 + .5 * FTP1
      GO TO 65
C
75      ICNT = ICNT + 1
      IF (MOD(ICNT,3) .EQ. 0) GO TO 115
      IF (MOD(ICNT,2) .EQ. 0) GO TO 95
C
      IF ((VMAX - VOL) / VMAX) 80,200,85
C DECREASE TFP
80      FTP3 = FTP2 - .5 * ABS(FTP2 - FTP1)
      GO TO 90
C INCREASE TFP
85      FTP3 = FTP2 + .5 * ABS(FTP2 - FTP1)
90      TFP = FTP3
      GO TO 15
200      C
95      IF ((VMAX - VOL) / VMAX) 100,200,105
C DECREASE TFP
100      FTP1 = FTP3 - .5 * ABS(FTP3 - FTP2)
      GO TO 110
C INCREASE TFP
105      FTP1 = FTP3 + .5 * ABS(FTP3 - FTP2)
110      TFP = FTP1
      GO TO 15
210      C
115      IF ((VMAX - VOL) / VMAX) 120,200,125
C DECREASE TFP
120      FTP2 = FTP1 - .5 * ABS(FTP1 - FTP3)
      GO TO 130
C INCREASE TFP
125      FTP2 = FTP1 + .5 * ABS(FTP1 - FTP3)
130      TFP = FTP2
      ICNT = 0
      GO TO 15
220      C
135      WRITE(5,170)
      GO TO 199
145      WRITE(5,175)
      GO TO 199
155      WRITE(5,180)
      GO TO 199
165      WRITE(5,185)
      GO TO 199
225

```

PAGE 157

83/11/07 09 41.53

FTN 4 6+428

74/74 OPT=1

SUBROUTINE PCHUTFT

```

230      170 FORMAT(2X, //72(1H*))/4X, "FATAL ERROR(SUBROUTINE PCHUTFT)*** ./,
      + "VMAX CALCULATED TO BE EQUAL TO ZERO RESULTS IN DIVISION BY 0",
      + /.72(1H*) )
      175 FORMAT(2X, //72(1H*))/4X, "FATAL ERROR(SUBROUTINE PCHUTFT)*** ./,
      + "A CALCULATED TO BE EQUAL TO ZERO RESULTS IN DIVISION BY ZERO",
      + /.72(1H*) )
235      180 FORMAT(2X, //72(1H*))/4X, "FATAL ERROR(SUBROUTINE PCHUTFT)*** ./,
      + "(P2-R2+P3/A) CALC TO BE EQUAL TO 0 RESULTS IN DIVISION BY 0",
      + /.72(1H*) )
      185 FORMAT(2X, //72(1H*))/4X, "FATAL ERROR(SUBROUTINE PCHUTFT)*** ./,
      + "AREA CALCULATED TO BE EQUAL TO ZERO RESULTS IN DIVISION BY 0",
      + /.72(1H*) )
240      190 PRINT 195
      195 FORMAT(5X, 36HPA -VOL HAS GONE THRU 500 ITERATIONS)
      199 CONTINUE
      IERRFLG = 1
245      200 CONTINUE
      RETURN
      END

```

```

SUBROUTINE PLOTBIN
C .....
C DESCRIPTION - LEVEL 2
C FUNCTION - CREATE PLOTTING FILE
C METHOD - WRITES TRAJECTORY DATA TO PLOTTING FILE EVERY .05 SEC
C          IN A FORMAT COMPATIBLE WITH DRAS PLOTTING PROGRAM
C COMMUNICATIONS :
C   CALLED BY :
C         GESS
C NON-COMMON VARIABLES DEFINED -
C   NONE
C POTENTIAL ERROR CONDITIONS -
C   NONE
C .....
C CONSTANTS COMMON BLOCK
C .....
COMMON /CONSTANT / GRAVITY , RADDEG , DEGRAD , PI
C .....
C SECTION 1 COMMON BLOCK
C .....
COMMON /CONTROL / TSTART , TSTOP , ESTOP , IRESTRT , UNITS ,
+
+
+ ISCATTR , ISOSEP , IPLOT , IDRIFLG ,
+
+ IPHASE1 , IPHASE2 , IPHASE3
C .....
INTEGER      ESTOP
C .....
C SECTION 3 COMMON BLOCK
C .....
COMMON /DELTA / DTPHASE1 , DTPHASE2 , DTPHASE3
C .....
C SECTION 2 COMMON BLOCK
C .....
COMMON /IREPORT / IREPTS(31) , PRFRQ , PT1 , PT2 , PT3
C .....
INTEGER
C .....
C MISCELLANEOUS DATA COMMON BLOCK
C .....
COMMON /MISC / PAGECT(31) , LINECT(31) , IPTCNT(31) ,
+
+ MAXLINE , MAXREPT , MAXEVRT , LU
+
+ IEVLNE , IERRFLG , LEU
+
+ IDATE , HEADALT , HEADVEL ,
+
+ HEADSR , HEADYAW , HEADPIT , BIAS
+
+ HEADROL , HEADWTG , BIAS
+
+ REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
+
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+
+ IMVDC , PRTEMP( 2 )
+
+ PRTMASS(2) , PRTINDX , PKZVEL ,
+
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+
+ XACCEL(3) , YACCEL(3) ,
+
+ REPTYPE , BIAS , ZACCEL(3) ,
+
+ PRTWGT , PRTINDX , PRTMASS , PRTINDX
+
+ PRTEMP
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193)

```

5-160

AD-A148 363

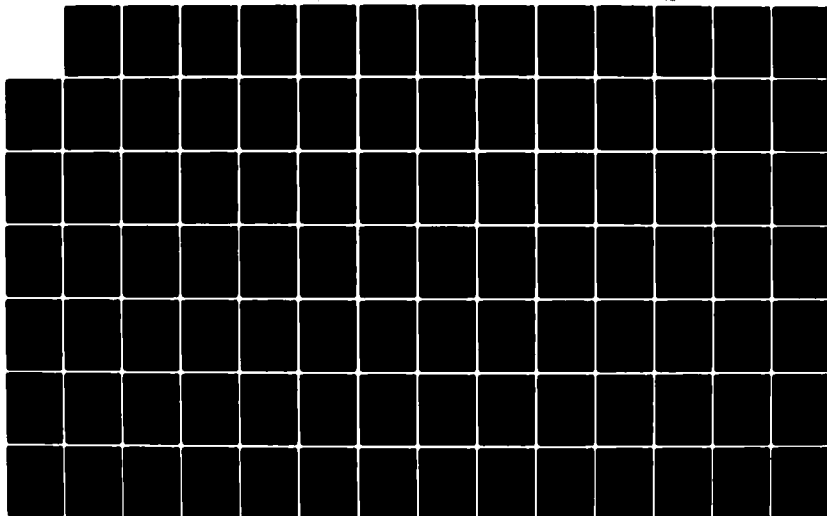
A GENERALIZED ESCAPE SYSTEM SIMULATION (GESS) COMPUTER
PROGRAM VOLUME 2 G.I.U) KETRON INC WARMINSTER PA
L A D'AULERIO ET AL. APR 84 NADC-84068-60

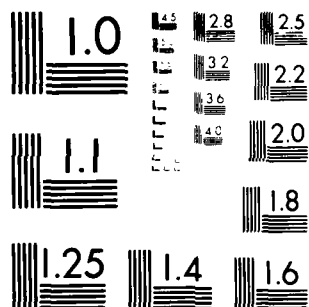
3/4

UNCLASSIFIED N62269-81-C-0206

F/G 9/2

NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```

115 SENSNAM(4,1)=10HTOTAL ACCE
    SENSNAM(4,2)=10HLERATION
    SENSNAM(4,3)=10H(SO/DA)
    SENSNAM(5,1)=10H-X- VELOC
    SENSNAM(5,2)=10HTY (SO/DA)
    SENSNAM(5,3)=10H
    SENSNAM(6,1)=10H-Y- VELOC
    SENSNAM(6,2)=10HTY (SO/DA)
    SENSNAM(6,3)=10H
    SENSNAM(7,1)=10H-Z- VELOC
    SENSNAM(7,2)=10HTY (SO/DA)
    SENSNAM(7,3)=10H
    SENSNAM(8,1)=10HTOTAL VELO
    SENSNAM(8,2)=10HCITY (SO/
    SENSNAM(8,3)=10HDA)
    SENSNAM(9,1)=10H-X- POSITI
    SENSNAM(9,2)=10HON (SO/DA)
    SENSNAM(9,3)=10H
    SENSNAM(10,1)=10H-Y- POSITI
    SENSNAM(10,2)=10HON (SO/DA)
    SENSNAM(10,3)=10H
    SENSNAM(11,1)=10H-Z- POSITI
    SENSNAM(11,2)=10HON (SO/DA)
    SENSNAM(11,3)=10H
    SENSNAM(12,1)=10HROLL RATE
    SENSNAM(12,2)=10H (SO/DA)
    SENSNAM(12,3)=10H
    SENSNAM(13,1)=10HPITCH RATE
    SENSNAM(13,2)=10H (SO/DA)
    SENSNAM(13,3)=10H
    SENSNAM(14,1)=10HYAW RATE
    SENSNAM(14,2)=10H(SO/DA)
    SENSNAM(14,3)=10H
    SENSNAM(15,1)=10HROLL ANGLE
    SENSNAM(15,2)=10H (SO/DA)
    SENSNAM(15,3)=10H
    SENSNAM(16,1)=10HPITCH ANGL
    SENSNAM(16,2)=10HE (SO/DA)
    SENSNAM(16,3)=10H
    SENSNAM(17,1)=10HYAW ANGLE
    SENSNAM(17,2)=10H (SO/DA)
    SENSNAM(17,3)=10H
    C
    IF(ISEATTR.EQ.0) GO TO 30
    C
    NCHANFR(1) = 35
    NSENSOR(1) = 34
    SENSNAM(18,1)=10H-X- ACCELE
    SENSNAM(18,2)=10HRATION
    SENSNAM(18,3)=10H(SO/SA)
    SENSNAM(19,1)=10H-Y- ACCELE
    SENSNAM(19,2)=10HRATION
    SENSNAM(19,3)=10H(SO/SA)
    SENSNAM(20,1)=10H-Z- ACCELE
    SENSNAM(20,2)=10HRATION
    SENSNAM(20,3)=10H(SO/SA)
    SENSNAM(21,1)=10HTOTAL ACCE

```

```

175 SENSNAM(21,2)=10HLERATION
    SENSNAM(21,3)=10H(SO/SA)
    SENSNAM(22,1)=10H-X- VELOC
    SENSNAM(22,2)=10HTY (SO/SA)
    SENSNAM(22,3)=10H
    SENSNAM(23,1)=10H-Y- VELOC
    SENSNAM(23,2)=10HTY (SO/SA)
    SENSNAM(23,3)=10H
    SENSNAM(24,1)=10H-Z- VELOC
    SENSNAM(24,2)=10HTY (SO/SA)
    SENSNAM(24,3)=10H
    SENSNAM(25,1)=10HTOTAL VELO
    SENSNAM(25,2)=10HCITY (SO/
    SENSNAM(25,3)=10HSA)
    SENSNAM(26,1)=10H-X- PCSITI
    SENSNAM(26,2)=10HON (SO/SA)
    SENSNAM(26,3)=10H
    SENSNAM(27,1)=10H-Y- POSITI
    SENSNAM(27,2)=10HON (SO/SA)
    SENSNAM(27,3)=10H
    SENSNAM(28,1)=10H-Z- POSITI
    SENSNAM(28,2)=10HON (SO/SA)
    SENSNAM(28,3)=10H
    SENSNAM(29,1)=10HROLL RATE
    SENSNAM(29,2)=10H (SO/SA)
    SENSNAM(29,3)=10H
    SENSNAM(30,1)=10HPITCH RATE
    SENSNAM(30,2)=10H (SO/SA)
    SENSNAM(30,3)=10H
    SENSNAM(31,1)=10HYAW RATE
    SENSNAM(31,2)=10H(SO/SA)
    SENSNAM(31,3)=10H
    SENSNAM(32,1)=10HROLL ANGLE
    SENSNAM(32,2)=10H (SO/SA)
    SENSNAM(32,3)=10H
    SENSNAM(33,1)=10HPITCH ANGL
    SENSNAM(33,2)=10HE (SO/SA)
    SENSNAM(33,3)=10H
    SENSNAM(34,1)=10HYAW ANGLE
    SENSNAM(34,2)=10H (SO/SA)
    SENSNAM(34,3)=10H
C
30 CONTINUE
C
NHEADER(1)= 1
NCHAN = NCHANFR(1)
BAUD=10HTIMEFORMAT
WORDLEN=0
TIMINT=TIME
TDelta=DIPHAS1
C
WRITE(42)TEXT1,TEXT2,TEXT3,BAUD,WORDLEN,NCHANFR(1),NSENSOR(1),
+ IDUMMY
WRITE(42)((SENSNAM(I,J),J=1,3),I=1,20)
WRITE(42)((SENSNAM(I,J),J=1,3),I=21,40)
WRITE(42)TIMINT,TDelta,TPOINT
C

```

```

230      C          50 CONTINUE
      PLTIME(1) = TIME
      IF (EVENTS(28).NE.0)GOTO 100
      RECORD(1)=TIME
      RECORD(2)=XACC(1)
      RECORD(3)=YACC(1)
      RECORD(4)=ZACC(1)
      RECORD(5)=SORT(XACC(1)*XACC(1) + YACC(1)*YACC(1) +
      + ZACC(1)*ZACC(1))
      RECORD(6)=TRAJSD(14)
      RECORD(7)=TRAJSD(15)
      RECORD(8)=TRAJSD(16)
      RECORD(9)=SORT(TRAJSD(14)*TRAJSD(14) + TRAJSD(15)*TRAJSD(15) +
      + TRAJSD(16)*TRAJSD(16))
      RECORD(10)=TRAJSD(2)
      RECORD(11)=TRAJSD(3)
      RECORD(12)=TRAJSD(4)
      RECORD(13)=PVL(1)
      RECORD(14)=QVL(1)
      RECORD(15)=RVL(1)
      RECORD(16)=FROLL(1)
      RECORD(17)=FPITCH(1)
      RECORD(18)=FYAW(1)
      C
      IF (ISEATTR.EQ.0) GOTO 150
      C
255      C          100 CONTINUE
      RECORD(19)=RECORD(2)
      RECORD(20)=RECORD(3)
      RECORD(21)=RECORD(4)
      RECORD(22)=RECORD(5)
      RECORD(23)=RECORD(6)
      RECORD(24)=RECORD(7)
      RECORD(25)=RECORD(8)
      RECORD(26)=RECORD(9)
      RECORD(27)=RECORD(10)
      RECORD(28)=RECORD(11)
      RECORD(29)=RECORD(12)
      RECORD(30)=RECORD(13)
      RECORD(31)=RECORD(14)
      RECORD(32)=RECORD(15)
      RECORD(33)=RECORD(16)
      RECORD(34)=RECORD(17)
      RECORD(35)=RECORD(18)
      GO TO 150
      C
275      C          100 CONTINUE
      C
      RECORD(1)=TIME
      RECORD(2)=XACC(2)
      RECORD(3)=YACC(2)
      RECORD(4)=ZACC(2)
      RECORD(5)=SORT(XACC(2)*XACC(2) + YACC(2)*YACC(2) +
      + ZACC(2)*ZACC(2))
      RECORD(6)=TRAJSD(14)
      RECORD(7)=TRAJSD(15)
      RECORD(8)=TRAJSD(16)
285

```

```

290      RECORD(9)=SORT(TRAJDA(14)*TRAJDA(14) + TRAJDA(15)*TRAJDA(15) +
      + TRAJDA(16)*TRAJDA(16))
      RECORD(10)=TRAJDA(2)
      RECORD(11)=TRAJDA(3)
      RECORD(12)=TRAJDA(4)
      RECORD(13)=PVL(2)
      RECORD(14)=QVL(2)
      RECORD(15)=RVL(2)
      RECORD(16)=FROLL(2)
      RECORD(17)=FPITCH(2)
      RECORD(18)=FYAW(2)

295      C IF(1SEATR.EQ.0) GO TO 150
      C

300      RECORD(19)=XACC(3)
      RECORD(20)=YACC(3)
      RECORD(21)=ZACC(3)
      RECORD(22)=SORT(XACC(3)*XACC(3) + YACC(3)*YACC(3) +
      + ZACC(3)*ZACC(3))
      RECORD(23)=TRAJSA(14)
      RECORD(24)=TRAJSA(15)
      RECORD(25)=TRAJSA(16)
      RECORD(26)=SORT(TRAJSA(14)*TRAJSA(14) + TRAJSA(15)*TRAJSA(15) +
      + TRAJSA(16)*TRAJSA(16))
      RECORD(27)=TRAJSA(2)
      RECORD(28)=TRAJSA(3)
      RECORD(29)=TRAJSA(4)
      RECORD(30)=PVL(3)
      RECORD(31)=QVL(3)
      RECORD(32)=RVL(3)
      RECORD(33)=FROLL(3)
      RECORD(34)=FPITCH(3)
      RECORD(35)=FYAW(3)

310      C

315      C 150 CONTINUE
      WRITE(42) (RECORD(I),I=1,NCHAN)
      9000 CONTINUE
      RETURN
      END

320

```

```

1  SUBROUTINE PLOTWAC
C .....
C DESCRIPTION - LEVEL 2
C FUNCTION - CREATE PLOTTING FILE OF INFORMATION WRT AIRCRAFT
C METHOD - WRITES TRAJECTORY DATA TO PLOTTING FILE EVERY .05 SEC
C IN A FORMAT COMPATIBLE WITH DRAS PLOTTING PROGRAM
C COMMUNICATIONS -
C CALLED BY: GESS
C NON-COMMON VARIABLES DEFINED -
C NONE
C POTENTIAL ERROR CONDITIONS -
C NONE
C .....
C CONSTANTS COMMON BLOCK
C .....
C COMMON /CONSTANT / GRAVITY , RADDEG , DEGRAD , PI
C .....
C SECTION 1 COMMON BLOCK
C .....
C COMMON /ICONTROL / TSTART , TSTOP , ESTOP , IRESTRT, IUNITS ,
+ ISEATTR, ISOSEP , IPLOT , IDRIFLG,
+ IPHASE1, IPHASE2, IPHASE3
C .....
C INTEGER
C .....
C SECTION 3 COMMON BLOCK
C .....
C COMMON /IDELTAT / DTPHASE1, DTPHASE2, DTPHASE3
C .....
C SECTION 2 COMMON BLOCK
C .....
C COMMON /IREPORT / IREPTS(31) , PRTRFQ,PI1,PI2,PI3
C .....
C INTEGER
C .....
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31)
+ MAXREPT , MAXEVNT
+ IEVLN , LU
+ IDATE , IERRFLG , HEADVEL
+ HEADSR , HEADYAW , HEADPIT
+ HEADROL , HEADWGT , BIAS
+ REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2)
+ IHEADER(24) , IEVENTS(38) , TIMES(38)
+ IMVDC , PRTEMP( 2)
+ PRTMAS(2) , PRINDX , PKZVEL
+ ZVECT(3) , XYZ(3) , SAVTIME
+ XACCEL(3) , YACCEL(3) , ZACCEL(3)
+ REPTYPE , BIAS , PRTLNCT
+ PRTWGT , PRTEMP , PRINDX
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
C COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193)

```

PAGE 165

83/11/07. 09.41.53

FTN 4.6+428

SUBROUTINE PLOTWAC 74/74 OPT=1

```

+      TRAJSA(193)      . TRAJDA(193)      . TRAJCH(97,3) .
+      TRAJAC(193)      . TVGEQS(225)      . QUATSO(65)  .
+      QUATSA(65)       . QUATOA(65)       . QUATAC(65)  .
+      INTSTP           . IPCPASS          . IRKPASS      .
+      IPOINTS          . IYX              . IYPRX        .
+      IKX              . IKSUMX           . IKPASSX      .
+      IYIX             . IYIX            . IYI2X        .
+      IVI3X            . IVPRI3X         . IVPRI1X      .
+      IVPRI2X          . IYIX            . IYI1X        .
+      ICYIX            . ICYIX           . IREIN        .
C.....
C TOF PLOTTING FILE COMMON BLOCK
C.....
COMMON /TITLES / SENSAM(40,6) . TEXT1(6)      . TEXT2(6)      .
+      TEXT3(3)         . BAUD            . WORDLEN       .
+      NCHANFR(2)       . NSENSOR(2)      . IDUMMY(40)    .
+      RECORD(35)       . TIMINT          . TDELTA        .
+      TPOINT           . PLTIME(2)       . NHEADER(2)    .
+      INTEGER          . TEXT1          . TEXT2        .
+      BAUD             . WORDLEN        . TEXT3         .
C.....
C PLOT FILE VARIABLES COMMON BLOCK
C.....
COMMON /PLOT / XACC(3) . YACC(3) . ZACC(3) . ACCR(3) .
+      PVL(3) . QVL(3) . RVL(3) . VELR(3) .
+      FROLL(3) . FPITCH(3) . FVAC(3) . RPOS(3) .
+      RXACC(3) . RYACC(3) . RZACC(3) . RLACR(3) .
+      RXVEL(3) . RYVEL(3) . RZVEL(3) . RLVR(3) .
+      RPOS(3) . RQPOS(3) . RZPOS(3) . RLPSR(3) .
+      RPVEL(2) . RQVEL(2) . RZVEL(2) .
+      RPOS(2) . RQPOS(2) . RRPOS(2) .
C
      IF(IPLOT .LT. 2) GO TO 9000
      IF(INTSTP .EQ. 0) GO TO 9000
      IF(TIMES .LE. 0.0) GO TO 20
      IF((TIME-PLTIME(2)) .LT. 0.05) GO TO 9000
20 CONTINUE
C
      IF(NHEADER(2) .EQ. 1) GOTO 50
      DO 25 I=1,6
      TEXT1(I)=I*HEADER(1)
      TEXT2(I)=I*HEADER(1+8)
      IF(I .GT. 3) GOTO 25
      TEXT3(I)=I*HEADER(1+16)
25 CONTINUE
C
      NCHANFR(2) = 12
      NSENSOR(2) = 11
      SENSAM(1,4)=IOH-X- ACCEL
      SENSAM(1,5)=IOH(SO/OA) WR
      SENSAM(1,6)=IOHT AC
      SENSAM(2,4)=IOH-Y- ACCEL
      SENSAM(2,5)=IOH(SO/OA) WR
      SENSAM(2,6)=IOHT AC
      SENSAM(3,4)=IOH-Z- ACCEL
      SENSAM(3,5)=IOH(SO/OA) WR
      SENSAM(3,6)=IOHT AC

```



```

115 SENSNAM(4,4)=10HTOTAL ACCE
    SENSNAM(4,5)=10HL (SD/OA)
    SENSNAM(4,6)=10HRT AC
    SENSNAM(5,4)=10H-X- VELOC
    SENSNAM(5,5)=10HTY (SD/OA)
    SENSNAM(5,6)=10H) WRT AC
    SENSNAM(6,4)=10H-Y- VELOC
    SENSNAM(6,5)=10HTY (SD/OA)
    SENSNAM(6,6)=10H) WRT AC
    SENSNAM(7,4)=10H-Z- VELOC
    SENSNAM(7,5)=10HTY (SD/OA)
    SENSNAM(7,6)=10H) WRT AC
    SENSNAM(8,4)=10HTOTAL VELO
    SENSNAM(8,5)=10HCITY (SD/
    SENSNAM(8,6)=10HDA) WRT AC
    SENSNAM(9,4)=10H-X- POSITI
    SENSNAM(9,5)=10HDN (SD/OA)
    SENSNAM(9,6)=10H) WRT AC
    SENSNAM(10,4)=10H-Y- POSITI
    SENSNAM(10,5)=10HDN (SD/OA)
    SENSNAM(10,6)=10H) WRT AC
    SENSNAM(11,4)=10H-Z- POSITI
    SENSNAM(11,5)=10HDN (SD/OA)
    SENSNAM(11,6)=10H) WRT AC
    C
140 IF(1SEATTR.EQ.O) GO TO 30
    C
    NCHANFR(2) = 29
    NSENSOR(2) = 28
    SENSNAM(12,4)=10H-X- ACCEL
    SENSNAM(12,5)=10H(SD/SA) WR
    SENSNAM(12,6)=10HT AC
    SENSNAM(13,4)=10H-Y- ACCEL
    SENSNAM(13,5)=10H(SD/SA) WR
    SENSNAM(13,6)=10HT AC
    SENSNAM(14,4)=10H-Z- ACCEL
    SENSNAM(14,5)=10H(SD/SA) WR
    SENSNAM(14,6)=10HT AC
    SENSNAM(15,4)=10HTOTAL ACCE
    SENSNAM(15,5)=10HL (SD/SA)
    SENSNAM(15,6)=10HRT AC
    SENSNAM(16,4)=10H-X- VELOC
    SENSNAM(16,5)=10HTY (SD/SA)
    SENSNAM(16,6)=10H) WRT AC
    SENSNAM(17,4)=10H-Y- VELOC
    SENSNAM(17,5)=10HTY (SD/SA)
    SENSNAM(17,6)=10H) WRT AC
    SENSNAM(18,4)=10H-Z- VELOC
    SENSNAM(18,5)=10HTY (SD/SA)
    SENSNAM(18,6)=10H) WRT AC
    SENSNAM(19,4)=10HTOTAL VELO
    SENSNAM(19,5)=10HCITY (SD/
    SENSNAM(19,6)=10HSA) WRT AC
    SENSNAM(20,4)=10H-X- POSITI
    SENSNAM(20,5)=10HDN (SD/SA)
    SENSNAM(20,6)=10H) WRT AC
    SENSNAM(21,4)=10H-Y- POSITI

```

```

175 SENSNAM(21,5)=10HON (SO/SA
    SENSNAM(21,6)=10H) WRT AC
    SENSNAM(22,4)=10H-2- POSITI
    SENSNAM(22,5)=10HON (SO/SA
    SENSNAM(22,6)=10H) WRT AC
    SENSNAM(23,4)=10HROLL RATE
    SENSNAM(23,5)=10H (SO/SA)
    SENSNAM(23,6)=10HWRT AC
    SENSNAM(24,4)=10HPITCH RATE
    SENSNAM(24,5)=10H (SO/SA)
    SENSNAM(24,6)=10HWRT AC
    SENSNAM(25,4)=10HYAW RATE
    SENSNAM(25,5)=10H(SO/SA)
    SENSNAM(25,6)=10HWRT AC
    SENSNAM(26,4)=10HROLL ANGLE
    SENSNAM(26,5)=10H (SO/SA)
    SENSNAM(26,6)=10HWRT AC
    SENSNAM(27,4)=10HPITCH ANGL
    SENSNAM(27,5)=10HE (SO/SA)
    SENSNAM(27,6)=10HWRT AC
    SENSNAM(28,4)=10HYAW ANGLE
    SENSNAM(28,5)=10H (SO/SA)
    SENSNAM(28,6)=10HWRT AC

180
185
190
195 C 30 CONTINUE
    C
    NCHAN = NCHANFR(2)
    NHEADER(2)= 1
    BAUD=10HTIMEFORMAT
    WORDLEN=0
    TIMINT=TIME
    TDELTA=DTPHAS1
    C
    WRITE(41)TEXT1,TEXT2,TEXT3,BAUD,WORDLEN,NCHANFR(2),NSENSOR(2),
    + IDUMMY
    WRITE(41)((SENSNAM(I,J),J=4,8),I=1,20)
    WRITE(41)((SENSNAM(I,J),J=4,6),I=21,40)
    WRITE(41)TIMINT,TDELTA,TPPOINT
    C
    C 50 CONTINUE
    C
    PLTIME(2) = TIME
    IF(IEVENTS(28).NE.0)GOTO 100
    RECORD(1)=TIME
    RECORD(2)=RXACC(1)
    RECORD(3)=RYACC(1)
    RECORD(4)=RZACC(1)
    RECORD(5)=SORT(RXACC(1)*RXACC(1) + RYACC(1)*RYACC(1) +
    + RZACC(1)*RZACC(1))
    RECORD(6)=RXVEL(1)
    RECORD(7)=RYVEL(1)
    RECORD(8)=RZVEL(1)
    RECORD(9)=SORT(RXVEL(1)*RXVEL(1) + RYVEL(1)*RYVEL(1) +
    + RZVEL(1)*RZVEL(1))
    RECORD(10)=RXPOS(1)
    RECORD(11)=RYPOS(1)
    RECORD(12)=RZPOS(1)

```

```

230 IF(1SEATTR.EQ.O) GOTO 150
    RECORD(13)=RECORD(2)
    RECORD(14)=RECORD(3)
    RECORD(15)=RECORD(4)
    RECORD(16)=RECORD(5)
    RECORD(17)=RECORD(6)
    RECORD(18)=RECORD(7)
    RECORD(19)=RECORD(8)
    RECORD(20)=RECORD(9)
    RECORD(21)=RECORD(10)
    RECORD(22)=RECORD(11)
    RECORD(23)=RECORD(12)
    RECORD(24)=RPVEL(1)
    RECORD(25)=RPVEL(1)
    RECORD(26)=RPPOS(1)
    RECORD(27)=RPPOS(1)
    RECORD(28)=RPPOS(1)
    RECORD(29)=RPPOS(1)
    GO TO 150
C
250 C 100 CONTINUE
    RECORD(1)=TIME
    RECORD(2)=RXACC(2)
    RECORD(3)=RYACC(2)
    RECORD(4)=RZACC(2)
    RECORD(5)=SORT(RXACC(2)*RXACC(2)+RYACC(2)*RYACC(2)+
    + RZACC(2)*RZACC(2))
    RECORD(6)=RXVEL(2)
    RECORD(7)=RYVEL(2)
    RECORD(8)=RZVEL(2)
    RECORD(9)=SORT(RXVEL(2)*RXVEL(2)+RYVEL(2)*RYVEL(2)+
    + RZVEL(2)*RZVEL(2))
    RECORD(10)=RXPOS(2)
    RECORD(11)=RYPOS(2)
    RECORD(12)=RZPOS(2)
C
265 IF(1SEATTR.EQ.O) GO TO 150
C
    RECORD(13)=RXACC(3)
    RECORD(14)=RYACC(3)
    RECORD(15)=RZACC(3)
    RECORD(16)=SORT(RXACC(3)*RXACC(3)+RYACC(3)*RYACC(3)+
    + RZACC(3)*RZACC(3))
    RECORD(17)=RXVEL(3)
    RECORD(18)=RYVEL(3)
    RECORD(19)=RZVEL(3)
    RECORD(20)=SORT(RXVEL(3)*RXVEL(3)+RYVEL(3)*RYVEL(3)+
    + RZVEL(3)*RZVEL(3))
    RECORD(21)=RXPOS(3)
    RECORD(22)=RYPOS(3)
    RECORD(23)=RZPOS(3)
    RECORD(24)=RPVEL(2)
    RECORD(25)=RPVEL(2)
    RECORD(26)=RPVEL(2)
    RECORD(27)=RPPOS(2)
    RECORD(28)=RPPOS(2)
285

```

PAGE 169

83/11/07. 09.41.53

FTN 4.6+428

SUBROUTINE PLOTWAC 74/74 OPT=1

RECORD(29)=RRPOS(2)

C

150 CONTINUE

C

WRITE(41) (RECORD(I),I=1,NCHAN)

9000 CONTINUE

RETURN

END

290

```

1 SUBROUTINE QUAT
2 .....
3 C DESCRIPTION - LEVEL 2
4 C FUNCTION - COMPUTES DERIVATIVES OF QUATERNIONS USED TO UPDATE
5 C TRANSFORMATION MATRICES
6 C METHOD - PRIOR TO RAIL SEPARATION, DERIVATIVES OF QUATERNIONS
7 C USED TO UPDATE BOTH DCMSE(EFGS TO SCS) AND DCMAE
8 C (EFGS TO ACS), ARE COMPUTED. FOLLOWING RAIL
9 C SEPARATION, THE USER HAS THE OPTION OF CHOOSING TO
10 C UPDATE ONLY DCMSE.
11 C COMMUNICATIONS -
12 C CALLED BY: GESS
13 C CALLS:
14 C NONE
15 C NON-COMMON VARIABLES DEFINED: NONE
16 C POTENTIAL ERROR CONDITIONS: NONE
17 C .....
18 C SECTION 1 COMMON BLOCK
19 C .....
20 C COMMON /ICONTROL / TSTART, TSTOP, ESTOP, IRESTR, IUNITS,
21 C ISEATTR, ISOSEP, IPLOT, IDIRFLO,
22 C IPHASE1, IPHASE2, IPHASE3
23 C ESTOP
24 C INTEGER
25 C .....
26 C MISCELLANEOUS DATA COMMON BLOCK
27 C .....
28 C COMMON /MISC / IPAGECT(31), LINECT(31), IPRTCNT(31),
29 C MAXLINE, MAXREPT, MAXEVNT,
30 C IEVLNE, TERFLG, LU,
31 C IDATE, HEADALT, HEADVEL,
32 C HEADSR, HEADYAW, HEADPIT,
33 C HEADROL, HEADWGT, BIAS,
34 C REPTYPE(5,31), PRTLNCT(2), PRTWGT(2),
35 C IHEADER(24), IEVENTS(38), PRTWGT(2),
36 C IMVOC, PRTWGT(2), PRTLNCT(2),
37 C PRTWGT(2), XYZ(3), PKZVEL,
38 C XACCEL(3), YACCEL(3), SAVTIME,
39 C REPTYPE, ZACCEL(3),
40 C PRTWGT, BIAS, PRTLNCT,
41 C PRTWGT, PRTWGT, PRTWGT,
42 C PRTWGT, PRTWGT, PRTWGT,
43 C PRTWGT, PRTWGT, PRTWGT,
44 C PRTWGT, PRTWGT, PRTWGT,
45 C PRTWGT, PRTWGT, PRTWGT,
46 C PRTWGT, PRTWGT, PRTWGT,
47 C PRTWGT, PRTWGT, PRTWGT,
48 C PRTWGT, PRTWGT, PRTWGT,
49 C PRTWGT, PRTWGT, PRTWGT,
50 C PRTWGT, PRTWGT, PRTWGT,
51 C PRTWGT, PRTWGT, PRTWGT,
52 C PRTWGT, PRTWGT, PRTWGT,
53 C PRTWGT, PRTWGT, PRTWGT,
54 C PRTWGT, PRTWGT, PRTWGT,
55 C PRTWGT, PRTWGT, PRTWGT,
56 C PRTWGT, PRTWGT, PRTWGT,
57 C PRTWGT, PRTWGT, PRTWGT,
58 C PRTWGT, PRTWGT, PRTWGT,
59 C PRTWGT, PRTWGT, PRTWGT,
60 C PRTWGT, PRTWGT, PRTWGT,
61 C PRTWGT, PRTWGT, PRTWGT,
62 C PRTWGT, PRTWGT, PRTWGT,
63 C PRTWGT, PRTWGT, PRTWGT,
64 C PRTWGT, PRTWGT, PRTWGT,
65 C PRTWGT, PRTWGT, PRTWGT,
66 C PRTWGT, PRTWGT, PRTWGT,
67 C PRTWGT, PRTWGT, PRTWGT,
68 C PRTWGT, PRTWGT, PRTWGT,
69 C PRTWGT, PRTWGT, PRTWGT,
70 C PRTWGT, PRTWGT, PRTWGT,
71 C PRTWGT, PRTWGT, PRTWGT,
72 C PRTWGT, PRTWGT, PRTWGT,
73 C PRTWGT, PRTWGT, PRTWGT,
74 C PRTWGT, PRTWGT, PRTWGT,
75 C PRTWGT, PRTWGT, PRTWGT,
76 C PRTWGT, PRTWGT, PRTWGT,
77 C PRTWGT, PRTWGT, PRTWGT,
78 C PRTWGT, PRTWGT, PRTWGT,
79 C PRTWGT, PRTWGT, PRTWGT,
80 C PRTWGT, PRTWGT, PRTWGT,
81 C PRTWGT, PRTWGT, PRTWGT,
82 C PRTWGT, PRTWGT, PRTWGT,
83 C PRTWGT, PRTWGT, PRTWGT,
84 C PRTWGT, PRTWGT, PRTWGT,
85 C PRTWGT, PRTWGT, PRTWGT,
86 C PRTWGT, PRTWGT, PRTWGT,
87 C PRTWGT, PRTWGT, PRTWGT,
88 C PRTWGT, PRTWGT, PRTWGT,
89 C PRTWGT, PRTWGT, PRTWGT,
90 C PRTWGT, PRTWGT, PRTWGT,
91 C PRTWGT, PRTWGT, PRTWGT,
92 C PRTWGT, PRTWGT, PRTWGT,
93 C PRTWGT, PRTWGT, PRTWGT,
94 C PRTWGT, PRTWGT, PRTWGT,
95 C PRTWGT, PRTWGT, PRTWGT,
96 C PRTWGT, PRTWGT, PRTWGT,
97 C PRTWGT, PRTWGT, PRTWGT,
98 C PRTWGT, PRTWGT, PRTWGT,
99 C PRTWGT, PRTWGT, PRTWGT,
100 C PRTWGT, PRTWGT, PRTWGT,

```

```

C
C
C
60      IF (QUATSO(1) .EQ. 0.) GO TO 200
        QUATSO(6) = -.5*( QUATSO(3)*TRAJSD(11) + QUATSO(4)*TRAJSD(12)
        + QUATSO(5)*TRAJSD(13))
        QUATSO(7) = 0.5*( QUATSO(2)*TRAJSD(11) + QUATSO(4)*TRAJSD(13)
        + QUATSO(5)*TRAJSD(12))
        QUATSO(8) = 0.5*( QUATSO(2)*TRAJSD(12) - QUATSO(3)*TRAJSD(13)
        - QUATSO(5)*TRAJSD(11))
        QUATSO(9) = 0.5*( QUATSO(2)*TRAJSD(13) + QUATSO(3)*TRAJSD(12)
        + QUATSO(4)*TRAJSD(11))
200      CONTINUE
        IF (QUATAC(1) .EQ. 0.) GO TO 300
        QUATAC(6) = -.5*( QUATAC(3)*TRAJAC(11) + QUATAC(4)*TRAJAC(12)
        + QUATAC(5)*TRAJAC(13))
        QUATAC(7) = 0.5*( QUATAC(2)*TRAJAC(11) + QUATAC(4)*TRAJAC(13)
        + QUATAC(5)*TRAJAC(12))
        QUATAC(8) = 0.5*( QUATAC(2)*TRAJAC(12) - QUATAC(3)*TRAJAC(13)
        - QUATAC(5)*TRAJAC(11))
        QUATAC(9) = 0.5*( QUATAC(2)*TRAJAC(13) + QUATAC(3)*TRAJAC(12)
        + QUATAC(4)*TRAJAC(11))
300      CONTINUE
        IF (QUATOA(1) .EQ. 0.) GO TO 400
        QUATOA(6) = -.5*( QUATOA(3)*TRAJOA(11) + QUATOA(4)*TRAJOA(12)
        + QUATOA(5)*TRAJOA(13))
        QUATOA(7) = 0.5*( QUATOA(2)*TRAJOA(11) + QUATOA(4)*TRAJOA(13)
        + QUATOA(5)*TRAJOA(12))
        QUATOA(8) = 0.5*(QUATOA(2)*TRAJOA(12) - QUATOA(3)*TRAJOA(13)
        - QUATOA(5)*TRAJOA(11))
        QUATOA(9) = 0.5*( QUATOA(2)*TRAJOA(13) + QUATOA(3)*TRAJOA(12)
        + QUATOA(4)*TRAJOA(11))
400      CONTINUE
        IF (QUATSA(1) .EQ. 0.) GO TO 9000
        QUATSA(6) = -.5*( QUATSA(3)*TRAJSA(11) + QUATSA(4)*TRAJSA(12)
        + QUATSA(5)*TRAJSA(13))
        QUATSA(7) = 0.5*( QUATSA(2)*TRAJSA(11) + QUATSA(4)*TRAJSA(13)
        + QUATSA(5)*TRAJSA(12))
        QUATSA(8) = 0.5*(QUATSA(2)*TRAJSA(12) - QUATSA(3)*TRAJSA(13)
        - QUATSA(5)*TRAJSA(11))
        QUATSA(9) = 0.5*( QUATSA(2)*TRAJSA(13) + QUATSA(3)*TRAJSA(12)
        + QUATSA(4)*TRAJSA(11))
9000      CONTINUE
        RETURN
        END

```

```

1  SUBROUTINE RAILFM
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - COMPUTES FORCES AND MOMENTS ON SEAT/OCCUPANT DUE TO
C RAIL INTERACTION
C METHOD - FOR CALCULATING FORCES, ASSUMES SLIPPERS (SLIDER B
C MOVE WITH THE SEAT. HOWEVER, BEFORE COMPUTING FORCES
C AND MOMENTS DUE TO EACH SLIPPER (SLIDER BLOCK), IT
C DECIDES IF THE SLIPPER (SLIDER BLOCK) IS ACTIVE AS
C FOLLOWS:
C IF ON THE RAILS, THE POSITION OF THE SEAT BOTTOM IS
C COMPARED WITH THE INITIAL POSITION OF THE SLIDER BLOCK.
C I THE SEAT HAS PASSED IT, THE SLIDER BLOCK IS INACTIVE.
C IF ON THE SEAT, THE CURRENT POSITION OF THE SLIPPER
C IS COMPARED WITH THE RAIL LENGTH. IF PAST THE TOP OF
C THE RAILS, THE SLIPPER IS INACTIVE.
C FOR 'CONTINUOUS' SLIPPERS, THE PROGRAM ASSUMES THAT
C SLIPPERS (SLIDER BLOCKS) ARE ACTIVE AT THE TOP OF
C THE SEAT (OR THE TOP OF THE RAILS ONCE THE SEAT BEGINS
C TO LEAVE THE RAILS) AND AT THE SEAT BOTTOM, UNTIL
C THE SEAT IS CLEAR OF THE RAILS.
C COMMUNICATIONS -
C CALLED BY: SEATOCC
C CALLS:
C ROTATE, RKTFM
C NON-COMMON VARIABLES DEFINED -
C NONE
C POTENTIAL ERROR CONDITIONS -
C NONE
C .....
C CONSTANTS COMMON BLOCK
C .....
C COMMON /CONSTNT / GRAVITY , RADDEG , DEGRAD , PI
C SEAT/OCCUPANT FORCES COMMON BLOCK
C .....
C COMMON /FORCESO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
C FXTUBSO , FYTUBSO , FZTUBSO ,
C FXSLSO(6) , FYSLSO(6) , FZSLSO(6) ,
C FXRKSO(6) , FYRKSO(6) , FZRKSO(6) ,
C FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
C FXAESO , FYAESO , FZAESO ,
C FXDRTSO , FYDRTSO , FZDRTSO
C .....
C SECTION 7 COMMON BLOCK
C .....
C COMMON /IRAIL / RAILNTH , RAILANG , ISTRL , NSLBKS ,
C KXSB , KYSB , MUSB , YKTOR ,
C XPOSRR , YPOSRR , ZPOSRR ,
C XPOSRL , YPOSRL , ZPOSRL ,
C XPOSSB(6) , YPOSSB(6) , ZPOSSB(6) ,
C REAL
C KXSB , KYSB , MUSB
C .....
C SECTION 5 COMMON BLOCK
C .....

