

AD-A085 147

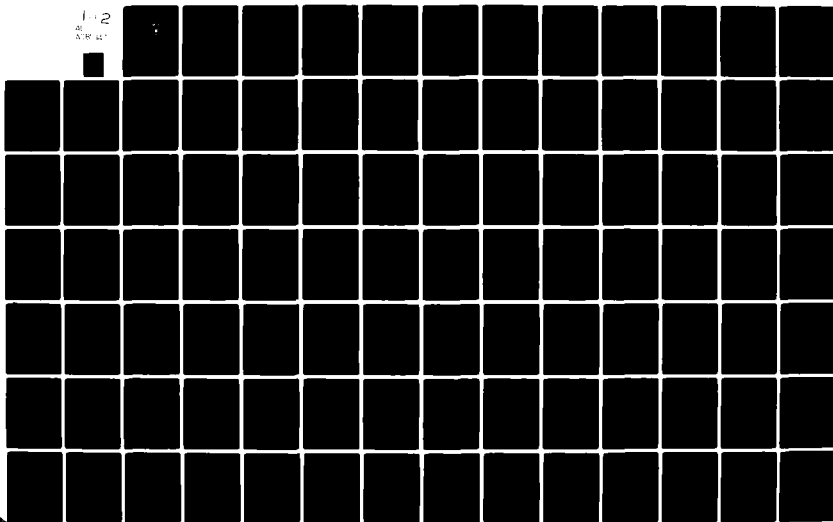
NAVAL POSTGRADUATE SCHOOL MONTEREY CA
GUIDANCE AND CONTROL OF TACTICAL MISSILES.(U)
DEC 79 T A GROTE

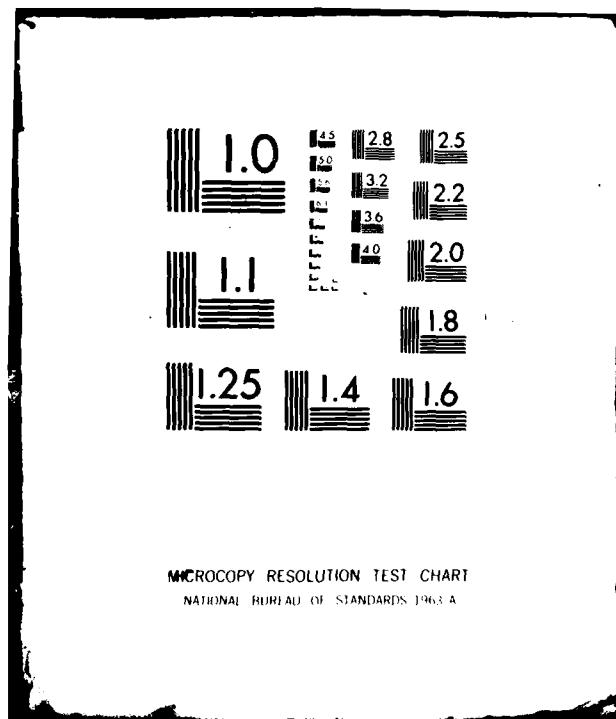
F/G 16/2

UNCLASSIFIED

NL

1-2
AD-A085 147





154
ADA 085147

LEVEL II (2)

NAVAL POSTGRADUATE SCHOOL

Monterey, California



DTIC
ELECTE
S JUN 5 1980

THESIS

A

GUIDANCE AND CONTROL OF TACTICAL MISSILES

by

Thomas Alan Grote

December 1979

Thesis Advisor:

D. J. Collins

Approved for public release; distribution unlimited

DDC FILE COPY

80 6 3 03

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM								
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD A085 147	3. RECIPIENT'S CATALOG NUMBER (9)								
4. TITLE (and Subtitle) Guidance and Control of Tactical Missiles.		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis (December 1979)								
7. AUTHOR(s) Thomas Alan Grote		6. PERFORMING ORG. REPORT NUMBER								
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		8. CONTRACT OR GRANT NUMBER(s)								
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS								
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940		12. REPORT DATE December 1979								
		13. NUMBER OF PAGES 173								
		15. SECURITY CLASS. (of this report) Unclassified								
		16a. DECLASSIFICATION/DOWNGRADING SCHEDULE								
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.										
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) Accession For										
<table border="1"> <tr> <td>NTIS Final</td> <td>☒</td> </tr> <tr> <td>Pub. Rep.</td> <td>☐</td> </tr> <tr> <td>Unannounced</td> <td>☐</td> </tr> <tr> <td>Justification</td> <td>☐</td> </tr> </table>			NTIS Final	☒	Pub. Rep.	☐	Unannounced	☐	Justification	☐
NTIS Final	☒									
Pub. Rep.	☐									
Unannounced	☐									
Justification	☐									
18. SUPPLEMENTARY NOTES										
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) MOD6DF - STM										
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) This thesis discusses the conversion of the MOD6DF computer program for use on the IBM-360 computer at the Naval Postgraduate School. The functioning program was modified to investigate the impact miss distance for the Supersonic Tactical Missile. When the initial y-displacement error exceeded 1800 feet, the missile did not acquire the target. All errors smaller than this resulted in miss distances within 0.5 feet of the target. The midcourse guidance reference altitude was changed to reflect a sea-skimming missile. This										

251452

simulation ran and impact was recorded. An attempt at adding random noise to the homing seeker was tried, but revealed that more information is required on this topic. The MOD6DF computer program was successfully converted and altered to run using the simplified ramjet model.



Approved for public release; distribution unlimited

Guidance and Control of Tactical Missiles

by

Thomas Alan Grote
Lieutenant, United States Navy
B.S.A.E., United States Naval Academy, 1974

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from

NAVAL POSTGRADUATE SCHOOL
December 1979

Author

Thomas Alan Grote

Approved by:

Daniel J. Collins

(Thesis Advisor)

W. F. Platte

Chairman, Department of Aeronautical Engineering

William M. Tolles

Dean of Science and Engineering

ABSTRACT

This thesis discusses the conversion of the MOD6DF computer program for use on the IEM-360 computer at the Naval Postgraduate School. The functioning program was modified to investigate the impact miss distance for the Supersonic Tactical Missile