```



```

115 C .....+XRSDAC ,YRSDAC ,ZRSOAC ,XSCPAP(2),YSCPAP(2),ZSCPAP(2)
C .....C RAIL VARIABLES COMMON BLOCK
C .....COMMON /RAILVRB / FXR , FYR , FZR , XDISP , YDISP
120 C .....C INTEGRATION ROUTINE COMMON BLOCK
C .....COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193)
+ .....TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3)
+ .....TRAJAC(193) , TVCEOS(225) , QUATSO(65)
125 C .....+ .....QUATOA(65) , QUATAC(65)
+ .....INTSTP , IPCPASS , IRKPASS
+ .....IPOINTS , IYX , IYPRX
+ .....IKX , IKSUMX , IKPASSX
130 C .....+ .....IYIX , IYIIX , IYI2X
+ .....IYI3X , IYPRIX , IYPRIIX
+ .....IYPR12X , IYIYX , IYI1X
+ .....ICYIYX , ICYIIX , IREIN
135 C .....C TORQUE SEAT/OCCUPANT COMMON BLOCK
C .....COMMON /TORQSO / TLCASO(2) , TMCASO(2) , TNCASO(2)
+ .....TLTUBSO , TMTUBSO , TNTUBSO
+ .....TLSLSO(6) , TMSLSO(6) , TNSLSO(6)
140 C .....+ .....TLRKS(6) , TMRKS(6) , TNRKS(6)
+ .....TLCHSO(3) , TMCHSO(3) , TNCHSO(3)
+ .....TLAESO , TMAESO , TNAESO
+ .....TLDRTSO , TMDRTSO , TNDRTSO
145 C .....C IF ALREADY CLEAR OF THE RAILS, JUST RETURN
C .....C IF (IEVENTS(5) .NE. 0) GO TO 9000
150 C .....C TEST FOR SEAT FIRST MOTION
C .....IF (IEVENTS(38) .NE. 0) GO TO 150
IF (IEVENTS(1) .EQ. 0) GO TO 9000
IF ((ZRRSROT - ZPOSROT) .LT. 0.000001) GO TO 125
155 C .....IEVENTS(38) = 1
TIMES(38) = TIME
GO TO 150
C .....C ROTATE GRAVITY TO S.C.S.
C .....125 CONTINUE
ZGRAV = GRAVITY * DCMS(3.3)
160 C .....C IF CATAPULT IGNITION HAS OCCURRED AND THE SEAT IS UPRIGHT
C .....C RETURN
165 C .....IF (ZGRAV .LT. 0.) GO TO 9000
C .....C TEST FOR RAIL CLEARANCE
170 C .....150 CONTINUE

```

SUBROUTINE RAILFM 74/74 OPT=1 FTN 4.6+428 83/11/07. 09.41.53 PAGE 175

```

175 IF (ABS(ZRRSBOY) .LT. RAILNTH .OR. INTSTP .EQ. 0) GO TO 200
    IEVENTS(5) = 1
    TIMES(5) = TIME
    C .....
    C ZERO OUT ALL FORCES AND MOMENTS DUE TO RAIL INTERACTION
    C .....
    DO 100 I=1,6
        FXLSO(I) = FYLSO(I) = FZLSO(I) = 0.0
        TSLSO(I) = TMSLSO(I) = TNSLSO(I) = 0.0
    100 CONTINUE
    GO TO 9000
    200 CONTINUE
    C .....
    C COMPUTE ANGLE BETWEEN SEAT AND RAILS
    C .....
    C .....
    PSISR = PSISA - RAILANG
    ANGSR = -ASIN(DCMSR(1,3))
    IF (PI .LE. ANGSR .AND. ANGSR .LE. 2.*PI) ANGSR = ANGSR-2.*PI
    C .....
    C .....
    C COMPUTE FORCES AND MOMENTS AT EACH SLIPPER (SLIDER BLOCK)
    C .....
    N=NSLBKS
    IF (NSLBKS .EQ. 0) N=4
    DO 300 I=1,N
        IF (NSLBKS .NE. 0 .AND. 1STRL .NE. 0) GO TO 210
        XRRSB=XSSOSB(I)
        YRRSB=YSSOSB(I)
        ZRRSB=ZSSOSB(I)
        CALL ROTATE (XRRSB,XRRSB,XSSOMRE,DCMSR,1)
    300 CONTINUE
    C .....
    C TEST HERE IS FOR ON SEAT - TEST IF SLIPPER IS PAST TOP OF RAILS
    C .....
    IF (NSLBKS .NE. 0 .OR. ABS(ZRRSB) .LE. RAILNTH) GO TO 220
    C .....
    C FOR CONTINUOUS SLIPPERS, IF SLIPPER HAS MOVED OFF RAILS, PUT IT AT
    C TOP OF RAILS AND UPDATE MOMENT ARM
    C .....
    ZRRSB=RAILNTH
    CALL ROTATE (XRRSB,XYZ,ZVECT,DCMSR,0)
    XSSOSB(1)=XYZ(1)+XSSOMRE
    YSSOSB(1)=XYZ(2)+YSSOMRE
    ZSSOSB(1)=XYZ(3)+ZSSOMRE
    220 CONTINUE
    IF (ABS(ZRRSB) .GT. RAILNTH) GO TO 300
    C .....
    C GET POSITION OF SLIPPER IN RCS
    C .....
    XDISP=XRRSB-XRRSBO(1)
    YDISP=YRRSB-YRRSBO(1)
    XYZ(1)=XSSOSB(1)
    XYZ(2)=YSSOSB(1)
    XYZ(3)=ZSSOSB(1)
    GO TO 240
    210 CONTINUE
    C .....
    C TEST IF SEAT BOTTOM IS PAST INITIAL POSITION OF SLIDER BLOCK

```

```

C *****
230 IF (ZRRSBO(1) .GE. ZRRSBO(1)) GO TO 300
    CALL ROTATE(XSSOMRE,XRRSB,ZVECT,DCMSR,1)
    XRRSB=XRRSB+XRRSBO(1)
    YRRSB=YRRSB+YRRSBO(1)
    ZRRSB=ZRRSB+ZRRSBO(1)
    XYZ(1)=XSSOSB(1)
    XYZ(2)=YSSOSB(1)
    XYZ(3)=ZSSOSB(1)
    CALL ROTATE (XYZ,XYZ,ZVECT,DCMSR,1)
    XDISP=XYZ(1)-XRRSB
    YDISP=XYZ(2)-YRRSB
    CALL ROTATE(XRRSB,XYZ,ZVECT,DCMSR,0)
C *****
240 C COMPUTE FORCES IN RCS, THEN ROTATE TO SCS
    C *****
    240 CONTINUE
    FXR=-KXSB*XDISP
    FYR=-KYSB*YDISP
    FZR=-MUSB*SORT(FXR*FXR+FYR*FYR)
    CALL ROTATE (FXR,FXR,ZVECT,DCMSR,0)
    FXLSO(1)=FXR
    FYLSO(1)=FYR
    FZLSO(1)=FZR
C *****
255 C COMPUTE MOMENTS
    C *****
    275 CONTINUE
    TLISO(1)=XYZ(2)*FZLSO(1) - XYZ(3)*FYLSO(1)
    TMISO(1)=XYZ(3)*FXLSO(1) - XYZ(1)*FZLSO(1)
    TNISO(1)=XYZ(1)*FYLSO(1) - XYZ(2)*FXLSO(1)
C *****
260 C COMPUTE TORSIONAL SPRING MOMENTS IN PITCH PLANE
    C *****
    300 CONTINUE
    TORPTCH = -YKTOR*(ANGSR-PSISR)
    TMSLO(1) = TMSLO(1)+TORPTCH
    9000 RETURN
    END

```

```

1 SUBROUTINE RECOV
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - CONTROLS SEQUENCING OF A RECOVERY CHUTE SYSTEM
C METHOD - THIS ROUTINE CHECKS FOR PROJECTION, LINE STRETCH
C AND FULL INFLATION OF THE RECOVERY CHUTE, THEN
C COMPUTES THE CHUTE'S POSITION
C COMMUNICATIONS -
C CALLED BY:
C CHUTES
C CALLS:
C CHUTFM
C PCHUTFT
C ROTATE
C ZLININT
C NON-COMMON VARIABLES DEFINED -
C TVEL - TOTAL VELOCITY OF THE RECOVERY CHUTE SYSTEM
C
C XD1S -
C YD1S - COMPONENTS OF THE VECTOR FROM THE RECOVERY CHUTE
C ZD1S - ATTACHMENT POINT TO THE RECOVERY CHUTE
C
C POTENTIAL ERROR CONDITIONS: NONE
C.....
C MATRIX COMMON BLOCK
C.....
C COMMON /MATRIX / DCM4E(3,3) , DCMRA(3,3) , DCM5A(3,3) ,
+ DCMSE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,
+ DCM5AE(3,3) , DCMOAE(3,3) , DCM5R(3,3) ,
+ DCMOUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
C COMMON /MISC / IPAGECT(31) , LINECT(31) , IPTCNT(31) ,
+ MAXLINE , MAXREPT , MAXEVNT ,
+ IEVFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS ,
+ REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ INVDC , PRTMP( 2) ,
+ PRTMAS(2) , PRTINDX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+ REPTYPE , BIAS , PRTLNGT ,
+ PRTWGT , PRTEMP , PRTMAS , PRTINDX
C.....
C MOMARMS COMMON BLOCK
C.....
C COMMON /MOMARMS /
+ REFLNSO , REFLNOA , REFLNSA , URX(6) , URY(6) , URZ(6) ,
+ XSSOCA(2) , YSSOCA(2) , ZSSOCA(2) , XSSORK(6) , YSSORK(6) , ZSSORK(6) ,
+ XSSORRE , YSSORRE , ZSSORRE , XSSOLRE , YSSOLRE , ZSSOLRE ,
+ XSSOMRE , YSSOMRE , ZSSOMRE , XSSOBOT , YSSOBOT , ZSSOBOT ,

```

```

60  +XSSOSB(6),YSSOSB(6),ZSSOSB(6),XRRCSAC ,YRRCSAC ,ZRRCSAC ,
    +XSSCSAC ,YSSCSAC ,ZSSCSAC ,XSSOSRP ,YSSOSRP ,ZSSOSRP ,
    +
    +XSSASRP ,YSSASRP ,ZSSASRP ,XRRDAP(2),YRRDAP(2),ZRRDAP(2),
    +XRRSBO(6),YRRSBO(6),ZRRSBO(6),XSSOCP(2),YSSOCP(2),ZSSOCP(2),
    +XSSDAP(2),YSSDAP(2),ZSSDAP(2),XESDAC ,YESDAC ,ZESDAC ,
    +XSRCSAC ,YSRCSAC ,ZSRCSAC ,XSSDAC ,YSSDAC ,ZSSDAC ,
    +XRSOSB ,YRSOSB ,ZRSOSB ,XRRSBO ,YRRSBO ,ZRRSBO ,
    +XRRSB ,YRRSB ,ZRRSB ,XSSOCH(3),YSSOCH(3),ZSSOCH(3),
    +XAACSD ,YAACSD ,ZAACSD ,XASDAC ,YASDAC ,ZASDAC ,
    +XRSOAC ,YRSOAC ,ZRSOAC ,XSCAP(2),YSCAP(2),ZSCAP(2),
70  C .....
    C SECTION 14 COMMON BLOCK
    C .....
    COMMON /PARCHUT / IRECOV , TROPLOY , RECOVLL ,
    + XRECAP , YRECAP , ZRECAP , POROSR ,
    + NPTSRLS , RECOVLS(2,25) , IFTRECV ,
    + NPTSRTF , RECOVFT(2,25) , SEPERCE ,
    + IDROGUE , DRDRAG2 , DRDGP2 ,
    + POROSD2 , VELCON , IFTDR02 ,
    + NPTDFT1 , DRDGT2(2,25) , IFTDR01 ,
    + NPTSCLS , DRDGT1(2,25) , IDROGLS ,
    + DISPLAY , DRDGLS(2,25) , TROPLOY ,
    + DRDGP1 , DRDGLL , DRDRAG1 ,
    + DROVELY , POROS01 , DROVELX ,
    + YDROGAP , ZDROGAP , XDR0GAP ,
    + CHALT2 , GLMIT , CHALT1 ,
    + AREADC , WGHTRC , TDELAY ,
    + TFP2 , TFP3 , TFP1 ,
    + CDDC , NPTSRTD , TDRGLS ,
    + RECOVFT(2,25) ,
90  C .....
    C INTEGRATION ROUTINE COMMON BLOCK
    C .....
    COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSD(193) ,
    + TRAJA(193) , TRAJDA(193) , TRAJCH(97,3) ,
    + TRAJAC(193) , TVCEQS(225) , QUATSD(65) ,
    + QUATSA(65) , QUATOA(65) , QUATAC(65) ,
    + INTSTP , IPCPASS , IRKPASS ,
    + IPOINTS , IYX , IYPRX ,
    + IKX , IKSUMX , IKPASSX ,
    + IYIX , IY1X , IY12X ,
    + IYPR12X , IY1X , IY11X ,
    + IY13X , IYPR1X , IYPR11X ,
    + ICYIX , ICY11X , IREIN ,
100  C .....
    C SEAT/OCCUPANT FORCES COMMON BLOCK
    C .....
    COMMON /FORCESO / FXCASO(2) , FZCASO(2) ,
    + FXTUBSO , FZTUBSO ,
    + FXLSO(6) , FZLSO(6) ,
    + FXRKS(6) , FZRKS(6) ,
    + FXCHSO(3) , FZCHSO(3) ,
    + FXAESO , FZAESO ,
    + FXDRTSO , FZDRTSO ,
110  C .....

```

```

115 C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
COMMON /TORQSO / TLCSO(2) , TMCASO(2) , TNCASO(2) ,
+ TLUBSO , INTUBSO ,
+ TMSLSO(6) , TMSLSO(6) , TNSLSO(6) ,
+ TMRKSO(6) , TMRKSO(6) , TNRKSO(6) ,
+ TMSO(3) , TMSO(3) , TMSO(3) ,
+ TMAESO , TMAESO , TMAESO ,
+ TMDTSO , TMDTSO , TMDTSO
C.....

120

125 C.....
C CHECK FOR RECOVERY CHUTE PROJECTION
C.....
IF(IEVENTS(24) .GE. 1) GOTO 10
IF(INTSTP .EQ. 0) GOTO 500
IF(TRAJSD(4) .LT. CHALT1) GO TO 5
IF(TRAJSD(4) .LT. CHALT2) GO TO 5
C.....
C BETWEEN THE UPPER AND LOWER ALTITUDE LIMITS (CHALT1 & CHALT2, RES-
C PECTIVELY). THE TRECQV=2 OPTION ALLOWS A DELAY OF THE RECOVERY CHUTE
C DEPLOYMENT UNTIL THE SEAT ACCELERATION ALONG THE Z-AXIS HAS FALLEN
C BELOW A PRESCRIBED LIMIT (GLIMIT). THE G-LIMIT TEST CAN BE
C DELAYED BY THE USE OF TDELAY, WHICH CAN PRESCRIBE A FIXED TIME
C DELAY PRECEDING THE GLIMIT TEST. THIS FEATURE IS USEFUL IN PREVENT--
C ING GLIMIT DELAYS DURING CATAPULT, ROCKET, AND DART ACCELERATIONS.
C IF USING TRECQV=1, THEN THE DEPLOYMENT OF THE CHUTE IS DELAYED BY
C TDELAY IF BETWEEN CHALT1 AND CHALT2.
C.....
IF(TRECQV .GE. 2) GOTO 2
IF(TIMES .LT. TDELAY) GOTO 500
IF(TDELAY .LT. 0.0) GOTO 5
TROPLOY = TDELAY
TDELAY = -1.0
2 CONTINUE
IF(TIMES .LT. TDELAY) GOTO 5
IF(ABS(ZACCEL(1)) .GE. GLIMIT) TROPLOY = TROPLOY + DELTAT
5 CONTINUE
IF(TIMES .LT. TROPLOY) GOTO 500
IEVENTS(24) = 1
TIMES(24) = TIME
TVEL = SORT(TRAJSD(14)*TRAJSD(14) + TRAJSD(15)*TRAJSD(15) +
+ TRAJSD(16)*TRAJSD(16))
CALL ZLININT(TVEL,RECOVLS,NPTSRLS,25,TRECCLS,2)
C.....
C SET ALL CHUTE FORCES AND MOMENTS TO ZERO
C.....
DO 8 I=1,3
FXCHSO(I) = FYCHSO(I) = FZCHSO(I) = 0.0
TLCHSO(I) = TMCHSO(I) = TNCHSO(I) = 0.0
8 CONTINUE
GOTO 500
C.....
C CHECK FOR RECOVERY CHUTE LINE STRETCH
C.....

```

```

C
10 CONTINUE
  IF (IEVENTS(25) .NE. 0) GOTO 40
  IF (INSTP .EQ. 0) GOTO 500
  IF (TIMES .LT. (TIMES(24) + TRECCLS)) GOTO 500
  IEVENTS(25) = 1
  TIMES(25) = TIME
  TRAJCH(5,3) = TRAJSO(14)
  TRAJCH(6,3) = TRAJSO(15)
  TRAJCH(7,3) = TRAJSO(16)
  GOTO 15
15 CONTINUE
  TVEL = SORT(TRAJCH(5,3)*TRAJCH(5,3) + TRAJCH(6,3)*TRAJCH(6,3) +
+ TRAJCH(7,3)*TRAJCH(7,3))
  IF (IFIRECV .EQ. 0) GOTO 25
  CALL ZLININT(TVEL, RECOVFT, NPTSRT, 25, TFP3, 2)
  GOTO 500
25 CONTINUE
  CALL PCHUTFT(TVEL, RECOVPD, RECDRAG, POROSR, TFP3)
  GOTO 500
C
C.....
C CHECK FOR RECOVERY CHUTE FULL INFLATION
C.....
195 C
40 CONTINUE
  IF (IEVENTS(26) .NE. 0) GOTO 60
  IF (INSTP .EQ. 0) GOTO 60
  IF (TIMES .LT. (TIMES(25) + TFP3)) GOTO 60
  IEVENTS(26) = 1
  TIMES(26) = TIME
C.....
C COMPUTE RECOVERY CHUTE FORCE
C.....
205 C
60 CONTINUE
  ICHUTE = 3
  CALL CHUTEM(ICHUTE, RECOVPD)
C.....
C COMPUTE RECOVERY CHUTE POSITION
C.....
215 C
  XYZ(1) = XSSOCH(3)
  XYZ(2) = YSSOCH(3)
  XYZ(3) = ZSSOCH(3)
C
  CALL ROTATE(XYZ(1), XYZ(1), ZVECT(1), DOMSE, 1)
C
  IF (IEVENTS(28) .NE. 0) GOTO 75
C
  R = SORT(TRAJSO(14)*TRAJSO(14) + TRAJSO(15)*TRAJSO(15) +
+ TRAJSO(16)*TRAJSO(16))
  IF (R .EQ. 0.0) GOTO 400
C
  SIN2 = TRAJSO(16)/R

```

PAGE 181

83/11/07 09.41.53

FTN 4.6+428

SUBROUTINE RECOV 74/74 OPT=1

```

230      COS2 = COS(ASIN(SIN2))
          BETA = ZARCTAN(TRAJSD(15), TRAJSD(14))
          C
          C
          XD1S = -SIGN((RECOVLL+COS2*COS(BETA)), TRAJSD(14))
          YD1S = -SIGN((RECOVLL+COS2*SIN(BETA)), TRAJSD(15))
          ZD1S = -SIGN((RECOVLL*SIN2), TRAJSD(16))
          C
          TRAJCH(2,3) = XYZ(1) + XD1S + TRAJSD(2)
          TRAJCH(3,3) = XYZ(2) + YD1S + TRAJSD(3)
          TRAJCH(4,3) = XYZ(3) + ZD1S + TRAJSD(4)
          C
          GOTO 800
          C
          75 CONTINUE
          C
          R = SORT(TRAJDA(5)*TRAJDA(5) + TRAJDA(6)*TRAJDA(6) +
          + TRAJDA(7)*TRAJDA(7))
          IF(R .EQ. 0.0) GOTO 400
          C
          SIN2 = TRAJDA(7)/R
          COS2 = COS(ASIN(SIN2))
          BETA = ZARCTAN(TRAJDA(6), TRAJDA(5))
          C
          XD1S = -SIGN((RECOVLL+COS2*COS(BETA)), TRAJDA(5))
          YD1S = -SIGN((RECOVLL+COS2*SIN(BETA)), TRAJDA(6))
          ZD1S = -SIGN((RECOVLL*SIN2), TRAJDA(7))
          C
          TRAJCH(2,3) = XYZ(1) + XD1S + TRAJDA(2)
          TRAJCH(3,3) = XYZ(2) + YD1S + TRAJDA(3)
          TRAJCH(4,3) = XYZ(3) + ZD1S + TRAJDA(4)
          C
          GOTO 500
          C
          400 CONTINUE
          WRITE(5,410)
          410 FORMAT(2X, '//72(1H*)//.4X, "FATAL ERROR(SUBROUTINE RECOV)***
          + "R EQUAL TO ZERO RESULTS IN DIVISION BY ZERO",//.72(1H*))
          IERRFLG = 1
          500 CONTINUE
          RETURN
          END

```


5-183

SUBROUTINE REINZRO 74/74 OPT=1 FTN 4 6+428 83/11/07 09 41 53 PAGE 183

```

1      SUBROUTINE REINZRO(ARRAY)
C .....
C DESCRIPTION - FUNCTIONAL SUBROUTINE
C FUNCTION - ZERO OUT INTEGRATION ARRAYS
5      METHOD - ZERES OUT ALL VALUES PAST THE DERIVATIVES IN THE ARRAY
C          PASSED IN THE CALL. BASED ON THE NUMBER OF EQUATIONS IN
C          THE ARRAY.
C COMMUNICATIONS -
C CALLED BY:
10     C          REINTEG
C          CALLS:
C          NONE
C POTENTIAL ERROR CONDITIONS -
15     C          NONE
C .....
C          DIMENSION ARRAY(225)
C          IF (ARRAY(1).EQ. 0) GO TO 9000
C          NSTART=ARRAY(1)+2
C          NSTOP =16*ARRAY(1)+1
C          DO 100 I=NSTART,NSTOP
C             ARRAY(I)=0.0
C          100 CONTINUE
C          9000 RETURN
C          END
20

```

```

1 SUBROUTINE REPORTS
C.....
C VERSION KGESSAB - DATED 7 NOVEMBER 1983
C.....
5 C.....
C DESCRIPTION - LEVEL 2
C.....
C FUNCTION - CONTROLS THE CREATION OF OUTPUT REPORTS, AS
C SPECIFIED BY THE INPUT FLAGS.
C.....
C METHOD - THIS ROUTINE WILL CALL ALL OF THE REPORT WRIT-
C ING SUBROUTINES BASED ON A VALUE STORED IN
C ARRAY IREPTS.
C EACH POSITION OF THE ARRAY REPRESENTS ONE
C OF THE REPORT SUBROUTINES (I.E. IREPTS(2) - SUBROUTINE
C REPT2). IF THE VALUE OF A SLOT IN ARRAY IREPTS
C IS POSITIVE THEN THE CORRESPONDING SUBROUTINE
C IS EXECUTED. IF THE VALUE IS ZERO THEN THE
C CORRESPONDING SUBROUTINE IS IGNORED.
C NOTE: REPORT1 IS GENERATED BY ROUTINE INPUT AND IT
C IS GENERATED EVERY TIME. THUS IREPTS(1) IS A DUMMY
C LOCATION.
C.....
C EX. CALL REPORTS
C.....
C I.E. IF IREPTS(2) = 1 ROUTINE REPT2 IS EXECUTED
C IF IREPTS(3) = 1 ROUTINE REPT3 IS EXECUTED
C.....
C IF IREPTS(2) = 0 ROUTINE REPT2 IS IGNORED
C.....
C NOTE: REPT13 IS CALLED FOR EACH OF THE 6 ROCKET
C REPORTS. THE ACTUAL ROCKET NUMBER IS PASSED
C AS A FORMAL PARAMETER TO IDENTIFY WHICH OF THE
C 6 REPORTS IS BEING PROCESSED.
C EX. CALL REPT13(4)
C.....
C.....USED FOR ROCKET 4 FORCES, MOMENTS REPORT.
C.....
C COMMUNICATIONS -
C.....
C CALLED BY GESS
C INTLZ
C.....
C CALLS
C EVENTS
C REPT2
C REPT3
C REPT4
C REPT5
C REPT6
C REPT7
C REPT8
C REPT9
C REPT10
C.....
95 C.....

```

5-186

```

115      +      TRAJSA(193)      TRAJDA(193)      TRAJCH(97,3)
      +      TRAJAC(193)      TVCEQS(225)      QUATSO(65)
      +      QUATSA(65)      QUATOA(65)      QUATAC(65)
      +      INTSTP      IPCPASS      IRKPASS
      +      IPOINIS      IYX      IYPRX
      +      IKX      IKSUMX      IKPASSX
      +      IYIX      IYIIX      IYI2X
      +      IYI3X      IYPRIX      IYPRIIX
      +      IYPR12X      IYIIX      IYIIX
      +      ICYIX      ICYIIX      IREIN
      +
      IF(INTSTP.EQ.0) GO TO 500
      IF(TIME5.EQ.0.0) GO TO 10
      IF(PRTFRO.NE.0) GO TO 10
C.....
C IF PRTFRO IS ZERO, PRINT EVERY 0.1 SECOND
      IF (TIME5 .LT. SAVTIME) GO TO 500
      SAVTIME = TIME + 0.1
      GO CONTINUE
      CALL EVENT
135      IF(IREPTS( 2) .EQ. 1) CALL REPT2
      IF(IREPTS( 3) .EQ. 1) CALL REPT3
      IF(IREPTS( 4) .EQ. 1) CALL REPT4
      IF(IREPTS( 5) .EQ. 1) CALL REPT5
      IF(IREPTS( 6) .EQ. 1) CALL REPT6
      IF(IREPTS( 7) .EQ. 1) CALL REPT7
      IF(IREPTS( 8) .EQ. 1) CALL REPT8
      IF(IREPTS( 9) .EQ. 1) CALL REPT9
      IF(IREPTS(10) .EQ. 1) CALL REPT10
      IF(IREPTS(11) .EQ. 1) CALL REPT11
      IF(IREPTS(12) .EQ. 1) CALL REPT12
      IF(IREPTS(13) .EQ. 1) CALL REPT13(1)
      IF(IREPTS(14) .EQ. 1) CALL REPT13(2)
      IF(IREPTS(15) .EQ. 1) CALL REPT13(3)
      IF(IREPTS(16) .EQ. 1) CALL REPT13(4)
      IF(IREPTS(17) .EQ. 1) CALL REPT13(5)
      IF(IREPTS(18) .EQ. 1) CALL REPT13(6)
      IF(IREPTS(19) .EQ. 1) CALL REPT19
      IF(IREPTS(20) .EQ. 1) CALL REPT20
      IF(IREPTS(21) .EQ. 1) CALL REPT21
      IF(IREPTS(22) .EQ. 1) CALL REPT22
      IF(IREPTS(23) .EQ. 1) CALL REPT23
      IF(IREPTS(24) .EQ. 1) CALL REPT24
      IF(IREPTS(25) .EQ. 1) CALL REPT25
      IF(IREPTS(26) .EQ. 1) CALL REPT26
      IF(IREPTS(27) .EQ. 1) CALL REPT27
      IF(IREPTS(28) .EQ. 1) CALL REPT28
      IF(IREPTS(29) .EQ. 1) CALL REPT29
      IF(IREPTS(30) .EQ. 1) CALL REPT30
      IF(IREPTS(31) .EQ. 1) CALL REPT31
      500 CONTINUE
      RETURN
      END
165

```

```

1      SUBROUTINE REPR11
C.....
C
C INPUT VALIDATION REPORT
C
C CALLED BY: INITLZ
C
C.....
C COEFFICIENTS (USED IN SUBROUTINE AERDIN) COMMON BLOCK
C.....
C COMMON /COEF / COEF(700,6)
C.....
C CONSTANTS COMMON BLOCK
C.....
C COMMON /CONSTNT / GRAVITY , RADDEG , DEGRAD , PI
C.....
C SECTION 13 COMMON BLOCK
C.....
C COMMON /DYNCGIN / IDYNGC , WY , WYY ,
+ CX , XSLACK , SKP , SXN ,
+ CY , SY , CZ , ZSLACK ,
+ SZP , ZBOT , SZN1 , SZN2
C.....
C EVENT MESSAGES COMMON BLOCK
C.....
C COMMON /EVNES / IEVNES(3,38) , ISPNES(4,6) , ISPECL(6)
C.....
C OCCUPANT ALONE FORCES COMMON BLOCK
C.....
C COMMON /FORCEA / FXCHOA(3) , FYCHOA(3) , FZCHOA(3) ,
+ FXAEOA , FYAEOA , FZAEOA
C.....
C SEAT ALONE FORCES COMMON BLOCK
C.....
C COMMON /FORCEA / FXAESA , FYAESA , FZAESA
C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
C COMMON /FORCEO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+ FXTUBSO , FYTUBSO , FZTUBSO ,
+ FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+ FXRKS(6) , FYRKS(6) , FZRKS(6) ,
+ FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+ FXAESO , FYAESO , FZAESO ,
+ FXORTSO , FYORTSO , FZORTSO
C.....
C SECTION 4 COMMON BLOCK
C.....
C COMMON /AIRCRT / TEMP , PRESSUR , ZACVEL , XPOS , YPOS ,
+ ZPOS , XTAIL , VTAIL , ZTAIL , YAW ,
+ PITCH , ROLL , RVEL , OVEL , PVEL ,
+ WINDX , WINDY , WINDZ , XACVEL , CKPITHT ,
+ DENSITY , NPTSAAT , AAT(4,50) , NPTSLAT , LAT(4,50) ,
+ IACSFLG
C.....
C SECTION 9 COMMON BLOCK
C.....

```

```

60      COMMON /ICATPLT / INCAT      , CATLNT(2), CATSTK(2), TCI      (2),
      +      XPOSAP(2)      , YPOSAP(2), ZPOSAP(2), NPTSTCT(2),
      +      CATHRST(2,25,2), ITUBEND , KTUBE , CTUBE ,
      +      PTUBE      , MUTUBE , EXTNGT , ICATOUT
      REAL
      KTUBE
      MUTUBE
      ICATOUT
C.....
C SECTION 1 COMMON BLOCK
C.....
65      COMMON /ICONTRL / TSTART , ISTOP , ESTOP , IRESTR, IUNITS ,
      +      ISEATTR, IJOSEP , IPLOT , IDRIFLG,
      +      IPHASE1, IPHASE2, IPHASE3
      INTEGER
      ESTOP
C.....
C SECTION 11 COMMON BLOCK
C.....
70      COMMON /IDARTIN / IDART      , DRTFRCE , DRTSTRT , DRTSTOP ,
      +      XDRTAP(2), YDRTAP(2), ZDRTAP(2),
      +      XDRTCP(2), YDRTCP(2), ZDRTCP(2)
C.....
C SECTION 3 COMMON BLOCK
C.....
75      COMMON /IDELTAT / DTPHASE1, DTPHASE2, DTPHASE3
C.....
C INFO04 DATA (USED IN SUBROUTINE AERDIN) COMMON BLOCK
C.....
80      COMMON /INFO04 / NCXS(12) , NCYS(12) , NCZS(12) ,
      +      DLTC(3,12) , ENDRPC(6,12) , IAERCSQ(12)
C.....
C SECTION 7 COMMON BLOCK
C.....
85      COMMON /IRAIL / RAILNTH , RAILANG , ISTRL , NSLBKS ,
      +      KXSB , KYSB , MUSB , YKTOR ,
      +      XPOSRR , YPOSRR , ZPOSRR ,
      +      XPOSRL , YPOSRL , ZPOSRL ,
      +      XPOSSB(6), YPOSSB(6), ZPOSSB(6)
      REAL
      KXSB , KYSB , MUSB
C.....
C SECTION 2 COMMON BLOCK
C.....
95      COMMON /IREPORT / IREPTS(31) , PRTRQ,PI1,PI2,PI3
      INTEGER
      PRTRQ,PI1,PI2,PI3
C.....
C RECALCULATED ROCKET THRUST TABLE COMMON BLOCK
C.....
100     COMMON /IRKTOUT / RKTOUT(2,25,6)
C.....
C SECTION 10 COMMON BLOCK
C.....
105     COMMON /IROCKET / INRKT      , RKDELY(6), RKNPTS(6), IROKOUT ,
      +      RKIGN(6) , RKWGT(6), RKBURN(6), TSTAR(6) ,
      +      XPOSRK(6), YPOSRK(6), ZPOSRK(6),
      +      RKALPH(6), RKBETA(6), RKGAMA(6), RKTHRST(2,25,6)
      INTEGER
      RKNPTS
C.....
C SECTION 6 COMMON BLOCK
C.....
110     COMMON /ISEATOC / IPCNLT , XCGSO , YCGSO , ZCGSO , IXXSO ,

```

```

115      + IXYSO , IXZSO , IYYSO , IY2SO , IZ2SO ,
      + AREASO , AREASO , WGHIOAB , WGHIOAB ,
      + IXXOA , IXXOA , IXXOA , IXXOA , IY2OA , IY2OA ,
      + IZ2OA , IZ2OA , XGQOA , XGQOA , ZGQOA , ZGQOA , SOSEP ,
      + REAL
      + IXYSO , IXZSO , IYYSO , IY2SO , IZ2SO ,
      + IXXOA , IXXOA , IXXOA , IXXOA , IY2OA , IY2OA ,
      + IZ2OA , IZ2OA
C.....
C SECTION 5 COMMON BLOCK
C.....
      COMMON /ISETALN / XPOSSRP , YPOSSRP , ZPOSSRP , XCGSA , YCGSA ,
      + ZCGSA , IXXSA , IXXSA , IXXSA , IXXSA , IY2SA , IY2SA ,
      + IZ2SA , IZ2SA , PHISA , PHISA , PSISA , PSISA , THESA ,
      + AREASA , HGHTSA , WGHISA , XPOSBOT , YPOSBOT ,
      + ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS ,
      + REAL
      + IXXSA , IXXSA , IXXSA , IXXSA , IY2SA , IY2SA ,
      + IZ2SA , IZ2SA
C.....
C SECTION 12 COMMON BLOCK
C.....
      COMMON /ITVCIN / ITVC , MPHI , MPHI , MPSI , MPHI , MTHE ,
      + ROLLRL , PITCHRL , SMPLRAT , TVCDLAY ,
      + RKANG
      + REAL
      + MPHI , MPHI , MPHI , MPHI , MTHE
C.....
C MASSES COMMON BLOCK
C.....
      COMMON /MASSES / MASSOA1 , MASSOA2 , MASSO , MASSO , MASSO ,
      + MASSA , MASSRK(6) , MASSDC ,
      + REAL
      + MASSOA1 , MASSOA2 , MASSO , MASSO ,
      + MASSA , MASSRK , MASSRK , MASSDC
C.....
C MATRIX COMMON BLOCK
C.....
      COMMON /MATRIX / DCMAE(3,3) , DCMAE(3,3) , DCMSA(3,3) ,
      + DCMSA(3,3) , DCMTS(3,3) , DCMTS(3,3) , DCMTS(3,3) ,
      + DCMSAE(3,3) , DCMSAE(3,3) , DCMSR(3,3) ,
      + DCMDUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
      COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCTNT(31) ,
      + MAXLINE , MAXREPT , MAXEVT , LU ,
      + IEVLIN , TERFLG , LU ,
      + IDATE , HEADALT , HEADVEL ,
      + HEADSR , HEADYAW , HEADPIT ,
      + HEADROL , HEADWGT , BIAS ,
      + REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
      + IHEADER(24) , IEVENTS(36) , TIMES(36) ,
      + IMVOC , PRTIMP( 2) ,
      + PRIMASS(2) , PRTINDX ,
      + ZVECT(3) , XYZ(3) , PKZVEL ,
      + XACCEL(3) , YACCEL(3) , SAVTIME ,
      + INTEGER
      + REPTYPE , BIAS , ZACCEL(3) ,
      + PRTWGT , PRTIMP , PRTWGT , PRTLNCT ,
      + PRTIMP , PRIMASS , PRTINDX
C.....

```


5-191

```

230 C.....
C TORQUE OCCUPANT ALONE COMMON BLOCK
C.....
COMMON /TORQA / TLCHOA(3) , TMCHOA(3) , TNCHOA(3) ,
+      TLAEOA , TMAEOA , TNAEOA
C.....
235 C TORQUE SEAT ALONE COMMON BLOCK
C.....
COMMON /TORQA / TLAESA , TMAESA , TNAESA
C.....
240 C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
COMMON /TORQSO / TLCASO(2) , TMCASO(2) , TNCASO(2) ,
+      TLTUBSO , TMTUBSO , TINTUBSO ,
+      TSLISO(6) , TMSISO(6) , TNSISO(6) ,
+      TLRKSO(6) , TMRKSO(6) , TNRKSO(6) ,
+      TLCHSO(3) , TMCHSO(3) , TNCHSO(3) ,
+      TLAESO , TMAESO , TNAESO ,
+      TLDRTSO , TMDRTSO , TNDRTSO
C.....
C DAMPING COEFFICIENT COMMON BLOCK
C.....
COMMON /DAMPING / DMPGF2 , DMPGF3 , DMPGC
DIMENSION MEASURE(6)
IBLANK = 1H
LU = BIAS + 1
IF(TSTOP.NE.O.O.AND.TIMES.GE.TSTOP) GOTO 650
IF(ESTOP.NE.O.O.AND.IEVENTS(ESTOP).NE.O) GOTO 650
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C FOR PROGRAM CONTROL VARIABLES
C.....
CALL HEADER
WRITE(5,1100)
WRITE(5,1101) TSTART
IF(TSTOP.NE.O.O) WRITE(5,1102) TSTOP
IF (TSTOP.EQ.O.O) WRITE(5,1103)(IEUVES(I,ESTOP),I=1,3)
WRITE(5,1104)
WRITE(5,1105) IRESTRT
WRITE(5,1106) IUNITS
WRITE(5,1108) ISEATTR
WRITE(5,1110) IPLOT
WRITE(5,1111) ISOSEP
WRITE(5,1112) IDRIFLG
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C FOR REPORT CONTROL FLAGS
C.....
CALL HEADER
WRITE(5,1210)
WRITE(5,1215)
DO 20 K = 1,MAXREPT
IF(IREPTS(K).NE.1) GO TO 20
WRITE(5,1220) K, (REPTYPE(I,K),I=1,5)
20 CONTINUE
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT

```

C FOR INTEGRATION TIME STEPS

```

C.....
CALL HEADER
WRITE(5,1300)
WRITE(5,1310) DTPHAS1,P11
WRITE(5,1320) DTPHAS2,P12
WRITE(5,1330) DTPHAS3,P13
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C FOR AIRCRAFT INITIAL CONDITIONS
C.....
CALL HEADER
WRITE(5,1400)
WRITE(5,1410) TEMP
+
+ DENSITY
+ CKPITHI
+ WINDX
+ WINDY
+ WINDZ
+
WRITE(5,1420) PRTLNGT(PRTINDX), PRTLNGT(PRTINDX)
PVEL = RVEL*RADDEG
PPVEL = QVEL*RADDEG
WRITE(5,1430) XPOS , HEADYAW, XACVEL , PRVEL
WRITE(5,1440) YPOS , HEADPIT, PVEL
WRITE(5,1450) ZPOS , HEADROL, ZACVEL , PPVEL
IF(NPTSLEAT.EQ.0.AND.NPTSAT.EQ.0) GOTO 48
CSL = 4HEFC5
IF(NPTSLEAT.LT.0) CSL = 4H ACS
CALL HEADER
WRITE(5,1480) CSL,PRTLNGT(PRTINDX)
NPTS = IABS(NPTSLEAT)
IF(IABS(NPTSLEAT).LT.IABS(NPTSAT)) NPTS = NPTSAT
DO 40 K=1,NPTS
WRITE(5,1490) (LAT(I,K),I=1,4), (AAT(I,K),I=1,4)
40 CONTINUE
48 CONTINUE
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C FOR SEAT ALONE INITIAL CONDITIONS
C.....
CALL HEADER
WRITE(5,1500)
WRITE(5,1505) AREASA, PRTLNGT(PRTINDX), HGHTSA , PRTLNGT(PRTINDX),
+
+ WGTSA, PRTWGT(PRTINDX)
WRITE(5,1506) PRTLNGT(PRTINDX), PRTLNGT(PRTINDX)
WRITE(5,1507) XPOSSCS,YPOSBOT,ZPOSSCS,ZPOSBOT,ZPOSSCS
WRITE(5,1510) PRTLNGT(PRTINDX), PRTLNGT(PRTINDX)
PPHISA = PHISA*RADDEG
PPSISA = PSISA*RADDEG
PTHESA = THESA*RADDEG
WRITE(5,1515) XPOSSRP, PPHISA , XCGSA
WRITE(5,1520) YPOSSRP, PPSISA , YCGSA
WRITE(5,1525) ZPOSSRP, PTHESA , ZCGSA
WRITE(5,1530) PRTMASS(PRTINDX), PRTLNGT(PRTINDX),
+
+ IXXSA, IXXSA, IXXSA, IXXSA, IXXSA, IXXSA, IXXSA, IXXSA
C.....

```

```

345 C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
346 C FOR SEAT/OCCUPANT, OCCUPANT ALONE INITIAL CONDITIONS
347 C .....
348 CALL HEADER
349 WRITE(5,1600)
350 WRITE(5,1608) AREASO, PRTLNGT(PRTINDX),
351 + AREAOA, PRTLNGT(PRTINDX),
352 + HEADWGT, PRTWGT(PRTINDX),
353 + WGHIOAB, PRTWGT(PRTINDX),
354 + WGHIOAA, PRTWGT(PRTINDX)
355 IF(ISOSEP.EQ.0) WRITE(5,1606)
356 IF(ISOSEP.EQ.1) WRITE(5,1607) SOSEP
357 IF(ISOSEP.EQ.2) WRITE(5,1608) SOSEP,PRTWGT(PRTINDX)
358 WRITE(5,1609) DMPCG
359 WRITE(5,1610) PRTLNGT(PRTINDX), XCGSO, YCGSO, ZCGSO
360 WRITE(5,1615) PRTMAS(PRTINDX), PRTLNGT(PRTINDX)
361 WRITE(5,1620) IXXSO, IYXSO,
362 + IXXSO, IYXSO,
363 + IYZSO, IZZSO
364 C .....
365 C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
366 C FOR RAIL INITIAL CONDITIONS
367 C .....
368 CALL HEADER
369 WRITE(5,1700)
370 WRITE(5,1710) RAILNTH, PRTLNGT(PRTINDX), RAILANG,RADDEG,
371 + KXSB, PRTWGT(PRTINDX), PRTLNGT(PRTINDX),
372 + KYSB, PRTWGT(PRTINDX), PRTLNGT(PRTINDX),
373 + MUSB, YKTOR,
374 + PRTLNGT(PRTINDX), PRTWGT(PRTINDX)
375 WRITE(5,1715) PRTLNGT(PRTINDX)
376 WRITE(5,1716) XPOSRE, XPOSLE,
377 + YPOSRE, YPOSLE,
378 + ZPOSRE, ZPOSLE
379 IF (NSLBKS.EQ.0) GO TO 50
380 IF (ISTRL.EQ.0) WRITE (5,1712) NSLBKS
381 IF (ISTRL.NE.0) WRITE (5,1711) NSLBKS
382 IF (ISTRL.EQ.0)
383 +WRITE(5,1721) PRTLNGT(PRTINDX),PRTLNGT(PRTINDX),PRTLNGT(PRTINDX)
384 IF (ISTRL.NE.0)
385 +WRITE(5,1720) PRTLNGT(PRTINDX),PRTLNGT(PRTINDX),PRTLNGT(PRTINDX)
386 WRITE(5,1730) XPOSSB(1), XPOSSB(2), XPOSSB(3),
387 + YPOSSB(1), YPOSSB(2), YPOSSB(3),
388 + ZPOSSB(1), ZPOSSB(2), ZPOSSB(3)
389 IF (ISTRL.EQ.0)
390 +WRITE(5,1741) PRTLNGT(PRTINDX),PRTLNGT(PRTINDX),PRTLNGT(PRTINDX)
391 IF (ISTRL.NE.0)
392 +WRITE(5,1740) PRTLNGT(PRTINDX),PRTLNGT(PRTINDX),PRTLNGT(PRTINDX)
393 WRITE(5,1750) XPOSSB(4), XPOSSB(5), XPOSSB(6),
394 + YPOSSB(4), YPOSSB(5), YPOSSB(6),
395 + ZPOSSB(4), ZPOSSB(5), ZPOSSB(6)
396 GO TO 60
397 50 WRITE (5,1712) NSLBKS
398 WRITE(5,1713)
399 60 CONTINUE
400 C .....
401 C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT

```

```

400 C FOR CATAPULT PARAMETERS
C.....*
      IF(INCAT.EQ.0) GO TO 101
      CALL HEADER
      WRITE(5,1900)
      WRITE(5,1905) INCAT, PRTLNKT(PRTINDX)
      DO 75 I=1, INCAT
      WRITE(5,1910) I, CATLNT(I), PRTLNKT(PRTINDX), XPOSAP(I)
      WRITE(5,1920) CATSTK(I), PRTLNKT(PRTINDX), YPOSAP(I)
      WRITE(5,1930) TCI(I), ZPOSAP(I)
75 CONTINUE
      IF (ITUBEND.NE.2) GO TO 85
      CALL HEADER
      WRITE(5,1935)
      WRITE(5,1940) TUBE, PRTWGT(PRTINDX), PRTLNKT(PRTINDX),
+ TUBE, PRTWGT(PRTINDX), PRTLNKT(PRTINDX),
+ MUTUBE, PTUBE
+ EXTLNKT, PRTWGT(PRTINDX), PRTLNKT(PRTINDX)
85 CONTINUE
      CALL HEADER
      IF(INCAT.EQ.2) GO TO 95
      WRITE(5,1975) PRTWGT(PRTINDX)
      INDX = NPTSCT(1)
      DO 90 K=1, INDX
      WRITE(5,1980) CATHRST(1,K,1), CATHRST(2,K,1)
90 CONTINUE
      GO TO 101
95 WRITE(5,1985) PRTWGT(PRTINDX), PRTWGT(PRTINDX)
      INDX=NPTSCT(1)
      IF(NPTSCT(2).GT.INDX) INDX=NPTSCT(2)
      DO 100 K=1, INDX
      WRITE(5,1990) CATHRST(1,K,1), CATHRST(2,K,1),
+ CATHRST(1,K,2), CATHRST(2,K,2)
100 CONTINUE
101 CONTINUE
C.....*
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C.....*
C FOR ROCKET PARAMETERS
C.....*
      IF(INRKT.EQ.0) GO TO 200
      CALL HEADER
      WRITE(5,11000)
      WRITE(5,11010) INRKT, PRTLNKT(PRTINDX)
      DO 102 I=1, INRKT
      WRITE(5,11020) I, RKIGN(I), PRTLNKT(PRTINDX), XPOSRK(I),
+ (RKALPH(I)*RADDEG)
      WRITE(5,11030) RKWGT(I), PRTWGT(PRTINDX), YPOSRK(I),
+ (RKBETA(I)*RADDEG)
      WRITE(5,11040) RKBURN(I), ZPOSRK(I),
+ (RKGAMA(I)*RADDEG)
      WRITE(5,11050) RKDELY(I)
102 CONTINUE
      DO 150 K=1, INRKT
      IF(RKNPTS(K).LT.1) GO TO 150
      CALL HEADER
      WRITE(5,11070) K, PRTWGT(PRTINDX), PRTWGT(PRTINDX)
      INDX = RKNPTS(K)

```

```

DO 140 J = 1,INDX
  WRITE(5,11075) RKTHRST(1,J,K), RKTHRST(2,J,K),
  + RKOUT(1,J,K), RKOUT(2,J,K)
140 CONTINUE
150 CONTINUE
200 CONTINUE
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C FOR DART PARAMETERS
C.....
  IF(1DART.EQ.0) GO TO 250
  CALL HEADER
  WRITE(5,12000)
  WRITE(5,12010) DRTFRCE, PRTWGT(PRTINDX),
  + DRTSTRT, PRTLNGT(PRTINDX),
  + DRTSTOP, PRTLNGT(PRTINDX)
  WRITE(5,12020) PRTLNGT(PRTINDX), PRTLNGT(PRTINDX)
  WRITE(5,12030) XDRTCP(2),
  + XDRTCP(1)
  WRITE(5,12035) YDRTCP(2),
  + YDRTCP(1)
  WRITE(5,12040) ZDRTCP(2),
  + ZDRTCP(1)
  WRITE(5,12045) PRTLNGT(PRTINDX), PRTLNGT(PRTINDX)
  WRITE(5,12050) XDRTAP(2),
  + XDRTAP(1)
  WRITE(5,12055) YDRTAP(2),
  + YDRTAP(1)
  WRITE(5,12060) ZDRTAP(2),
  + ZDRTAP(1)
250 CONTINUE
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C FOR THRUST VECTOR CONTROL DATA
C.....
  IF(1ITVC.EQ.0) GO TO 300
  CALL HEADER
  PMPHI = MPHI *RADDEG
  PMPSI = MPSI *RADDEG
  PMTHE = MTHE *RADDEG
  PROLLRL = ROLLRL *RADDEG
  PPTICRL = PITCHRL *RADDEG
  PSMPPLRT = SMPLRAT *RADDEG
  WRITE(5,13000)
  WRITE(5,13010) PMPHI, PMPSI, PMTHE
  WRITE(5,13015) PROLLRL
  WRITE(5,13020) PPTICRL
  WRITE(5,13025) PSMPPLRT
  WRITE(5,13030) TVCOLAY
  WRITE(5,13035) RKANG
300 CONTINUE
C.....
C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT
C FOR DYNAMIC CG VARIANCES
C.....
  IF(1DYNCG.EQ.0) GO TO 400
  CALL HEADER

```



```

575      IF(IDROGLS .NE. 1) GOTO 435
        WRITE(5,15500)
        WRITE(5,15505)
        NPI = 1
        430 CONTINUE
        NPF = NPI + 7
        IF(NPF .GT. NPISDLS) NPF = NPISDLS
        IF(NPF .GT. NPISDLS) NPF = NPISDLS
        WRITE(5,15510) VELUNIT , (DROGLS(1,1),1 = NPI,NPF)
        WRITE(5,15512)      (DROGLS(2,1),1 = NPI,NPF)
        IF(NPF .GE. NPISDLS) GOTO 435
        NPI = NPI + 8
        GOTO 430
        435 CONTINUE
        IF(IFDRO1 .NE. 1) GOTO 455
        IF(IDROQUE .EQ. 3) GOTO 440
        WRITE(5,15500)
        WRITE(5,15520)
        GOTO 445
        440 CONTINUE
        WRITE(5,15530)
        445 CONTINUE
        NPI = 1
        450 CONTINUE
        NPF = NPI + 4
        IF(NPF .GT. NPISDLS) NPF = NPISDLS
        WRITE(5,15510) VELUNIT , (DROGLT(1,1),1 = NPI,NPF)
        WRITE(5,15512)      (DROGLT(2,1),1 = NPI,NPF)
        IF(NPF .GE. NPISDLS) GOTO 455
        NPI = NPI + 5
        GOTO 450
        455 CONTINUE
        IF(IFDRO2 .NE. 1) GOTO 500
        IF(NPISDLS + NPISDLS + NPISDLS) .GE. 40) CALL HEADER
        WRITE(5,15500)
        WRITE(5,15540)
        NPI = 1
        460 CONTINUE
        NPF = NPI + 4
        IF(NPF .GT. NPISDLS) NPF = NPISDLS
        WRITE(5,15510) VELUNIT , (DROGLT2(1,1),1 = NPI,NPF)
        WRITE(5,15512)      (DROGLT2(2,1),1 = NPI,NPF)
        IF(NPF .GE. NPISDLS) GOTO 500
        NPI = NPI + 5
        GOTO 460
        500 CONTINUE
        C.....
        C WRITE HEADER INFO (VIA SUBROUTINE HEADER) AND VALIDATION REPORT *
        C FOR RECOVERY CHUTE INITIAL CONDITIONS
        C.....
        IF(IRECOV .EQ. 0) GO TO 600
        CALL HEADER
        WRITE(5,16000)
        IF(IRECOV .EQ. 2) GOTO 510
        WRITE(5,16007)
        GOTO 520
        510 CONTINUE
        WRITE(5,16008)

```



```

520 CONTINUE
  WRITE(5,16010) RECOVLL , PRTLNLT(PRTINDX) ,
+ RECDRAG , RECOVPD ,
+ PRTLNLT(PRTINDX) , POROSR ,
+ CHALT1 ,
+ PRTLNLT(PRTINDX) , CHALT2 ,
+ PRTLNLT(PRTINDX) ,
+ PRTLNLT(PRTINDX) , DELAY
  WRITE(5,16005) IDLAY
  IF(IRECOV.EQ.2) WRITE(5,16006) GLIMIT/GRAVITY
  IF(NPTSRT.GT.1) WRITE(5,16003) TRDPLOY
  IF(NPTSRT.GT.1) WRITE(5,16004)
  WRITE(5,16020) PRTLNLT(PRTINDX) ,
+ XRECAP ,
+ YRECAP ,
+ ZRECAP
  IF(NPTSRT.GT.1) GOTO 540
  WRITE(5,16400) PRTLNLT(PRTINDX) , PRTLNLT(PRTINDX)
  NUM = NPTSRLS
  IF(NPTSRT.GT.1) NUM = NPTSRT
  DO 530 I=1,NUM
    WRITE(5,16410) RECOVLS(1,I) , RECOVLS(2,I) ,
+ RECOVFT(1,I) , RECOVFT(2,I)
  530 CONTINUE
  GOTO 600
640 CONTINUE
  WRITE(5,16500) PRTLNLT(PRTINDX) , PRTLNLT(PRTINDX) ,
+ PRTLNLT(PRTINDX)
  NUM = NPTSRT
  IF(NPTSRLS.GT.NUM) NUM = NPTSRLS
  IF(NPTSRT.GT.NUM) NUM = NPTSRT
  DO 560 I=1,NUM
    WRITE(5,16510) RECOVDT(1,I) , RECOVDT(2,I) ,
+ RECOVLS(1,I) , RECOVLS(2,I) ,
+ RECOVFT(1,I) , RECOVFT(2,I)
  560 CONTINUE
  GOTO 600
650 CONTINUE
  GO TO 9000
C.....
C FORMAT STATEMENTS FOR PROGRAM CONTROL VARIABLES VALIDATION REPORT
C.....
650 CONTINUE
1100 FORMAT(10('),52X,"PROGRAM CONTROL VARIABLES",5('/))
1101 FORMAT(3X,"EJECTION SIMULATION START TIME = ",F10.4,
+ "(IF > 0 FROM RESTART FILE)")
1102 FORMAT(3X,"EJECTION SIMULATION STOP TIME = ",F10.4////)
1103 FORMAT(3X,"EJECTION SIMULATION STOP EVENT = ",8X,3(A10)////)
1104 FORMAT(5X,"VARIABLE",20X,"VALUE",20X,"DEFINITION"/)
1105 FORMAT(5X,"IRESTRT",23X,12,18X,"RESTART",
+ "FILE CREATION SWITCH (0 = NOT CREATED, 1 = CREATED)"/)
1106 FORMAT(5X,"IUNITS",23X,12,18X,"UNITS OF MEASUREMENT (",
+ "0 = METRIC, 1 = ENGLISH)"/)
1108 FORMAT(5X,"ISEATTR",23X,12,18X,"SEAT ALONE TRAJECTORY SWITCH (",
+ "0 = TRAJ. NOT GENERATED, 1 = TRAJ. GENERATED)"/)
1110 FORMAT(5X,"IPLOT",23X,12,18X,"PLOT FILE SWITCH (",
+ "0 = NOT CREATED, 1 = CREATED WRT EFCS, 2 = 94X,
+ "2 = CREATED WRT EFCS AND ACS)"/)
1111 FORMAT(5X,"ISOS.P",23X,12,18X,"SEAT/OCCUPANT SEP. SWITCH (",

```

```

885      * O = WILL NOT OCCUR, 1 = WILL OCCUR BASED ON TIME"/.
      * 102X,"2 = WILL OCCUR BASED ON FORCE")//
      1112 FORMAT(5X,"IDRIFLG",23X,12,18X,"DYNAMIC RESPONSE INDEX SWITCH",
      * (O= NOT COMPUTED 1= COMPUTED))
      C.....
      C FORMAT STATEMENTS FOR REPORT FLAGS VALIDATION REPORT
      C.....
      1210 FORMAT(3//),30X,"OUTPUT REPORTS: THE FOLLOWING REPORTS WILL BE ",
      * "CREATED"/)
      1215 FORMAT(30X,"REPORT NO.",150,"TITLE"/,30X,10(1H-),150,5(1H-))
      1220 FORMAT(34X,13,145,5(A10))
      C.....
      C FORMAT STATEMENTS FOR INTEGRATION TIME STEPS VALIDATION REPORT
      C.....
      1300 FORMAT(10//),44X,"INTEGRATION TIME STEPS AND PRINT FREQUENCIES",
      * 3//),77X,"INTEGRATION STEP",14X,"PRINT"/,82X,"(SEC)",18X,
      * "FREQUENCY")
      1310 FORMAT(/,10X,"PHASE 1 (INITIATION TO RAIL CLEARANCE)",27X,
      * "DTPHAS1 = ",F10.5,9X,"PI1 = ",15)
      1320 FORMAT(/,10X,"PHASE 2 (RAIL CLEARANCE TO SEAT/OCCUPANT",
      * " SEPARATION)",13X,"DTPHAS2 = ",F10.5,9X,"PI2 = ",15)
      1330 FORMAT(/,10X,"PHASE 3 (SEAT/OCCUPANT SEPARATION TO COMPLETION)",
      * 17X,"DTPHAS3 = ",F10.5,9X,"PI3 = ",15)
      C.....
      C FORMAT STATEMENTS FOR AIRCRAFT INITIAL CONDITIONS VALIDATION REPORT
      C.....
      1400 FORMAT(3//),52X,"AIRCRAFT INITIAL CONDITIONS",31//)
      1410 FORMAT(54X,"ATMOSPHERIC CONDITIONS"/,37X,"TEMPERATURE",
      * 7X,"(TEMP)",F10.4," DEG ",A10/,
      * 37X,"BAROM PRESSURE (PRESSUR)",F10.4," MILLIBAR"/,
      * 37X,"AIR DENSITY (DENSITY)",F10.4,1X,A4,"/",A2,"**3"/,
      * 37X,"COCKPIT HEIGHT (CKPHTHT)",F10.4," ",A2/,
      * 37X,"WIND VELOCITY (WINDX)",F10.4," ",A2,"/SEC"/,
      * 54X,"(WINDY)",F10.4," ",A2,"/SEC"/,
      * 54X,"(WINDZ)",F10.4," ",A2,"/SEC"/,///)
      1420 FORMAT(14X,"POSITION (EFGS)",13X,"ORIENTATION (EFCS)",
      * 10X,"LINEAR VELOCITY (EFGS)",7X,"ANGULAR VELOCITY (ACS)",/,
      * 19X,"(",A2,")",25X,"(DEG)",23X,"(",A2,"/SEC)",20X,
      * "(DEG/SEC)",/)
      1430 FORMAT(14X,"XPOS",F10.4,12X,"YAW",F10.4,10X,
      * "XACVEL",F10.4,8X,"YAW (RVEL)",F10.4,/,
      1440 FORMAT(14X,"YPOS",F10.4,12X,"PITCH",F10.4,10X,
      * "YACVEL",F10.4,8X,"PITCH (QVEL)",F10.4,/,
      1450 FORMAT(14X,"ZPOS",F10.4,12X,"ROLL",F10.4,10X,
      * "ZACVEL",F10.4,8X,"ROLL (PVEL)",F10.4,/,
      1460 FORMAT(5//),55X,"AIRCRAFT ACCELERATIONS"/,28X,
      * "LINEAR ACCELERATIONS (",A4,")",28X,
      * "ANGULAR ACCELERATIONS (ACS)",/,36X,
      * "(",A2,"/SEC 2)",45X,"(RAD/SEC 2)",/,19X,
      * "TIME (SEC)",8X,"X",11X,"Y",11X,"Z",13X,
      * "TIME (SEC)",8X,"R",11X,"Q",11X,"P"/)
      1490 FORMAT(15X,4(2X,F10.4),8X,4(2X,F10.4))
      C.....
      C FORMAT STATEMENTS FOR SEAT ALONE INITIAL CONDITIONS
      C.....
      C VALIDATION REPORT
      C.....
      1500 FORMAT(5//),52X,"SEAT ALONE INITIAL CONDITIONS",51//)

```

```

1505 FORMAT(46X,"REFERENCE AREA (AREASA): ",F10.4,1X,A2,"**2"/.
+ 54X,"HEIGHT (HGTISA): ",F10.4,1X,A2,/,
+ 54X,"WEIGHT (WGTISA): ",F10.4,1X,A2,////)
1506 FORMAT(25X,"LOCATION OF SEAT BOTTOM (RCS)",
+ 24X,"ORIGIN OF RCS (RCS)",/
+ 38X,("A2,"),46X,("A2,"))//
1507 FORMAT(28X,"XPOSBOT: ",F10.4,31X,"YPOSSCS: ",F10.4,/,
+ 28X,"YPOSBOT: ",F10.4,31X,"YPOSSCS: ",F10.4,/,
+ 28X,"ZPOSBOT: ",F10.4,31X,"ZPOSSCS: ",F10.4,////)
1510 FORMAT(5X,"LOCATION OF SEAT REFERENCE POINT (SCS)",
+ 15X,"ORIENTATION (ACS)",22X,"LOCATION OF SEAT ALONE CG (SCS)",/
+ 20X,("A2,"),40X,(DEG),40X,("A2,"))//
1515 FORMAT(15X,"XPOSSRP: ",F10.4,22X,"ROLL (PHISA): ",F10.4,22X,
+ "XCGSA: ",F10.4)
1520 FORMAT(15X,"YPOSSRP: ",F10.4,22X,"PITCH (PSISA): ",F10.4,22X,
+ "YCGSA: ",F10.4)
1525 FORMAT(15X,"ZPOSSRP: ",F10.4,22X,"YAW (THESA): ",F10.4,22X,
+ "ZCGSA: ",F10.4,////)
1530 FORMAT(58X,"MOMENTS OF INERTIA",61X,("A4,"),A2,"**2")//,
+ 58X,"IXXSA: ",F10.4/.58X,"IYXSA: ",F10.4/,
+ 58X,"IXZSA: ",F10.4/.58X,"IYXSA: ",F10.4/,
+ 58X,"IYZSA: ",F10.4/.58X,"IZZSA: ",F10.4)
C.....
C FORMAT STATEMENTS FOR SEAT/OCCUPANT, OCCUPANT ALONE
C INITIAL CONDITIONS VALIDATION REPORT
C.....
1600 FORMAT(5(/),43X,"SEAT/OCCUPANT, OCCUPANT ALONE INITIAL",
+ " CONDITIONS",5(/))
1605 FORMAT(42X,"SEAT/OCCUPANT REFERENCE AREA (AREASO): ",
+ F10.4,1X,A2,"**2"/,
+ 36X,"OCCUPANT ALONE EFFECTIVE DRAG AREA (AREADA): ",
+ F10.4,1X,A2,"**2"/,
+ 50X,"SEAT/OCCUPANT WEIGHT (WGHTSD): ",
+ F10.4,1X,A2/,
+ 22X,"OCCUPANT ALONE WEIGHT BEFORE SEAT/OCC SEPARATION(WGHTOAB): ",
+ F10.4,1X,A2/,
+ 22X,"OCCUPANT ALONE WEIGHT AFTER SEAT/OCC SEPARATION(WGHTOAA): ",
+ F10.4,1X,A2)
1606 FORMAT(40X,"SEAT/OCCUPANT SEPARATION DOES NOT OCCUR"////)
1607 FORMAT(40X,"SEAT/OCCUPANT SEPARATION OCCURS ("
+ "SOSEP): ",F10.4," SEC AFTER INITIATION")
1608 FORMAT(37X,"SEAT/OCCUPANT SEPARATION OCCURS AT ("
+ "SOSEP): ",F10.4,1X,A2," PARACHUTE FORCE")
1609 FORMAT(40X,"AERODYNAMIC DAMPING COEFFICIENT (DMPGC): ",
+ F10.4,////)
1610 FORMAT(44X,"LOCATION OF SEAT/OCCUPANT C.G. (SCS)",/
+ 59X,("A2,"),53X,"XCGSO: ",F10.4/,
+ 53X,"YCGSO: ",F10.4/.53X,"ZCGSO: ",F10.4,/)
1615 FORMAT(45X,"SEAT/OCCUPANT MOMENTS OF INERTIA"/,
+ 54X,("A4,"),A2,"**2")//
1620 FORMAT(53X,"IXXSO: ",F10.4/,
+ 53X,"IYXSO: ",F10.4/,
+ 53X,"IXZSO: ",F10.4/,
+ 53X,"IYXSO: ",F10.4/,
+ 53X,"IYZSO: ",F10.4/,
+ 53X,"IZZSO: ",F10.4)
C.....

```

```

800      C FORMAT STATEMENTS FOR RAIL INITIAL CONDITIONS VALIDATION REPORT
801      C.....
802      1700 FORMAT(5(/),52X,"RAIL INITIAL CONDITIONS",3(/))
803      1710 FORMAT(45X,"RAIL LENGTH (RAILNTH) ",F12.4,1X,A2,/,
804      + 33X,"RAIL ANGLE WRT AIRCRAFT (RAILANG): ",F12.4," DEG",/,
805      + 32X,"X-DIRECTION SPRING CONSTANT(KXSB): ",F12.4,1X,A2,"/",
806      + A2,/,
807      + 32X,"Y-DIRECTION SPRING CONSTANT(KYSB): ",F12.4,1X,A2,"/",
808      + A2,/,
809      + 36X,"COEFFICIENT OF FRICTION(MUSB): ",F12.4,/,
810      + 33X,"TORSIONAL SPRING CONSTANT(YKTOR): ",F12.4,1X,A2,"-",A2,
811      + "/DEG")
812
813      1715 FORMAT (2(/),42X,"LOCATION OF RAIL ATTACHMENT POINTS (ACS)",/
814      1716 FORMAT(25X,"XPOSRR: ",F10.4,30X,"XPOSRLRE: ",F10.4,/,
815      25X,"YPOSRR: ",F10.4,30X,"YPOSRLRE: ",F10.4,/,
816      25X,"ZPOSRR: ",F10.4,30X,"ZPOSRLRE: ",F10.4,/)
817
818      1711 FORMAT (10X,"NUMBER OF SLIDER BLOCKS ON AIRCRAFT: ",I3////)
819      1712 FORMAT(45X,"NUMBER OF SLIPPERS ON SEAT: ",I3////)
820      1713 FORMAT(40X,"10H(PROGRAM A, 10H(SUMS A C, 10H(NTINUOUS
821      + 10H"RAIL WITH, 10HIN A RAIL", 1H))
822      1720 FORMAT(5X,"LOCATION OF SLIDER BLOCK 1 ("A2,"), 10X,
823      + "LOCATION OF SLIDER BLOCK 2 ("A2,"), 10X,
824      + "LOCATION OF SLIDER BLOCK 3 ("A2,"),/
825      + 21X,"(SCS)",2(36X,"(SCS)"),//)
826
827      1721 FORMAT(10X,"LOCATION OF SLIPPER 1 ("A2,"), 15X,
828      + "LOCATION OF SLIPPER 2 ("A2,"), 15X,
829      + "LOCATION OF SLIPPER 3 ("A2,"),/
830      + 21X,"(SCS)",2(36X,"(SCS)"),//)
831
832      1730 FORMAT(15X,"XPOSSB1: ",F10.4,26X,"XPOSSB2: ",F10.4,
833      + 26X,"XPOSSB3: ",F10.4,/,
834      + 15X,"YPOSSB1: ",F10.4,26X,"YPOSSB2: ",F10.4,
835      + 26X,"YPOSSB3: ",F10.4,/,
836      + 15X,"ZPOSSB1: ",F10.4,26X,"ZPOSSB2: ",F10.4,26X,"ZPOSSB3
837      + F10.4//)
838
839      1740 FORMAT(15X,"LOCATION OF SLIDER BLOCK 4 ("A2,"),
840      + 10X,"LOCATION OF SLIDER BLOCK 5 ("A2,"),
841      + 10X,"LOCATION OF SLIDER BLOCK 6 ("A2,"),/
842      + 21X,"(SCS)",2(36X,"(SCS)"),//)
843
844      1741 FORMAT(10X,"LOCATION OF SLIPPER 4 ("A2,"),
845      + 15X,"LOCATION OF SLIPPER 5 ("A2,"),
846      + 10X,"LOCATION OF SLIPPER 6 ("A2,"),/
847      + 21X,"(SCS)",2(36X,"(SCS)"),//)
848
849      1750 FORMAT(15X,"XPOSSB4: ",F10.4,26X,"XPOSSB5: ",F10.4
850      + 26X,"XPOSSB6: ",F10.4,/,
851      + 15X,"YPOSSB4: ",F10.4,26X,"YPOSSB5: ",F10.4,
852      + 26X,"YPOSSB6: ",F10.4,/,
853      + 15X,"ZPOSSB4: ",F10.4,26X,"ZPOSSB5: ",F10.4,
854      + 26X,"ZPOSSB6: ",F10.4)
855
856      C.....
857      C FORMAT STATEMENTS FOR CATAPULT PARAMETERS VALIDATION REPORT
858      C.....
859      1900 FORMAT(5(/),56X,"CATAPULT PARAMETERS",4(/))
860      1905 FORMAT(5X,"NUMBER OF CATAPULTS: ",I3,
861      + 90X,"LOCATION OF CATAPULT"/,89X,"ATTACHMENT POINTS (SCS)",/
862      + 98X,"(A2,"A2,"),")
863      1910 FORMAT(/,10X,"CATAPULT ",I1,"",9X,"LENGHT (CATLNT) ",

```

```

      + F9.4,1X,A2,25X,"XPOSAP:",F9.4,1X,A2)
1920 FORMAT(30X,"STROKE (CATSIK):",F9.4,1X,A2,25X,"YPOSAP:",
      + F9.4,1X)
1930 FORMAT(30X,"TIME OF IGNITION (TCI):",F9.4," SEC",24X,"ZPOSAP:",
      + F9.4,1X)
1935 FORMAT(54X,"TUBE BENDING PARAMETERS")
1940 FORMAT (38X,"K TUBE SPRING STIFFNESS CONSTANT (KTUBE):",
      + F10.4,1X,A2,"/",A2/,
      + 38X,"C TUBE SPRING DAMPING COEFFICIENT (CTUBE):",
      + F10.4,1X,A2,"-SEC/",A2/,
      + 40X,"EMPIRICAL TUBE BENDING CONSTANT (PTUBE):",F10.4/,
      + 48X,"COEFFICIENT OF FRICTION (MUTUBE):",F10.4/,
      + 33X,"RESTORING FORCE STIFFNESS COEFFICIENT (EXTLNGT):",
      + F10.4,1X,A2,"/",A2)
1975 FORMAT(100,53X,"CATAPULT 1 THRUST TABLE")
      + 52X,"TIME",20X,"THRUST"/,52X,"(SEC)",21X,"(",A2,")"/)
1980 FORMAT(47X,F10.4,15X,F10.4)
1985 FORMAT(100,15X,"CATAPULT 1 THRUST TABLE",T82,
      + "CATAPULT 2 THRUST TABLE"/T15,"TIME",20X,
      + "THRUST",T81,"TIME",20X,"THRUST"/,T15,
      + "(SEC)",20X,"(",A2,")",T81,
      + "(SEC)",20X,"(",A2,")"/)
1990 FORMAT(10X,F10.4,15X,F10.4,177,F10.4,15X,F10.4)
C.....
C FORMAT STATEMENTS FOR ROCKET PARAMETERS VALIDATION REPORT
C.....
11000 FORMAT(5//),57X,"ROCKET PARAMETERS",4(//)
11010 FORMAT(5X,"NUMBER OF ROCKETS:",11//)
      + 104X,"ROCKET THRUST LINE"/,65X,"ROCKET NOZZLE LOCATION (SCS)",
      + 6X,"DIRECTION COSINE ANGLES (SCS)/,77X,"(",A2,")",30X,"(DEG)",
11020 FORMAT(/,10X,"ROCKET",1X,11,"",6X,"IGNITION (RKIGN):",
      + F9.4,1X,A2,12X,"XPOSRK:",F9.4,4X,14X,"RKALPH:",
      + F9.4)
11030 FORMAT(26X,"FUEL WT (RKWGT):",F9.4,1X,A2,12X,"YPOSRK:",F9.4,
      + 4X,14X,"RKBETA:",F9.4)
11040 FORMAT(26X,"BURN TIME (RKBURN):",F9.4," SEC",11X,"ZPOSRK:",F9.4,
      + 4X,14X,"RKGAMA:",F9.4)
11050 FORMAT(26X,"IGN DELAY (RKDELY):",F9.4," SEC")
11070 FORMAT(5//),52X,"ROCKET THRUST TABLES"/)
      + 58X,"ROCKET",11//)
      + 40X,"NOMINAL",28X,"ADJUSTED"/,
      + 35X,"TIME",9X,"THRUST",16X,"TIME",9X,"THRUST"/,
      + 35X,"(SEC)",9X,"(",A2,")",17X,"(SEC)",9X,"(",A2,")"/)
11075 FORMAT(30X,F10.4,5X,F10.4,10X,F10.4,5X,F10.4)
C.....
C FORMAT STATEMENTS FOR DART PARAMETERS VALIDATION REPORT
C.....
12000 FORMAT(5//),56X,"DART PARAMETERS",6(//)
12010 FORMAT(48X,"DART FORCE (DRTFRC):",F10.4,1X,A2,/
      + 39X,"DART START DISTANCE (DRTSIRT):",F10.4,1X,A2,/
      + 40X,"DART STOP DISTANCE (DRTSTOP):",F10.4,1X,A2,////)
12020 FORMAT(5X,"LEFT DART CONFLUENCE POINT (SCS)",41X,"RIGHT DART",
      + "CONFLUENCE POINT (SCS)/,19X,"(",A2,")",67X,"(",A2,")"/)
12030 FORMAT(10X,"XDRTCPL:",F10.4,54X,
      + "XDRTCPR:",F10.4)
12035 FORMAT(10X,"YDRTCPL:",F10.4,54X,
      + "YDTCPR:",F10.4)

```

```

12040 FORMAT(10X,"ZDRTCP: ",F10.4,.54X,
+ "ZDRTCP: ",F10.4,////)
12045 FORMAT(1X,"LEFT DART COCKPIT ATTACHMENT POINT (SCS)",31X,
+ "RIGHT DART COCKPIT ATTACHMENT POINT (SCS)",/
+ 19X,"( ",A2," )",.67X,"( ",A2," )",/
12050 FORMAT(10X,"XDRTP: ",F10.4,.54X,
+ "XDRTP: ",F10.4)
12055 FORMAT(10X,"YDRTP: ",F10.4,.54X,
+ "YDRTP: ",F10.4)
12060 FORMAT(10X,"ZDRTP: ",F10.4,.54X,
+ "ZDRTP: ",F10.4,////)
C .....
C FORMAT STATEMENTS FOR THRUST VECTOR CONTROL DATA VALIDATION REPORT *
C .....
13000 FORMAT(5(//),51X,"THRUST VECTOR CONTROL DATA"6(//))
13010 FORMAT(48X,"TVC COORDINATE SYSTEM DEFINITION"/
+ 58X,"(WRT SCS)",.54X,"MPHI: ",F10.4," DEG",.54X,"MPSI: ",F10.4,
+ " DEG",.54X,"MTH: ",F10.4," DEG",////)
13015 FORMAT(33X,"TVC ROLL CONTROL ROCKET GIMBALLING LIMIT (ROLLRL): ",
+ F10.4," DEG"/)
13020 FORMAT(31X,"TVC PITCH CONTROL ROCKET GIMBALLING LIMIT ",
+ "(PITCHRL): ",F10.4," DEG"/)
13025 FORMAT(41X,"MAXIMUM GYROSCOPE SAMPLING RATE (SMPLRAT): ",F10.4,
+ " DEG/SEC"/)
13030 FORMAT(32X,"TIME DELAY FOR TVC AFTER ROCKET IGNITION (TVCCLAY): ",
+ F10.4," SEC"/)
13035 FORMAT(49X,"ROCKET ANGLE WRT SEAT BACK (RKANG): ",F10.4," DEG")
C .....
C FORMAT STATEMENTS FOR DYNAMIC CG VARIANCES VALIDATION REPORT *
C .....
14000 FORMAT(///,95X,"DYNAMIC CG VARIABLES",4(//))
14002 FORMAT(50X,"DEFAULT VALUES USED: ",////)
14003 FORMAT(51X,"INPUT VALUES USED: ",////)
14005 FORMAT(42X,"DAMPING CONSTANT (CX): ",F10.4/)
14010 FORMAT(42X,"DAMPING CONSTANT (CY): ",F10.4/)
14015 FORMAT(42X,"DAMPING CONSTANT (CZ): ",F10.4/)
14020 FORMAT(32X,"X SPRING MODULUS CONSTANT (SXP): ",F10.4,1X,A2,"/",
+ A2/)
14025 FORMAT(32X,"Y SPRING MODULUS CONSTANT (SYN): ",F10.4,1X,A2,"/",
+ A2/)
14030 FORMAT(33X,"Z SPRING MODULUS CONSTANT (SZ): ",F10.4,1X,A2,"/",A2/)
14035 FORMAT(31X,"Z SPRING MODULUS CONSTANT (SZN1): ",F10.4,1X,A2,"/",
+ A2/)
14040 FORMAT(31X,"Z SPRING MODULUS CONSTANT (SZN2): ",F10.4,1X,A2,"/",
+ A2/)
14045 FORMAT(33X,"X DIRECTION DEAD ZONE (XSLACK): ",F10.4,1X,A2/)
14050 FORMAT(33X,"Z DIRECTION DEAD ZONE (ZSLACK): ",F10.4,1X,A2/)
14055 FORMAT(30X,"Z DIRECTION BOTTOMING ZONE (ZBOT): ",F10.4,1X,A2)
C .....
C FORMAT STATEMENTS FOR DROGUE CHUTE VALIDATION REPORT *
C .....
15000 FORMAT(5(//),49X,"DROGUE CHUTE(S) PARAMETERS",4(//))
15005 FORMAT(25X,"DROGUE TYPE",33X,"(DROGUE = 1) :",.8X,
+ "STANDARD SINGLE DROGUE")
15006 FORMAT(25X,"DROGUE TYPE",33X,"(DROGUE = 2) :",.8X,
+ "VELCON DOUBLE DROGUE")
15010 FORMAT(25X,"DROGUE CONTAINER/SLUG REFERENCE AREA",13X,"(AREADC) "

```

```

970      + .F14.4,1X,A2)
15015 FORMAT(25X,"DROGUE CONTAINER/SLUG WEIGHT",21X,"(WGHTDC) ",F14.4,
+ 1X,A2)
15020 FORMAT(25X,"DROGUE CONTAINER/SLUG DRAG COEFFICIENT",13X,"(CDDC) ",
+ .F14.4)
15025 FORMAT(25X,"DROGUE CHUTE DRAG COEFFICIENT",19X,"(DRDRAG1) ",
+ .F14.4)
15030 FORMAT(25X,"DROGUE CHUTE EFFECTIVE POROSITY",17X,"(POROSD1) ",
+ .F14.4)
15035 FORMAT(25X,"DROGUE CONTAINER/SLUG RELEASE DELAY ",10X,
+ "(TDDPLY) ",F14.4," SEC")
15040 FORMAT(25X,"SEAT TRAVEL TO DROGUE CONTAINER/SLUG RELEASE ",
+ "(DISPLY) ",F14.4,1X,A2)
15045 FORMAT(25X,"DROGUE CHUTE LINE LENGTH",22X," (DROGLL) ",F14.4,
+ 1X,A2)
15050 FORMAT(25X,"DROGUE CHUTE PROJECTED DIAMETER",17X,"(DROGPD1) ",
+ F14.4,1X,A2)
15075 FORMAT(//,44X,"DROGUE CHUTE ATTACHMENT POINT (SCS)"/,60X,"(",
+ A2,"")"/,50X,"XDRGAP: ",F10.4/,50X,"YDRGAP: ",F10.4,
+ /,50X,"ZDRGAP: ",F10.4)
15080 FORMAT(//,38X,"DROGUE CONTAINER/SLUG PROJECTION VELOCITY (SCS)"/,
+ 57X,"(",A2,"/SEC)"/,
+ 50X,"DROVELX: ",F10.4/,50X,"DROVELY: ",F10.4,
+ /,50X,"DROVELZ: ",F10.4)
15085 FORMAT(//,44X,"SECOND VELCON DROGUE CHUTE PARAMETERS"/,25X,
+ "VELCON CHUTE DRAG COEFFICIENT",16X,"(DRDRAG2) ",F14.4/,25X,
+ "VELCON CHUTE EFFECTIVE POROSITY",14X,"(POROSD2) ",F14.4/,25X,
+ "VELCON CHUTE PROJECTED DIAMETER",14X,"(DROGPD2) ",F14.4,1X,
+ A2/,25X,"VELCON CHUTE SEPARATION VELOCITY",14X,"(VELCON) ",
+ F14.4,1X,A2,"/SEC")
15500 FORMAT(51//,56X,"TABLE OF TIMES FROM")
15505 FORMAT(50X,"DROGUE CONTAINER/SLUG DEPLOYMENT"/,52X,
+ "TO DROGUE CHUTE LINE STRETCH"/,62X,"(DROGLS)")
15510 FORMAT(//,10X,"VELOCITY AT DROGUE DEPLOYMENT ",A9,"",8F10.1)
15512 FORMAT(10X,"TIME TO DROGUE LINE STRETCH",6X,"(SEC) ",8F10.4)
15520 FORMAT(53X,"DROGUE CHUTE LINE STRETCH"/,57X,"TO FULL ",
+ "INFLATION"/,61X,"(DROGFT1)")
15530 FORMAT(50X,"FIRST DROGUE CHUTE LINE STRETCH"/,57X,
+ "TO FULL INFLATION"/,61X,"(DROGFT1)")
15540 FORMAT(50X,"SECOND DROGUE CHUTE LINE STRETCH"/,57X,
+ "TO FULL INFLATION"/,61X,"(DROGFT2)")
C*****
C FMT STATEMENTS FOR RECOVERY CHUTE VALIDATION REPORT
C*****
16000 FORMAT(21//,50X,"RECOVERY PARACHUTE PARAMETERS",21//)
16003 FORMAT(40X,"PARACHUTE RELEASE TIME (TROPLOY) ",F10.4," SEC")
16004 FORMAT(40X,"PARACHUTE RELEASE TIME (TROPLOY) : SEE TABLE BELOW")
16005 FORMAT(40X,"UPPER ALTITUDE DELAY (TDELAY) ",F10.4," SEC")
16006 FORMAT(40X,"G-LIMIT VALUE (GLIMIT) ",F10.4," G/S")
16007 FORMAT(40X,"TYPE OF CONTROL (IRECOV = 1): FIXED TIME DELAY")
16008 FORMAT(40X,"TYPE OF CONTROL (IRECOV = 2): GLIMIT EXTENDED",
+ " DELAY")
16010 FORMAT(
+ 40X,"LINE LENGTH",164,"(RECOVLL) ",F10.4,1X,A2,/,
+ 40X,"DRAG COEFFICIENT (RECDRAG) ",F10.4,/,
+ 40X,"PROJECTED DIAMETER (RECOVPD) ",F10.4,1X,A2,/,
+ 40X,"EFFECTIVE POROSITY (POROSR) ",F10.4,/.

```

PAGE 205

83/11/07 09.41 53

FIN 4.6+428

SUBROUTINE REPT11 74/74 OPT=1

```

1030      + 40X,"LOWER ALTITUDE LIMIT (CHALT1): ",F10.4,1X,A2,/,
      + 40X,"UPPER ALTITUDE LIMIT (CHALT2) ",F10.4,1X,A2,/,
16020  FORMAT(1H,2(//),46X,"RECOVERY CHUTE ATTACHMENT POINT (SCS)"/,
      + 61X,"(",A2,")"/,
      + 53X,"XRECAP:",1X,F10.4,/,
      + 53X,"YRECAP:",1X,F10.4,/,
      + 53X,"ZRECAP:",1X,F10.4)
16400  FORMAT(//,12X,2(16X,"TABLE OF TIMES FROM RECOVERY")/,
      + 30X,"PARACHUTE RELEASE TO LINE",16X,"PARACHUTE LINE STRETCH",
      + " TO FULL"/,
      + 39X,"STRETCH",35X,"INFLATION"/,12X,2(20X,"VELOCITY",9X,"TIME"),
      + //,12X,2(20X,"(",A2,")/SEC)",8X,"(SEC)")
16410  FORMAT(12X,2(20X,F7.1,4X,F10.4))
16500  FORMAT(//,12X,"TABLE OF TIMES FOR RECOVERY",13X,
      + "TABLE OF TIMES FROM RECOVERY",12X,"TABLE OF TIMES ",
      + "FROM RECOVERY",18X,"CHUTE DEPLOYMENT",22X,
      + "CHUTE RELEASE TO LINE",16X,"CHUTE LINE STRETCH TO FULL"/,
      + 63X,"STRETCH",31X,"INFLATION"/,3(14X,"VELOCITY",9X,"TIME",
      + 4X,)/,3(14X,"(",A2,")/SEC)",9X,"(SEC)",3X))
16510  FORMAT(3(10X,2(3X,F10.4),3X))
9000  RETURN
      END

```


5-207

```

60      IF(IEVENTS(28) .NE. 0) GO TO 9900
      LU=BIAS+2
      IF (PRTFRQ .LE. 0) GO TO 100
      IPRCNT(2) = MOD (IPRCNT(2)+1, PRTFRQ)
      IF(IEVLINE .NE. 0) GO TO 100
      IF(IPRCNT(2) .NE. 0) GO TO 9900
      100 CONTINUE
      IF(LINECT(2) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE(LU,7000) PRTINGT(PRTINDX) , PRTINGT(PRTINDX)
      IF(PRTFRQ .LT. 0) GOTO 9900
      200 CONTINUE
      XACC(1) = XACCEL(1)/GRAVITY
      YACC(1) = YACCEL(1)/GRAVITY
      ZACC(1) = ZACCEL(1)/GRAVITY
      ACCR(1) = SORT(XACC(1)+YACC(1)+YACC(1)+ZACC(1)+ZACC(1))
      VELR(1) = SORT(TRAJSD(14)+TRAJSD(14) + TRAJSD(15)+TRAJSD(15)
      + + TRAJSD(16)+TRAJSD(16))
      RPOS(1) = SORT(TRAJSD( 2)+TRAJSD( 2) + TRAJSD( 3)+TRAJSD( 3)
      + + TRAJSD( 4)+TRAJSD( 4))
      WRITE(LU,7010) TIME , XACC(1) , YACC(1) , ZACC(1) , ACCR(1),
      + TRAJSD(14) , TRAJSD(15) , TRAJSD(16) , VELR(1),
      + TRAJSD( 2) , TRAJSD( 3) , TRAJSD( 4) , RPOS(1)
      LINECT(2) = LINECT(2) + 1
      9900 CONTINUE
      RETURN
      C.....
      C WRITE FORMAT STATEMENTS
      C.....
      7000 FORMAT(/,BX,"TIME",17X,"ACCELERATION (SCS)",22X,"VELOCITY(EFCS)",
      + 29X,"POSITION (EFCS)",5X,"(SEC)",22X,"(GIS)",32X,"(",
      + A2,"/SEC)",38X,"(",A2,")",/,
      + 17X,3(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),/)
      7010 FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
      END

```



```

60      + PVL(3) , QVL(3) , RVL(3) , VELR(3) ,
      + FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3) ,
      + RXACC(3) , RYACC(3) , RZACC(3) , RLACR(3) ,
      + RXVEL(3) , RYVEL(3) , RZVEL(3) , RLVL(3) ,
      + RXPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3) ,
      + RPVEL(2) , RPOVEL(2) , RRVEL(2) ,
      + RPPOS(2) , ROPOS(2) , RRPOS(2)
65      IF(IEVENTS(28) .NE. 0) GO TO 9900
      LU=BIAS+3
      IF (PRTRFR .LE. 0) GO TO 100
      IPRCNT(3) = MOD (IPRCNT(3)+1, PRTRFR)
      IF(IEVLIN .NE. 0) GO TO 100
      IF(IPRCNT(3) .NE. 0) GO TO 9900
70      100 CONTINUE
      IF(LINECT(3) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE (LU,7000)
75      IF(PRTRFR .LT. 0) GOTO 9900
      200 CONTINUE
      FROLL(1) = ZARTAN(DCMSE(2,3),DCMSE(3,3)) * RADDEG
      FPITCH(1) = -ASIN( DCMSE(1,3)) * RADDEG
      FYAW(1) = ZARTAN(DCMSE(1,2),DCMSE(1,1)) * RADDEG
      PACC = TRAJSO(23) * RADDEG
      QACC = TRAJSO(24) * RADDEG
      RACC = TRAJSO(25) * RADDEG
      PVL(1) = TRAJSO(11) * RADDEG
      QVL(1) = TRAJSO(12) * RADDEG
      RVL(1) = TRAJSO(13) * RADDEG
85      RESULT1 = SORT(PACC+PACC + QACC+QACC + RACC+RACC)
      RESULT2 = SORT(PVL(1)+PVL(1)+QVL(1)+QVL(1)+RVL(1))
      RESULT3 = SORT(FROLL(1)+FROLL(1)+FPITCH(1)+FPITCH(1)+FYAW(1)
      + FYAW(1))
      WRITE(LU,7010) TIME
      PACC , QACC , RACC , RESULT1,
      + PVL(1) , QVL(1) , RVL(1) , RESULT2,
      + FROLL(1) , FPITCH(1) , FYAW(1) , RESULT3
90      LINECT(3) = LINECT(3) + 1
      9900 CONTINUE
      RETURN
      C*****
      C WRITE FORMAT STATEMENTS
      C*****
100      7000 FORMAT(/,5X,"TIME",17X,"ACCELERATION (SCS)",24X,"RATE (EFC)",
      + 28X,"ORIENTATION (EFC)",5X,"(SEC)",18X,"(DEG/SEC 2)",
      + 30X,"(DEG/SEC)",36X,"(DEG)",/,
      + 17X,2(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),
      + 3X,"ROLL",5X,"PITCH",6X,"YAW",6X,"RES"/)
105      7010 FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
      END

```

```

1 SUBROUTINE REPR4
C .....
C
C OCCUPANT ALONE LINEAR TIME HISTORY REPORT
C .....
5 C CONSTANTS COMMON BLOCK
C .....
C
C COMMON /CONSTANT / GRAVITY , RADEG , DEGRAD , PI
C .....
C
C SECTION 2 COMMON BLOCK
C .....
10 C COMMON /IREPORT / IREPTS(31) , PRTRFQ,P11,P12,P13
C
C INTEGER
C .....
C
C MISCELLANEOUS DATA COMMON BLOCK
C .....
15 C
C COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRTCNT(31)
C
C + MAXLINE , MAXREPT , MAXEVRT
C + IEVLNE , TERRFLG , LU
C + IDEATE , HEADALT , HEADVEL
C + HEADSR , HEADYAW , HEADPIT
C + HEADROL , HEADWGT , BIAS
C + REPTYPE(5,31) , PRTWGT(2)
C + IHEADER(24) , TEVENTS(38) , TIMES(38)
C + IMVDC , PRTEMP( 2)
C + PRTMASS(2) , PRTINDX
C + ZVECT(3) , XYZ(3) , SAVTIME
C + XACCEL(3) , YACCEL(3)
C + REPTYPE , BIAS , PRTLNGT
C + PRTWGT , PRTMASS , PRTINDX
C + PRTEMP
C .....
C
C INTEGRATION ROUTINE COMMON BLOCK
C .....
30 C
C COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSOI(193)
C
C + TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3)
C + TRAJAC(193) , TVCEOS(225) , QUATSOI(65)
C + QUATSA(65) , QUATOA(65) , QUATAC(65)
C + INTSTP , IPCPASS , IRKPASS
C + IPOINTS , IYX , IYPRX
C + IKX , IKSUMX , IKPASSX
C + IYIX , IYIIX , IYI2X
C + IYI3X , IYPRIX , IYPR1IX
C + IYPR2X , IYPR1X , IYPR11X
C + IYI1X , IYI11X , IYI11X , IYI11X , IYI11X
C .....
C
C PLOT FILE VARIABLES COMMON BLOCK
C .....
35 C
C COMMON /PLOT / XACC(3) , YACC(3) , ZACC(3) , ACCR(3)
C
C + PVL(3) , QVL(3) , RVL(3) , VELR(3)
C + FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3)
C + RXACC(3) , RYACC(3) , RZACC(3) , RLACR(3)
C + RXVEL(3) , RYVEL(3) , RZVEL(3) , RLVL(3)
C + RXPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3)
C + RPVEL(2) , RQVEL(2) , RRVEL(2)
C + RPPOS(2) , RQPOS(2) , RRPOS(2)
C .....

```

SUBROUTINE REPR4 74/74 OPT=1 FIN 4 6+428 83/11/07 09 41 53 PAGE 211

```

60      LU=BIAS + 4
      IF (PRTRQ .LT. 0) GO TO 100
      IF (IEVENTS(28) .EQ. 0) GO TO 9900
      IF (PRTRQ .EQ. 0) GO TO 100
      IPRCNT(4) = MOD (IPRCNT(4)+1, PRTRQ)
      IF (IEVLIN .NE. 0) GO TO 100
      IF (IPRCNT(4) .NE. 0) GO TO 9900
65      100 CONTINUE
      IF (LINECT(4) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE(LU,7000) PRTLNCT(PRTINDX) , PRTLNCT(PRTINDX)
      IF (PRTRQ .LT. 0) GO TO 9900
70      200 CONTINUE
      XACC(2) = XACCEL(2)/GRAVITY
      YACC(2) = YACCEL(2)/GRAVITY
      ZACC(2) = ZACCEL(2)/GRAVITY
      ACCR(2)=SORT( XACC(2)+XACC(2)+YACC(2)+YACC(2)+ZACC(2)+ZACC(2))
      VELR(2)=SORT(TRAJDA(14)+TRAJDA(14)+TRAJDA(15)+TRAJDA(15)+
      + TRAJDA(16)+TRAJDA(16))
      RPOS(2)=SORT(TRAJDA( 2)+TRAJDA( 2)+TRAJDA( 3)+TRAJDA( 3)+
      + TRAJDA( 4)+TRAJDA( 4))
      WRITE(LU,7010) TIME
      + XACC(2) , YACC(2) , ZACC(2) , ACCR(2),
      + TRAJDA(14), TRAJDA(15), TRAJDA(16), VELR(2),
      + TRAJDA( 2), TRAJDA( 3), TRAJDA( 4), RPOS(2)
      LINECT(4) = LINECT(4) + 1
      9900 CONTINUE
      RETURN
85      C*****
      C WRITE FORMAT STATEMENTS
      C*****
      7000 FORMAT(/5X,"TIME",17X,"ACCELERATION (SCS)",22X,"VELOCITY(EFCS)",
      + 29X,"POSITION (EFCS)",/5X,"(SEC)",22X,"(G[S]",32X,"((",
      + A2,"/SEC)",38X,"(",A2,")"/,
      + 17X,3(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),/)
      7010 FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
      END

```

```

1 SUBROUTINE REPRTS
C .....
C
5 C SEAT ALONE LINEAR TIME HISTORY REPORT
C .....
C
C CONSTANTS COMMON BLOCK
C .....
C
C COMMON /CONSTANT / GRAVITY , RADDEG , DEGRAD , PI
C .....
10 C SECTION 2 COMMON BLOCK
C .....
C
C COMMON /IREPORT / IREPTS(31) , PRIFRO,P11,P12,P13
C .....
15 C INTEGER
C .....
C
C MATRIX COMMON BLOCK
C .....
C
C COMMON /MATRIX / DCMAE(3,3) , DCNRA(3,3) , DCMSAI(3,3) ,
C .....
C + DCMSE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,
C .....
20 C + DCMSAE(3,3) , DCMDAE(3,3) , DCMSRI(3,3) ,
C .....
C + DCMDSM(3,3)
C .....
C
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C
C COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31)
C .....
C + MAXLINE , MAXREPT , MAXEVT ,
C .....
25 C + IEVLNE , IERRFLG , LU ,
C .....
C + IDATE , HEADALT , HEADVEL ,
C .....
C + HEADSR , HEADYAW , HEADPIT ,
C .....
C + HEADROL , HEADWGT , BIAS ,
C .....
30 C + REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
C .....
C + IHEADER(24) , IEVENTS(38) , TIMES(38) ,
C .....
C + IMVDC , PRTMP( 2) ,
C .....
C + PRIMASS(2) , PRINDX , PKZVEL ,
C .....
35 C + ZVCT(3) , XYZ(3) , SAVTIME ,
C .....
C + XACCEL(3) , YACCEL(3) ,
C .....
C + REPTYPE , BIAS , PRTLNGT ,
C .....
C + PRTWGT , PRTEMP ,
C .....
40 C + PRIMASS , PRINDX ,
C .....
C
C INTEGRATION ROUTINE COMMON BLOCK
C .....
C
C COMMON /RKUTTA / TIME , TIME5 , DELTAT , TRAJSO(193)
C .....
C + TRAJSA(193) , TRAJOA(193) , TRAJCHI(97,3) ,
C .....
45 C + TRAJAC(193) , TVEOS(225) , TRAJSO(65) ,
C .....
C + QUATOA(65) , QUATAC(65) ,
C .....
C + INSTP , IPCPASS ,
C .....
C + IPPOINTS , IYX , IRKPASS ,
C .....
50 C + IKX , IKSUMX , IKPASSX ,
C .....
C + IYX , IYIX , IY12X ,
C .....
C + IY13X , IYPR1X , IYPR1X ,
C .....
C + IYPR12X , IYIX , IY11X ,
C .....
C + ICYIX , ICYIX , IY11X , IY11X ,
C .....
C
C PLOT FILE VARIABLES COMMON BLOCK
C .....
C
55 C COMMON /PLOT / XACC(3) , YACC(3) , ZACC(3) , ACCR(3) ,
C .....

```

```

60      + PVL(3) , QVL(3) , RVL(3) , VELR(3) ,
      + FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3) ,
      + RXACC(3) , RYACC(3) , RZACC(3) , RLACR(3) ,
      + RXVEL(3) , RYVEL(3) , RZVEL(3) , RLVLRI(3) ,
      + RXPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3) ,
      + RPVEL(2) , ROVEL(2) , RRVEL(2) ,
      + RPPS(2) , RQPOS(2) , RRPOS(2)
      LU = BIAS + 5
      IF (IEVENTS(31) .NE. 0) GO TO 9900
      IF (PRIFRQ .LT. 0) GO TO 100
      IF (IEVENTS(28) .EQ. 0) GO TO 9900
      IF (PRIFRQ .EQ. 0) GO TO 100
      IPRCNT(5) = MOD (IPRCNT(5)+1, PRIFRQ)
      IF (IEVLIN .NE. 0) GO TO 100
      IF (IPRCNT(5) .NE. 0) GO TO 9900
100    CONTINUE
      IF (LINECT(5) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE(LU,7000) PRTLNCT(PRTINDX) , PRTLNCT(PRTINDX)
      IF (PRIFRQ .LT. 0) GO TO 9900
200    CONTINUE
      XACC(3) = XACCEL(3)/GRAVITY
      YACC(3) = YACCEL(3)/GRAVITY
      ZACC(3) = ZACCEL(3)/GRAVITY
      ACCR(3) = SORT(XACC(3)+YACC(3)+ZACC(3)+ZACC(3))
      VELR(3) = SORT(TRAUSA(14)+TRAUSA(14)+TRAUSA(15)+TRAUSA(15) +
      + TRAUSA(16)+TRAUSA(16))
      RPOS(3) = SORT(TRAUSA(2)+TRAUSA(2)+TRAUSA(3)+TRAUSA(3) +
      + TRAUSA(4)+TRAUSA(4))
      WRITE(LU,7010) TIME
      + XACC(3) , YACC(3) , ZACC(3) , ACCR(3) ,
      + TRAUSA(14) , TRAUSA(15) , TRAUSA(16) , VELR(3) ,
      + TRAUSA(2) , TRAUSA(3) , TRAUSA(4) , RPOS(3)
      LINECT(5) = LINECT(5) + 1
9900    CONTINUE
      RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....
7000    FORMAT(1X,"TIME",17X,"ACCELERATION (GCS)",22X,"VELOCITY(EFCS)",
      + 29X,"POSITION (EFCS)",15X,"(SEC)",22X,"(G(S))",32X,
      + "(",A2,"/SEC)",38X,"(",A2,")",
      + 17X,3(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),/)
7010    FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
      END

```


5-215

```

60      + PVL(3) , QVL(3) , RVL(3) , VELR(3) ,
      + FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3) ,
      + RXACC(3) , RVACC(3) , RZACC(3) , RLACR(3) ,
      + RXVEL(3) , RYVEL(3) , RZVEL(3) , RLVL(3) ,
      + RXPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3) ,
      + RPVEL(2) , ROVEL(2) , RRVEL(2) ,
      + RPPOS(2) , RQPOS(2) , RRPOS(2)
65      LU = BIAS + 6
      IF (IEVENTS(31) .NE. 0) GO TO 9900
      IF (PRTRQ .LT. 0) GO TO 100
      IF (IEVENTS(28) .EQ. 0) GO TO 9900
      IF (PRTRQ .EQ. 0) GO TO 100
      IPRCNT(6) = MOD (IPRCNT(6)+1, PRTRQ)
      IF (IEVLN .NE. 0) GO TO 100
      IF (IPRCNT(6) .NE. 0) GO TO 9900
100     CONTINUE
      IF (LINECT(6) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE(LU,7000)
      IF (PRTRQ .LT. 0) GO TO 9900
200     CONTINUE
      FROLL(3) = (ZARCtan(DCMSAE(2,3),DCMSAE(3,3))) * RADDEG
      FPITCH(3) = -ASIN( DCMSAE(1,3)) * RADDEG
      FYAW(3) = ZARCtan(DCMSAE(1,2),DCMSAE(1,1)) * RADDEG
      PACC = TRAJSA(23) * RADDEG
      QACC = TRAJSA(24) * RADDEG
      RACC = TRAJSA(25) * RADDEG
      PVL(3) = TRAJSA(11) * RADDEG
      QVL(3) = TRAJSA(12) * RADDEG
      RVL(3) = TRAJSA(13) * RADDEG
      RESULT1 = SORT(PACC+PACC + QACC+QACC + RACC+RACC)
      RESULT2 = SORT(PVL(3)+PVL(3)+QVL(3)+QVL(3)+RVL(3)+RVL(3))
      RESULT3 = SORT(FROLL(3)+FROLL(3)+FPITCH(3)+FPITCH(3)+FYAW(3)+FYAW(3))
      WRITE(LU,7010) TIME
      + PACC , QACC , RACC , RESULT1,
      + PVL(3) , QVL(3) , RVL(3) , RESULT2,
      + FROLL(3) , FPITCH(3) , FYAW(3) , RESULT3
95      LINECT(6) = LINECT(6) + 1
9900    CONTINUE
      RETURN
C*****
C WRITE FORMAT STATEMENTS
C*****
7000    FORMAT(/,5X,"TIME",17X,"ACCELERATION (SCS)",24X,"RATE (SCS)",
      + 28X,"ORIENTATION (EFC)",5X,"(SEC)",18X,"(DEG/SEC 2)",30X,
      + "(DEG/SEC)",36X,"(DEG)",/,
      + 17X,2(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),
      + 3X,"ROLL",5X,"PITCH",6X,"YAW",6X,"RES",/)
7010    FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
      END

```

```

1 SUBROUTINE REPR17
C .....
C
C SEAT/OCCUPANT LINEAR TIME HISTORY WRT AIRCRAFT REPORT
C .....
5 C .....
C
C CONSTANTS COMMON BLOCK
C .....
C
C COMMON /CONSTANT / GRAVITY , RADDEG , DEGRAD , PI
C .....
10 C .....
C
C SECTION 2 COMMON BLOCK
C .....
C
C COMMON /REPORT / IREPTS(31) , PRTRFQ,P11,P12,P13
C
C INTEGER
C .....
15 C .....
C
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C
C COMMON /MISC / IPAGECT(31) , LINECT(31) , MAXREPT , IPRTCNT(31)
C
C + , MAXLINE , MAXREPT , LU
C
C + , IEVLIN , IERRFLG , HEADVEL
C
C + , IDATE , HEADALT , HEADPI1
C
C + , HEADSR , HEADYAW , HEADPI1
C
C + , HEADROL , HEADWGT , BIAS
C
C + , REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2)
C
C + , IHEADER(24) , IEVENTS(38) , TIMES(38)
C
C + , IMVDC , PRTEMP( 2)
C
C + , PRTMASS(2) , PRINDX , PKZVEL
C
C + , ZVECT(3) , XYZ(3) , SAVTIME
C
C + , XACCEL(3) , YACCEL(3) , ZACCEL(3)
C
C + , REPTYPE , BIAS , PRTLNGT
C
C + , PRTWGT , PRTEMP , PRTMASS , PRINDX
C
C + , PRTEMP , PRINDX
C .....
C
C INTEGRATION ROUTINE COMMON BLOCK
C .....
C
C COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJ50(193)
C
C + , TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3)
C
C + , TRAJAC(193) , TVCEOS(225) , QUAT50(65)
C
C + , QUATSA(65) , QUATDA(65) , QUATAC(65)
C
C + , INSTP , IPCPASS , IRKPASS
C
C + , IPOINTS , IYX , IYPRX
C
C + , IKX , IKSUNX , IKPASSX
C
C + , IY1X , IY1X , IY12X
C
C + , IY13X , IY1X , IYPR11X
C
C + , IYPR12X , IY1X , IYVI1X
C
C + , IYV1X , IY1X , IYV1X
C
C + , IYV1X , IYV1X , IYV1X
C .....
C
C PLOT FILE VARIABLES COMMON BLOCK
C .....
C
C COMMON /PLOT / XACC(3) , YACC(3) , ZACC(3) , ACCR(3)
C
C + , PVL(3) , QVL(3) , RVL(3) , VELR(3)
C
C + , FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3)
C
C + , RXACC(3) , RYACC(3) , RZACC(3) , RLACR(3)
C
C + , RXVEL(3) , RYVEL(3) , RZVEL(3) , RLVR(3)
C
C + , RXPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3)
C
C + , RPVEL(2) , ROVEL(2) , RRVEL(2)
C
C + , RPPOS(2) , RQPOS(2) , RRPOS(2)
C .....

```

```

60      IF(IEVENTS(28) .NE. 0) GO TO 9900
        LU=BIAS+7
        IF (PRTRQ .LE. 0) GO TO 100
        IPRCNT(7) = MOD (IPRCNT(7)+1, PRTRQ)
        IF(IEVLINE .NE. 0) GO TO 100
        IF(IPRCNT(7) .NE. 0) GO TO 9900
100      CONTINUE
        IF(LINECT(7) .LE. MAXLINE) GO TO 200
        CALL HEADER
        WRITE(LU,7000) PRTLNCT(PRTINDX) , PRTLNCT(PRTINDX)
        IF(PRTFRQ .LT. 0) GO TO 9900
200      CONTINUE
        RXACC(1) = (XACCEL(1)-TRAJAC(17))/GRAVITY
        RYACC(1) = (YACCEL(1)-TRAJAC(18))/GRAVITY
        RZACC(1) = (ZACCEL(1)-TRAJAC(19))/GRAVITY
        RLACR(1) = SORT(RXACC(1)*RXACC(1)+RYACC(1)*RYACC(1)+RZACC(1)
        + *RZACC(1))
        RXVEL(1) = TRAJSO(14) - TRAJAC(14)
        RYVEL(1) = TRAJSO(15) - TRAJAC(15)
        RZVEL(1) = TRAJSO(16) - TRAJAC(16)
        RLVL(1) = SORT(RXVEL(1)*RXVEL(1) + RYVEL(1)*RYVEL(1) + RZVEL(1)
        + *RZVEL(1))
        RXPOS(1) = TRAJSO(2) - TRAJAC(2)
        RYPOS(1) = TRAJSO(3) - TRAJAC(3)
        RZPOS(1) = TRAJSO(4) - TRAJAC(4)
        RLPSR(1) = SORT(RXPOS(1)*RXPOS(1) + RYPOS(1)*RYPOS(1) + RZPOS(1)
        + *RZPOS(1))
        WRITE(LU,7010) TIME
        + RXACC(1) , RYACC(1) , RZACC(1) , RLACR(1)
        + RXVEL(1) , RYVEL(1) , RZVEL(1) , RLVL(1)
        + RXPOS(1) , RYPOS(1) , RZPOS(1) , RLPSR(1)
        LINECT(7) = LINECT(7) + 1
9900      CONTINUE
        RETURN
C*****
C WRITE FORMAT STATEMENTS
C*****
7000      FORMAT(/,5X,"TIME",17X,"ACCELERATION (ACS)",22X,"VELOCITY(ACS)",
        + 30X,"POSITION (ACS)",/,5X,"(SEC)",22X,"(G(S)",32X,"(",
        + A2,"/SEC)",38X,"(",A2,")",/,
        + 17X,3(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),/)
7010      FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
        END
100

```

```

1 SUBROUTINE REPR18
C .....
C
5 C SEAT/OCCUPANT ANGULAR TIME HISTORY WRT AIRCRAFT REPORT
C .....
C
C CONSTANTS COMMON BLOCK
C .....
C
10 COMMON /CONSTANT / GRAVITY , RADEG , DEGRAD , PI
C .....
C
C SECTION 2 COMMON BLOCK
C .....
C
15 COMMON /IREPORT / IREPTS(31) , PRTRQ,P11,P12,P13
INTEGER
C .....
C
C MATRIX COMMON BLOCK
C .....
C
20 COMMON /MATRIX / DCMAE(3,3) , DCMRA(3,3) , DCMXA(3,3) ,
+ DCMSE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,
+ DCMXAE(3,3) , DCMOAE(3,3) , DCMXR(3,3) ,
+ DCMXUM(3,3)
C .....
C
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C
25 COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+ MAXLINE , MAXREPT , MAXEVRT ,
+ IEVLNE , IERRFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPTT ,
+ HEADROL , HEADWGT , BIAS ,
30 + REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ IMVDC , PRTEMP( 2) ,
+ PRTMSS(2) , PRINDX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) ,
35 + REPTYPE , ZACCEL(3) ,
INTEGER
+ PRTWGT , BIAS ,
+ PRTEMP , PRTLNGT ,
+ PRINDX , PRTMSS , PRTINDX
C .....
C
40 C INTEGRATION ROUTINE COMMON BLOCK
C .....
C
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSD(193) ,
+ TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3) ,
45 + TRAJAC(193) , TVEQS(225) , QUATSD(65) ,
+ QUATSA(65) , QUATOA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IPOINTS , IYX , IYPRX , IYPRX ,
+ IKX , IKSUMX , IKPASSX ,
50 + IYIX , IYIX , IYI2X ,
+ IYI3X , IYPRIX , IYPRIX ,
+ IYPR12X , IYIYX , IYIYX ,
+ ICYIX , ICYIX , IREIN
C .....
C
C PLOT FILE VARIABLES COMMON BLOCK
C .....
C
55 COMMON /PLOT / XACC(3) , YACC(3) , ZACC(3) , ACCR(3) ,

```

```

60      + PVL(3) , QVL(3) , RVL(3) , VELR(3) ,
      + FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3) ,
      + RXACC(3) , RYACC(3) , RZACC(3) , RLACR(3) ,
      + RXVEL(3) , RYVEL(3) , RZVEL(3) , RLVL(3) ,
      + RXPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3) ,
      + RPVEL(2) , RQVEL(2) , RRVEL(2) ,
      + RPPOS(2) , RQPOS(2) , RRPOS(2) ,
      +
65      C.....
      C SECTION 5 COMMON BLOCK
      C.....
      COMMON /SETALN / XPOSSRP, YPOSSRP, ZPOSSRP, XCGSA , YCGSA ,
      + ZCGSA , IXSA , IYSA , IZSA , IXSA , IYSA ,
      + IYSA , IZSA , PHISA , PSISA , THESA ,
      + AREAISA , HGHTISA , WGTISA , XPOSBOT , YPOSBOT ,
      + ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS ,
      + CISA , C2SA , C3SA , C4SA
      REAL
      + IXSA , IYSA , IZSA , IXSA , IYSA , IZSA ,
      + IZSA
      DIMENSION RPACC(2),RQACC(2),RRACC(2)
      IF(IEVENTS(28) .NE. 0) GO TO 9900
      LU=BIAS+8
      IF (PRTRQ .LE. 0) GO TO 100
      IPRCNT(8) = MOD (IPRCNT(8)+1, PRTRQ)
      IF(IEVLIN .NE. 0) GO TO 100
      IF(IPRCNT(8) .NE. 0) GO TO 9900
      100 CONTINUE
      IF(LINECT(8) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE (LU,7000)
      IF (PRTRQ .LT. 0) GOTO 9900
      200 CONTINUE
      RPPOS(1) = ZARCTAN(DCMSA(2,3),DCMSA(3,3)) * RADDEG
      RQPOS(1) = -ASIN(DCMSA(1,3)) * RADDEG
      RRPOS(1) = ZARCTAN(DCMSA(1,2),DCMSA(1,1)) * RADDEG
      RPACC(1) = (TRAJSOI(23) - TRAJAC(23)) * RADDEG
      RQACC(1) = (TRAJSOI(24) - TRAJAC(24)) * RADDEG
      RRACC(1) = (TRAJSOI(25) - TRAJAC(25)) * RADDEG
      RPVEL(1) = (TRAJSOI(11) - TRAJAC(11)) * RADDEG
      RQVEL(1) = (TRAJSOI(12) - TRAJAC(12)) * RADDEG
      RRVEL(1) = (TRAJSOI(13) - TRAJAC(13)) * RADDEG
      RESULT1 = SORT(RPACC(1)*RPACC(1) + RQACC(1)*RQACC(1) +
      + RRACC(1)*RRACC(1))
      RESULT2 = SORT(RPVEL(1)*RPVEL(1) + RQVEL(1)*RQVEL(1) +
      + RRVEL(1)*RRVEL(1))
      RESULT3 = SORT(RPPOS(1)*RPPOS(1)+RQPOS(1)*RQPOS(1)+RRPOS(1)
      + *RRPOS(1))
      WRITE(LU,7010) TIME
      + RPACC(1) , RQACC(1) , RRACC(1) , RESULT1
      + RPVEL(1) , RQVEL(1) , RRVEL(1) , RESULT2
      + RPPOS(1) , RQPOS(1) , RRPOS(1) , RESULT3
      LINECT(8) = LINECT(8) + 1
      9900 CONTINUE
      RETURN
      C.....
      C WRITE FORMAT STATEMENTS
      C.....
      7000 FORMAT(/.5X,"TIME",17X,"ACCELERATION (ACS)",.24X,"RATE (ACS)",

```

```

115      + 29X,"ORIENTATION (ACS)"./.5X,"(SEC)".18X,"(DEG/SEC 2)".,
      + 30X,"(DEG/SEC)".,38X,"(DEG)"./,
      + 17X,2(4X,"X".9X,"Y".9X,"Z".7X,"RES".5X),
      + 3X,"ROLL".5X,"PITCH".6X,"YAW".6X,"RES"/)
      7010 FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
120      END

```

5-222


```

60 LU=BIAS+9
   IF (PRTFRO .LT. 0) GO TO 100
   IF (EVENTS(28) .EQ. 0) GO TO 9900
   IF (PRTFRO .EQ. 0) GOTO 100
   IPRCNT(9) = MOD (IPRCNT(9)+1, PRTFRO)
   IF (EVLINE .NE. 0) GO TO 100
   IF (IPRCNT(9) .NE. 0) GO TO 9900
100 CONTINUE
   IF (LINECT(9) .LE. MAXLINE) GO TO 200
   CALL HEADER
   WRITE(LU,7000) PRTLNQ(TPRTINDX) , PRTLNQ(TPRTINDX)
   IF (PRTFRO .LT. 0) GOTO 9900
200 CONTINUE
   RXACC(2) = (XACCEL(2)-TRAJAC(17))/GRAVITY
   RYACC(2) = (YACCEL(2)-TRAJAC(18))/GRAVITY
   RZACC(2) = (ZACCEL(2)-TRAJAC(19))/GRAVITY
   RLACR(2) = SORT(RXACC(2)*RXACC(2)+RYACC(2)*RYACC(2)+RZACC(2)
+ *RZACC(2))
   RXVEL(2) = TRAJOA(14) - TRAJAC(14)
   RYVEL(2) = TRAJOA(15) - TRAJAC(15)
   RZVEL(2) = TRAJOA(16) - TRAJAC(16)
   RLVL(2) = SORT(RXVEL(2)*RXVEL(2) + RYVEL(2)*RYVEL(2) + RZVEL(2)
+ *RZVEL(2))
   RXPOS(2) = TRAJOA(2) - TRAJAC(2)
   RYPOS(2) = TRAJOA(3) - TRAJAC(3)
   RZPOS(2) = TRAJOA(4) - TRAJAC(4)
   RLPSR(2) = SORT(RXPOS(2)*RXPOS(2) + RYPOS(2)*RYPOS(2) + RZPOS(2)
+ *RZPOS(2))
   WRITE(LU,7010) TIME
+ , RXACC(2) , RYACC(2) , RZACC(2) , RLACR(2)
+ , RXVEL(2) , RYVEL(2) , RZVEL(2) , RLVL(2)
+ , RXPOS(2) , RYPOS(2) , RZPOS(2) , RLPSR(2)
   LINECT(9) = LINECT(9) + 1
9900 CONTINUE
   RETURN
C*****
C WRITE FORMAT STATEMENTS
C*****
7000 FORMAT(1/.5X,"TIME",17X,"ACCELERATION (ACS)",22X,"VELOCITY (ACS)",
+ 30X,"POSITION (ACS)",/.5X,"(SEC)",22X,"(G[S)",.32X,"("
+ A2,"/SEC)",.38X,"(A2)",/.
+ 17X,3(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),/)
7010 FORMAT(1X,F9.4,2X,3(4F10.2),1X)
END

```

5-224

```

60 LU=BIAS+10
   IF (PRTFRQ .LT. 0) GO TO 100
   IF (IEVENTS(28) .EQ. 0) GO TO 9900
   IF (PRTFRQ .EQ. 0) GOTO 100
   IPRCNT(10) = MOD (IPRCNT(10)+1, PRTFRQ)
   IF (IEVLN .NE. 0) GO TO 100
   IF (IPRCNT(10) .NE. 0) GO TO 9900
65 100 CONTINUE
   IF (LINECT(10) .LE. MAXLINE) GO TO 200
   CALL HEADER
   WRITE(LU,7000) PRTLNCT(PRTINDX), PRTLNCT(PRTINDX)
   IF (PRTFRQ .LT. 0) GOTO 9900
70 200 CONTINUE
   RXACC(3) = (TRAUSA(17)-TRAJAC(17))/GRAVITY
   RYACC(3) = (TRAUSA(18)-TRAJAC(18))/GRAVITY
   RZACC(3) = (TRAUSA(19)-TRAJAC(19))/GRAVITY
   RLACR(3) = SORT(RXACC(3)*RXACC(3)+RYACC(3)*RYACC(3)+RZACC(3)
   + *RZACC(3))
   RXVEL(3) = TRAUSA(14) - TRAJAC(14)
   RYVEL(3) = TRAUSA(15) - TRAJAC(15)
   RZVEL(3) = TRAUSA(16) - TRAJAC(16)
   RLVL(3) = SORT(RXVEL(3)*RXVEL(3) + RYVEL(3)*RYVEL(3) + RZVEL(3)
   + *RZVEL(3))
   RXPOS(3) = TRAUSA(2) - TPAJAC(2)
   RYPOS(3) = TRAUSA(3) - TPAJAC(3)
   RZPOS(3) = TRAUSA(4) - TPAJAC(4)
   RLPSR(3) = SORT(RXPOS(3)*RXPOS(3) + RYPOS(3)*RYPOS(3) + RZPOS(3)
   + *RZPOS(3))
   WRITE(LU,7010) TIME
   + RXACC(3), RYACC(3), RZACC(3), RLACR(3)
   + RXVEL(3), RYVEL(3), RZVEL(3), RLVL(3)
   + RXPOS(3), RYPOS(3), RZPOS(3), RLPSR(3)
90 LINECT(10) = LINECT(10) + 1
   9900 CONTINUE
   RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....
7000 FORMAT(/,5X,"TIME",17X,"ACCELERATION (ACS)",22X,"VELOCITY(ACS)",
   + 30X,"POSITION (ACS)",/5X,"(SEC)",22X,"(G(S)",32X,"("
   + A2,"/SEC)",38X,"(",A2,")",
   + 17X,3(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),/)
7010 FORMAT(1X,F9.4,2X,3(4F10.2),1X)
END

```

```

1 SUBROUTINE REPR11
C .....
C SEAT ALONE ANGULAR TIME HISTORY WRT AIRCRAFT REPORT
C .....
5 C .....
C CONSTANTS COMMON BLOCK
C .....
COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C .....
10 C SECTION 2 COMMON BLOCK
C .....
COMMON /IREPORT / IREPTS(31) , PRTRFQ,P11,P12,P13
INTEGER
PRTRFQ,P11,P12,P13
C .....
15 C MATRIX COMMON BLOCK
C .....
COMMON /MATRIX / DCMAE(3,3) , DCMRA(3,3) , DCMXA(3,3) ,
+ DCMSE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,
20 + DCMXAE(3,3) , DCMOAE(3,3) , DCMNR(3,3) ,
+ DCMOM(3,3)
C .....
C MISCELLANEOUS DATA COMMON BLOCK
C .....
25 COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+ MAXLINE , MAXREPT , MAXEVNT ,
+ IEVLNE , TERFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
30 + HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS ,
+ REPTYPE(5,31) , PRTLNGT(2) , PRTWGHT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ IMVDC , PRAP( 2) ,
35 + PRIMASS(2) , PRINDX , PKEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+ REPTYPE , BIAS , PRTLNGT ,
INTEGR
+ PRTWGHT , PRTMSS , PRINDX ,
40 + PRTEMP
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193) ,
+ TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3) ,
45 + TRAJAQ(193) , TVCEQS(225) , QUATSO(65) ,
+ QUATSA(65) , QUATAO(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , TRKPASS ,
+ IPINTS , IVX , IVPRX ,
+ IKX , IKSUMX , IKPASSX ,
50 + IVIX , IVIIX , IVI2X ,
+ IVI3X , IVPRIX ,
+ IVP12X , IPVIX , JPV11X ,
+ ICYIX , ICYIIX , IREIN ,
C .....
55 C PLOT FILE VARIABLES COMMON BLOCK
C .....
COMMON /PLOT / XACC(3) , YACC(3) , ZACC(3) , ACCR(3)

```

```

60      PVL(3) , QVL(3) , RVL(3) , VEL(3) ,
      FROLL(3) , FPITCH(3) , FYAW(3) , RPOS(3) ,
      RXACC(3) , RYACC(3) , RZACC(3) , RLACR(3) ,
      RVEL(3) , RYVEL(3) , RZVEL(3) , RLVR(3) ,
      RPOS(3) , RYPOS(3) , RZPOS(3) , RLPSR(3) ,
      RPVEL(2) , RQVEL(2) , RRVEL(2) ,
      RPPOS(2) , RQPOS(2) , RRPPOS(2) ,
      C.....
65      C SECTION 5 COMMON BLOCK
      C.....
      COMMON /ISETALN / XPOSSRP , YPOSSRP , ZPOSSRP , XCGSA , YCGSA ,
      ZCGSA , IXZSA , IYXSA , IXZSA , IYXSA ,
      IYZSA , IZZSA , PHISA , PSISA , THESA ,
      AREASA , HGHTSA , WHTISA , XPOSBOT , YPOSBOT ,
      ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS ,
      CISA , C2SA , C3SA , C4SA ,
      REAL
      IXZSA , IYXSA , IXZSA , IYXSA , IYZSA ,
      IZZSA
      DIMENSION RPACC(2),RQACC(2),RRACC(2)
      LU=BIAS+11
      IF (PRIFRQ .LT. 0) GO TO 100
      IF (IEVENTS(2P) .EQ. 0) GO TO 9900
      IF (PRIFRQ .EQ. 0) GOTO 100
      IPTCNT(11) = MOD (IPTCNT(11)+1, PRIFRQ)
      IF (IEVLIN .NE. 0) GO TO 100
      IF (IPTCNT(11) .NE. 0) GO TO 9900
      100 CONTINUE
      IF (LINECT(11) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE (LU,7000)
      IF (PRIFRQ .LT. 0) GOTO 9900
      200 CONTINUE
      RPPOS(2) = ZARCTAN(DCMSA(2,3),DCMSA(3,3)) * RADDEG
      RQPOS(2) = -ASIN(DCMSA(1,3)) * RADDEG
      RRPOS(2) = ZARCTAN(DCMSA(1,2),DCMSA(1,1)) * RADDEG
      RPACC(2) = (TRAUSA(23) - TRAJAC(23)) * RADDEG
      RQACC(2) = (TRAUSA(24) - TRAJAC(24)) * RADDEG
      RRACC(2) = (TRAUSA(25) - TRAJAC(25)) * RADDEG
      RPVEL(2) = (TRAUSA(11) - TRAJAC(11)) * RADDEG
      RQVEL(2) = (TRAUSA(12) - TRAJAC(12)) * RADDEG
      RRVEL(2) = (TRAUSA(13) - TRAJAC(13)) * RADDEG
      RESULT1 = SORT(RPACC(2)*RPACC(2) + RQACC(2)*RQACC(2) +
      + RRACC(2)*RRACC(2))
      RESULT2 = SORT(RPVEL(2)*RPVEL(2) + RQVEL(2)*RQVEL(2) +
      + RRVEL(2)*RRVEL(2))
      RESULT3 = SORT(RPPOS(2)*RPPOS(2)+RQPOS(2)*RQPOS(2)+RRPOS(2)
      + *RRPOS(2))
      WRITE(LU,7010) TIME
      RPACC(2) , RQACC(2) , RRACC(2) , RESULT1
      RPVEL(2) , RQVEL(2) , RRVEL(2) , RESULT2
      RPPOS(2) , RQPOS(2) , RRPOS(2) , RESULT3
      LINECT(8) = LINECT(8) + 1
      9900 CONTINUE
      RETURN
      C.....
      C WRITE FORMAT STATEMENTS
      C.....

```

PAGE 227

83/11/07 09.41.53

FTN 4.6+428

74/74 OPT=1

SUBROUTINE REPRI11

```

115      7000 FORMAT(/, 5X, "TIME", 17X, "ACCELERATION (ACS)", 24X, "RATE (ACS)",
+      29X, "ORIENTATION (ACS)", 5X, " (SEC)", 18X, " (DEG/SEC 2)",
+      30X, " (DEG/SEC)", 36X, " (DEG)", 5X, " (DEG)", 5X, " (DEG)",
+      17X, 2(4X, "X", 9X, "Y", 9X, "Z", 7X, "RES", 5X),
+      3X, "ROLL", 5X, "PITCH", 6X, "YAW", 6X, "RES"/)
120      7010 FORMAT(1X, F9.4, 2X, 3(4(F10.2), 1X))
      END

```

```

1  SUBROUTINE REPRT12
C.....
C
C CATAPULT FORCES AND MOMENTS REPORT
C
5  C
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
10  COMMON /FORCESO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
      + FXTUBSO , FYTUBSO , FZTUBSO ,
      + FXSLSO(6) , FYSLSO(6) , FZSLSO(6) ,
      + FXRKSO(6) , FYRKSO(6) , FZRKSO(6) ,
      + FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
      + FXAESO , FYAESO , FZAESO ,
      + FXDRISO , FYDRISO , FZDRISO
15  C
C SECTION 9 COMMON BLOCK
C.....
20  COMMON /ICATPLT / INCAT , CATLNT(2) , CAISTK(2),TCI (2),
      + XPOSAP(2) , YPOSAP(2) , ZPOSAP(2),NPTSCT(2),
      + CATHRST(2,25,2), ITUBEND , KTUBE ,CTUBE ,
      + PTUBE , MUTUBE , EXTINGT , ICATOUT
      + REAL
      + KTUBE , MUTUBE
25  C
C SECTION 2 COMMON BLOCK
C.....
30  COMMON /IREPT / IREPTS(31) , PRIFRQ,P11,P12,P13
      + INTEGER
      + PRIFRQ,P11,P12,P13
C MISCELLANEOUS DATA COMMON BLOCK
C.....
35  COMMON /MISC / IPAGECT(31) , IINECT(31) , IPRICT(31) ,
      + MAXLINE , MAXREPT , TERRFLG , LU , MAXEVRT ,
      + IDATE , HEADALT , HEADVEL ,
      + HEADSR , HEADYAW , HEADPIT ,
      + HEADROL , HEADWGT , BIAS ,
      + REPTYPE(5,31) , PRTINGT(2) ,
      + IHEADER(24) , IEVENTS(38) ,
      + IMVDC , PRTINDX ,
      + PRTMASS(2) , PRTINDX ,
      + ZVECT(3) , XYZ(3) , SAVTIME ,
      + XACCEL(3) , YACCEL(3) ,
      + REPTYPE , BIAS ,
      + PRTWGT , PRTINGT ,
      + PRIEMP , PRTMASS , PRTINDX
45  C
C INTEGRATION ROUTINE COMMON BLOCK
C.....
50  COMMON /RKUTTA / TIME , TIME5 , DELTAT , TRAJSO(193) ,
      + TRAJSA(193) , TRAJUA(193) , TRAJCH(97,3) ,
      + TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
      + QUATSA(65) , QUATQA(65) , QUATAC(65) ,
      + INTSTP , IPCPASS , IRKPASS ,
      + IPOINTS , IYX , IYPRX ,
      + IKX , IKSUMX ,
      + IYIX , IYIIX
55  C

```

```

60      + IYI3X      IYPR1X      IYPR11X
      + IYPR12X      IYI1X      IYI11X
      + IYI1X      IYI1X      IYI11X
C.....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
      COMMON /TORSO / TLCASO(2) , TMCASO(2) , TNCASO(2) ,
      + TLTUBSO , TMTUBSO , TINTUBSO ,
      + TILSLO(6) , TMSLSO(6) , TNSLSO(6) ,
      + TMRKSO(6) , TNRKSO(6) ,
      + TLCHSO(3) , TMCHSO(3) , TNCCHSO(3) ,
      + TLAESO , TMAESO , TNAESO ,
      + TLDRTSO , TMDRTSO , TNDRTSO
      LU=BIAS+12
      IF (PRFRQ .LT. 0) GO TO 100
      IF ((IEVENTS(1) .EQ. 0) .OR.
      + (IEVENTS(3) .NE. 0)) GO TO 9900
      IF (PRFRQ .EQ. 0) GOTO 100
      IPRCNT(12) = MOD (IPRCNT(12)+1, PRFRQ)
      IF (IEVLNE .NE. 0) GO TO 100
      IF (IPRCNT(12) .NE. 0) GO TO 9900
100 CONTINUE
      IF (LINECT(12) .LE. MAXLINE) GO TO 200
      CALL HEADER
      IF (INCAT .EQ. 1) WRITE(LU,7001) PRTWGT(PRTINDX),PRTWGT(PRTINDX),
      + PRTLNGT(PRTINDX)
      IF (INCAT .EQ. 2) WRITE(LU,7002) PRTWGT(PRTINDX),PRTWGT(PRTINDX),
      + PRTLNGT(PRTINDX),PRTWGT(PRTINDX),
      + PRTWGT(PRTINDX),PRTLNGT(PRTINDX)
      IF (PRFRQ .LT. 0) GOTO 9900
200 CONTINUE
      RESULT1=SQRT(FXCASO(1)+FXCASO(1)+FXCASO(1)+FXCASO(1)+
      + FZCASO(1)+FZCASO(1))
      RESULT2=SQRT(TLCASO(1)+TLCASO(1)+TMCASO(1)+TMCASO(1)+
      + TNCASO(1)+TNCASO(1))
      IF (INCAT .NE. 2) GO TO 230
      RESULT3 = SQRT(FXCASO(2)+FXCASO(2) + FXCASO(2)+FXCASO(2) +
      + FZCASO(2)+FZCASO(2))
      RESULT4 = SQRT(TLCASO(2)+TLCASO(2) + TMCASO(2)+TMCASO(2) +
      + TNCASO(2)+TNCASO(2))
      WRITE(LU,7012) TIME
      + FXCASO(1) , FXCASO(1) , FZCASO(1) , RESULT1
      + TLCASO(1) , TMCASO(1) , TNCASO(1) , RESULT2
      + FXCASO(2) , FXCASO(2) , FZCASO(2) , RESULT3
      + TLCASO(2) , TMCASO(2) , TNCASO(2) , RESULT4
      GO TO 350
230 CONTINUE
      WRITE(LU,7011) TIME
      + FXCASO(1) , FXCASO(1) , FZCASO(1) , RESULT1
      + TLCASO(1) , TMCASO(1) , TNCASO(1) , RESULT2
350 CONTINUE
      LINECT(12)=LINECT(12)+1
9900 CONTINUE
      RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....

```



```

115      7001 FORMAT(/5X,"TIME",40X,"FORCES (SCS)",43X,"MOMENTS (SCS)"/,
+          5X,"(SEC)",42X,"(",A2,")",51X,"(",A2,")",A2,")",/,
+          38X,"X",10X,"Y",10X,"Z",8X,"RES",21X,"L",10X,"M",10X,"N",
+          8X,"RES",/)
120      7002 FORMAT(/3X,"TIME",2(12X,"FORCES (SCS)",19X,"MOMENTS (SCS)",5X),
+          /3X,"(SEC)",2(14X,"(",A2,")",27X,"(",A2,")",A2,")",9X),/,
+          18X,2("X",6X,"Y",6X,"Z",5X,"RES",6X,"L",7X,"M",7X,"N",6X,
+          "RES",5X),/)
125      7011 FORMAT(1X,F9.4,24X,4(F9.2,2X),13X,4(F9.2,2X))
      7012 FORMAT(1X,F9.4,1X,2(4(F7.1),1X,4(F8.1)))
      END

```

```

1      SUBROUTINE REPT13(RKTI)
C.....
C
C ROCKET FORCES AND MOMENTS REPORT
C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
COMMON /FORCESO / FXCASO(2) , FZCASO(2) ,
+   FYTUBSO , FZTUBSO ,
+   FXLSO(6) , FZLSO(6) ,
+   FXRKS(6) , FZRKS(6) ,
+   FXCHSO(3) , FZCHSO(3) ,
+   FXAESO , FZAESO ,
+   FXORTSO , FZORTSO
C.....
C SECTION 2 COMMON BLOCK
C.....
COMMON /IREPT / IREPTS(31) , PRIFRO,PI1,PI2,PI3
+   PRIFRO,PI1,PI2,PI3
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGET(31) , LINECT(31) , IPRCNT(31) ,
+   MAXLINE , MAXREPT , MAXEVT ,
+   IEVLIN , TERRFLG , LU ,
+   IDATE , HEADALT , HEADVEL ,
+   HEADSR , HEADYAW , HEADPIT ,
+   HEADROL , HEADWGT , BIAS ,
+   REPTYPE(5,31) , PRTLNGT(2) ,
+   IHEADER(24) , IEVENTS(38) ,
+   INVDC , PRTEMP( 2) ,
+   PRTMAS(2) , PRTIINDX , PKZVEL ,
+   ZVECT(3) , XYZ(3) , SAVTIME ,
+   XACCEL(3) , YACCEL(3) ,
+   REPTYPE , BIAS , PRTLNGT ,
+   PRTWGHT , PRTEMP , PRTMAS , PRTIINDX
C.....
C INTEGRATION ROUTINE COMMON BLOCK
C.....
COMMON /RKUTTA / TIME , TIMES , DELTAT ,
+   TRAJSA(193) , TRAJDA(193) , TRAJSO(193) ,
+   TRAJAC(193) , TVCEQS(225) , TRAJCH(97.3) ,
+   QUATSA(65) , QUATOA(65) , QUATAC(65) ,
+   INTSTP , IPCPASS , IRKPASS ,
+   IPOINTS , IVX , IVPRX ,
+   IKX , IKSUMX , IKPASSX ,
+   IVIX , IVIIX , IV12X ,
+   IV13X , IVPRIX , IVPRIIX ,
+   IVPR12X , IVYIX , IPYIIX ,
+   ICYIX , ICYIIX , IREIN
C.....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
COMMON /TORQSO / TMCASO(2) , TNCASO(2) ,
+   TLTUBSO , TMTUBSO ,

```

5-233

```

1 SUBROUTINE REPR19
C .....
C
C DART FORCES AND MOMENTS REPORT
C .....
C
C SEAT/OCCUPANT FORCES COMMON BLOCK
C .....
C
COMMON /FORCESO / FXCASO(2) , FVCASO(2) , FZCASO(2) ,
+ FXTUBSO , FYTUBSO , FZTUBSO ,
+ FXLSO(6) , FVLSO(6) , FZLSO(6) ,
+ FXRKS(6) , FVRKS(6) , FZRKS(6) ,
+ FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+ FXAESO , FYAESO , FZAESO ,
+ FXORTSO , FYORTSO , FZORTSO
C .....
C
C SECTION 2 COMMON BLOCK
C .....
C
COMMON /IREPORT / IREPTS(31) , PRTRFQ,P11,P12,P13
INTEGER
C .....
C
C MISCELLANEOUS DATA COMMON BLOCK
C .....
C
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRTCNT(31) ,
+ MAXLINE , MAXREPT , MAXEVT ,
+ IEVLNE , IERRFLG , LU ,
+ IDATE , HEADLT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS ,
+ REPTYPE(5,31) , PRTLNGT(2) ,
+ IHEADER(24) , IEVENTS(38) ,
+ IMVDC , PRTEMP( 2) ,
+ PRTHSS(2) , PRINDX ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+ REPTYPE , BIAS , PRTLNGT
INTEGER
+ PRTWGHT , PRTHSS , PRINDX
C .....
C
C INTEGRATION ROUTINE COMMON BLOCK
C .....
C
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193) ,
+ TRAJA(193) , TRAJCH(97,3) ,
+ TRAJAC(193) , TVCEOS(225) , QUATSO(65) ,
+ QUATSA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IPOINTS , IYX , IYPRX ,
+ IKX , IKSUMX , IKPASSX ,
+ IYIX , IYIIX , IYI2X ,
+ IYI3X , IYPRIX , IYPRI1X ,
+ IYPR12X , IYPIX , IYI1X ,
+ IYI1X , IYI1X , IREIN
C .....
C
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C .....
C
COMMON /TORQSO / TLCASO(2) , TMCASO(2) , TNCASO(2) ,
+ TLTUBSO , TMTUBSO , TINTUBSO

```

```

+      TMSLSO(6) , TMSLSO(6) , TMSLSO(6) ,
+      TMRKSO(6) , TMRKSO(6) , TMRKSO(6) ,
+      TMSLSO(3) , TMSLSO(3) , TMSLSO(3) ,
+      TMAESO , TMAESO , TMAESO ,
+      TMDRTSO , TMDRTSO , TMDRTSO
+
LU = BIAS+18
IF (PRTRQ .LT. 0) GO TO 100
IF ((IEVENTS(33) .EQ. 0) .AND. (IEVENTS(34) .EQ. 0)) .OR.
+ ((IEVENTS(35) .NE. 0) .AND. (IEVENTS(36) .NE. 0)))
+
+
IF (PRTRQ .EQ. 0) GO TO 100
IPRTRQ(19) = MOD (IPRTRQ(19)+1, PRTRQ)
IF (IEVLINE .NE. 0) GO TO 100
IF (IPRTRQ(19) .NE. 0) GO TO 9900
100 CONTINUE
IF (LINECT(19) .LE. MAXLINE) GO TO 200
CALL HEADER
WRITE(LU,7000) PRTRQ(PRTINDX) , PRTRQ(PRTINDX) ,
+
+
IF (PRTRQ .LT. 0) GO TO 9900
200 CONTINUE
RESULT1=SQRT(FXDRTO*FXDRTO+FYDRTO*FYDRTO+TZDRTO*TZDRTO)
RESULT2=SQRT(TLDRTO*TLDRTO+TMDRTO*TMDRTO+TNDRTO*TNDRTO)
WRITE(LU,7010) TIME
+
+
FXDRTO , FYDRTO , TZDRTO , RESULT1 ,
+
TLDRTO , TMDRTO , TNDRTO , RESULT2
+
LINECT(19) = LINECT(19) + 1
9900 CONTINUE
RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....
7000 FORMAT(/5X,"TIME",28X,"FORCES (SCS)",42X,"MOMENTS (SCS)"/.
+
+
5X,"(SEC)",30X,"(A2)",50X,"(A2)",50X,"(A2)"/.
+
28X,"X",10X,"Y",10X,"Z",9X,"RES",21X,"L",10X,"M",10X,"N",
+
9X,"RES"/.
7010 FORMAT(1X,F9.4,12X,F9.1,2X),12X,F9.1,2X)
END

```

```

1 SUBROUTINE REPR120
C .....
C
C C DROQUE FORCES AND MOMENTS REPORT
C .....
5
C
C C SEAT/OCCUPANT FORCES COMMON BLOCK
C .....
10 COMMON /FORCESO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+ FXTUBSO , FYTUBSO , FZTUBSO ,
+ FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+ FXKSO(6) , FYKSO(6) , FZKSO(6) ,
+ FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+ FXAESO , FYAESO , FZAESO ,
+ FXORTSO , FYORTSO , FZORTSO
15
C .....
C
C C SECTION 2 COMMON BLOCK
C .....
20 COMMON /IREPORT / IREPTS(31) , PRTRQ,P11,P12,P13
+ PRTRQ,P11,P12,P13
+ INTEGER
C .....
C C MISCELLANEOUS DATA COMMON BLOCK
C .....
25 COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+ MAXLINE , MAXREPT , MAXEVT , MAXEVT ,
+ JEVLNE , IERRFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS ,
30 + REPTYPE(5,31) , PRTNGT(2) , PRTNGT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ IMVDC , PRTEMP( 2) ,
+ PRTMASS(2) , PRINDX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
35 + REPTYPE , BIAS , PRTNGT ,
+ PRWGT , PRTEMP , PRTMASS , PRINDX
C .....
C
C C INTEGRATION ROUTINE COMMON BLOCK
C .....
40 COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSOI(193) ,
+ TRAJSA(193) , TRAJDA(193) , TRAJCHI(97,3) ,
+ TRAJJAC(193) , TVCEOS(225) , QUATSOI(65) ,
45 + QUATOA(65) , QUATAC(65) ,
+ INISTP , IPCPASS , IRKPASS ,
+ IPOINTS , IYX , IYPRX ,
+ IKX , IKSUNX , IMPASSX ,
+ IYIX , IYIIX , IYI2X ,
50 + IYI3X , IYPRIX , IYPIIX ,
+ IYPR12X , IYPIX , IYPIIX ,
+ ICYIX , ICYIIX , IREIN ,
C .....
C
C C TORQUE SEAT/OCCUPANT COMMON BLOCK
C .....
55 COMMON /TORQSO / TLCASO(2) , TMCASO(2) , TNCASO(2) ,
+ TLTUBSO , TMTUBSO , INTUBSO ,

```

```

+      TSLSO(6) ; TMSLSO(6) ; TNSLSO(6) ;
+      TLRKSO(6) ; TMRKSO(6) ; TNRKSO(6) ;
+      TLCHSO(3) ; TMCHSO(3) ; TNCHSO(3) ;
+      TLAESO ; TMAESO ; TNAESO ;
+      TLDRTSO ; TMDRTSO ; TNDRTSO ;

LU=BIAS+20
IF (PRTFRQ .LT. 0) GO TO 100
IF ((IEVENTS(19) .EQ. 0) .OR.
+   (IEVENTS(24) .NE. 0)) GO TO 9900
IF (PRTFRQ .EQ. 0) GO TO 100
IPRTCNT(20) = MOD (IPRTCNT(20)+1, PRTFRQ)
IF (IEVLIN .NE. 0) GO TO 100
IF (IPRTCNT(20) .NE. 0) GO TO 9900
100 CONTINUE
IF (LINECT(20) .LE. MAXLINE) GO TO 200
CALL HEADER
WRITE(LU,7000) PRTWGT(PRTINDX) ; PRTWGT(PRTINDX) ;
+   PRTLNGT(PRTINDX) ; PRTLNGT(PRTINDX)
IF (PRTFRQ .LT. 0) GO TO 9900
200 CONTINUE
X = FCHSO(1) + FCHSO(2)
Y = FYCHSO(1) + FYCHSO(2)
Z = FZCHSO(1) + FZCHSO(2)
FL = TLCHSO(1) + TLCHSO(2)
FM = TMCHSO(1) + TMCHSO(2)
FN = TNCHSO(1) + TNCHSO(2)
PX = TRAJCH(2,1) + TRAJCH(2,2)
PY = TRAJCH(3,1) + TRAJCH(3,2)
PZ = TRAJCH(4,1) + TRAJCH(4,2)
RESULT1 = SORT(X*FL + Y*FM + Z*FN)
RESULT2 = SORT(FL*FL + FM*FM + FN*FN)
RESULT3 = SORT(PX*PX + PY*PY + PZ*PZ)
WRITE(LU,7010) TIME
+   X ; Y ; Z ; RESULT1.
+   FL ; FM ; FN ; RESULT2.
+   PX ; PY ; PZ ; RESULT3

9900 CONTINUE
LINECT(20) = LINECT(20) + 1
RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....
7000 FORMAT(/5X,"TIME",17X,"FORCES (SCS)",28X,"MOMENTS (SCS)",29X,
+ "POSITION (EFC)",5X,"(SEC)",19X,"(A2)",36X,"(A2)",A2,
+ ".37X,"(A2)",18X,"X",9X,"Y",9X,"Z",8X,"RES",10X,"L",9X,
+ "M",9X,"N",8X,"RES",10X,"X",9X,"Y",9X,"Z",8X,"RES"/)
7010 FORMAT(1X,F9.4,3(2X,4(F9.1,1X)))
END

```

```

1 SUBROUTINE REPRT21
C
C
C PARACHUTE FORCES AND MOMENTS
C
C
C OCCUPANT ALONE FORCES COMMON BLOCK
C
C COMMON /FORCEOA / FXCHOA(3) , FYCHOA(3) , FZCHOA(3) ,
C + FXAEOA , FYAEOA , FZAEOA
C
C SEAT/OCCUPANT FORCES COMMON BLOCK
C
C COMMON /FORCESO / FXCASO(2) , FYCASSO(2) , FZCASO(2) ,
C + FXTUBSO , FYTUBSO , FZTUBSO ,
C + FXLSO(6) , FYSLO(6) , FZLSO(6) ,
C + FXRKS(6) , FYRKS(6) , FZRKS(6) ,
C + FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
C + FXAESO , FYAESO , FZAESO ,
C + FXDRTSO , FYDRTSO , FZDRTSO
C
C SECTION 2 COMMON BLOCK
C
C COMMON /IREPT / IREPTS(31) , PRIFRQ,P11,P12,P13
C INTEGER
C
C MISCELLANEOUS DATA COMMON BLOCK
C
C COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
C + MAXLINE , MAXREPT , MAXEVT ,
C + IEVLNE , TERRFLG , LU ,
C + IDATE , HEADAL , HEADVEL ,
C + HEADSR , HEADVAV , HEADPIT ,
C + HEADROL , HEADWGT , BIAS ,
C + REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
C + IHEADER(24) , TEVENTS(38) ,
C + INVDC , PRIEMP( 2) ,
C + PRTMASS(2) , PRINDX , PKZVEL ,
C + ZVECT(3) , XYZ(3) , SAVTIME ,
C + XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
C + REPTYPE , BIAS , PRTLNCT ,
C + PRTWGT , PRIEMP , PRTLNCT ,
C + PRIEMP , PRINDX
C
C INTEGRATION ROUTINE COMMON BLOCK
C
C COMMON /RKUTTA / TIME , TIMES ,
C + TRAJSA(193) , TRAJOAT(193) , TRAJSD(193) ,
C + TRAJAC(193) , TVCEQS(225) , TRAJCH(97,3) ,
C + QUATSA(65) , QUATOAI(65) , QUATSD(65) ,
C + INTSTP , IPCPASS , QUATAC(65) ,
C + IPOINTS , IYX , IRKPASS ,
C + IKX , IYX , IYPRX , IYPRX ,
C + IYIX , IYIX , IYPRIX , IYPRIX ,
C + IYIX , IYIX , IYPRIX , IYPRIX ,
C + ICVIX , ICVIX , ICVIX , ICVIX

```



```

C.....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
COMMON /TORQSO / TLCASO(2) , TMCASO(2) , TNCASO(2) ,
+   TLTUBSO , TMTUBSO , TNCASO(2) ,
+   TMSLSO(6) , TMSLSO(6) , TMSLSO(6) ,
+   TMRKSO(6) , TMRKSO(6) , TMRKSO(6) ,
+   TMLCHSO(3) , TMLCHSO(3) , TMLCHSO(3) ,
+   TLAESO , TMAESO , TMAESO ,
+   TLDRTSO , TMDRTSO , TMDRTSO
C.....
C TORQUE OCCUPANT ALONE COMMON BLOCK
C.....
COMMON /TORQOA / TLCHOA(3) , TMCHOA(3) , TNCCHOA(3) ,
+   TLAEOA , TMAEOA , TMAEOA
REAL L , M , N
LU=BIAS+21
IF (PRTRQ .LT. 0) GO TO 100
IF (IEVENTS(25) .EQ. 0) GO TO 9900
IF (IPTRFRQ .EQ. 0) GO TO 100
IPRTCNT(21) = MOD (IPRTCNT(21)+1, PRTRFRQ)
IF (IEVLINE .NE. 0) GO TO 100
IF (IPRTCNT(21) .NE. 0) GO TO 9900
100 CONTINUE
IF (LINECT(21) .LE. MAXLINE) GO TO 200
CALL HEADER
WRITE(LU,7000) PRTWGHT(PRTINDX)
IF (PRTRFRQ .LT. 0) GO TO 9900
200 CONTINUE
IF (IEVENTS(28) .GE. 1) GO TO 300
X = FXCHSD(3)
Y = FYCHSD(3)
Z = FZCHSD(3)
L = TLCHSD(3)
M = TMCHSD(3)
N = TNCHSD(3)
GO TO 400
300 CONTINUE
X = FXCHOA(3)
Y = FYCHOA(3)
Z = FZCHOA(3)
L = TLCHOA(3)
M = TMCHOA(3)
N = TNCHOA(3)
400 CONTINUE
RES1 = SORT(X*X + Y*Y + Z*Z)
RES2 = SORT(L*L + M*M + N*N)
WRITE(LU,7010) TIME, X, Y, Z, RES1, L, M, N, RES2
LINECT(21) = LINECT(21) + 1
9900 CONTINUE
RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....
7000 FORMAT(/5X,"TIME",28X,"FORCES (SCS)",42X,"MOMENTS (SCS)"/,
+   5X,"(SEC)",30X,"(-.A2,")",50X,"(-.A2,")",/
+   28X,"X",10X,"Y",10X,"Z",9X,"RES",21X,"L",10X,"M",10X,"N",

```

PAGE 239

83/11/07. 09.41.53

FTN 4 6+428

SUBROUTINE REPRT21 74/74 OPT=1

115 9X,"RES",/)
 7010 FORMAT(1X,F9.4,12X,4(F9.1,2X),12X,4(F9.1,2X))
 END

```

1      SUBROUTINE REPRT22
C.....
C      C TVC MICROPROCESSOR DATA REPORT
C.....
5      C.....
C.....
C      C SECTION 2 COMMON BLOCK
C.....
      COMMON /IREPORT / IREPTS(31) , PRIFRQ,P11,P12,P13
      INTEGER
      PRIFRQ,P11,P12,P13
10     C.....
C      C SECTION 12 COMMON BLOCK
C.....
      COMMON /ITVCIN / ITVC , MPH1 , MPS1 , MTHE ,
      ROLLRL , PITCHRL , SMPLRAT , TVCDLAY ,
      RKANG
      REAL
      MPH1 , MPS1 , MTHE
15     C.....
C      C MISCELLANEOUS DATA COMMON BLOCK
C.....
      COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
      MAXLINE , MAXREPT , MAXEVNT ,
      IEVLIN , IERFLG , LU ,
      IDATE , HEADALT , HEADVEL ,
      HEADSR , HEADYAW , HEADPIT ,
      HEADROL , HEADWGT , BIAS ,
      REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
      IHEADER(24) , TEVENTS(38) ,
      IMVDC , PRTEMP( 2) ,
      PRTHASS(2) , PRTINDX , PKZVEL ,
      ZVECT(3) , XYZ(3) , SAVTIME ,
      XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
      REPTYPE , BIAS , PRTLNGT ,
      PRTWGT ,
      PRTEMP , PRTHASS , PRTINDX
35     C.....
C      C MOMARMS COMMON BLOCK
C.....
      COMMON /MOMARMS /
      +REFLNSO ,REFLNDA ,REFLNSA ,URX(6) ,URY(6) ,URZ(6) ,
      +XSSOCA(2) ,YSSOCA(2) ,ZSSOCA(2) ,XSORK(6) ,YSSORK(6) ,ZSSORK(6) ,
      +XSORRE ,YSSORRE ,ZSSORRE ,XSOLRE ,YSSOLRE ,ZSSOLRE ,
      +XSOMRE ,YSSOMRE ,ZSSOMRE ,XSORBOT ,YSSORBOT ,ZSSORBOT ,
      +XSSOSB(6) ,YSSOSB(6) ,ZSSOSB(6) ,XRCSAC ,YRCSAC ,ZRCSAC ,
      +XSSCSAC ,YSSCSAC ,ZSSCSAC ,XSOSRP ,YSSOSRP ,ZSSOSRP ,
      +
      +XSSASRP ,YSSASRP ,ZSSASRP ,XRDPAP(2) ,YRDPAP(2) ,ZRDPAP(2) ,
      +XRSBOT(6) ,YRSBOT(6) ,ZRSBOT(6) ,XSOCAP(2) ,YSOCAP(2) ,ZSOCAP(2) ,
      +XSDAP(2) ,YSDAP(2) ,ZSDAP(2) ,XESDAC ,YESDAC ,ZESDAC ,
      +XSRCSAC ,YSRCSAC ,ZSRCSAC ,XSOCAC ,YSOCAC ,ZSOCAC ,
      +XRSOSB ,YRSOSB ,ZRSOSB ,XRSBOT ,YRSBOT ,ZRSBOT ,
      +XRRSB ,YRRSB ,ZRRSB ,XSOCH(3) ,YSOCH(3) ,ZSOCH(3) ,
      +XAACSO ,YAACSO ,ZAACSO ,XASOAC ,YASOAC ,ZASOAC ,
      +XRSOAC ,YRSOAC ,ZRSOAC ,XSCAP(2) ,YSCAP(2) ,ZSCAP(2) ,
      +
55     C.....
C      C INTEGRATION ROUTINE COMMON BLOCK
C.....

```

```

COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSA(193) , TRAJOA(193) , TRAJSD(193) ,
+ TRAJAC(193) , TRAJCH(97,3) ,
+ TRAJEQS(225) , QUATSO(65) ,
+ QUATSA(65) , QUATOAI(65) , QUATAC(65) ,
+ INTSTP , ICPASS , IRKPASS ,
+ IPOINT5 , IYX , IYPRX ,
+ IYPR12X , IYIYX , IYPIYX ,
+ IKK , IKSUMX , IKPASSX ,
+ IYIX , IYI2X ,
+ IYI3X , IYPR11X ,
+ ICYIX , ICYI1X , IREIN ,
C .....
C THRUST VECTOR CONTROL VARIABLES COMMON BLOCK
C .....
COMMON/TVCVRB / ITVCFLG , CMPVAL , D3(3) , RKTCMND(3) ,
+ C29 , C30 , C31 , C32 ,
+ DTH(3,2) , ANGR(3)
C .....
LU=BIAS+22
IF (PRTRFQ .LT. 0) GO TO 100
IF (ITVC .NE. 1) OR
+ EVENTS(6) .EQ. 0 .OR. IEVENTS(12) .NE. 0)
+ GO TO 9900
IF (PRTRFQ .EQ. 0) GO TO 100
IPRTCNT(22) = MOD (IPRTCNT(22)+1, PRTRFQ)
IF(IEVLNE .NE. 0) GO TO 100
IF (IPRTCNT(22) .NE. 0) GO TO 9900
100 CONTINUE
IF (LINECT(22) .LE. MAXLINE) GO TO 200
CALL HEADER
WRITE (LU,7000)
IF(PRTFRQ .LT. 0) GO TO 9900
200 CONTINUE
WRITE (LU,7010) TIME , URX(1) , URY(1) , URZ(1) ,
+ D3(1) , D3(2) , D3(3) ,
+ RKTCMND(1) , RKTCMND(2) , RKTCMND(3)
LINECT(22) = LINECT(22) + 1
9900 CONTINUE
RETURN
C .....
C WRITE FORMAT STATEMENTS
C .....
7000 FORMAT (1/5X,"TIME",17X,"ROCKET THRUST LINE",20X,"DIRECTION",
+ "COSINES",23X,"ROCKET GIMBALLING COMMANDS",/5X,"(SEC)"/)
7010 FORMAT (1X,F9.4,3(10X,3(F10.4)))
END

```

```

1      SUBROUTINE REPRT23
C.....
C
C RAIL FORCES AND MOMENTS REPORT
C.....
5      C
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
10     COMMON /FORCESO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+      FXTUBSO , FYTUBSO , FZTUBSO ,
+      FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+      FXRKS(6) , FYRKS(6) , FZRKS(6) ,
+      FXCHS(3) , FYCHS(3) , FZCHS(3) ,
+      FXAESO , FYAESO , FZAESO ,
+      FXDRTSO , FYDRTSO , FZDRTSO
15     C
C.....
C SECTION 7 COMMON BLOCK
C.....
20     COMMON /IRAIL / RAILNTH , RAILANG , ISTRL , NSLBKS ,
+      KXSB , KYSB , MUSB , YKTOR ,
+      XPOSRR , YPOSRR , ZPOSRR ,
+      XPOSRL , YPOSRL , ZPOSRL ,
+      XPOSSB(6) , YPOSSB(6) , ZPOSSB(6)
25     REAL
+      KXSB , KYSB , MUSB
C.....
C SECTION 2 COMMON BLOCK
C.....
30     COMMON /IREPT / IREPTS(31) , PRIFRO,P11,P12,P13
+      INTEGER
+      PRIFRO,P11,P12,P13
C MISCELLANEOUS DATA COMMON BLOCK
C.....
35     COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+      MAXLINE , MAXREPT , MAXEVT ,
+      TEVLIN , TERRFLG , LU ,
+      IDATE , HEADALT , HEADVEL ,
+      HEADSR , HEADYAW , HEADPIT ,
+      HEADROL , HEADWGT , BIAS ,
+      REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
+      IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+      INVDC , PRTEMP( 2) ,
+      PRTMAS(2) , PRTINDX ,
+      ZVECT(3) , XYZ(3) , SAVTIME ,
+      XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+      REPTYPE , BIAS , PRTLNCT ,
+      PRTWGT , PRTEMP , PRTMAS , PRTINDX
45     INTEGER
+      PRTWGT , PRTEMP , PRTMAS , PRTINDX
C.....
C INTEGRATION ROUTINE COMMON BLOCK
C.....
50     COMMON /RKU.TA / TIME , TIMES , DELTAT , TRAUJO(193) ,
+      TRAJSA(193) , TRAJOA(193) , TRAUCH(97,3) ,
+      TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
+      QUATSA(65) , QUATOA(65) , QUATAC(65) ,
+      INTSTP , IPCPASS , IRKPASS ,
+      IPC.NTS , IYX , IYPRX ,
+      IKX , IKSUMX , IKPASSX
55     C

```

[illegible]

```

1      SUBROUTINE REPR24
C.....
C
C SEAT/OCCUPANT AERODYNAMIC FORCES AND MOMENTS REPORT
C.....
C
C OCCUPANT ALONE FORCES COMMON BLOCK
C.....
C
COMMON /FORCEDA / FXCHOA(3) , FYCHOA(3) , FZCHOA(3) ,
+   FXAEOA , FYAEOA , FZAEOA
C.....
C SEAT ALONE FORCES COMMON BLOCK
C.....
C
COMMON /FORCESA / FXAESA , FYAESA , FZAESA
C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
C
COMMON /FORCESO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+   FXTUBSO , FYTUBSO , FZTUBSO ,
+   FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+   FXRKS(6) , FYRKS(6) , FZRKS(6) ,
+   FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+   FXAESO , FYAESO , FZAESO ,
+   FXORTSO , FYORTSO , FZORTSO
C.....
C SECTION 2 COMMON BLOCK
C.....
COMMON /IREPORT / IREPTS(31) , PRTRQ,P11,P12,P13
INTEGER
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCTNT(31) ,
+   MAXLINE , MAXREPT , MAXEVT ,
+   IEVLNE , IERRFLG , LU ,
+   IDATE , HEADALT , HEADVEL ,
+   HEADSR , HEADYAW , HEADPIT ,
+   HEADROL , HEADWGT , BIAS ,
+   REPTYPE(5,31) , PRTLNGT(2) , PRTWGHT(2) ,
+   IHEADER(24) , TEVENTS(38) , TIMES(38) ,
+   INVDC , PRTEMP( 2) ,
+   PRTMASX(2) , PRTINDX , PKZVEL ,
+   ZVECT(3) , XYZ(3) , SAVTIME ,
+   XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+   REPTYPE , BIAS , PRTLNGT ,
+   PRTWGHT , PRTEMP , PRTMASX , PRTINDX
C.....
C INTEGRATION ROUTINE COMMON BLOCK
C.....
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAUJO(193) ,
+   TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3) ,
+   TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
+   QUATOA(65) , QUATAC(65) ,
+   INTSTP , IPCPASS , IRKPASS ,
+   IPNTS , IVX , IYPRX ,
+   IKX , IKSUMX , IKPASSX

```

```

+      . IYIX      . IYIX      . IY12X      .
+      . IY13X      . IYPR1X      . IYPR11X      .
+      . IYPR12X      . IYPR1X      . IYPR11X      .
+      . IYIX      . IYIX      . IREIN      .
C.....
C TORQUE OCCUPANT ALONE COMMON BLOCK
C.....
+      COMMON /TORQQA / TLCHOA(3) , TMCHOA(3) , TNCHOA(3) ,
+      TLAEOA , TMAEOA , TNAEOA
C.....
C TORQUE SEAT ALONE COMMON BLOCK
C.....
+      COMMON /TORQSA / TLAESA , TMAESA , TNAESA
C.....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
+      COMMON /TORQSD / TLCASD(2) , TMCASD(2) , TNCASD(2) ,
+      TLTUBSD , TMTUBSD , TINTUBSD ,
+      TLSLSD(6) , TMSLSD(6) , TNSLSD(6) ,
+      TLRKSD(6) , TMRKSD(6) , TNRKSD(6) ,
+      TLCHSD(3) , TMCHSD(3) , TNCHSD(3) ,
+      TLAESO , TMAESO , TNAESO ,
+      TLDRTSD , TMDRTSD , TNDRTSD
C.....
C AERODYNAMICS INFORMATION COMMON BLOCK
C.....
+      COMMON /AEROCFS / QALPH , QABETA , QAVEL , QAMACH ,
+      SAALPH , SABETA , SAVEL , SAMACH ,
+      SOALPH , SOBETA , SOVEL , SOMACH ,
+      CXQA , CYQA , CZQA , CLOA , CMOA , CNOA ,
+      CXSA , CYSA , CZSA , CLSA , CMSA , CNSA ,
+      CXSO , CYSO , CZSO , CLSO , CMSO , CNSO
LU = 8IAS + 24
IF(1EVENTS(28) .NE. 0) GO TO 9900
IF (PRTFRQ .LE. 0) GO TO 100
IPRTCNT(24) = MOD (IPRTCNT(24)+1, PRTFRQ)
IF(1EVLNE .NE. 0) GO TO 100
IF(IPRTCNT(24) .NE. 0) GO TO 9900
100 CONTINUE
IF(1LINECT(24) .LE. MAXLINE) GO TO 200
CALL HEADER
WRITE(LU,7000) PRTWGHT(PRTINDX) , PRTWGHT(PRTINDX) ,
+      PRTLNGT(PRTINDX) , PRTLNGT(PRTINDX)
IF(PRTFRQ LT. 0) GO TO 9900
200 CONTINUE
RESULT1=SORT(FXAESD*FXAESD + FYAESD*FYAESD + FZAESD*FZAESD)
RESULT2=SORT(TLAESO*TLAESO + TMAESO*TMAESO + TNAESO*TNAESO)
WRITE(LU,7010) TIME
+      FXAESD , FYAESD , FZAESD , RESULT1
+      TLAESO , TMAESO , TNAESO , RESULT2
+      SOALPH , SOBETA , SOVEL , SOMACH
1LINECT(24) = LINECT(24) + 1
GO TO 9900
9900 CONTINUE
RETURN
C.....
C WRITE FORMAT STATEMENTS
C.....

```



```

115 C.....
7000 FORMAT(/5X,"TIME",20X,"FORCES (SCS)",29X,"MOMENTS (SCS)",26X,
+ "AERODYNAMICS (WCS)",5X,"(SEC)",22X,"( ",A2," )",37X,"( ",A2," ",A2,
+ " )",21X,"ALPHA BETA VEL MACH",19X,"X",9X,"Y",9X,
+ "Z",8X,"RES",10X,"L",8X,"M",9X,"N",8X,"RES",8X,"(DEG)",5X,"(DEG)",
+ "X",(" ",A2,"-SEC)",6X,"NO.",/)
120 7010 FORMAT(1X,F9.4,4X,4(F8.1,2X),2X,4(F8.1,2X),4(2X,F8.2))
      END

```

```

1  SUBROUTINE REPT25
C .....
C .....
C OCCUPANT ALONE AERODYNAMIC FORCES AND MOMENTS REPORT
C .....
5  C OCCUPANT ALONE FORCES COMMON BLOCK
C .....
COMMON /FORCEA / FXCHOA(3) , FYCHOA(3) , FZCHOA(3) ,
+   FXAEOA , FYAEOA , FZAEOA
C .....
10 C SEAT ALONE FORCES COMMON BLOCK
C .....
COMMON /FORCEB / FXAESB , FYAESB , FZAESB
C .....
15 C SEAT/OCCUPANT FORCES COMMON BLOCK
C .....
COMMON /FORCEC / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+   FXTUBSO , FYTUBSO , FZTUBSO ,
+   FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+   FXRKS(6) , FYRKS(6) , FZRKS(6) ,
+   FXCHSO(3) , FYCHSO(3) , FZCHSO(3) ,
+   FXAESO , FYAESO , FZAESO ,
+   FYDRTSO , FZDRTSO
C .....
25 C SECTION 2 COMMON BLOCK
C .....
COMMON /IREPT / IREPTS(31) , PRTRFO,P11,P12,P13
INTEGER
C .....
30 C MISCELLANEOUS DATA COMMON BLOCK
C .....
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+   MAXLINE , MAXREPT , MAXEVRT ,
+   IEVLINE , IERRFLG , LU ,
+   IDATE , HEADALT , HEADVEL ,
+   HEADSR , HEADYAW , HEADPIT ,
+   HEADROL , HEADWGT , BIAS ,
+   REPTYPE(5,31) , PRTLNCT(2) , PRTWGT(2) ,
+   IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+   IMVDC , PRTMP( 2) ,
+   PRTMASS(2) , PRTINDX , PKZVEL ,
+   ZVECT(3) , XYZ(3) , SAVTIME ,
+   XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+   REPTYPE , BIAS , PRTLNCT ,
+   PRTWGT , PRTWGT , PRTWGT ,
+   PRTMP , PRTMASS , PRTINDX
C .....
50 C INTEGRATION ROUTINE COMMON BLOCK
C .....
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193) ,
+   TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3) ,
+   TRAJAC(193) , TVCEOS(225) , QUATSO(65) ,
+   QUATSA(65) , QUATDA(65) , QUATAC(65) ,
+   INTSTP , IPCPASS , IRKPASS ,
+   IPOINTS , IYPRX , IYPRX ,
+   IKX , IKSUMX , IKPASSX ,
+   IYIX , IYIX , IYIX

```

```

60      + IY13X      , IYPR1X      , IYPR11X      ,
      + IYPR12X      , IYV1X      , IYV11X      ,
      + IYV1X      , IYV11X      , IYV11X      ,
C.....
C TORQUE OCCUPANT ALONE COMMON BLOCK
C.....
65      COMMON /TORQDA / TLCHOA(3) , TNCCHOA(3) , TNCCHOA(3) ,
      + TLAEDA      , TMAEDA      , TNAEDA      ,
C.....
C TORQUE SEAT ALONE COMMON BLOCK
C.....
70      COMMON /TORQSA / TLAESA      , TMAESA      , TNAESA      ,
C.....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
75      COMMON /TORQSO / TMCASO(2) , TMCASO(2) , TNCASO(2) ,
      + TLTUBSO      , TMTUBSO      , TMTUBSO      ,
      + TMSLSO(6)      , TMSLSO(6)      , TMSLSO(6)      ,
      + TMRKSO(6)      , TMRKSO(6)      , TMRKSO(6)      ,
      + TLCHSO(3)      , TLCHSO(3)      , TNCCHSO(3)      ,
      + TLAESO      , TMAESO      , TNAESO      ,
      + TLDRTSO      , TMDRTSO      , TNDRTSO      ,
C.....
C AERODYNAMICS INFORMATION COMMON BLOCK
C.....
80      COMMON /AEROCFS / OALPH      , OABETA      , DAVEL      , OAMACH      ,
      + SAALPH      , SABETA      , SAVED      , SAMACH      ,
      + SOALPH      , SOBETA      , SOVEL      , SOMACH      ,
      + CXOA      , CYOA      , CZOA      , CMOA      , CNOA      ,
      + CXSA      , CYSA      , CZSA      , CLSA      , CNSA      ,
      + CXSO      , CYSO      , CZSO      , CLSO      , CNSO      ,
      LU = BIAS + 25
      IF (PRTFRQ .LT. O) GO TO 100
      IF (IEVENTS(28) .EQ. O) GO TO 9900
      IF (PRTFRQ .EQ. O) GO TO 100
      IPRCNT(25) = MOD (IPRCNT(25)+1, PRTFRQ)
      IF (IEVLN .NE. O) GO TO 100
      IF (IPRCNT(25) .NE. O) GO TO 9900
100     CONTINUE
      IF (LINECT(25) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE(LU,7000) PRTWGHT(PRTINDX) , PRTWGHT(PRTINDX) ,
      + PRTLNGT(PRTINDX) , PRTLNGT(PRTINDX)
200     CONTINUE
      RESULT1=SQRT(FXAEOA*FXAEOA + FYAEOA*FYAEOA + FZAEOA*FZAEOA)
      RESULT2=SQRT(TLAEDA*TLAEDA + TMAEDA*TMAEDA + TNAEDA*TNAEDA)
      WRITE(LU,7010) TIME
      + FXAEOA      , FYAEOA      , FZAEOA      , RESULT1      ,
      + TLAEDA      , TMAEDA      , TNAEDA      , RESULT2      ,
      + OALPH      , OABETA      , OAVEL      , OAMACH      ,
      LINECT(25) = LINECT(25) + 1
      GO TO 9900
9900    CONTINUE
      RETURN
C.....
C WRITE FORMAT STATEMENTS

```

PAGE 249

83/11/07 09.41.53

FTN 4 6+428

SUBROUTINE REPT25 74/74 OPT=1

```

115 C.....
      7000 FORMAT(/5X,"TIME",20X,"FORCES (SCS)",29X,"MOMENTS (SCS)",26X,
      + "AERODYNAMICS (WCS)",5X,"(SEC)",22X,"(",A2,")",37X,"(",A2,")",A2,
      + ")",21X,"ALPHA",BETA,VEL,MACH"/,19X,"X",9X,"Y",9X,
      + "Z",8X,"RES",10X,"L",9X,"M",9X,"N",8X,"RES",8X,"(DEG)",5X,"(DEG)",
      + "X",("A2,"-SEC)",6X,"NO.",/)
120 7010 FORMAT(1X,F9.4,4X,4(F8.1,2X),2X,4(F8.1,2X),4(2X,F8.2))
      END

```


5-252

```

115      C WRITE FORMAT STATEMENTS
      C.....
      7000 FORMAT(1/5X,"TIME",20X,"FORCES (SCS)",29X,"MOMENTS (SCS)",26X,
      + "AERODYNAMICS (WCS)",5X,"(SEC)",22X,"(A2)",37X,"(A2)",A2,
      + "A2",21X,"ALPHA BETA VEL MACH",19X,"X",9X,"Y",9X,
      + "Z",8X,"RES",10X,"L",9X,"M",9X,"N",8X,"RES",8X,"(DEG)",5X,"(DEG)",
      + "3X,"(A2)",A2,"-SEC)",6X,"NO.",/)
      7010 FORMAT(1X,F9.4,4X,4(F8.1,2X),2X,4(F8.1,2X),4(2X,F8.2))
      END
120

```

PAGE 253

83/11/07 09.41.53

FTN 4 6*428

SUBROUTINE REPT27 74/74 DPT=1

```

1      SUBROUTINE REPT27
      C.....
      C
      C OCCUPANT ALONE ANGULAR TIME HISTORY REPORT
      C ***** NOT CURRENTLY IMPLEMENTED *****
      C.....
      RETURN
      END
5

```


5-255

5-256

AD-A148 363

A GENERALIZED ESCAPE SYSTEM SIMULATION (GESS) COMPUTER
PROGRAM VOLUME 2 G... (U) KETRON INC WARMINSTER PA
L A D' AULERIO ET AL. APR 84 NADC-84068-60

4/4

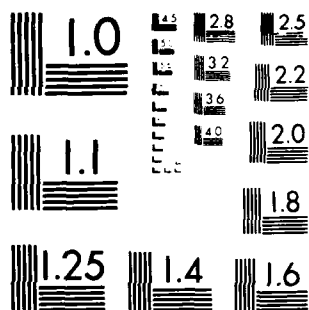
UNCLASSIFIED

N62269-81-C-0206

F/G 9/2

NL

END
DATE
FILMED
1-85
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

5-257

```

60      IF(IEVLIN .NE. 0) GO TO 100
      IF(IPRINT(29) .NE. 0) GO TO 9900
      100 CONTINUE
      IF(LINECT(29) .LE. MAXLINE) GO TO 200
      CALL HEADER
      WRITE (LU,7000)
      IF(PRTFRQ .LT. 0) GOTO 9900
      200 CONTINUE
      FROLLA = ZARCTAN(DCMAE(2,3),DCMAE(3,3)) * RADDEG
      FPITCHA = -ASIN( DCMAE(1,3)) * RADDEG
      FYAWA = ZARCTAN(DCMAE(1,2),DCMAE(1,1)) * RADDEG
      PACC = TRAJAC(23) * RADDEG
      QACC = TRAJAC(24) * RADDEG
      RACC = TRAJAC(25) * RADDEG
      PVEL = TRAJAC(11) * RADDEG
      OVEL = TRAJAC(12) * RADDEG
      RVEL = TRAJAC(13) * RADDEG
      QVELA = OVEL
      RVELA = RVEL
      RESULT1 = SORT(PACC*PACC + QACC*QACC + RACC*RACC)
      RESULT2 = SORT(PVEL*PVEL + QVELA*QVELA + RVELA*RVELA)
      RESULT3 = SORT(FROLLA*FROLLA + FPITCHA*FPITCHA + FYAWA*FYAWA)
      80      WRITE(LU,7010) TIME
      +      PACC      QACC      RACC      RESULT1
      +      PVEL      QVELA      RVELA      RESULT2
      +      FROLLA      FPITCHA      FYAWA      RESULT3
      LINECT(29) = LINECT(29) + 1
      9000 CONTINUE
      RETURN
      C*****
      C WRITE FORMAT STATEMENTS
      C*****
      7000 FORMAT(/,5X,"TIME",17X,"ACCELERATION (ACS)",24X,"RATE (ACS)",
      +      28X,"ORIENTATION (EFGS)",5X,"(SEC)",18X,"(DEG/SEC 2)",30X,
      +      -(DEG/SEC)",36X,"(DEG)",/,
      +      17X,2(4X,"X",9X,"Y",9X,"Z",7X,"RES",5X),
      +      3X,"ROLL",5X,"PITCH",6X,"YAW",6X,"RES",/)
      7010 FORMAT(1X,F9.4,2X,3(4(F10.2),1X))
      END

```

```

1 SUBROUTINE REPRT30
C
C
5 C AERODYNAMIC COEFFICIENTS TIME HISTORY REPORT
C
C
C SECTION 2 COMMON BLOCK
C
COMMON /IREPORT / IREPTS(31) , PRTRQ,P11,P12,P13
INTEGER PRTRQ,P11,P12,P13
C
C MISCELLANEOUS DATA COMMON BLOCK
C
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+ MAXLINE , MAXREPT , MAXEVNT ,
+ IERRFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWTG , BIAS ,
+ REPTYPE(5,31) , PRTLNGT(2) , PRTWGHT(2) ,
+ IHEADER(24) , IEVENTS(38) , TIMES(38) ,
+ IMVDC , PRTEMP( 2) ,
+ PRTMASS(2) , PRTINDX , PKZVEL ,
+ ZVECT(3) , XYZ(3) , SAVTIME ,
+ XACCEL(3) , YACCEL(3) , ZACCEL(3) ,
+ REPTYPE , BIAS , PRTLNGT ,
+ PRTWGHT , PRTEMP , PRTMASS , PRTINDX
+
C
30 C INTEGRATION ROUTINE COMMON BLOCK
C
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193) ,
+ TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3) ,
+ TRAJAC(193) , TVCEOS(225) , QUATSO(65) ,
+ QUATSA(65) , QUATOA(65) , QUATAC(65) ,
+ INTSTP , IPCPASS , IRKPASS ,
+ IPOINTS , IYX , IYPRX ,
+ IKX , IKSUNX , IKPASSX ,
+ IYIX , IY13X , IY12X ,
+ IYPRIX , IYPR1X ,
+ IYPR12X , IY11X ,
+ ICYIX , ICY1X , IREIN
+
C
45 C AERODYNAMICS INFORMATION COMMON BLOCK
C
COMMON /AEROCFS / QALPH , QABETA , QAVEL , QAMACH ,
+ SIALPH , SABETA , SVEL , SAMACH ,
+ SDALPH , SOBETA , SOVEL , SOMACH ,
+ CXOA , CYOA , CZOA , CMOA , CNOA ,
+ CXSA , CYSA , CZSA , CLSA , CMSA ,
+ CXSO , CYSO , CZSO , CLSO , CMSO , CNSO
+
REAL MACH
IF(INTSTP .EQ. 0) RETURN
LU = BIAS + 30
IF(IEVENTS(28) .NE. 0) GOTO 60
ALPHA = SIALPH
BETA = SOBETA

```

SUBROUTINE REPR130 74/74 OPT=1 FTN 4.6*428 83/11/OT. 09.41.53 PAGE 259

```

60      MACH = SOMACH
        GOTO 51
50      ALPHA = SAALPH
        BETA = SABETA
        MACH = SAMACH
51      IF (PRTRQ .LE. 0) GO TO 100
        IPRCNT(30) = MOD (IPRCNT(30)+1, PRTRQ)
        IF (IEVLIN .NE. 0) GO TO 100
        IF (IPRCNT(30) .NE. 0) GO TO 9900
100     CONTINUE
        IF (LINECT(30) .LE. MAXLINE) GOTO 200
        CALL HEADER
        WRITE(LU,7000)
        IF (PRTRQ .LT. 0) GOTO 9900
200     CONTINUE
        WRITE(LU,7010) TIME , ALPHA , BETA , MACH
        +      CXSO , CYSO , CZSO , CLSO , CNSO
        +      CXSA , CYSa , CZSA , CLSA , CNSA , CNSA
        LINECT(30) = LINECT(30) + 1
        GO TO 9900
9900    CONTINUE
        RETURN
C*****
C WRITE FORMAT STATEMENTS
C*****
7000    FORMAT(/5X,"TIME",6X,"ALPHA",3X,"BETA",4X,"MACH",21X,
+ "SEAT/OCCUPANT",38X,"SEAT ALONE",/5X,"(SEC)",5X,"(DEG)",
+ 3X,"(DEG)",4X,"NO.",25X,"(WCS)",45X,"(WCS)",/33X,2(9X,
+ "X",7X,"Y",7X,"Z",7X,"L",7X,"M",7X,"N"),/)
7010    FORMAT(1X,F8.4,2X,3F8.2,2(2X,6F8.4))
        END

```



```

1 SUBROUTINE REPT31
C
C DYNAMIC CG REPORT
C
C SECTION 2 COMMON BLOCK
C
COMMON /IREPORT / IREPTS(31) , PRTRQ,PI1,PI2,PI3
INTEGER
C
C CONSTANTS COMMON BLOCK
C
COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C
C MISCELLANEOUS DATA COMMON BLOCK
C
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31)
+ MAXREPT , MAXEVRT , LU
+ IEVLNE , IERRFLG , HEADVEL , HEADPIT
+ HEADSR , HEADALT , HEADYAW , HEADWGT , BIAS
+ HEADROL , REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2)
+ IHEADER(24) , IEVENTS(38) , TIMES(38)
+ INVDC , PRTMP( 2)
+ PRIMASS(2) , PRTINDX , PKZVEL
+ ZVECT(3) , XYZ(3) , SAVTIME
+ XACCEL(3) , YACCEL(3) , ZACCEL(3)
+ INTEGER , REPTYPE , BIAS , PRTLNGT
+ PRTWGT , PRTMP , PRIMASS , PRTINDX
C
C INTEGRATION ROUTINE COMMON BLOCK
C
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193)
+ TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3)
+ TRAJAC(193) , TVCEQS(225) , QUATSO(65)
+ QUATSA(65) , QUATA(65) , IRKPASS
+ INTSTP , IPCPASS , IYPRX , IYPRX
+ IPPOINTS , IKX , IKSUMX , IKPASSX
+ IKX , IYIX , IYIX , IYIX
+ IY13X , IYPR1X , IYPR1X
+ IYPR12X , IYPR1X , IYPR1X
+ ICYIX , ICYIX , IREIN
C
C SECTION 6 COMMON BLOCK
C
COMMON /ISEATOC / IPCNTL , XCGSD , YCGSD , ZCGSD , IXXSO
+ IAYSO , IYXSO , IYYSO , IZZSO
+ AREASO , AREAOA , WGTTOAB , WGTTOAA
+ IXXDA , IAYDA , IYODA , IYODA , IYODA
+ IZZDA , XCGDA , YCGDA , ZCGDA , SOSEP
+ C150 , C250 , C350 , C450
+ C10A , C20A , C30A , C40A
+ REAL , IAXSO , IAYSO , IYXSO , IYXSO , IYXSO

```

```

+      IZZSO , IXAOA , IXVOA , IXZDA , IYVOA ,
+      IYZOA , IZZOA
C .....
C DYNAMIC CG VARIABLES COMMON BLOCK
C .....
COMMON /DYNCGVB / CGVAL(8) , CGDERV(6) , ZCGDAO
+      XCGDAO , YCGDAO , ZCGDAO ,
+      DXCG , DYCG , DZCG
C
DIMENSION SOACCEL(4)
IF(INTSTP.EQ.0) RETURN
LU = BIAS + 31
IF(IEVENTS(28).EQ.1) RETURN
IF (PRTRQ .LE. 0) GOTO 100
IPRTCNT(31) = MOD (IPRTCNT(31)+1, PRTRQ)
IF(IEVLNE.NE.0) GOTO 100
IF(IPRTCNT(31).NE.0) RETURN
100 CONTINUE
IF(LINECT(31).LE.MAXLINE) GOTO 200
CALL HEADER
WRITE(LU,7000) PRTLNCT(PRTINDX) , PRTLNCT(PRTINDX)
IF(PRTFRQ.LT.0) RETURN
200 CONTINUE
SOACCEL(1) = XACCEL(1) / GRAVITY
SOACCEL(2) = YACCEL(1) / GRAVITY
SOACCEL(3) = ZACCEL(1) / GRAVITY
SOACCEL(4) = SORT(SOACCEL(1) + SOACCEL(2) +
+ SOACCEL(3) + SOACCEL(4))
DISSDCG = SORT(CGVAL(1) * CGVAL(1) + CGVAL(3) * CGVAL(3) +
+ CGVAL(5) * CGVAL(5))
SOCGLC = SORT(XCGSO * XCGSO + YCGSO * YCGSO + ZCGSO * ZCGSO)
C
WRITE(LU,7010) TIME , SOACCEL ,
+      CGVAL(1) , CGVAL(3) , CGVAL(5) , DISSDCG ,
+      XCGSO , YCGSO , ZCGSO , SOCGLC
LINECT(31) = LINECT(31) + 1
RETURN
C .....
C WRITE FORMAT STATEMENTS
C .....
7000 FORMAT(/,5X,"TIME",T20,"SEAT/OCCUPANT ACCELERATION (SCS)",T65,
+ "DISPLACEMENT OF O/A CG (SCS)",T107,"S/O CG LOCATION (SCS)",T134,
+ "(GLS)",T175,"(A2,")",T116,"(A2,")/",T10X,"(10X,"X",9X,"Y",
+ 9X,"Z",8X,"RES")")
7010 FORMAT(1X,F9.4,1X,3(2X,F10.4))
END

```

```

1      SUBROUTINE REPTDRI
C.....
C
C POST SIMULATION REPORT OF MAXIMUM VALUES OF DRI VARIABLES
C
5      C CALLED BY: DRICALC
C.....
C
10     C CONSTANTS COMMON BLOCK
C.....
C      COMMON /CONSTNT / GRAVITY , RADEG , DEGRAD , PI
C MISCELLANEOUS DATA COMMON BLOCK
C.....
15     C      COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCTNT(31) ,
+      MAXLINE , MAXREPT , MAXEVNT
+      IEVLINE , IERRFLG , LU
+      IDATE , HEADALT , HEADVEL
+      HEADSR , HEADYAW , HEADPIT
+      HEADROL , HEADWGT , BIAS
+      REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2)
+      IHEADER(24) , IEVENTS(38) , TIMES(38)
+      PRTMP( 2)
+      PRTHSS(2) , PRTHNDX
+      ZVECT(3) , XYZ(3) , SAVTIME
+      XACCEL(3) , YACCEL(3) , ZACCEL(3)
+      REPTYPE , BIAS , PRTLNGT
+      INTEGER
+      PRTWGT , PRTEMP , PRTHSS , PRTHNDX
C.....
30     C DYNAMIC RESPONSE INDEX VARIABLES COMMON BLOCK
C.....
C      COMMON /DRIVRB / DRIVAL(2) , DRIDERV(2) , ACCEL1
+      DRI , DRIMAX , TMAX
+      ZACCMAX , DRICON
+      LU = BIAS + 1
+      CALL HEADER
+      WRITE(5,900)
+      900 FORMAT(5(/),52X,"END OF SIMULATION"/,47X,"DRI VARIABLE MAXIMUM ",
+      "VALUES")
+      999 FORMAT(5(/),50X,"MAX DRI = ",F10.4//,50X,"AT TIME = ",F10.4,
+      " SEC")
+      RETURN
+      END

```

SUBROUTINE RESTART 74/74 OPT=1 FTN 4.6+428 83/11/07. 09.41.53 PAGE 263

```

1      SUBROUTINE RESTART
      C .....
      C VERSION KGESSAB - DATED 7 NOVEMBER 1983
      C .....
      RETURN
      END
5

```

5-265

[illegible]

```

115 C .....
C CHECK NUMBER OF ROCKETS - RETURN IF NONE
C .....
C IF(IINRKT.EQ. IROKOUT) RETURN
C
C DO 500 I=1,IINRKT
C .....
C CHECK FOR ROCKET IGNITION - ONLY AT THE END OF SUCCESSFUL
C INTEGRATION TIME STEP
C .....
IF(IEVENTS(I+11).GE. 1) GOTO 500
IF(IEVENTS(I+ 5).GE. 1) GOTO 30
IF(INTSTP.EQ. 0) GOTO 500
IF(RKIGN(I).GT. ABS(ZRRSBO1)) GOTO 500
C .....
C ROCKET IGNITION
C .....
IF(TSTAR(I).LT. 0.0) TSTAR(I) = TIME + RKDELY(I)
IF(TSTAR(I).GT. TIME5) GOTO 500
IEVENTS(I+5) = 1
TIME5(I+5) = TIME
GOTO 250
30 IF(INTSTP.EQ. 0) GOTO 250
C .....
C CHECK FOR ROCKET BURNOUT
C .....
IF(TIME5.LT. (TIME(I+5) + RKURN(I))) GOTO 250
C .....
C ROCKET BURNOUT
C .....
IEVENTS(I+11) = 1
TIME5(I+11) = TIME
IROKOUT = IROKOUT + 1
TVCEQS(I) = 0.0
MASSSO=MASSO-MASSRK(I)
MASSSA=MASSA-MASSRK(I)
FVRKSO(I) = FVRKSO(I) + FZRKSO(I) = 0.0
TLRKSO(I) = TMRKSO(I) = TMRKSO(I) = 0.0
GOTO 500
155 C .....
C .....
C COMPUTE ROCKET THRUST COMPONENTS IN SCS
C .....
C .....
250 CONTINUE
RKTIME = TIME - TIME(I+5)
CALL ZLININT(RKTIME, RKOUT(1,1,1), RKNPPTS(1),25,CTR,2)
FVRKSO(I) = CTR*URX(I)
FVRKSO(I) = CTR*URY(I)
FZRKSO(I) = CTR*URZ(I)
C .....
C .....
C COMPUTE ROCKET MOMENTS
C .....
C .....

```

SUBROUTINE RKTFM 74/74 OPT=1 FTN 4 6+428 83/11/07 09.41.53 PAGE 267

```

175      TLKSO(I) = YSSORK(I) * FZKSO(I) - ZSSORK(I) * FYKSO(I)
      TRKSO(I) = ZSSORK(I) * FXKSO(I) - XSSORK(I) * FZKSO(I)
      TRKSO(I) = XSSORK(I) * FYKSO(I) - YSSORK(I) * FXKSO(I)
      C .....
      C .....
      C COMPUTE ROCKET FUEL BURNOFF FROM SEAT/OCCUPANT MASS
      C .....
      C .....
      C      MASSO = MASSO - RKTIME/RKBURN(I) * MASSRK(I)
      C
      C 500 CONTINUE
      RETURN
      END

```



```

1 SUBROUTINE ROTATE(XFROM,XTO,XSUB,MATRIX,1)
C.....
C DESCRIPTION - LEVEL 4
C FUNCTION - TRANSFORMS A VECTOR FROM ONE COORDINATE SYSTEM TO
C ANOTHER
C METHOD - IN TWO SEPARATE FUNCTIONS, ROTATE SUBTRACTS ONE
C VECTOR FROM ANOTHER, AND TRANSFORMS THE COORDINATES
C OF A VECTOR PRESENTED IN ONE COORDINATE SYSTEM,
C TO ANOTHER COORDINATE SYSTEM.
C XFROM - VECTOR IN ORIGINAL COORDINATE SYSTEM
C XTO - VECTOR IN REQUESTED COORDINATE SYSTEM
C XSUB - VECTOR TO BE SUBTRACTED FROM XFROM
C MATRIX - MATRIX USED TO PERFORM THE TRANSFORMATION
C I - INDICATES WHETHER THE ORIGINAL MATRIX, OR
C ITS TRANSPOSE, SHOULD BE USED
C = 0, USE ORIGINAL MATRIX
C = 1, USE TRANSPOSE OF ORIGINAL MATRIX
C COMMUNICATIONS -
C CALLED BY:
C AERFMSA, AERFMSO, AIRCRAFT, CHUINIT, CHUTFM,
C DARTEN, DROGUE1, DROGUE2, DYNAMICG, INITRAJ,
C INIVRBL, RAILFM, SEATALN, SEATOC, SEPINIT,
C SLUGCON, TUBEND, UPOVECT, VERTSK
C CALLS:
C NONE
C NON-COMMON VARIABLES DEFINED:
C XDIF - DIFFERENCE BETWEEN THE COORDINATES OF THE VECTOR IN
C YDIF - THE ORIGINAL COORDINATE SYSTEM AND THOSE IN THE
C ZDIF - REQUESTED SYSTEM
C POTENTIAL ERROR CONDITIONS: NONE
C.....
C REAL MATRIX
C DIMENSION XFROM(3), XTO(3), XSUB(3), MATRIX(3,3)
C
C XDIF = XFROM(1) - XSUB(1)
C YDIF = XFROM(2) - XSUB(2)
C ZDIF = XFROM(3) - XSUB(3)
C
C IF(1.EQ. 1) GOTO 10
C
C XTO(1) = MATRIX(1,1) * XDIF + MATRIX(1,2) * YDIF +
C +MATRIX(1,3) * ZDIF
C XTO(2) = MATRIX(2,1) * XDIF + MATRIX(2,2) * YDIF +
C +MATRIX(2,3) * ZDIF
C XTO(3) = MATRIX(3,1) * XDIF + MATRIX(3,2) * YDIF +
C +MATRIX(3,3) * ZDIF
C GOTO 20
C
C 10 XTO(1) = MATRIX(1,1) * XDIF + MATRIX(2,1) * YDIF +
C +MATRIX(3,1) * ZDIF
C XTO(2) = MATRIX(1,2) * XDIF + MATRIX(2,2) * YDIF +
C +MATRIX(3,2) * ZDIF
C XTO(3) = MATRIX(1,3) * XDIF + MATRIX(2,3) * YDIF +
C +MATRIX(3,3) * ZDIF
C 20 CONTINUE

```

NADC-84068-60

SUBROUTINE ROTATE 74/74 OPT-1 FIN 4.6428 83/11/07. 09.41.53 PAGE 269

RETURN
END

```

1 SUBROUTINE RUNGE(N,Y,F,X,H,IP)
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - INTEGRATE THE DYNAMIC CG AND DYNAMIC RESPONSE INDEX
C EQUATIONS OF MOTION
C METHOD - THE SUBROUTINE RUNGE EMPLOYS THE FOURTH ORDER
C RUNGE KUTTA METHOD WITH RUNGE'S COEFFICIENTS
C TO INTEGRATE A SYSTEM OF N SIMULTANEOUS FIRST ORDER
C DIFFERENTIAL EQUATIONS  $F(J)=DY(J)/DX$ ,  $J=(1,2,...,N)$ .
C ACROSS ONE STEP OF LENGTH H IN THE INDEPENDENT
C VARIABLE X, SUBJECT TO INITIAL CONDITIONS  $Y(J)$ ,
C ( $J=1,2,...,N$ ). EACH  $F(J)$ , THE DERIVATIVES OF  $Y(J)$ ,
C MUST BE COMPUTED FOUR TIMES PER INTEGRATION STEP BY
C THE CALLING PROGRAM. THE ROUTINE MUST BE CALLED 4
C TIMES PER STEP (PASS(1)...PASS(4)) SO THAT THE
C INDEPENDENT VARIABLE (X) AND THE SOLUTION VALUES
C ( $Y(1)...Y(N)$ ) CAN BE UPDATED USING THE RUNGE-KUTTA
C ALGORITHM. THIS ROUTINE WAS DERIVED FROM
C PAGE 374 OF CARNAHAN, LUTHER AND WILKES:
C APPLIED NUMERICAL METHODS.
C
C COMMUNICATIONS -
C CALLED BY: DRICALC
C DYNAMCG
C
C CALLS: NONE
C
C NON-COMMON VARIABLES DEFINED -
C SAVEF - SAVES PREVIOUS VALUES OF  $F(J)$ 
C SAVEY - SAVES PREVIOUS VALUES OF  $Y(J)$ 
C Y - SIMULTANEOUS DIFFERENTIAL EQUATIONS
C F - DERIVATIVES FOR Y
C N - NUMBER OF EQUATIONS
C X - PREVIOUS TIME
C H - TIME STEP
C IP - PASS COUNTER
C
C POTENTIAL ERROR CONDITIONS
C NONE
C.....
C DIMENSION SAVEF(50),SAVEY(50),Y(N),F(N)
C GO TO (10,20,30,40) IP
C
C PASS 1
C
10 CONTINUE
C DO 11 J=1,N
C SAVEY(J)=Y(J)
C SAVEF(J)=F(J)
C Y(J)=SAVEY(J)+.5*H*F(J)
C
11 CONTINUE
C X=X+.5*H
C RETURN
C
C PASS 2
C
20 CONTINUE
C DO 22 J=1,N
C SAVEF(J)=SAVEF(J)+2.*F(J)
C Y(J)=SAVEY(J)+.5*H*F(J)
C
22 CONTINUE

```

PAGE 271

83/11/07 09.41.53

FTN 4.6+428

SUBROUTINE RUNGE 74/74 OPT=1

```

60      C
        C PASS 3
        C
        30      CONTINUE
              DO 33 J=1,N
                SAVEF(J)=SAVEF(J)+2.0*F(J)
                Y(J)=SAVEY(J)+H*F(J)
              33 CONTINUE
                X=X+.5*H
                RETURN
        C
        C PASS 4
        C
        40      CONTINUE
              DO 44 J=1,N
                Y(J)=SAVEY(J)+(SAVEF(J)+F(J))*H/6.0
              44 CONTINUE
                RETURN
        END

```

```

1  SUBROUTINE SEATLN
C.....
C DESCRIPTION - LEVEL 2
C FUNCTION - CALCULATES SEAT ALONE TRAJECTORY PARAMETERS
C METHOD - SEAT ALONE EQUATIONS OF MOTION ARE COMPUTED FROM
C SEAT/OCCUPANT SEPARATION TO SEAT IMPACT. AERODYNAMIC
C FORCES ARE USED TO CALCULATE LINEAR ACCELERATIONS
C IN THE SEAT COORDINATE SYSTEM.
C COMMUNICATIONS -
C CALLED BY:
C GESS
C CALLS:
C ATMOS, AERFMSA, ROTATE
C NON-COMMON VARIABLES DEFINED:
C OLDALT(3) - LATEST PREVIOUS ALTITUDE OF THE SEAT ALONE
C GRAVITY - ENGLISH OR METRIC VALUE OF ACCELERATION DUE
C TO GRAVITY
C POTENTIAL ERROR CONDITIONS: NONE
C.....
C CONSTANTS COMMON BLOCK
C.....
C COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C.....
C DENSITY COMMON BLOCK
C.....
C COMMON /DENSITY / IATMOS , OLDALT(3) , RHOS
C PRESALT(3) , DTEMP ,
C TEMPS ,
C VFWIND , VYVWIND , VZWIND
C.....
C SEAT ALONE FORCES COMMON BLOCK
C.....
C COMMON /FORCESA / FXAESA , FYAESA , FZAESA
C.....
C SECTION 1 COMMON BLOCK
C.....
C COMMON /ICONTRL / TSTART , TSTOP , ESTOP , IRESTRT, IUNITS ,
C ISEATTR, ISOSEP , IPLOT , IDRIFLG,
C IPHASE1, IPHASE2, IPHASE3
C.....
C INTEGER
C ESTOP
C.....
C SECTION 5 COMMON BLOCK
C.....
C COMMON /ISETALN / XPOSSRP, YPOSSRP, ZPOSSRP, XCGSA , YCGSA ,
C ZCGSA , IXXSA , IXYSA , IXZSA , IYVSA ,
C IYZSA , IZZSA , PHISA , PSISA , THESA ,
C AREASA , HGHTSA , WGH TSA , XPOSBOT, YPOSBOT,
C ZPOSBOT, XPOSSCS, YPOSSCS, ZPOSSCS,
C C1SA , C2SA , C3SA , C4SA
C.....
C REAL
C IXXSA , IXYSA , IXZSA , IYVSA , IYZSA ,
C IZZSA
C.....
C MASSES COMMON BLOCK
C.....
C COMMON /MASSES / MASSOA1 , MASSOA2 , MASSSO , MASSO
C MASSSA , MASSRK(6) , MASSDC
C.....

```

```

      REAL
      + MASSOA1 , MASSOA2 , MASSSO , MASSO
      + MASSA , MASSRK , MASSDC
C.....
C MATRIX COMMON BLOCK
C.....
      COMMON /MATRIX / DCMAE(3,3) , DCMA(3,3) , DCMSA(3,3) ,
      + DCME(3,3) , DCMTS(3,3) , DCMTS(3,3) ,
      + DCMSAE(3,3) , DCMDAE(3,3) , DCMSR(3,3) ,
      + DCMDUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
      COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
      + MAXLINE , MAXREPT , MAXEVT ,
      + TEVLIN , TERRFLG , LU ,
      + IDATE , HEADALT , HEADVEL ,
      + HEADSR , HEADWGT , HEADPIT ,
      + HEADROL , HEADWGT , BIAS ,
      + REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
      + IHEADER(24) , IEVENTS(38) , TIMES(38) ,
      + INVDC , PRTMP( 2) ,
      + PRTMAS(2) , PRTINDX , PKZVEL ,
      + ZVECT(3) , XYZ(3) , SAVTIME ,
      + XACCEL(3) , YACCEL(3) ,
      + REPTYPE , ZACCEL(3) ,
      + PRTWGT , BIAS , PRTLNGT ,
      + PRTMP , PRTINDX , PRTINDX
C.....
C INTEGRATION ROUTINE COMMON BLOCK
C.....
      COMMON /RKUTTA / TIME , TIMEB , DELTAT , TRAJSO(193) ,
      + TRAJSA(193) , TRAJOA(193) , TRAJCH(97,3) ,
      + TRAJAC(193) , TVCEOS(225) , QUATSO(65) ,
      + QUATSA(65) , QUATOA(65) , QUATAC(65) ,
      + INTSTP , IPCPASS , IRKPASS ,
      + IPOINTS , IVX , IYPRX ,
      + IKX , INSUMX , IKPASSX ,
      + IVIX , IVIIX , IVI2X ,
      + IVI3X , IVPRIIX , IVPRIIX ,
      + IVPRI2X , IVPRIIX , IVPRIIX ,
      + ICYIX , ICYIIX , IREIN
C.....
C TORQUE SEAT ALONE COMMON BLOCK
C.....
      COMMON /TOROSA / TLAESA , TMAESA , TNAESA
C.....
      IF(IEVENTS(28) .EQ. 0) GOTO 500
      IF(IEVENTS(31) .NE. 0) GO TO 500
      IF(INTSTP .EQ. 0) GOTO 10
      IF(TRAJSA(4) .GT. 0.0) GOTO 10
      TRAJSA(1) = 0.0
      IEVENTS(31) = 1
      TIMES(31) = TIME
      GOTO 500
C.....
C.....
C.....

```

```

115 C COMPUTE VELOCITY COMPONENTS IN EFCS
116 C.....
117 C
118 C 10 CALL ROTATE(TRAJSA(5),TRAJSA(14),ZVECT(1),DCMSAE,1)
119 C TRAJSA(20) = TRAJSA(11)
120 C TRAJSA(21) = TRAJSA(12)
121 C TRAJSA(22) = TRAJSA(13)
122 C.....
123 C
124 C COMPUTE FORCES AND MOMENTS ON SEAT ALONE
125 C.....
126 C
127 C CALL ATMOS(TRAJSA(4),OLDALT(3),PRESALT(3))
128 C OLDALT(3) = TRAJSA(4)
129 C CALL AERFMSA
130 C.....
131 C
132 C COMPUTE ACCELERATIONS IN SCS
133 C.....
134 C XACCEL(3)=FXAESA/MASSSA-GRAVITY*DCMSAE(1,3)
135 C TRAJSA(17) = XACCEL(3)
136 C + *TRAJSA(13)*TRAJSA(6) - TRAJSA(12)*TRAJSA(7)
137 C YACCEL(3) = FYAESA/MASSSA-GRAVITY*DCMSAE(2,3)
138 C TRAJSA(18) = YACCEL(3)
139 C + *TRAJSA(11)*TRAJSA(7) - TRAJSA(13)*TRAJSA(5)
140 C ZACCEL(3) = FZAESA/MASSSA - GRAVITY*DCMSAE(3,3)
141 C TRAJSA(19) = ZACCEL(3)
142 C + *TRAJSA(12)*TRAJSA(5) - TRAJSA(11)*TRAJSA(6)
143 C.....
144 C
145 C ANGULAR MOMENTUM EQUATIONS
146 C.....
147 C
148 C IXZP = IXZSA * TRAJSA(11)
149 C IYZQ = IYZSA * TRAJSA(12)
150 C IZZR = IZZSA * TRAJSA(13)
151 C IXXP = IXXSA * TRAJSA(11)
152 C IXYQ = IXYSA * TRAJSA(12)
153 C IXPZ = IXPZSA * TRAJSA(11)
154 C IYZR = IYZSA * TRAJSA(13)
155 C.....
156 C
157 C TAUZ = TRAJSA(12) * (IXZP + IYZQ - IZZR) - TRAJSA(13) *
158 C (IXYP - IYVQ + IYZR) + TLAESA
159 C TAUZ = TRAJSA(13) * (IXZR + IXVQ - IXXP) - TRAJSA(11) *
160 C (IXZP + IYZQ - IZZR) + TMAESA
161 C TAUZ = TRAJSA(11) * (IXYP - IYVQ + IYZR) - TRAJSA(12) *
162 C (IXZR + IXVQ - IXXP) + TMAESA
163 C TRAJSA(24) = ((TAUX * CISA + TAUZ * C2SA + TAUZ * C3SA)/C4SA)
164 C TRAJSA(23) = ((I1ZZSA * (TAUX + IXYSA * TRAJSA(24)) +
165 C IXZSA * (TAUZ + IYZSA * TRAJSA(24)))/C3SA)
166 C TRAJSA(25) = ((TAUZ + IXZSA * TRAJSA(23) + IYZSA *
167 C TRAJSA(24))/I1ZZSA)
168 C.....
169 C
170 C 500 CONTINUE
171 C RETURN

```

NADC-84068-60

PAGE 275

PAGE

83/11/07. 09.41.53

FTN 4.6+428

OPT-1

74/74

SUBROUTINE SEATALN

END


```

1 SUBROUTINE SEATOC
2 .....
3 C DESCRIPTION - LEVEL 2
4 C FUNCTION - CALCULATES SEAT/OCCUPANT TRAJECTORY PARAMETERS
5 C METHOD - SEAT/OCCUPANT EQUATIONS OF MOTION ARE COMPUTED FROM
6 C CATAPULT IGNITION TO SEAT/OCCUPANT SEPARATION.
7 C CATAPULT, ROCKET, RAIL, AERODYNAMIC, DART, DROGUE
8 C CHUTE, AND RECOVERY CHUTE FORCES AND MOMENTS ACT ON
9 C THE SEAT/OCCUPANT. FORCES ARE SUMMED AND USED TO
10 C CALCULATE ACCELERATIONS IN THE SEAT COORDINATE SYSTEM.
11 C MOMENTS ARE ALSO SUMMED AND USED IN THE ANGULAR
12 C MOMENTUM EQUATIONS TO GET ANGULAR ACCELERATIONS OF THE
13 C SEAT/OCCUPANT.
14 C COMMUNICATIONS:
15 C CALLED BY: GESS
16 C CALLS:
17 C AERFMSO
18 C ATMOS
19 C CATAFM
20 C CHUTES
21 C DARTFM
22 C RAILFM
23 C RKTFM
24 C ROTATE
25 C TUBEND
26 C VERTSK
27 C NON-COMMON VARIABLES DEFINED:
28 C OLDALT(1) - LATEST PREVIOUS ALTITUDE OF THE SEAT/OCCUPANT
29 C COMBINATION
30 C PRESALT(1) - LATEST PREVIOUS PRESSURE ALTITUDE OF THE SEAT/OCCUPANT
31 C COMBINATION
32 C
33 C FX
34 C FY - COMPONENTS OF SUMMED CATAPULT, ROCKET, RAIL,
35 C FZ - AERODYNAMIC, DART, DROGUE CHUTE, AND RECOVERY CHUTE
36 C FORCES
37 C
38 C TISO
39 C TMSO - COMPONENTS OF SUMMED CATAPULT, ROCKET, RAIL,
40 C TNSO - AERODYNAMIC, DART, DROGUE CHUTE, AND RECOVERY CHUTE
41 C MOMENTS
42 C GRAVITY - ENGLISH OR METRIC VALUE OF ACCELERATION DUE TO GRAVITY
43 C MASS - MASS OF THE SEAT/OCCUPANT COMBINATION MINUS TOTAL
44 C BURNED ROCKET PROPELLANT
45 C
46 C IAZP
47 C IYZQ
48 C IZZR
49 C IXXP
50 C IXYQ
51 C IXPZ
52 C IXZR
53 C IYYQ
54 C IYZR
55 C TAUZ
56 C TAUZ
57 C C1

```

5-278


```

C ..... COMMON /RKUTTA / TIME . TIMES . DELTAT . TRAJSO(193) .
+ TRAJSA(193) . TRAJQA(193) . TRAJCH(97,3) .
+ TRAJAC(193) . TVCEQS(225) . QUATSO(65) .
+ QUATSA(65) . QUATOA(65) . QUATAC(65) .
+ INTSTP . IPCPASS . IRKPASS .
+ IPOINTS . IYX . IYPRX .
+ IKX . IKSUMX . IKPASSX .
+ IYIX . IYIIX . IYI2X .
+ IYI3X . IYPRIX . IYPRIIX .
+ IYPR12X . IYIYX . IYIYIX .
+ ICYIX . ICYIIX . IREIN .
C .....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C .....
COMMON /TORQSO / TMCASO(2) . TMCASO(2) . TNCASO(2) .
+ TLTUBSO . TMTUBSO . TINTUBSO .
+ TMSLSO(6) . TMSLSO(6) . TNSLSO(6) .
+ TLRKSO(6) . TMRKSO(6) . TNRKSO(6) .
+ TLRKSO(3) . TMRKSO(3) . TNRKSO(3) .
+ TLAESO . TMAESO . TNAESO .
+ TLRTOISO . TMRTOISO . TNRTOISO .
C .....
C REAL MASS,IXZP,IYVQ,IZZR,IXXP,IYVQ,IXYP,
+ IYZR,IYVQ,IYZR
C .....
IF (IYEVENTS(28) .NE. 0) GO TO 5000
IF (INTSTP .EQ. 0) GO TO 25
C .....
C SAVE PREVIOUS Z-VELOCITY FOR PEAK TRAJECTORY
C .....
C .....
C PKZVEL=TRAUSO(16)
C .....
C .....
C CHECK FOR SEAT/OCCUPANT IMPACT OR SEPARATION
C .....
IF (TRAUSO(4) .GT. 0.0) GO TO 15
TRAUSO(1) = 0.0
IYEVENTS(29) = 1
TIMES(29) = TIME
GO TO 5000
15 CONTINUE
PFORCE = SORT(FXCHSO(3) + FXCHSO(3) + FYCHSO(3) + FYCHSO(3) +
+ FZCHSO(3) + FZCHSO(3))
IF ((ISOSEP .EQ. 1) AND. (TIMES .GT. SOSEP)) GO TO 20
IF ((ISOSEP .EQ. 2 AND. PFORCE .GT. SOSEP AND.
+ IYEVENTS(24) .NE. 0) GO TO 20
GO TO 25
C .....
20 IYEVENTS(28) = 1
TIMES(28) = TIME
GO TO 5000
25 CONTINUE
C .....
C COMPUTE VELOCITY COMPONENTS IN EFCS
C .....

```

```

C.....
230 C
    CALL ROTATE(TRAJSO(5),TRAJSO(14),ZVECT(1),DCMSE,1)
    TRAJSD(20) = TRAJSD(11)
    TRAJSD(21) = TRAJSD(12)
    TRAJSD(22) = TRAJSD(13)

235 C
C.....
C COMPUTE FORCES AND MOMENTS ON SEAT/OCCUPANT
C.....
240 C
    CALL ATMOS(TRAJSO(4),OLDALT(1),PRESALT(1))
    OLDALT(1) = TRAJSD(4)
    CALL RAILFM
    CALL CATAFM
    CALL TUBEND
    CALL AERFMSO
    CALL DARTFM
    CALL VERTSK
    CALL RKTFM
    CALL CHUTES

245 C
C.....
250 C
C.....
C SUM FORCES AND MOMENTS
C.....
255 C
    FX = FXAESO + FXORTSO + FXTUBSO
    FY = FYAESO + FYDRTSO + FYTUBSO
    FZ = FZAESO + FZDRTSO + FZTUBSO

260 C
    TLSO = TLAESO + TLDRTSO + TLTUBSO
    TMSO = TMAESO + TMDRTSO + TMTUBSO
    TNSO = TNAESO + TNDRTSO + TNTUBSO

265 C
    DO 30 I = 1,2
    FX = FX + FXCASO(I)
    FY = FY + FYCASO(I)
    FZ = FZ + FZCASO(I)

270 C
    TLSO = TLSO + TLCASO(I)
    TMSO = TMSO + TMCASO(I)
    TNSO = TNSO + TNCASO(I)
30 CONTINUE

275 C
    DO 35 I = 1,3
    FX = FX + FXCHSO(I)
    FY = FY + FYCHSO(I)
    FZ = FZ + FZCHSO(I)

280 C
    TLSO = TLSO + TLCHSO(I)
    TMSO = TMSO + TMCHSO(I)
    TNSO = TNSO + TNCHSO(I)
35 CONTINUE

285 C
    DO 40 I = 1,6
    FX = FX + FXLSO(I) + FXRKSO(I)
    FY = FY + FYLSO(I) + FYRKSO(I)

```

```

      FZ = FZ + FZSLSO(1) + FZRKSO(1)
      C
      TLSD = TLSD + TSLSD(1) + TLRKSO(1)
      TMSO = TMSO + TMSLSO(1) + TMRKSO(1)
      TNSO = TNSO + TNSLSO(1) + TNRKSO(1)
      40 CONTINUE
      C
      C.....
      C COMPUTE ACCELERATIONS IN SCS
      C EQUATIONS ALLOW FOR ACCELERATION OF AIRCRAFT WHILE STILL IN COCKPIT
      C.....
      C
      IF(MASSO .LE. 0.0) GOTO 75
      XACCEL(1)=YACCEL(1)=ZACCEL(1)=0.
      TRAJSO(17) = TRAJSO(18) = TRAJSO(19) = 0.0
      IF (IEVENTS(37) .NE. 0) GO TO 63
      CALL ROTATE(TRAJAC(17),TRAJSD(17),ZVECT(1),DCMSA,0)
      XACCEL(1) = TRAJSD(17)
      YACCEL(1) = TRAJSD(18)
      ZACCEL(1) = TRAJSD(19)
      63 CONTINUE
      XGRAV = GRAVITY*DCMSE(1,3)
      YGRAV = GRAVITY*DCMSE(2,3)
      ZGRAV = GRAVITY*DCMSE(3,3)
      IF (IEVENTS(37) .EQ. 0) XGRAV = YGRAV = 0.0
      IF (IEVENTS(1) .EQ. 0) ZGRAV = 0.0
      IF (IEVENTS(1) .NE. 0 .AND. IEVENTS(38) .EQ. 0 .AND.
      + ZGRAV .GT. (FZ/MASSO)) ZGRAV = FZ/MASSO
      XACCEL(1)=XACCEL(1)+FX/MASSO-XGRAV
      TRAJSD(17) = XACCEL(1)
      + +TRAJSD(13)+TRAJSD(6) - TRAJSD(12)+TRAJSD(7)
      YACCEL(1) = YACCEL(1)+FY/MASSO-YGRAV
      TRAJSD(18) = YACCEL(1)
      + +TRAJSD(11)+TRAJSD(7) - TRAJSD(13)+TRAJSD(5)
      ZACCEL(1) = ZACCEL(1)+FZ/MASSO-ZGRAV
      TRAJSD(19) = ZACCEL(1)
      + +TRAJSD(12)+TRAJSD(5)-TRAJSD(11)+TRAJSD(6)
      C.....
      C ANGULAR MOMENTUM EQUATIONS
      C.....
      C
      IXZP = IXZSO + TRAJSD(11)
      IYZO = IYZSO + TRAJSD(12)
      IZZR = IZZSO + TRAJSD(13)
      IXXP = IXXSO + TRAJSD(11)
      IXYO = IXYSO + TRAJSD(12)
      IXYP = IXYSO + TRAJSD(11)
      IAZR = IAZSO + TRAJSD(13)
      IYQO = IYQSO + TRAJSD(12)
      IYZR = IYZSO + TRAJSD(13)
      C
      TAUZ = TRAJSD(12) * (IXZP + IYZO - IZZR) - TRAJSD(13) *
      + (IXYP - IYQO + IYZR) + TLSD
      TAUZ = TRAJSD(13) * (IXZR + IXYO - IXXP) - TRAJSD(11) *
      + (IXZP + IYZO - IZZR) + TMSO
      TAUZ = TRAJSD(11) * (IXYP - IYQO + IYZR) - TRAJSD(12) *
      + (IXXP + IXYO + IXZR) + TNSO

```

```

C
C
345   TRAJSO(23) = TRAJSO(24) + TRAJSO(25) = 0.0
      IF (IEVENTS(37) .EQ. 0)
      + CALL ROTATE(TRAJAC(23),TRAJSO(23),ZVECT,DCMSA,O)
      TRAJSO(24) = ((TAUX + C150 + TAUY + C250 + TAUZ + C350) /
      + C450) +
      + TRAJSO(24)
350   TRAJSO(23) = ((1ZZSO + (TAUX + IXYSO + TRAJSO(24)) +
      + IXZSO + (TAUZ + IYZSO + TRAJSO(24))) / C250) +
      + TRAJSO(23)
355   TRAJSO(25) = ((TAUZ + IXZSO + TRAJSO(23) + IYZSO +
      + TRAJSO(24)) / 1ZZSO) + TRAJSO(25)
      GOTO 5000

C
C.....
C FATAL ERROR MESSAGES
C.....
360   75 WRITE(5,100)
      GOTO 1000

C.....
C FORMATS FOR FATAL ERROR MESSAGES
C.....
365   100 FORMAT(1X, //72(1H*)//, 4X, "FATAL ERROR(SUBROUTINE SEATOC)**", //,
      + "MASS CALCULATED TO BE EQUAL TO ZERO RESULTS IN DIVISION BY ZERO",
      + //, 72(1H*))

C
370   1000 CONTINUE
      TERRFLG = 1

C
375   5000 CONTINUE
      RETURN
      END

```

```

1      SUBROUTINE SEPINIT
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - INITIALIZES OCCUPANT/ALONE AND SEAT/ALONE CONDITIONS
C
5      C METHOD - INITIALIZES SEAT/OCCUPANT SEPARATION
C
C OF MOTION, INITIALIZES OCCUPANT/ALONE AND SEAT/ALONE EQUATIONS
C
C VALUES FOR LATEST PREVIOUS ALTITUDE AND PRESSURE
C
C ALTITUDE, AND SETS THE FLAG FOR SEAT/OCCUPANT
C
10     C SEPARATION AND TIME OF THE EVENT
C
C COMMUNICATIONS:
C CALLED BY:
C
C CALLS:
C
15     C NONE
C NON-COMMON VARIABLES DEFINED: NONE
C POTENTIAL ERROR CONDITIONS: NONE
C.....
C DENSITY COMMON BLOCK
C.....
C COMMON /DENSITY / IATMOS , OLDALT(3) , RHOS
C + PRESALT(3) , DTEMP
C + TEMPS
C + VXWIND , VYWIND , VZWIND
C.....
C SECTION 1 COMMON BLOCK
C.....
C COMMON /ICONTROL / TSTART , TSTOP , ESTOP , IRESTRT, IUNITS ,
C + ISEATTR, ISOSEP , IPLOT , IDRIFLG,
C + IPHASE1, IPHASE2, IPHASE3
C
C INTEGER
C ESTOP
C.....
C SECTION 3 COMMON BLOCK
C.....
C COMMON /IDELTAT / DPHAS1, DPHAS2, DPHAS3
C COMMON /IREPORT / IREPTS(31) , PRTRFO, PI1, PI2, PI3
C INTEGER
C PRTRFO, PI1, PI2, PI3
C.....
C MATRIX COMMON BLOCK
C.....
C COMMON /MATRIX / DCMAR(3,3) , DCMRA(3,3) , DCMSE(3,3) ,
C + DCMSE(3,3) , DCMYS(3,3) , DCMTE(3,3) ,
C + DCMOAE(3,3) , DCMOAE(3,3) , DCMOAE(3,3) ,
C + DCMOUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
C COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31)
C + MAXLINE , MAXREPT , MAXEVT ,
C + IEVLIN , IERRFLG , LU
C + IDATE , HEADALT , HEADVEL
C + HEADSR , HEADYAW , HEADPTT
C + HEADROL , HEADWGT , BIAS
C + REPTYPE(5,31) , PRTLNGT(2) , PRTWGHT(2)
C + THEADER(24) , TEVENTS(38) , TIMES(38)
C + INVOC , PRTEMP( 2)

```



```

60      +      PRTMASS(2)      , PRTINDX      , PKZVEL      ,
      +      ZVECT(3)          , XYZ(3)        , SAVTIME     ,
      +      XACCEL(3)         , YACCEL(3)     , ZACCEL(3)    ,
      +      REPTYPE           , BIAS          , PRTLNKT      ,
      +      PRTWGT           , PRTMASS       , PRTINDX      ,
      +      PRTMP            , PRTINDX       , PRTINDX      ,
C *****
C INTEGRATION ROUTINE COMMON BLOCK
C *****
      COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSD(193) ,
      TRAJSA(193) , TRAJDA(193) , TRAJCH(97,3) ,
      TRAJAC(193) , TVCEQS(225) , QUATSO(65) ,
      QUATSA(65) , QUATAC(65) ,
      INITSTP , IPCPASS , IRKPASS ,
      IPOINTS , IYX , IKSUMX , IKPASSX ,
      IKX , IYIX , IYIIX , IYI2X ,
      IYI3X , IYPRIX , IYPR11X ,
      IYPR12X , IYVIX , IYV11X ,
      ICYIX , ICYI1X , IREIN ,
C *****
C AERODYNAMICS INFORMATION COMMON BLOCK
C *****
      COMMON /AEROCFS / OALPH , OABETA , OAVEL , OAMACH ,
      SAALPH , SABETA , SAVEL , SAMACH ,
      SOALPH , SOBETA , SOVEL , SOMACH ,
      CXOA , CYOA , CZOA , CLDA , CMOA , CNOA ,
      CXSA , CYSA , CZSA , CLSA , CMSA , CNSA ,
      CXSO , CYSO , CZSO , CLSO , CMSO , CNSO ,
C *****
C SEAT/OCCUPANT FORCES COMMON BLOCK
C *****
      COMMON /FORCESO / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
      FXTUBSO , FYTUBSO , FZTUBSO ,
      FXLSO(6) , FYLSO(6) , FZLSO(6) ,
      FXRKSQ(6) , FYRKSQ(6) , FZRKSQ(6) ,
      FXCHSQ(3) , FYCHSQ(3) , FZCHSQ(3) ,
      FXAESO , FYAESO , FZAESO ,
      FXDRTSO , FYDRTSO , FZDRTSO ,
C *****
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C *****
      COMMON /TORQSO / TLCASO(2) , TMCASO(2) , TNCASO(2) ,
      TLTUBSO , TMTUBSO , TNTUBSO ,
      TLLSQ(6) , TMLSQ(6) , TNSLSQ(6) ,
      TLRKSO(6) , TMRKSO(6) , TNRKSO(6) ,
      TLCHSQ(3) , TMCHSQ(3) , TNCHSQ(3) ,
      TLAESO , TMAESO , TNAESO ,
      TLDRTSO , TMDRTSO , TNDRTSO ,
C *****
C *****
      IF(INITSTP .EQ. 0) GO TO 9999
      IF(IEVENTS(28) .EQ. 0) GO TO 9999
      IF(IPHASE3 .GT. 0) GO TO 100
      IREIN = 1
      QUATOA(1) = QUATSA(1) = QUATSO(1) = 0.0
      TRAJSA(1) = TRAJSD(1) = 0.0

```

```

115      TRAJOA(1)= 12.0
      C .....
      C SET OCCUPANT/ALONE AND SEAT/ALONE EQUATIONS OF MOTION EQUAL *
      C TO SEAT/OCCUPANT EQUATIONS OF MOTION *
      C .....
      C
      DO 10 I=2,13
        TRAJOA(I) = TRAJSA(I) = TRAJSO(I)
      10 CONTINUE
      C .....
      C SET OCCUPANT ALONE ANGULAR VELOCITIES TO ZERO *
      C .....
      TRAJOA(11)= TRAJDA(12)= TRAJDA(13)= 0.0
      CALL ROTATE (TRAJOA(5),TRAJOA(5),ZVECT(1),DCMSE,1)
      C
      IF (ISEATTR.EQ.0) GO TO 50
      TRAJSA(1)= 12.0
      QUATSA(1)= 4.0
      DO 20 I=2,5
        QUATSA(I) = QUATSO(I)
      20 CONTINUE
      DO 30 I=1,3
        DO 30 J=1,3
          DCMSE(I,J) = DCMSE(I,J)
      30 CONTINUE
      50 CONTINUE
      C
      C .....
      C SET LATEST VALUES OF ALTITUDE AND PRESSURE ALTITUDE FOR BOTH *
      C OCCUPANT/ALONE AND SEAT/ALONE EQUAL TO LATEST VALUES OF *
      C ALTITUDE AND PRESSURE ALTITUDE FOR SEAT/OCCUPANT *
      C .....
      OLDALT(3) = OLDALT(2) = OLDALT(1)
      PRESALT(3) = PRESALT(2) = PRESALT(1)
      C
      C
      IPHASE3 = 1
      100 CONTINUE
      IF (IPHASE3.GT.1) GO TO 150
      L=IFIX(1.E6*TIME5)
      K=IFIX(1.E6*(DTPHASE3+5.E-10))
      IF (MOD(L,K).NE.0) GO TO 9999
      IPHASE3 = 2
      DELTAT = DTPHASE3
      150 CONTINUE
      IF (IPHASE3.GT.2) GO TO 9999
      L = IFIX(1.E6*TIME5)
      K = IFIX(1.E6*(DTPHASE3+PI3+5.E-10))
      IF (MOD(L,K).NE.0) GO TO 9999
      PRTRQ = PI3
      DO 200 I=1,31
        IPRTCNT(I) = PRTRQ - 1
      200 CONTINUE
      IPHASE3 = 3
      C

```

SUBROUTINE SEPINIT 74/74 OPT=1 FTN 4.6+428 83/11/07. 09.41.53 PAGE 286

```

C.....
C SET AERODYNAMIC COEFFICIENTS FOR SEAT/OCCUPANT EQUAL TO ZERO
C.....
175 C
C CXSO = CYSO = CZSO = CLSO = CMSO = CNSO = 0.0000
C
C
C.....
180 C SET SEAT/OCCUPANT FORCES AND MOMENTS EQUAL TO ZERO
C.....
C
C FXAESO = FYAESO = FZAESO = TLAESO = TMAESO = 0.0
9999 RETURN
185 END

```

83/11/07 09.41.53

FIN 4 6+428

74/74 OPI=1

```

1  SUBROUTINE SLUGCON
C .....
C DESCRIPTION - LEVEL 2
C FUNCTION - CALCULATES THE DROGUE SLUG/CONTAINER EQUATIONS OF
C MOTION
C METHOD - THE DROGUE SLUG/CONTAINER EQUATIONS OF MOTION ARE
C INTEGRATED FROM DROGUE PROJECTION TO LINE STRETCH.
C WHEN TIME TO DROGUE LINE STRETCH IS CALCULATED.
C IF TIME TO DROGUE LINE STRETCH IS DETERMINED FROM A
C TABLE, THE DROGUE SLUG/CONTAINER POSITION IS DETER-
C MINED BY FIRST ALIGNING THE CHUTE WITH THE WIND.
C COMMUNICATIONS -
C CALLED BY:
C GESS
C CALLS:
C ROTATE, ZARCTAN
C NON-COMMON VARIABLES DEFINED -
C DVELX - X AND Y COMPONENTS OF THE DROGUE SLUG/CONTAINER
C DVELY - VELOCITY, RELATIVE TO THE AIR
C DVEL - TOTAL VELOCITY OF THE DROGUE SLUG/CONTAINER, RELATIVE
C TO THE AIR
C ACC - INTERMEDIATE VALUE IN CALCULATION OF DROGUE SLUG/
C CONTAINER ACCELERATION
C R - TOTAL VELOCITY OF THE DROGUE SLUG/CONTAINER
C LINELTH - TOTAL LENGTH OF THE EXTENDED DROGUE LINES PRIOR TO
C LINE STRETCH
C ANG1 - ANGLES USED IN ALIGNING THE DROGUE CHUTE WITH THE WIND
C ANG2 -
C XD15 - COMPONENTS OF THE VECTOR FROM THE DROGUE CHUTE
C YD15 - ATTACHMENT POINT TO THE DROGUE CHUTE
C ZD15 -
C POTENTIAL ERROR CONDITIONS - NONE
C .....
C CONSTANTS COMMON BLOCK
C .....
C COMMON /CONST / GRAVITY , RADDEG , DEGRAD , PI
C .....
C DENSITY COMMON BLOCK
C .....
C COMMON /DENSITY / ATMOS , OLDALT(3) , RHOS
C + PRESALT(3) , DTEMP ,
C + VWIND , VZWIND ,
C + VWIND , VZWIND
C .....
C SECTION 4 COMMON BLOCK
C .....
C COMMON /IATRCRT / TEMP , PRESSUR, ZACVEL , XPOS , YPOS ,
C + ZPOS , XTAL , YTAIL , ZTAIL , YAW ,
C + PITCH , ROLL , RVEL , QVEL , PVEL ,
C + WINDX , WINDY , WINDZ , XACVEL , CKPITH ,
C + DENSITY, NPTSAAT, AAT(4,50), NPTSLAT,LAT(4,50),
C + IACSFLG
C .....
C SECTION 1 COMMON BLOCK
C .....
C COMMON /ICONTROL / TSTART , TSTOP , ESTOP , IRESTRY, IUNITS ,

```

```

+ ISATTR, ISOSEP , IPLOT , IDRIFLG,
+ IPHASE1, IPHASE2, IPHASE3
+ ESTOP
C.....
60      INTEGER
C MASSES COMMON BLOCK
C.....
COMMON /MASSES / MASSOAI , MASSOA2 , MASSSO , MASSO .
+ MASSA , MASSRK(6) , MASSDC
65      REAL
+ MASSOA1 , MASSOA2 , MASSSO , MASSO
+ MASSA , MASSRK , MASSDC
C.....
C MATRIX COMMON BLOCK
C.....
COMMON /MATRIX / DCMAE(3,3) , DCMRA(3,3) , DCMSA(3,3) ,
+ DCMSE(3,3) , DCMTS(3,3) , DCMTL(3,3) , DCMTD(3,3) ,
70      + DCMSE(3,3) , DCMOE(3,3) , DCMER(3,3) ,
+ DCDUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGCT(31) , LINECT(31) , IPRCNT(31)
+ MAXLINE , MAXREPT , MAXEVT
80      + IEVLNE , TERFLG , LU
+ IDATE , HEADALT , HEADVEL
+ HEADSR , HEADYAW , HEADPIT
+ HEADROL , HEADWGT , BIAS
85      + REPTYE(5,31) , PRTNGT(2) , PRTWGHT(2)
+ IHEADER(24) , EVENTS(38) , TIMES(38)
+ IMVDC , PRTEMP ( 2)
+ PRTMASS(2) , PRTINDX , PKZVEL
+ ZVECT(3) , XYZ(3) , SAVTIME
90      + XACCEL(3) , YACCEL(3) , ZACCEL(3)
+ REPTYE , BIAS , PRTNGT
+ PRTWGHT , PRTMP , PRTMASS , PRTINDX
C.....
C MOMARMS COMMON BLOCK
C.....
COMMON /MOMARMS /
+REFLNQ ,REFLNQA ,REFLNSA ,URX(6) ,URY(6) ,URZ(6) ,
+XSSOCA(2) ,YSSOCA(2) ,ZSSOCA(2) ,XSORK(6) ,YSORK(6) ,ZSORK(6) ,
+XSORRE ,YSORRE ,ZSORRE ,XSOLRE ,YSOLRE ,ZSOLRE ,
+XSOMRE ,YSOMRE ,ZSOMRE ,XSOBOT ,YSOBOT ,ZSOBOT ,
100    +XSOSB(6) ,YSSOSB(6) ,ZSSOSB(6) ,XRCSAC ,YRCSAC ,ZRCSAC
+XSOSCAC ,YSSCAC ,ZSSCAC ,XSOSRP ,YSSORP ,ZSSORP
+
+XSSASRP ,YSSASRP ,ZSSASRP ,XRDPAP(2) ,YRDDAP(2) ,ZRRDAP(2) ,
+XRSBO(6) ,YRSBO(6) ,ZRSBO(6) ,XSOCAP(2) ,YSOCAP(2) ,ZSOCAP(2) ,
+XSDAP(2) ,YSDAP(2) ,ZSDAP(2) ,XESDAC ,YESDAC ,ZESDAC ,
+XSRCSAC ,YRCSAC ,ZRCSAC ,XSOSAC ,YSOSAC ,ZSOSAC ,
+XRSOSB ,YRSOSB ,ZRSOSB ,XRSBOT ,YRSBOT ,ZRSBOT ,
110    +XRSRB ,YRSRB ,ZRSRB ,XSCHCH(3) ,YSCHCH(3) ,ZSCHCH(3) ,
+XAACSQ ,YAACSQ ,ZAACSQ ,XASDAC ,YASDAC ,ZASDAC ,
+XRSOAC ,YRSOAC ,ZRSOAC ,XSCAP(2) ,YSCAP(2) ,ZSCAP(2)
C.....
C SECTION 14 COMMON BLOCK
C.....

```

```

115      COMMON /PARACHUT / IRECOV      RECOVLL      POROSR      .
      +      XRECAP      ZRECAP      .
      +      NPTSRLS      IFTRCV      .
      +      NPTSRTF      SEPRCE      .
120      +      IDROGUE      DROGPD2      .
      +      POROSD2      IFDR02      .
      +      NPTDFT1      IFDR01      .
      +      NPTSLS      IDROGLS      .
125      +      DISPLAY      IDROPLO      .
      +      DROGPD1      DROVELX      .
      +      DROVELY      XDR0GAP      .
      +      YDROGAP      CHALT1      .
130      +      CHALT2      TDELAY      .
      +      AREADC      TFP1      .
      +      TFP2      TFP3      .
      +      CDDC      NPTSRTD      .
      +      RECOVDT(2.25)      .
      +      .
C *****
C INTEGRATION ROUTINE COMMON BLOCK
C *****
      COMMON /RKUTTA / TIME , TIMES      DELTAT      TRAJSO(193)      .
      +      TRAJA(193)      TRAJDA(193)      .
      +      TRAJAC(193)      TVCEQS(225)      .
      +      QUATSA(65)      QUATOA(65)      .
140      +      INTSTP      IPCPASS      .
      +      IPOINTS      IYX      .
      +      IKX      IKSUMX      .
      +      IY1X      IY12X      .
      +      IY13X      IYPR1X      .
145      +      IYPR12X      IYV1X      .
      +      ICY1X      ICY11X      .
      +      .
      REAL LINELTH
      IF(IEVENTS(18) .EQ. 0) GOTO 500
      IF(IDROGLS .EQ. 1) GOTO 30

150      C *****
      C COMPUTE VELOCITY OF DROGUE SLUG/CONTAINER, RELATIVE TO THE AIR
      C *****
      C
      DVELX = TRAJCH(5,1) - WINDX
      DVELY = TRAJCH(6,1) - WINDY
      DVELZ = TRAJCH(7,1) - WINDZ

155      C *****
      C COMPUTE TOTAL VELOCITY OF DROGUE SLUG/CONTAINER
      C *****
      C
      DVEL = SORT(DVELX*DVELX+DVELY*DVELY+DVELZ*DVELZ)

160      C *****
      C COMPUTE VELOCITIES AND ACCELERATIONS OF DROGUE SLUG/CONTAINER
      C *****
      C
      ACC = .5 * RHOS * CDDC * AREADC * DVEL/MASSDC

170      C *****
      C 20 TRAJCH(11,1) = -ACC * DVELX

```

```

175      TRAJCH(12,1) = -ACC * DVELY
      TRAJCH(13,1) = -ACC * TRAJCH(7,1) - GRAVITY
      C
      TRAJCH(8,1) = TRAJCH(5,1)
      TRAJCH(9,1) = TRAJCH(6,1)
      TRAJCH(10,1) = TRAJCH(7,1)
      C
      GO TO 500
180      C
      C.....
      C TRANSFORM VECTOR FROM SEAT/OCCUPANT CG TO DROGUE ATTACHMENT POINT
      C FROM SCS TO EFCS
      C.....
      30 CONTINUE
185      C
      XYZ(1) = XSSOCH(1)
      XYZ(2) = YSSOCH(1)
      XYZ(3) = ZSSOCH(1)
      C
      CALL ROTATE(XYZ(1),XYZ(1),ZVECT(1),DCMSE,1)
190      C
      C.....
      C COMPUTE DROGUE CONTAINER/SLUG POSITION WHEN LINE STRETCH IS
      C DETERMINED FROM A TABLE
      C.....
      R = SORT(TRAJSD(14) * TRAJSD(14) + TRAJSD(15) * TRAJSD(15) +
      + TRAJSD(16) * TRAJSD(16))
200      C
      IF(R.EQ.0.0) GO TO 400
      SIN2 = TRAJSD(16)/R
      COS2 = COS(ASIN(SIN2))
      BETA = ZARCTAN(TRAJSD(15),TRAJSD(14))
205      C
      LINELTH = DROGLL * ((TIME - TIMES(18))/TDROGLS)
      C
      XD1S = -SIGN((LINELTH * COS2 * COS(BETA)),TRAJSD(14))
      YD1S = -SIGN((LINELTH * COS2 * SIN(BETA)),TRAJSD(15))
      ZD1S = -SIGN((LINELTH * SIN2),TRAJSD(16))
210      C
      TRAJCH(2,1) = XYZ(1) + XD1S + TRAJSD(2)
      TRAJCH(3,1) = XYZ(2) + YD1S + TRAJSD(3)
      TRAJCH(4,1) = XYZ(3) + ZD1S + TRAJSD(4)
      C
      C.....
      C SET VELOCITIES OF DROGUE SLUG/CONTAINER EQUAL TO THOSE OF THE SEAT/
      C OCCUPANT WHEN LINE STRETCH IS DETERMINED FROM A TABLE
      C.....
215      C
      TRAJCH(5,1) = TRAJSD(14)
      TRAJCH(6,1) = TRAJSD(15)
      TRAJCH(7,1) = TRAJSD(16)
      GO TO 500
220      C
225      400 CONTINUE
      WRITE(5,410)
      410 FORMAT(2X,/,72(1H*),/,4X, "FATAL ERROR(SUBROUTINE SLUGCON)...."
      + "R EQUAL TO ZERO RESULTS IN DIVISION BY ZERO",/,72(1H*))

```

NADC-84068-60

PAGE 291

83/11/07 09.41.53

FTN 4 6:428

SUBROUTINE SLUGCON 74/74 OPT=1

500 CONTINUE
RETURN
END

230


```

1  SUBROUTINE THRUST
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - THIS SUBROUTINE COMPUTES A NEW THRUST VS TIME
C             TABLE FOR EACH ROCKET USING A STANDARD TABLE AND
C             THE NEW BURN TIME FOR THE ROCKET.
C METHOD - THE RATIO OF THE NEW BURN TIME TO THE
C          OLD(STANDARD) BURN TIME IS COMPUTED. THIS RATIO
C          IS USED TO CHANGE THE THRUST AND TIME AT EACH
C          POINT IN THE TABLE. THE NEW TIME IS DIRECTLY
C          PROPORTIONAL TO THE RATIO AND THE NEW THRUST
C          IS INVERSELY PROPORTIONAL TO THE RATIO. THE AREA
C          UNDER THE NEW AND OLD CURVES(TOTAL THRUST) IS
C          COMPUTED USING THE TRAPEZOIDAL RULE. IF THE RATIO OF
C          THE VALUES IS NOT BETWEEN 0.98 AND 1.02, THE BURN
C          RATIO IS MODIFIED AND THE PROCESS IS REPEATED. IF
C          THE RATIO DOES NOT APPROACH 1.000 AFTER FIVE TRIES
C          THIS IS REPEATED FOR EACH ROCKET.
C COMMUNICATIONS -
C CALLED BY: INPUT
C CALLS:
C          NONE
C NON-COMMON VARIABLES DEFINED:
C BTRATIO - RATIO OF NEW BURN TIME TO STANDARD BURN TIME
C ARATIO - RATIO OF AREAS UNDER CURVES(NEW/OLD)
C AREALD - AREA UNDER STANDARD THRUST CURVE
C AREANEW - AREA UNDER NEW THRUST CURVE
C TSTART - START TIME OF STANDARD TABLE
C TSTOP - STOP TIME OF STANDARD TABLE
C ITABX - ROCKET TABLE BEING PROCESSED
C NPNTS - LOCAL VARIABLE TO SAVE NUMBER OF POINTS
C          IN CURRENT TABLE
C NPASS - NUMBER OF PASSES THRU LOOP FOR CURRENT TABLE
C POTENTIAL ERROR CONDITIONS:
C IF THE RATIO OF THE NEW AREA TO THE OLD AREA DOES NOT
C GO TO WITHIN 1% OF 1.000 AFTER FIVE TRIES, THEN THERE IS A
C PROBLEM WITH THE TABLE.
C .....
C RECALCULATED ROCKET THRUST TABLE COMMON BLOCK
C .....
C COMMON /IRKTOUT / IRKTOUT(2,25.6)
C .....
C SECTION 10 COMMON BLOCK
C .....
C COMMON /IROCKET / INRKT , RKDELY(6), RKNPTS(6), IROKOUT ,
C          + RKIGN(6) , RKWGT(6), RKBURN(6), TSTAR(6) ,
C          + XPOSRK(6), YPOSRK(6), ZPOSRK(6),
C          + RKALPH(6), RKBETA(6), RKGAMA(6), RKTHRST(2,25.6)
C          INTEGER
C          RKNPTS
C .....
C LOOP THRU PROCESSING FOR EACH ROCKET IN SYSTEM
C .....
C DO 100 ITABX=1,INRKT
C   NPNTS = RKNPTS(ITABX)
C   NPASS = 1
C .....

```

```

C COMPUTE NEW/OLD BURN TIME RATIO
C .....
60  TSTART = RKTHRST(1,1,ITABX)
    TSTOP = RKTHRST(1,NPOINTS,ITABX)
    BTRATIO = RKBURN(ITABX)/(TSTOP-TSTART)
C .....
C INTEGRATE TO COMPUTE AREA UNDER OLD CURVE
C .....
65  AREALD = 0.0
    N1 = NPOINTS-1
    DO 10 IX = 1,N1
10  AREALD=AREALD+(RKTHRST(2,IX,ITABX)+RKTHRST(2,IX+1,ITABX))
    + *(RKTHRST(1,IX+1,ITABX)-RKTHRST(1,IX,ITABX))/2.0
C .....
C COMPUTE NEW POINTS AND INTEGRATE TO COMPUTE AREA UNDER NEW CURVE
C .....
70  AREANEW = 0.0
    RKOUT(1,1,ITABX) = RKTHRST(1,1,ITABX)
    RKOUT(2,1,ITABX) = RKTHRST(2,1,ITABX)/BTRATIO
    DO 30 IX = 2,NPOINTS
    IF (NPASS.EQ.1)RKOUT(1,IX,ITABX) = RKOUT(1,IX-1,ITABX)+
    + (RKTHRST(1,IX,ITABX)-RKTHRST(1,IX-1,ITABX))*BTRATIO
    RKOUT(2,IX,ITABX) = RKTHRST(2,IX,ITABX)/BTRATIO
    30 AREANEW = AREANEW+(RKOUT(2,IX-1,ITABX)+RKOUT(2,IX,ITABX))
    + *(RKOUT(1,IX,ITABX)-RKOUT(1,IX-1,ITABX))/2.0
C .....
C COMPUTE RATIO OF NEW AREA TO OLD AREA AND CHECK FOR DESIRED
C ACCURACY. IF NECESSARY, MODIFY BURN TIME RATIO AND RECOMPUTE TABLE
C .....
85  ARATIO = AREANEW/AREALD
    IF (ARATIO .GT. 0.98 .AND. ARATIO .LT. 1.02) GO TO 100
    BTRATIO = BTRATIO*ARATIO
    NPASS = NPASS+1
    IF (NPASS.LT.6) GO TO 20
    CONTINUE
    100 CONTINUE
    RETURN
    END
95

```

```

1 SUBROUTINE TMUPDAT
2 .....
3 C DESCRIPTION - LEVEL 2
4 C FUNCTION - NORMALIZES QUATERNIONS
5 C METHOD - PRIOR TO RAIL SEPARATION, QUATERNIONS USED TO UPDATE
6 C THE TRANSFORMATION MATRICES, DCMSE(EFCS TO ACS) AND
7 C DCMSE(EFCS TO SCS), ARE NORMALIZED. FOLLOWING RAIL
8 C SEPARATION, THE USER HAS THE OPTION OF CHOOSING TO
9 C UPDATE DCMSE ONLY.
10 C COMMUNICATIONS -
11 C CALLED BY: GESS
12 C CALLS:
13 C MATUPD, MATRIX
14 C NON-COMMON VARIABLES DEFINED:
15 C QUATS(1) -
16 C QUATS(2) - NORMALIZED QUATERNIONS
17 C QUATS(3) -
18 C QUATS(4) -
19 C ONORMS - NORMALIZING FACTOR
20 C POTENTIAL ERROR CONDITIONS:
21 C DIVISION BY ZERO IF ONORMS EQUALS ZERO. A MESSAGE IS
22 C PRINTED AND THE RUN IS TERMINATED.
23 C .....
24 C .....
25 C .....
26 C .....
27 C SECTION 1 COMMON BLOCK
28 C .....
29 C COMMON /ICONTROL / ISTART, TSTOP, ESTOP, IPRESTRT, IUNITS,
30 C ISEATTR, ISOSEP, IPLOT, IDIRIFLG,
31 C IPHASE1, IPHASE2, IPHASE3
32 C .....
33 C INTEGER
34 C ESTOP
35 C MATRIX COMMON BLOCK
36 C .....
37 C COMMON /MATRIX / DCMSE(3,3), DCMRA(3,3), DCMSE(3,3),
38 C DCMSE(3,3), DCMTE(3,3), DCMTE(3,3),
39 C DCMSE(3,3), DCMSE(3,3), DCMSE(3,3),
40 C DCMSE(3,3)
41 C .....
42 C MISCELLANEOUS DATA COMMON BLOCK
43 C .....
44 C COMMON /MISC / IPAGECT(31), LINECT(31), IPRTCNT(31),
45 C MAXREPT, MAXREPT, LU,
46 C IERRFLG, LU,
47 C HEADALT, HEADVEL,
48 C HEADYAW, HEADPIT,
49 C HEADROL, HEADWGT, BIAS,
50 C REPTYPE(5,31), PRTLNCT(2), PRTWGT(2),
51 C IHEADER(24), IEVENTS(38), TIMES(38),
52 C IMVDC, PRTEMP( 2),
53 C PRTEMP(2), PRINDX,
54 C ZVECT(3), PKZVEL,
55 C XACCEL(3), XYZ(3), SAVTIME,
56 C REPTYPE, YACCEL(3), ZACCEL(3),
57 C BIAS, PRTLNCT,
58 C PRTWGT
59 C .....

```



```

115      CALL MATUPD(QUATS,DCMSAE)
      500 CONTINUE
C.....
C THE TRANSPOSE OF DCMAE IS USED TO MULTIPLY DCMSA TO GET DCMSA .....
C.....
120      CALL MATRIX(DCMSE,DCMAE,DCMSA,2)
      CALL MATRIX(DCMSA,DCMRA,DCMSR,2)
      GO TO 9000
50 IONORMS=10*IONORMS-50
GO TO 55
51 IONORMS=10*IONORMS-AC
GO TO 55
52 IONORMS=10*IONORMS-QA
GO TO 55
53 IONORMS=10*IONORMS-SA
55 WRITE (5,100) IONORMS
      IERRFLG = 1
100 FORMAT (1X,/,/,72(1H*))/4X,"FATAL ERROR (SUBROUTINE TMUPDAT)***",/,
+ 4X,A10,"CALCULATED TO BE EQUAL TO ZERO-RESULTS IN DIVISION BY ",
+ "ZERO",/,72(1H*)
9000 CONTINUE
      RETURN
      END

```

```

1      SUBROUTINE TUBEND
C.....
C DESCRIPTION - LEVEL 3
C FUNCTION - TO CALCULATE THE FORCES AND MOMENTS IMPOSED ON THE
C SEAT/OCCUPANT DUE TO THE DEFLECTIONS OF THE CATAPULT
C TUBES
C COMMUNICATIONS:
C CALLED BY: SEATOCC
C CALLS: ROTATE
C.....
10     C.....
C SEAT/OCCUPANT FORCES COMMON BLOCK
C.....
15     COMMON /FORCES / FXCASO(2) , FYCASO(2) , FZCASO(2) ,
+      FXTUBSO , FYTUBSO , FZTUBSO ,
+      FXLSO(6) , FYLSO(6) , FZLSO(6) ,
+      FXKSO(6) , FYKSO(6) , FZKSO(6) ,
+      FXHSHO(3) , FYHSHO(3) , FZHSHO(3) ,
+      FXAESO , FYAESO , FZAESO ,
+      FXDRISO , FYDRISO , FZDRISO
C.....
20     C.....
C SECTION 9 COMMON BLOCK
C.....
25     COMMON /ICATPLT / INCAT , CATLNT(2) , CATSTK(2) , TCI (2) ,
+      XPOSAP(2) , YPOSAP(2) , ZPOSAP(2) , NPTSCT(2) ,
+      CATHRST(2,25,2) , ITUBEND , KTUBE , CTUBE ,
+      PTUBE , MUTUBE , EXTNGT , ICATOUT
+      REAL
+      KTUBE , MUTUBE
C.....
30     C.....
C MATRIX COMMON BLOCK
C.....
35     COMMON /MATRIX / DCMAT(3,3) , DCMAT(3,3) , DCMAT(3,3) ,
+      DCMSE(3,3) , DCMTS(3,3) , DCMTE(3,3) ,
+      DCMSE(3,3) , DCMOAE(3,3) , DCMSE(3,3) ,
+      DCMOUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
40     COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+      MAXLINE , MAXREPT , MAXEVNT ,
+      IEVLNE , IERRFLG , LU ,
+      IDATE , HEADALT , HEADVEL ,
+      HEADSR , HEADYAW , HEADPIT ,
+      HEADROL , HEADWTG , BIAS ,
+      REPTYPE(5,31) , PRTLNGT , PRTWGT ,
+      IHEADER(24) , IEVENTS(38) ,
+      IMVDC , TIMES(38) ,
+      PRTMASS(2) , PRTINDX , PKZVEL ,
+      ZVECT(3) , XYZ(3) , PRTEMP( 2) ,
+      XACCEL(3) , YACCEL(3) , SAVTIME ,
+      REPTYPE , BIAS , ZACCEL(3) ,
+      INTEGER , PRTWGT , PRTLNGT ,
+      PRTEMP , PRTMASS , PRTINDX
C.....
55     C.....
C MOMARMS COMMON BLOCK
C.....
COMMON /MOMARMS /

```

```

60  +REFLN50 ,REFLN0A ,REFLN5A ,URX(6) ,URY(6) ,URZ(6)
    +XSSOCA(2) ,YSSOCA(2) ,ZSSOCA(2) ,XSSORR(6) ,YSSORR(6) ,ZSSORR(6) ,
    +XSSORRE ,YSSORRE ,ZSSORRE ,XSSOLRE ,YSSOLRE ,ZSSOLRE ,
    +XSSOMRE ,YSSOMRE ,ZSSOMRE ,XSSOBOT ,YSSOBOT ,ZSSOBOT ,
    +XSSOSB(6) ,YSSOSB(6) ,ZSSOSB(6) ,XRRCSAC ,YRRCSAC ,ZRRCSAC ,
    +XSSCSAC ,YSSCSAC ,ZSSCSAC ,XSSOSRP ,YSSOSRP ,ZSSOSRP ,
    +
    +XSSASRP ,YSSASRP ,ZSSASRP ,XRRDAP(2) ,YRRDAP(2) ,ZRRDAP(2) ,
    +XRRSBO(6) ,YRRSBO(6) ,ZRRSBO(6) ,XSSOCP(2) ,YSSOCP(2) ,ZSSOCP(2) ,
    +XSSDAP(2) ,YSSDAP(2) ,ZSSDAP(2) ,XESOAC ,YESOAC ,ZESOAC ,
    +XSRCSAC ,YSRCSAC ,ZSRCSAC ,XSSOAC ,YSSOAC ,ZSSOAC ,
    +XRSOSB ,YRSOSB ,ZRSOSB ,XSSOCH(3) ,YSSOCH(3) ,ZSSOCH(3) ,
    +XRRSB ,YRRSB ,ZRRSB ,XASOAC ,YASOAC ,ZASOAC ,
    +XAACSO ,YAACSO ,ZAACSO ,XSCAP(2) ,YSCAP(2) ,ZSCAP(2) ,
    +XRSOAC ,YRSOAC ,ZRSOAC ,XSCAP(2) ,YSCAP(2) ,ZSCAP(2) ,
    +
C.....
C TORQUE SEAT/OCCUPANT COMMON BLOCK
C.....
    COMMON /TORQSO / TLCASO(2) , TMCASO(2) , TNCASO(2) ,
    + TLTUBSO , TMTUBSO , TINTUBSO ,
    + TLSLSO(6) , TMSLSO(6) , TNSLSO(6) ,
    + TLRKSO(6) , TMRKSO(6) , TNRKSO(6) ,
    + TLCHSO(3) , TMCHSO(3) , TNCHSO(3) ,
    + TLAESO , TMAESO , TNAESO ,
    + TLDRTSO , TMDRTSO , TNDRTSO
C.....
    FXTUBSO=FYTUBSO=FZTUBSO=0.0
    TLTUBSO=TMTUBSO=TINTUBSO=0.0
    IF (ITUBEND.EQ.0) DR. IEVENTS(3) .NE. 0) GO TO 8000
    XYZ(1) = XSSOCA(1)
    XYZ(2) = YSSOCA(1)
    XYZ(3) = ZSSOCA(1)
    CALL ROTATE(XYZ,XRRCSAC,XSSOMRE,DCMSR,1)
    TUBEEXT = SORT(XRRCSAC+YRRCSAC+YRRCSAC+YRRCSAC+
    + CATLNT(1))*(ZRRCSAC-CATLNT(1)))
    TUBEDEF = SORT(XRRCSAC+YRRCSAC+YRRCSAC+YRRCSAC)
    FTUBE = KTUBE+TUBEDEF*((1.-TUBEEXT/EXTLNGT)**PTUBE)
    IF (TUBEDEF.NE.0) FXTUBSO=-FTUBE+YRRCSAC/TUBEDEF
    IF (TUBEDEF.NE.0) FYTUBSO=-FTUBE+YRRCSAC/TUBEDEF
    FZTUBSO=-MTUBE+FTUBE
    CALL ROTATE(FXTUBSO,FXTUBSO,ZVECT,DCMSR,0)
    TLTUBSO=FYTUBSO+ZSSOCA(1)-FZTUBSO+YSSOCA(1)
    TMTUBSO=FZTUBSO+XSSOCA(1)-FXTUBSO+ZSSOCA(1)
    TINTUBSO=FYTUBSO+XSSOCA(1)-FXTUBSO+YSSOCA(1)
9000 CONTINUE
    RETURN
    END

```

```

1      SUBROUTINE UPDVECT
C.....
C DESCRIPTION - LEVEL 2
C FUNCTION - NORMALIZES VECTORS
C METHOD - CALCULATE NEW CG VECTOR VARIABLES
C COMMUNICATIONS -
C CALLED BY:
C      GESS
C      CALLS:
C      ROTATE
C NON-COMMON VARIABLES DEFINED:
C NONE
C POTENTIAL ERROR CONDITIONS:
C NONE
C.....
C.....
C.....
C SECTION 6 COMMON BLOCK
C.....
COMMON /ISEATOC / IPCNTL , XCGSO , YCGSO , ZCGSO , IXXSO ,
+ IXYSO , IYZSO , IYZSO , IZZSO ,
+ AREASO , AREAOA , WGHIOAB , WGHIOAA ,
+ IXXOA , IXYOA , IYZOA , IYZOA , IYZOA ,
+ IZZOA , XCGOA , YCGOA , ZCGOA , SOSEP ,
+ IXXSO , IXYSO , IYZSO , IYZSO ,
+ IZZSO , IXXOA , IXYOA , IYZOA , IYZOA ,
+ IYZOA , IZZOA
C.....
C SECTION 5 COMMON BLOCK
C.....
COMMON /ISETALN / XPOSSRP , YPOSSRP , ZPOSSRP , XCGSA , YCGSA ,
+ ZCGSA , IXXSA , IYXSA , IYZSA , IYZSA ,
+ IYZSA , IZZSA , PHISA , PSISA , THESA ,
+ AREASA , HGHISA , WGHISA , XPOSBOT , YPOSBOT ,
+ ZPOSBOT , XPOSSCS , YPOSSCS , ZPOSSCS ,
+ IXXSA , IYXSA , IYZSA , IYZSA ,
+ IZZSA
C.....
C MATRIX COMMON BLOCK
C.....
COMMON /MATRIX / DCHAE(3,3) , DCHRA(3,3) , DCHSA(3,3) ,
+ DCHSE(3,3) , DCHTS(3,3) , DCHTE(3,3) ,
+ DCHSAE(3,3) , DCHDAE(3,3) , DCHSR(3,3) ,
+ DCHDUM(3,3)
C.....
C MISCELLANEOUS DATA COMMON BLOCK
C.....
COMMON /MISC / IPAGECT(31) , LINECT(31) , IPRCNT(31) ,
+ MAXLINE , MAXREPT , MAXEVT ,
+ TERFLG , LU ,
+ IDATE , HEADALT , HEADVEL ,
+ HEADSR , HEADYAW , HEADPIT ,
+ HEADROL , HEADWGT , BIAS ,
+ REPTYPE(5,31) , PRTLNGT(2) , PRTWGT(2) ,
+ IHEADER(24) , TEVENTS(38) ,
+ IMVDC , PRTIMP( 2) ,
+ PRIMASS(2) , PRTINDX , PKZVEL

```



```

      +          ZVECT(3)       . XYZ(3)       . SAVTIME
      + XACCEL(3)       . ZACCEL(3)       . PRTNGT
      + REPTYPE         . BIAS           .
      + PRWGT          . PRIMASS        . PRTINDX
      + PREMP          .
C .....
C MOMARNS COMMON BLOCK
C .....
COMMON /MOMARNS /
+REFLNSD ,REFLNDA ,REFLNSA ,URX(6) ,URY(6) ,URZ(6)
+XSOCAC(2),YSSOCA(2),ZSSOCA(2),XSORK(6),YSORK(6),ZSORK(6),
+XSORRE ,YSSORRE ,ZSSORRE ,XSOLRE ,YSSOLRE ,ZSOLRE
+XSOMRE ,YSSOMRE ,ZSSOMRE ,XSOSBT ,YSSOBOT ,ZSSOBOT
+XSOSBI(6),YSSOSB(6),ZSSOSB(6),XRCSAC ,YRCSAC ,ZRCSAC
+XSCSAC ,YSCSAC ,ZSCSAC ,XSOSRP ,YSSOSRP ,ZSSOSRP
+
+XSASRP ,YSSASRP ,ZSSASRP ,XRDPAP(2),YRDDAP(2),ZRRDPAP(2),
+XRSBOT(6),YRSBO(6),ZRSBO(6),XSOCIP(2),YSSOCP(2),ZSSOCP(2),
+XSDDAP(2),YSSDAP(2),ZSSDAP(2),XESDAC ,YESDAC ,ZESDAC
+XSRCSAC ,YSRCSAC ,ZSRCSAC ,XRSBOT ,YRSBOT ,ZRSBOT
+XRSOB ,YRSOB ,ZRSOB ,XSOCH(3),YSSOCH(3),ZSSOCH(3),
+XRSB ,YRSB ,ZRSB
+XAACS0 ,YAACS0 ,ZAACS0 ,XASDAC ,YASDAC ,ZASDAC
+XRSOAC ,YRSOAC ,ZRSOAC ,XSCPAP(2),YSCPAP(2),ZSCPAP(2)
C .....
C INTEGRATION ROUTINE COMMON BLOCK
C .....
COMMON /RKUTTA / TIME , TIMES , DELTAT , TRAJSO(193)
+
+ TRAJSA(193) , TRAUDA(193) , TRAJCHI(97,3)
+ TRAJAC(193) , TVCEQS(225) , QUATSO(65)
+ QUATSA(65) , QUATOAA(68) , QUATAC(65)
+ INTSTP , IPCPASS , IRKPASS
+ IPOINTS , IVX , IYPRX , IYKPASS
+ IKX , KSUMX , IKPASSX
+ IVIX , IVIIX , IVI2X
+ IVI3X , IVPRIIX , IYPRIIX
+ IVPRI2X , IPYIX , IPY1IX
+ ICYIX , ICY1IX , IREIN
C
C IF (INTSTP .EQ. 0) RETURN
C .....
C CALCULATE VECTORS TO BE USED IN INVECT, INITRAJ, & DARTFM
C .....
XESOAC=TRAJAC(2)-TRAJSO(2)
YESDAC=TRAJAC(3)-TRAJSO(3)
ZEODAC=TRAJAC(4)-TRAJSO(4)
CALL ROTATE (XESOAC,XSDOAC,ZVECT,DCMSE,O)
CALL ROTATE (XESOAC,XSOAC,ZVECT,DCMAE,O)
CALL ROTATE (XESOAC,XRSOAC,ZVECT,DCMSR,1)
XAACS0=-XASOAC
YAACS0=-YASOAC
ZAACS0=-ZASOAC
XSOSCAC=XSSOAC+XCGSO
YSSCSAC=YSSOAC+YCGSO
ZSSCSAC=ZSSOAC+ZCGSO
CALL ROTATE (XRRCSCAC,XSRCSCAC,ZVECT,DCMSR,O)

```

PAGE 301

83/11/07. 09.41.53

FTN 4.6+428

74/74 OPT=1

SUBROUTINE UPDVECT

CALL ROTATE (XRSDAC,XSSOMRE,XRRCAC,DCMSR,0)
CALL ROTATE(XSSBOT,XRRSROT,XSSOMRE,DCMSR,1)
RETURN
END

115

```

1 SUBROUTINE VERTSK
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - CONTROLS SIMULATION OF THE VERTICAL SEEKING MANEUVER
C METHOD - SIMULATION OF THE VERTICAL SEEKING MANEUVER IS BASED
C ON WORK DONE AT CHINA LAKE. CURRENTLY, IT IS ASSUMED
C THAT THERE IS ONLY ONE ROCKET ON THE SEAT BOTTOM,
C WHICH CAN BE GIMBALED TO CONTROL ROLL AND PITCH
C MOVEMENT. ON THE SEAT, THE MICROPROCESSOR UPDATES
C THE DIRECTION COSINES SPECIFYING THE ORIENTATION OF
C OF THE SEAT, BASICALLY EVERY 1.25 MILLISECONDS FROM
C TIME ZERO TO ROCKET BURNOUT. FROM ROCKET IGNITION
C (+ AN INPUT TIME DELAY) UNTIL ROCKET BURNOUT
C THE DERIVATIVES FOR THE SIGNAL TRANSFER EQUATIONS
C ARE CALCULATED, AND THE TWO OUTPUT COMMANDS ARE
C GENERATED (EVERY OTHER TIME STEP)
C .....
C COMMUNICATIONS:
C CALLED BY: SEATOC
C CALLS: ROTATE
C NON-COMMON VARIABLES DEFINED:
C ANGR(1) = ROTATIONAL RATES OF SEAT/OCCUPANT IN THE TVC C.S.
C PHIT = INTERMEDIATE VALUE FOR ROLL ROCKET POSITION
C THETAT = INTERMEDIATE VALUE FOR PITCH ROCKET POSITION
C TOTTH = TOTAL ROLL, PITCH OR YAW ANGULAR DISPLACEMENT,
C COMPARED AGAINST MAXIMUM ALLOWED BEFORE
C UPDATING DIRECTION COSINES
C ADTH = ABSOLUTE VALUE OF TOTTH
C ANG1 = ONE OF TWO ANGLES GENERATED BY THE SIGNAL TRANSFER
C EQUATIONS, USED TO DETERMINE THE ROCKET THRUST LINE
C ANG2 = ONE OF TWO ANGLES GENERATED BY THE SIGNAL TRANSFER
C EQUATIONS, USED TO DETERMINE THE ROCKET THRUST LINE
C TVCDERV(14) = EQUIVALENCED TO TVCEOS(16) = THE DERIVATIVES OF THE
C THRUST VECTOR CONTROL SIGNAL TRANSFER EQUATIONS
C TVCVALS(14) = EQUIVALENCED TO TVCEOS(2) = THRUST VECTOR CONTROL
C SIGNAL TRANSFER EQUATIONS
C FOR EASE OF PROBABLE FUTURE IMPLEMENTATION OF YAW CONTROL,
C THESE EQUIVALENCE STATEMENTS ARE USED.
C .....
C THE FOLLOWING GIVES THE RELATIONSHIP BETWEEN THE
C VARIABLES IN THIS PROGRAM AND THE VARIABLES IN THE
C CHINA LAKE PROGRAM
C .....
C VALUES
C CHINA LAKE TVC ARRAY EQUIV TVC ARRAY EQUIV
C X19 TVCEOS(2) TVCVALS(1) XD19 TVCEOS(16) TVCDERV(1)
C X21 TVCEOS(3) TVCVALS(2) XD21 TVCEOS(17) TVCDERV(2)
C X27 TVCEOS(4) TVCVALS(3) XD27 TVCEOS(18) TVCDERV(3)
C X28 TVCEOS(5) TVCVALS(4) XD28 TVCEOS(19) TVCDERV(4)
C X29 TVCEOS(6) TVCVALS(5) XD29 TVCEOS(20) TVCDERV(5)
C X30 TVCEOS(7) TVCVALS(6) XD30 TVCEOS(21) TVCDERV(6)
C X31 TVCEOS(8) TVCVALS(7) XD31 TVCEOS(22) TVCDERV(7)
C X32 TVCEOS(9) TVCVALS(8) XD32 TVCEOS(23) TVCDERV(8)
C X33 TVCEOS(10) TVCVALS(9) XD33 TVCEOS(24) TVCDERV(9)
C X34 TVCEOS(11) TVCVALS(10) XD34 TVCEOS(25) TVCDERV(10)
C X35 TVCEOS(12) TVCVALS(11) XD35 TVCEOS(26) TVCDERV(11)
C X36 TVCEOS(13) TVCVALS(12) XD36 TVCEOS(27) TVCDERV(12)
C .....
C DERIVATIVES

```

5-304

5-305

```

C
175  THETAT = AMAX1 (THETAT, PITCHRL)
    THETAT = AMIN1 (THETAT, PITCHRL)
    PHIT = AMAX1 (PHIT, ROLLRL)
    PHIT = AMIN1 (PHIT, ROLLRL)
    TVCDERV(11) = 2.861 * 1000000.0 * THETAT - 31.45 * TVCVALS(11)
    TVCDERV(12) = TVCVALS(12)
    TVCDERV(13) = 2.861 * 1000000.0 * PHIT - 31.45 * TVCVALS(13)
    TVCDERV(14) = TVCVALS(14)
    TVCDERV(15) = TVCVALS(15)
    IF (TVCDERV(11) .LT. (PITCHRL * 0.25)) GO TO 120
    IF (TVCDERV(11) .LT. 0.) GO TO 130
    TVCDERV(11) = 0.
    GO TO 130
180  GO TO 130
185  GO TO 130
190  IF (TVCVALS(1) .LT. (PITCHRL * 0.25)) GO TO 130
    IF (TVCDERV(1) .GT. 0.) GO TO 130
    TVCDERV(1) = 0.
195  GO TO 130
200  IF (TVCVALS(2) .LT. (ROLLRL * 0.25)) GO TO 140
    IF (TVCDERV(2) .LT. 0.) GO TO 150
    TVCDERV(2) = 0.
    GO TO 150
205  GO TO 150
210  IF (TVCVALS(2) .LT. (ROLLRL * 0.25)) GO TO 150
    IF (TVCDERV(2) .GT. 0.) GO TO 150
    TVCDERV(2) = 0.
215  GO TO 150
220  ANG1 = TVCVALS(1) * RANG
    ANG2 = TVCVALS(2)
    URX(1) = SIN(ANG1) * COS(ANG2)
    URY(1) = SIN(ANG2)
    URZ(1) = COS(ANG1) * COS(ANG2)
225  GO TO 150
230  GO TO 150
235  GO TO 150
240  GO TO 150
245  GO TO 150
250  GO TO 150
255  GO TO 150
260  GO TO 150
265  GO TO 150
270  GO TO 150
275  GO TO 150
280  GO TO 150
285  GO TO 150
290  GO TO 150
295  GO TO 150
300  GO TO 150
305  GO TO 150
310  GO TO 150
315  GO TO 150
320  GO TO 150
325  GO TO 150
330  GO TO 150
335  GO TO 150
340  GO TO 150
345  GO TO 150
350  GO TO 150
355  GO TO 150
360  GO TO 150
365  GO TO 150
370  GO TO 150
375  GO TO 150
380  GO TO 150
385  GO TO 150
390  GO TO 150
395  GO TO 150
400  GO TO 150
405  GO TO 150
410  GO TO 150
415  GO TO 150
420  GO TO 150
425  GO TO 150
430  GO TO 150
435  GO TO 150
440  GO TO 150
445  GO TO 150
450  GO TO 150
455  GO TO 150
460  GO TO 150
465  GO TO 150
470  GO TO 150
475  GO TO 150
480  GO TO 150
485  GO TO 150
490  GO TO 150
495  GO TO 150
500  GO TO 150
505  GO TO 150
510  GO TO 150
515  GO TO 150
520  GO TO 150
525  GO TO 150
530  GO TO 150
535  GO TO 150
540  GO TO 150
545  GO TO 150
550  GO TO 150
555  GO TO 150
560  GO TO 150
565  GO TO 150
570  GO TO 150
575  GO TO 150
580  GO TO 150
585  GO TO 150
590  GO TO 150
595  GO TO 150
600  GO TO 150
605  GO TO 150
610  GO TO 150
615  GO TO 150
620  GO TO 150
625  GO TO 150
630  GO TO 150
635  GO TO 150
640  GO TO 150
645  GO TO 150
650  GO TO 150
655  GO TO 150
660  GO TO 150
665  GO TO 150
670  GO TO 150
675  GO TO 150
680  GO TO 150
685  GO TO 150
690  GO TO 150
695  GO TO 150
700  GO TO 150
705  GO TO 150
710  GO TO 150
715  GO TO 150
720  GO TO 150
725  GO TO 150
730  GO TO 150
735  GO TO 150
740  GO TO 150
745  GO TO 150
750  GO TO 150
755  GO TO 150
760  GO TO 150
765  GO TO 150
770  GO TO 150
775  GO TO 150
780  GO TO 150
785  GO TO 150
790  GO TO 150
795  GO TO 150
800  GO TO 150
805  GO TO 150
810  GO TO 150
815  GO TO 150
820  GO TO 150
825  GO TO 150
830  GO TO 150
835  GO TO 150
840  GO TO 150
845  GO TO 150
850  GO TO 150
855  GO TO 150
860  GO TO 150
865  GO TO 150
870  GO TO 150
875  GO TO 150
880  GO TO 150
885  GO TO 150
890  GO TO 150
895  GO TO 150
900  GO TO 150
905  GO TO 150
910  GO TO 150
915  GO TO 150
920  GO TO 150
925  GO TO 150
930  GO TO 150
935  GO TO 150
940  GO TO 150
945  GO TO 150
950  GO TO 150
955  GO TO 150
960  GO TO 150
965  GO TO 150
970  GO TO 150
975  GO TO 150
980  GO TO 150
985  GO TO 150
990  GO TO 150
995  GO TO 150
1000 GO TO 150

```

```

230      DTH(1,1) = TDTH - .0078125 * TDTH / ADTH
      210 CONTINUE
      C *****
      C UPDATE DIRECTION COSINES USING CROWDER-HESSION ALGORITHM
      C *****
      D3(2) = D3(2) - DTH(3,2) * D3(1)
      D3(3) = D3(3) + DTH(2,2) * D3(1)
      D3(1) = D3(1) + DTH(3,2) * D3(2)
      D3(3) = D3(3) - DTH(1,2) * D3(2)
      D3(1) = D3(1) - DTH(2,2) * D3(3)
      D3(2) = D3(2) + DTH(1,2) * D3(3)
      C *****
      C UPDATE COMMANDS EVERY OTHER TIME STEP IF BETWEEN ROCKET IGNITION
      C PLUS INPUT TIME DELAY AND ROCKET BURNOUT
      C *****
      IF (ITVCFLG .LT. 0) GO TO 9000
      IF (D3(3) .LT. 0.0) GO TO 310
      RKT CMD(1) = -D3(2)
      RKT CMD(2) = D3(1)
      GO TO 9000
      310 CONTINUE
      IF (D3(1) .NE. 0.0 .OR. D3(2) .NE. 0.0) GO TO 320
      RKT CMD(1) = 2.0
      IF (ANGR(1) .NE. 0.0) RKT CMD(1) = RKT CMD(1) * ANGR(1) / ABS(ANGR(1))
      RKT CMD(2) = 0.0
      GO TO 9000
      320 CONTINUE
      IF (ABS(D3(1)) .GE. ABS(D3(2))) GO TO 330
      RKT CMD(1) = 2.0 - ABS(D3(2))
      IF (D3(2) .NE. 0.0) RKT CMD(1) = -RKT CMD(1) * D3(2) / ABS(D3(2))
      RKT CMD(2) = D3(1)
      GO TO 9000
      330 CONTINUE
      RKT CMD(1) = -D3(2)
      RKT CMD(2) = 2.0 - ABS(D3(1))
      IF (D3(1) .NE. 0.0) RKT CMD(2) = RKT CMD(2) * D3(1) / ABS(D3(1))
      9000 CONTINUE
      RETURN
      END

```

FUNCTION ZARTAN 74/74 OPT=1 FTN 4.6+428 83/11/07. 09 41 53 PAGE 307

```

1  C ..... FUNCTION ZARTAN(A,B)
   C .....
   C DESCRIPTION - LEVEL 4
   C FUNCTION - COMPUTES THE ARCTANGENT OF A/B
   C METHOD - DETERMINES THE CORRECT QUADRANT AND USES THE SYSTEM
   C FUNCTION ATAN2
   C COMMUNICATIONS -
   C CALLED BY:
   C INITMS
   C CALLS:
   C NONE
   C NON-COMMON VARIABLES DEFINED -
   C A - SINE VALUE (PASSED IN CALL)
   C B - COSINE VALUE (PASSED IN CALL)
   C POTENTIAL ERROR CONDITIONS -
   C NONE
   C .....

20 C DATA W/1.5707963267949/
   C IF(B .NE. 0.0) GO TO 100
   C IF(A .GT. 0.0) ZARTAN = W
   C IF(A .LT. 0.0) ZARTAN = -W
   C IF(A .EQ. 0.0) ZARTAN = 0.0
   C GO TO 200
   C 100 ZARTAN = ATAN2(A,B)
   C 200 CONTINUE
   C RETURN
   C END
25

```



```

1  SUBROUTINE ZLININT(XINPUT, TABLE, NPOINTS, MAXPTS, ANSWER, INDEX)
C .....
C DESCRIPTION - LEVEL 3
C FUNCTION - THIS SUBROUTINE PERFORMS LINEAR INTERPOLATION ON A
C TWO DIMENSIONAL TABLE USING THE STANDARD LINEAR
C INTERPOLATION FORMULA
C METHOD - IN ORDER TO REDUCE THE NUMBER OF CALCULATIONS, THE
C SUBROUTINE FIRST SEARCHES FOR A FIVE ELEMENT RANGE
C THAT CONTAINS THE INDEPENDENT VARIABLE. THIS RANGE
C IS THEN SEARCHED TO FIND THE EXACT VALUE TO BE
C RETURNED. IF THE INDEPENDENT VARIABLE IS EQUAL TO
C A TABLE ELEMENT, THE CORRESPONDING ELEMENT IS
C RETURNED. IF THE INDEPENDENT VARIABLE LIES BETWEEN
C TWO TABLE ELEMENTS, THE RESULT IS DETERMINED USING
C THE STANDARD LINEAR INTERPOLATION FORMULA. THIS
C SUBROUTINE ALSO CHECKS FOR ERRORS IN THE TABLE
C SUCH AS ELEMENTS OUT OF ORDER.
C COMMUNICATIONS -
C CALLED BY: AIRCFT, CATAFM, RKTFM, DROGUE1, DROGUE2, INITRAJ, RECOV
C CALLS
C NONE
C NON-COMMON VARIABLES DEFINED:
C CALL PARAMETERS:
C XINPUT - INDEPENDENT VARIABLE TO BE INTERPOLATED
C TABLE - TWO DIMENSIONAL ARRAY WITH INTERPOLATION VALUES
C NPOINTS - NUMBER OF POINTS IN TABLE
C MAXPTS - MAXIMUM NUMBER OF POINTS IN TABLE
C ANSWER - VALUE TO BE RETURNED
C INDEX - NUMBER OF COLUMNS IN TABLE
C OTHER VARIABLES:
C ISTART - START POINT OF FIVE ELEMENT RANGE
C ISTOP - STOP POINT OF FIVE ELEMENT RANGE
C POTENTIAL ERROR CONDITIONS: NONE
C .....
35 DIMENSION TABLE(INDEX, MAXPTS), ANSWER(INDEX)
    ISTART = 1
C .....
C FIND FIVE ELEMENT RANGE IN WHICH VALUE LIES
C .....
40 10 IF (TABLE(1, ISTART) .EQ. XINPUT) GO TO 100
    ISTOP = ISTART + 5
    IF (ISTOP .GT. NPOINTS) ISTOP = NPOINTS
    IF (TABLE(1, ISTOP) .EQ. XINPUT) GO TO 200
    IF (TABLE(1, ISTOP) .GT. XINPUT) GO TO 20
    IF ((TABLE(1, ISTOP) .LT. XINPUT) .AND. (ISTOP .EQ. NPOINTS)) GO TO 30
    ISTART = ISTOP + 5
    GO TO 10
C .....
50 C NARROW SEARCH TO FIVE ELEMENT RANGE
C .....
20 ISTOP1 = ISTOP - 1
    DO 30 I = ISTART, ISTOP1
        IF (TABLE(1, I+1) .EQ. XINPUT) GO TO 300
        IF (TABLE(1, I) .LT. XINPUT .AND. TABLE(1, I+1) .GT. XINPUT)
            * GO TO 400
30 CONTINUE
    DO 40 J=2, INDEX

```

SUBROUTINE ZLININI 74/74 OPT=1 FIN 4 6+428 83/11/07 09.41.53 PAGE 309

```

60      ANSWER(J-1) = TABLE(J,NPOINTS)
        40 CONTINUE
        GO TO 1000
C .....
C RETURN ANSWER .....
C .....
65      100 DO 150 J=2,INDEX
        ANSWER(J-1) = TABLE(J,ISTART)
        150 CONTINUE
        GO TO 1000
        200 DO 250 J=2,INDEX
        ANSWER(J-1) = TABLE(J,ISTOP)
        250 CONTINUE
        GO TO 1000
        300 DO 350 J=2,INDEX
        ANSWER(J-1) = TABLE(J,I+1)
        350 CONTINUE
        GO TO 1000
75      400 DO 450 J=2,INDEX
        ANSWER(J-1) = TABLE(J,I) + ((XINPUT - TABLE(I,I))/
        + (TABLE(I,I+1) - TABLE(I,I))) * (TABLE(J,I+1) - TABLE(J,I))
        450 CONTINUE
        1000 RETURN
        END
80

```

6.0 REFERENCES

1. White, B. J., *Aeromechanical Properties of Ejection Seat Escape Systems*, Technical Report AFFDL-TR-74-57, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, April 1974.
2. ACT, A CDC 6700 Computer Program for Generating Random Files of Aerodynamic Coefficient Tables, Technical Note TN-K-1/74, Naval Weapons Laboratory, Dahlgren, Virginia, January 1974.
3. Hardy, S., unpublished notes of ICARUS program validation, Aerobalistics Division, Naval Weapons Laboratory, Dahlgren, Virginia.
4. Control Data Cyber 70 Series Models 72/73/74 6000 Series Computer Systems KRONOS 2.1 Reference Manual. Publication Number 60407000. Control Data Corporation, March 20, 1979.

NADC-84068-60

(THIS PAGE INTENTIONALLY LEFT BLANK)

PAGE 1

83/09/08 16.09.54

FTN 4 6+428

PROGRAM ACT 74/74 OPT=1

```

1  PROGRAM ACT(INPUT,OUTPUT,TAPE1=513,TAPE10)
   DIMENSION WORKA(700),FMT(8),INDEX(53)
   DIMENSION INFO2(30,10),INFO3(20,13)
   EQUIVALENCE (INFO2(1,1),INFO3(1,1))
5  COMMON/SHARE/FMT(8),NOFTAB,ITYPE,MODE,IC,ISEQNO,MTEST,IBLNK
   COMMON/RANGE/IDENT,NAL,NBE,NMA,DELALP,DELDET,DELMAC,ALMIN,
1  BEMIN,MAMIN,ALMAX,BEMAX,MAMAX,NTYP
   COMMON/NTABS/12NO,13NO,1TABNO
   COMMON/RFINFO/INFO2(30,10)
   COMMON/ERCOM/NODEF,IERTEST,1J,1K
   REAL INFO2,INFO3
   REAL NAL,NBE,NMA,MAMIN,MAMAX,IDENT
   IBLNK=10000000000
   CALL OPENMS(1,INDEX,53,0)
15  TIME=SECOND(SECS)
   IERTEST=0
   C THE FIRST CARD OF THE DATA DECK IS READ
   READ(10,100) ITYPE,MODE,NOFTAB,NTYP
   C ITYPE = 1 DATA FOR TWO-WAY TABLES
   C = 2 DATA FOR THREE-WAY TABLES
20  C MODE = 1 CREATION RUN
   C = 2 EXTENSION RUN
   C = 3 REPLACEMENT RUN
   C = 4 SHORT LISTING
   C = 5 LONG LISTING
25  C NOFTAB - NUMBER OF TABLES TO BE READ
   C NOT DEFINED FOR MODES 4 AND 5
   C NTYP - TYPE OF TWO-WAY TABLE
   C = 1 BETA VS MACH
   C = 2 BETA VS ALPHA
30  C = 3 ALPHA VS MACH
   C FORMAT(415)
   C IF(NTYP.EQ.0)NTYP=1
   C MTEST IS USED BY THE WRINFO AND PTOUT SUBROUTINES
35  C MTEST = 0 THE ENTIRE RANDOM FILE CONTAINING THE INFO ARRAY
   C WILL BE READ WHEN MODE = 2 OR 3
   C = 1 NOT NECESSARY TO READ INFO ARRAY AGAIN
   C = 9 INSTRUCTS THE PTOUT ROUTINE TO PRINT OUT
   C THE FILE DICTIONARY ONLY
40  MTEST=0
   IF(EOF(10).NE.0)GO TO 99
   IF(ITYPE.EQ.1)NAL=1
   C YOU INITIALIZE THE INFO ARRAY BY SETTING THE IDENT POSITION IN THE
   C INFO ARRAY TO 000000000 FOR LATER TESTING
45  DO 13 J=1,30
   INFO2(J,1)=IBLNK
13  CONTINUE
   C IS THIS JUST A LIST[ RUN
   C IF(MODE.GE.4)GO TO 98
   C TEST FOR MAX. NUMBER OF TABLES
50  IF(NOFTAB.GT.30.AND.ITYPE.EQ.1)CALL ERMMSG(1)
   IF(NOFTAB.GT.20.AND.ITYPE.EQ.2)CALL ERMMSG(1)
   C THE SECOND CARD OF THE DATA DECK GIVES THE FORMAT THAT
   C THE TABLES ARE TO BE READ UNDER
   READ(10,110) FMT
55  FORMAT(8A10)
   C BEGIN READING EACH TABLE. LET ICOUNT KEEP COUNT OF

```

```

60 C THE CURRENT TABLE NUMBER
   DO 11 ICOUNT=1,NOFTAB
     IF(MODE.NE.2)IC=ICOUNT
     IF(MODE.EQ.2.AND.MTEST.NE.O)IC=IC+1
65 C THE THIRD CARD OF THE DATA DECK DEFINES
   C IDENT - THE TABLE NAME
   C NCOEF - THE TOTAL NUMBER OF COEFFICIENTS,
   C ISEQNO - THE SEQUENCE NO. OF THE TABLE TO BE REPLACED(WHEN MODE=3)
119 READ(10,120) IDENT,NCOEF,ISEQNO
120 FORMAT(A10,I5,I5)
70 IF(MODE.EQ.3)GO TO 16
   IF(MODE.EQ.2.AND.MTEST.EQ.O)GO TO 16
   IF(ITYPE.EQ.1)INFO3(IC,1)=IDENT
   IF(ITYPE.EQ.2)INFO3(IC,1)=IDENT
16 IF(NCOEF.GT.700)CALL ERMSG(2)
   IF(IERTEST.EQ.3)GO TO 9
   IF(MODE.NE.3)GO TO 14
75 C PRIOR TO REPLACING A TABLE, THE RANDOM FILE CONTAINING
   C THE INFO ARRAY MUST BE READ
   CALL WRINFO(3)
   IF(IERTEST.EQ.3)GO TO 9
   IC=ITABNO
80 GO TO 15
   C READ THE INFO ARRAY ON THE FIRST PASS OF AN EXTENSION RUN
14 IF(MODE.EQ.2.AND.MTEST.EQ.O)CALL WRINFO(2)
   IF(IERTEST.EQ.1)GO TO 12
85 IF(ITYPE.EQ.1)INFO3(IC,1)=IDENT
   IF(ITYPE.EQ.2)INFO3(IC,1)=IDENT
   C CALL ROUTINE TO READ THE DATA FOR EACH TABLE
   CALL INPUTT(WORKA,NCOEF)
   IF(IERTEST.EQ.4)GO TO 95
90 INAL=NAL+.50
   INBE=NBE+.50
   INMA=NMA+.50
   C THE TABLE THAT WAS CREATED IS NOW PRINTED OUT
   CALL PTOUT(WORKA,INAL,INBE,INMA)
95 IF(IERTEST.EQ.3)IERTEST=O
   C THIS PROCESS IS CARRIED OUT UNTIL ALL TABLES, AS SPECIFIED
   C BY NOFTABS, HAVE BEEN READ
11 CONTINUE
   IF(IERTEST.EQ.2)CALL ERMSG(4)
100 C AFTER ALL TABLES HAVE BEEN CREATED, THE INFO ARRAY IS
   C WRITTEN TO THE PROPER RANDOM FILE
95 CALL WRINFO(1)
   IF(ITYPE.EQ.2)I3NO=IC
   IF(ITYPE.EQ.1)I2NO=IC
105 IF(IC.EQ.O)GO TO 94
   IF(MODE.EQ.3)CALL WRINFO(2)
   MTEST=9
   C THE PTOUT ROUTINE NOW PRINTS OUT THE FILE DICTIONARY
   CALL PTOUT(WORKA,1,1,1)
   GO TO 94
110 C ON LIST OPTION, DETERMINE THE NUMBER OF TABLES ON FILE
98 CALL WRINFO(2)
   IF(MODE.NE.5)GO TO 97
   IF(ITYPE.EQ.1)NT=I2NO

```

PROGRAM ACT 74/74 OPT=1

```

115 IF(1TYPE.EQ.2)NT=13ND
    MTEST=1
    DO 20 IC=1,NT
      C READ THE AERODYNAMIC COEFFICIENTS INTO MEMORY
      CALL READP(IC,WORKA)
      INAL=NAL+.50
      INBE=NBE+.50
      INMA=NMA+.50
      C PRINT OUT THE TABLE
      CALL PTOUT(WORKA,INAL,INBE,INMA)
20  CONTINUE
97  MTEST=9
      C PRINT OUT CP TIME USED
94  TIME=SECOND(SECS)-TIME
      IHOURS=TIME/3600.
      MINS=AMOD(TIME,3600.)/60.
      ISECS=TIME-FLOAT(MINS*60+IHOURS*3600)
      PRINT 93,IHOURS,MINS,ISECS
      FORMAT(1H0,15X,41HCENTRAL PROCESSOR TIME FOR THIS CASE WAS ,13,
1    6H HOURS,13.5H MINS,13.5H SECS)
      IF(1TEST.EQ.4)STOP
      GO TO 12
99  STOP
    END

```

```

1  SUBROUTINE ERMSG(IERNO)
C THE SUBROUTINE ERMSG IS CALLED WHENEVER AN ERROR IS DETECTED.
C IT PRINTS OUT AN APPROPRIATE ERROR MESSAGE AND TAKES THE NECESSARY
C CORRECTIVE ACTION. A FLAG (ERTEST) IS SET AND PASSED
C TO THE CALLING ROUTINE.
5  C ERTEST = 1 PRESENT CASE IGNORED. READ NEXT TYPE 1 CARD
C = 2 AFTER PROCESSING NOFTAB TABLES. RETURN TO ERMSG ROUTINE
C TO READ REMAINING TABLES INTO DUMMY
C = 3 DISREGARD PRESENT TABLE. CONTINUE READING NEXT TABLE
C = 4 STOP PROGRAM AFTER PROCESSING NOFTAB TABLES
10 C DIMENSION ITYP(2)
COMMON/ERCOM/NCOEFF,ERTEST,IJ,IK
COMMON/SHARE/FMT(8),NOFTAB,ITYPE,MODE,IC,ISEQNO,MTEST,IBLANK
INTEGER ERTEST
15 DATA ITYP/2,3/
GO TO (10,20,30,40,50,60,70,80,90),IERNO
PRINT 11
11 FORMAT(8H1)*** ERROR *** ATTEMPTING TO INPUT MORE THAN THE MAXIMUM
1 NUMBER OF TABLES ALLOWED)
GO TO (14,15),ITYPE
20 PRINT 12,ITYP(2)
12 FORMAT(15X,23HPROGRAM WILL END AFTER ,I1,26H0 TABLES HAVE BEEN CRE
ATED)
NOFTAB=30
GO TO 16
15 PRINT 12,ITYP(1)
NOFTAB=20
16 ERTEST=4
RETURN
30 PRINT 21
21 FORMAT(59H1)*** ERROR *** ATTEMPT TO DEFINE MORE THAN 700 COEFFICIE
NTS)
PRINT 22
22 FORMAT(15X,61HTHIS TABLE IS DISREGARDED - PROGRAM CONTINUES WITH N
EXT TABLE)
24 READ(10,25) DUMMY
25 FORMAT(A1)
READ(10,FMT) (DUMMY,I=1,NCOEFF)
ERTEST=3
RETURN
40 PRINT 31
31 FORMAT(68H1)*** ERROR *** ATTEMPTING TO USE THE REPLACE MODE TO REP
LACE A TABLE,/,19X,29H THAT HAS NOT YET BEEN DEFINED)
PRINT 22
GO TO 24
45 PRINT 51,ITYP(1)
51 FORMAT(29H1)*** ERROR *** MAX. NUMBER OF 12,27H-WAY TABLES ALREADY
DEFINED,/,15X,24H THEY CAN NOT BE EXTENDED)
PRINT 53
53 FORMAT(15X,56H EXTENSION TABLES ARE DISREGARDED AND EXECUTION CONTI
NUES)
52 READ(10,25) DUMMY
READ(10,FMT) (DUMMY,I=1,NCOEFF)
55 IF(NFTAB.EQ.1)GO TO 58
DO 55 1=2,NOFTAB
READ(10,56) NCoeff
56 FORMAT(10X,15)

```


83/09/08 16.09.54

FTN 4.6+428

SUBROUTINE ERMMSG 74/74 OPT=1

```

60      READ(10,25) DUMMY
55      READ(10,FMT) (DUMMY,J=1,NCOEF)
58      CONTINUE
      ERTEST=1
      RETURN
60      PRINT 61,ITYP(1)
81      FORMAT(39H1*** ERROR *** ATTEMPTING TO EXTEND THE 12.32H-WAY TABLE
        15 BEYOND THEIR MAXIMUM)
65      PRINT 62
      FORMAT(15X,70HTABLES WILL BE EXTENDED TO THEIR MAXIMUM AND THE REM
        1AINDER DISREGARDED)
70      LEFT=NOFTAB
      NOFTAB=31-IJ
      LEFT=LEFT-NOFTAB
65      ERTEST=2
      RETURN
70      PRINT 51,ITYP(2)
75      GO TO 52
80      PRINT 61,ITYP(2)
      LEFT=NOFTAB
      NOFTAB=21-IK
      GO TO 65
80      NOFTAB=LEFT+1
      GO TO 57
90      PRINT 91
91      FORMAT(77H1*** ERROR *** NUMBER OF GIVEN DOES NOT EQUAL NUMBER OF
        1COMPUTED COEFFICIENTS./,15X,49HPROGRAM WILL END AFTER WRITING INFO
        2 ARRAY TO FILE)
85      ERTEST=4
      RETURN
      END

```

```

1  SUBROUTINE INPUTT(WORKA,NCDEF)
C  INPUTT READS CARD TYPES 4 AND 5 AND DEFINES THE INFO ARRAY
C  DIMENSION WORKA(NCDEF)
C  EQUIVALENCE (INFO2(1,1),INFO3(1,1))
C  COMMON/SHARE/FMT(8),NOFTAB,ITYPE,MODE,IC,ISEQNO,MTEST,IBLNK
C  COMMON/RFINFO/INFO2(30,10)
C  COMMON/RANGE/IDENT,NAL,NBE,NMA,ALMAX,DELALP,DELALP,DELALP,ALMIN,
1  BEMIN,MAMIN,ALMAX,BEMAX,MAMAX,NTYP
C  COMMON/ERCOM/NUMCO,IERTEST,IJ,IK
C  REAL INFO2,INFO3
C  REAL NAL,NBE,NMA,MAMIN,MAMAX,IDENT
C  THE FOURTH CARD OF THE DATA DECK CONTAINS INFO. FOR
C  THE GENERATION OF THE TABLES
C  READ THE MAX, MIN, AND DELTA FOR ALPHA, BETA, AND MACH
C  READ(10,200) ALMIN,ALMAX,DELALP,BEMIN,BEMAX,DELALP,MAMIN,MAMAX,
1  DELMAC
200 FORMAT(19F8.0)
C  CALCULATE THE NUMBER OF BETA AND MACH
C  NBE=(BEMAX-BEMIN)/DELALP+1.0
C  NMA=(MAMAX-MAMIN)/DELMAC+1.0
C  IF(ITYPE.EQ.1)GO TO 20
C  INFO ARRAY FOR 3-WAY TABLES
C  COMPUTE NUMBER OF ALPHA WHEN CREATING 3-WAY TABLES
C  NAL=(ALMAX-ALMIN)/DELALP+1.0
C  INFO3(IC,2)=NAL
C  INFO3(IC,3)=NBE
C  INFO3(IC,4)=NMA
C  DOES THE NUMBER OF ALPHA, BETA, AND MACH AGREE WITH
C  THE GIVEN NUMBER OF COEFFICIENTS
C  MYCOEF=NAL*NBE*NMA+.50
C  IF(NCDEF.NE.MYCOEF)CALL ERMSG(9)
C  IF(ITERTEST.EQ.4)GO TO 32
C  INFO3(IC,5)=DELALP
C  INFO3(IC,6)=DELALP
C  INFO3(IC,7)=DELMAC
C  INFO3(IC,8)=ALMIN
C  INFO3(IC,9)=BEMIN
C  INFO3(IC,10)=MAMIN
C  INFO3(IC,11)=ALMAX
C  INFO3(IC,12)=BEMAX
C  INFO3(IC,13)=MAMAX
C  NUMA=NAL+.5
C  NUMB=NBE+.5
C  NUMM=NMA+.5
C  GO TO 21
C  INFO ARRAY FOR 2-WAY TABLES
20  INFO2(IC,2)=NBE
C  INFO2(IC,3)=NMA
C  MYCOEF=NBE*NMA+.50
C  IF(NCDEF.NE.MYCOEF)CALL ERMSG(9)
C  IF(ITERTEST.EQ.4)GO TO 31
C  INFO2(IC,4)=DELALP
C  INFO2(IC,5)=DELMAC
C  INFO2(IC,6)=BEMIN
C  INFO2(IC,7)=MAMIN
C  INFO2(IC,8)=BEMAX

```

PAGE 7

83/09/08. 16.09.54

FTN 4.6*428

SUBROUTINE INPUT 74/74 OPT=1

```

      INFO2(IC,9)=MAMAX
      INFO2(IC,10)=NTYP
      NUMA=NAI* 5
      NUMB=NBE* 5
      NUMM=NMA* 5
      C READ THE DATA FOR THE NEXT TABLE
      C STIPULATE THAT THE DATA MUST BE READ IN THE
      C FOLLOWING ORDER - C(ALPHA,BETA,MACH),ALPHA=1,NAL
      C BETA=1,NBE
      C MACH=1,NMA
      C READ TWO-WAY TABLE DATA
      DO 22 J=1,NUMM
      22 READ(10,FMT) (WORKA((J-1)*NUMB+1),I=1,NUMB)
      GO TO 23
      C READ THREE-WAY TABLE DATA
      21 DO 25 J=1,NUMM
      DO 25 K=1,NUMB
      25 READ(10,FMT) (WORKA((J-1)*NUMB+K*(K-1)+NUMA+1),I=1,NUMA)
      CONTINUE
      23 IRF=IC
      IF(ITYPE.EQ.1)IRF=20+IC
      IF(MODE.EQ.3)IRF=ISEONO
      C WRITE THE DATA ON THE RANDOM FILE SPECIFIED BY IRF
      CALL WRITMS(1,WORKA,NCOEF,IRF)
      RETURN
      C WHEN THE NUMBER OF COEFFICIENTS IS IN ERROR, BLANK OUT
      C IDENT FIELD BEFORE WRITING INFO ARRAY TO RANDOM FILE
      85 INFO2(IC,1)=IBLNK
      33 IC=IC-1
      RETURN
      32 INFO3(IC,1)=IBLNK
      GO TO 33
      90 END

```

```

1      SUBROUTINE PTOUT(WORKA,NAL,NBE,NMA)
C      THIS SUBROUTINE PRINTS THE AERODYNAMIC COEFFICIENT TABLES
C      OR THE FILE DICTIONARY
      DIMENSION WORKA(NAL,NBE,NMA)
      DIMENSION ALPHA(250),BETA(250),MACH(250)
      DIMENSION INFO2(30,10),INFO3(20,13)
      DIMENSION OPT(5),TABNO(2),DICT1(2),DICT2(11)
      EQUIVALENCE (INFO2(1,1),INFO3(1,1))
      COMMON/SHARE/FMT(8),NOFTAB,ITYPE,MODE,IC,ISEQNO,MTEST,IBLINK
      COMMON/RANGE/IDENT,NALP,NBET,NMAC,DELALP,DELBET,DELMAC,ALMIN,
1      BEMIN,MAMIN,ALMAX,BEMAX,MAMAX,NTYP
      COMMON/NTABS/12NO,13NO
      COMMON/RFINFO/INFO2(30,10)
      INTEGER OPT,TABNO,DICT1,DICT2,TABTST
      REAL INFO2,INFO3
      REAL NALP,NBET,NMAC,MAMIN,MAMAX,MACH,IDENT
      DATA DICT1/10HSEQUENCE,10HTABLE /
      DATA DICT2/10HNUMBER,10HIDENTIFIER,10H ALMIN,
1      10H ALMAX,10H DELALP,10H BEMIN,
2      10H BEMAX,10H DELBET,10H MAMIN,
3      10H MAMAX,10H DELMAC/
      DATA OPT,TABNO/7HICREATE,7HEXTEND,7HREPLACE,2*7HLIST
1      1      5H TWO,5HTHREE/
C      WHEN MTEST=9, YOU JUST PRINT OUT THE FILE DICTIONARY
      IF (MTEST.EQ.9)GO TO 40
      N1=1
      NMA=NMA
      TABTST=0
      NCARY=0
      IF (ITYPE.EQ.1)GO TO 31
C      COMPUTE ALPHA(1), BETA(1), AND MACH(1), USING THE
C      MIN., MAX., AND DELTA VALUES FOR THESE VARIABLES
      ALPHA(1)=ALMIN
      DO 23 I=2,NAL
      ALPHA(I)=ALPH(I-1)+DELALP
23      CONTINUE
      MACH(1)=MAMIN
      DO 20 I=2,NMA
      MACH(I)=MACH(I-1)+DELMAC
20      CONTINUE
      BETA(1)=BEMIN
      DO 21 I=2,NBE
      BETA(I)=BETA(I-1)+DELBET
21      CONTINUE
C      PRINT TABLE HEADER, NAME, AND NUMBER
      PRINT 160,OPT(MODE),TABNO(ITYPE)
160      FORMAT(11H1NOW UNDER, A8,26HOPTION FOR RANDOM FILE OF,
1      A5,12H-WAY TABLES,.)
      IN=IC
      IF (ITYPE.EQ.1)IN=IC+20
      PRINT 140,IN
140      FORMAT(/,16H TABLE NUMBER = ,I3)
      PRINT 150,IDENT
150      FORMAT(14H TABLE NAME = ,A10)
      IF (ITYPE.EQ.1)GO TO 26

```

```

60      ISAVMA=NMA
      DO 25 L=1,NAL
      PRINT 170,ALPH(L)
170     FORMAT(///.20X,18HANGLE OF ATTACK = .E10.4)
      C PRINT THE NBE * NMA TABLE FOR 3-WAY TABLES.
      C THERE WILL BE NAL SUB-TABLES
177     CONTINUE
26      PRINT 174
174     FORMAT(///.10H ANGLES OF)
      C PRINT HEADINGS ACCORDING TO [NTYP]
      GO TO (32,33,34),NTYP
32      PRINT 171
171     FORMAT(9H SIDESLIP,T30,12HMACH NUMBERS./)
181     IF(NMA.GT.8)GO TO 190
176     CONTINUE
172     PRINT 172,(MACH(I),I=N1,NNMA)
      FORMAT(T13.8(E10.4,5X))
175     PRINT 175
      FORMAT(2H )
      IF(I1TYPE.EQ.1)GO TO 27
      DO 22 M=1,NBE
      PRINT 173,BETA(M),(WORKA(L,M,N),N=N1,NNMA)
173     FORMAT(E11.4,T13.8(E15.8))
22     CONTINUE
29     IF(TABTST.EQ.1)GO TO 192
      N1=1
      NCARY=0
      NMA=ISAVMA
25     CONTINUE
      RETURN
27     DO 28 M=1,NBE
      PRINT 173,BETA(M),(WORKA(1,M,N),N=N1,NNMA)
28     CONTINUE
      IF(TABTST.EQ.1)GO TO 192
      RETURN
      C IF THERE ARE MORE THAN 8 MACH VALUES, THE TABLE MUST BE CONTINUED
      C ON ANOTHER LINE
190     NNMB=NMA-8
      NNMA=8+NCARY
      NCARY=8
      TABTST=1
      GO TO 176
192     NMA=NNMB
      TABTST=0
      N1=N1+8
      NNMA=NNMA+NMA
      GO TO 177
105     PRINT 180
180     FORMAT(9H SIDESLIP,T30,18HANGLE OF ATTACK./)
34     PRINT 182
182     FORMAT(7H ATTACK,T30,12HMACH NUMBERS./)
      GO TO 181
110
      C
      C
      C THE FOLLOWING CODE WILL PRINT THE FILE DICTIONARY
40     IF(I1TYPE.EQ.1)GO TO 101

```

```

115      PRINT 100, I3NO, TABNO(2)
          FORMAT(30H THIS RANDOM FILE NOW CONTAINS, I3, I1X, A5,
1          12H-WAY TABLES.)
          GO TO 102
101      PRINT 100, I2NO, TABNO(1)
102      PRINT 110
110      FORMAT(//, 24H FILE DICTIONARY FOLLOWS.//)
120      PRINT 120, DICT1
          FORMAT(1X, 11A10, /)
          PRINT 120, DICT2
          IF(I1TYPE.EQ.1) GO TO 131
          DO 121 J=1, I3NO
          IDENT=INFO3(J, 1)
          PRINT 130, J, IDENT, INFO3(J, 8), INFO3(J, 11), INFO3(J, 5),
1          INFO3(J, 9), INFO3(J, 12), INFO3(J, 6), INFO3(J, 10),
2          INFO3(J, 13), INFO3(J, 7)
130      FORMAT(I3, 8X, A10, 9F10.4)
121      CONTINUE
131      DO 132 J=1, I2NO
          K=J+20
          IDENT=INFO2(J, 1)
          NTYP=INFO2(J, 10)
          PRINT VALUES IN CORRECT COLUMNS
          CHECK WHICH TYPE OF TWO-WAY TABLE
          GO TO (123, 124, 125), NTYP
123      PRINT 134, K, IDENT, INFO2(J, 6), INFO2(J, 8), INFO2(J, 4),
1          INFO2(J, 7), INFO2(J, 9), INFO2(J, 5)
134      FORMAT(I3, 8X, A10, 30X, 6F10.4)
          GO TO 132
124      PRINT 130, K, IDENT, INFO2(J, 7), INFO2(J, 9), INFO2(J, 5),
1          INFO2(J, 6), INFO2(J, 8), INFO2(J, 4)
          GO TO 132
125      PRINT 135, K, IDENT, INFO2(J, 6), INFO2(J, 8), INFO2(J, 4),
1          INFO2(J, 7), INFO2(J, 9), INFO2(J, 5)
135      FORMAT(I3, 8X, A10, 3F10.4, 30X, 3F10.4)
132      CONTINUE
          RETURN
          END

```

```

1      SUBROUTINE READRF(IT,WORKA)
C      THIS ROUTINE READS THE TABLES FROM THE RANDOM FILE TO MEMORY
      DIMENSION WORKA(700)
5      EQUIVALENCE (IF2(1,1),IF3(1,1))
      COMMON/SHARE/FMT(8),NOFTAB,ITYPE,MODE,IC,ISEQNO,MTEST,IBLNK
      COMMON/RANGE/IDENT,NAL,NBE,NMA,DELALP,DELBET,DELMAC,ALMIN,
1      BEMIN,MAMIN,ALMAX,BEMAX,MAMAX,NTYP
      COMMON/RFINFO/IF2(30,10)
10     REAL IF2,IF3
      REAL NAL,NBE,NMA,MAMIN,MAMAX,IDENT
      GO TO (10,20),ITYPE
10     IDENT=IF2(IT,1)
      NBE=IF2(IT,2)
      NMA=IF2(IT,3)
      DELBET=IF2(IT,4)
      DELMAC=IF2(IT,5)
      BEMIN=IF2(IT,6)
      MAMIN=IF2(IT,7)
      BEMAX=IF2(IT,8)
      MAMAX=IF2(IT,9)
      NTYP=IF2(IT,10)
      RMCF=NBE*NMA
      MCF=RMCF+.50
      IRF=IT+20
25     CALL READMS(1,WORKA,MCF,IRF)
      RETURN
20     IDENT=IF3(IT,1)
      NAL=IF3(IT,2)
      NBE=IF3(IT,3)
      NMA=IF3(IT,4)
      DELALP=IF3(IT,5)
      DELBET=IF3(IT,6)
      DELMAC=IF3(IT,7)
      ALMIN=IF3(IT,8)
      BEMIN=IF3(IT,9)
      MAMIN=IF3(IT,10)
      ALMAX=IF3(IT,11)
      BEMAX=IF3(IT,12)
      MAMAX=IF3(IT,13)
      RMCF=NAL*NBE*NMA
      MCF=RMCF+.50
40     CALL READMS(1,WORKA,MCF,IT)
      RETURN
45     END

```

```

1  SUBROUTINE WRINFO(KTST)
C  WRINFO PERFORMS VARIOUS OPERATIONS ON THE INFO ARRAY
C  IT MAY BE USED TO WRITE IT ON DISK, READ THE INFO ARRAY FROM
C  DISK INTO MEMORY, OR CHECK THE CONTENTS OF A CERTAIN
5  TABLE NUMBER
C  DIMENSION INFO2(30,10),INFO3(20,13)
C  EQUIVALENCE (INFO2(1,1),INFO3(1,1))
C  COMMON/SHARE/FMT(8),NOFTAB,ITYPE,MODE,IC,ISEONO,MTEST,IBLNK
C  COMMON/RFINFO/INFO2(30,10)
C  COMMON/RANGE/IDENT,NAL,NBE,NMA,DELALP,DELBET,DELMAC,ALMIN,
1  BEMIN,MAMIN,ALMAX,BEMAX,MAMAX,NTYP
C  COMMON/NTABS/I2NO,I3NO,ITABNO
C  COMMON/ERCOM/NCDEF,IERTEST,IJ,IK
C  REAL INFO2,INFO3
15  REAL NAL,NBE,NMA,MAMIN,MAMAX,IDENT
C  KTST DETERMINES THE PROCEDURE TO BE USED TO READ
C  OR WRITE THE INFO ARRAY
C  GO TO (30,22,33),KTST
20  C  KTST = 1
C  AFTER ALL TABLES HAVE BEEN READ, WRITE THE INFO ARRAY
C  TO THE PROPER RANDOM FILE
30  GO TO (31,32),ITYPE
31  NUMB=300
C  CALL WRITMS(1,INFO2(1,1),NUMB,52)
C  RETURN
25  NUMB=260
C  CALL WRITMS(1,INFO3(1,1),NUMB,51)
C  RETURN
C  KTST = 2
30  C  PROCEDURE FOR CHANGING INFO ARRAY ON AN EXTENSION RUN
C  AND FOR DETERMINING THE NUMBER OF TABLES ON A LIST RUN
22  GO TO (23,24),ITYPE
C  READ THE INFO ARRAY FROM THE RANDOM FILE
23  CALL READMS(1,INFO2(1,1),300,52)
35  C  FIND THE LAST TABLE NUMBER BY COMPARING
C  THE IDENT FIELD TO IBLNK(*0000000000)
DO 25 IJ=1,30
IF(INFO2(IJ,1) EQ IBLNK)GO TO 26
25  CONTINUE
IF(MODE.NE.2)GO TO 50
CALL ERMSG(5)
IF(IERTEST.EQ.1)RETURN
26  IC=IJ
IF(MODE.NE.2)GO TO 50
C  TEST TO SEE WHETHER THE TABLES CAN BE EXTENDED
C  BY NOFTABS WITHOUT EXCEEDING THE LIMIT
LNTH2=IJ+NOFTAB
IF(LNTH2.GT.30)CALL ERMSG(6)
MTEST=1
RETURN
50  C  SAME PROCEDURE AS ABOVE IS FOLLOWED FOR 3-WAY TABLES
24  CALL READMS(1,INFO3(1,1),260,51)
DO 27 IK=1,20
IF(INFO3(IK,1).EQ.IBLNK)GO TO 28
27  CONTINUE
IF(MODE.NE.2)GO TO 51
CALL ERMSG(7)

```



```

28      IF(IERTST.EQ.1)RETURN
        IC=1K
        IF(MODE.NE.2)GO TO 51
        LNTH3=1K*NOFTAB
        IF(LNTH3.GT.20)CALL ERMSG(8)
        MTEST=1
        RETURN
65      C PROCEDURE FOR DETERMINING THE NUMBER OF TABLES
        C WHEN RUNNING UNDER LIST OPTION
        IF(1J.EQ.30)1J=1J+1
        C SET I2ND AS THE NUMBER OF 2-WAY TABLES ON RANDOM FILES
        I2ND=1J-1
        RETURN
70      IF(1K.EQ.20)1K=1K+1
        C SET I3ND AS THE NUMBER OF 3 WAY TABLES ON RANDOM FILES
        I3ND=1K+1
        RETURN
75      C KIST = 3
        C PROCEDURE FOR CHANGING INFO ARRAY ON A REPLACE RUN
        C READ INFO ARRAY FROM RANDOM FILE AND TEST TO BE SURE THAT
        C THE GIVEN SEQUENCE NUMBER IS A VALID TABLE
        IF(MTEST.EQ.1)GO TO (43,44),1TYPE
        GO TO (41,42),1TYPE
        CALL READMS(1,INFO2(1,1),300,52)
        ITABND=ISEQNO-20
        IF(INFO2(ITABND,1).EQ.IBLNK)CALL ERMSG(3)
        MTEST=1
        RETURN
85      CALL READMS(1,INFO3(1,1),260,51)
        ITABND=ISEQNO
        IF(INFO3(ITABND,1).EQ.IBLNK)CALL ERMSG(3)
        MTEST=1
        RETURN
90      END

```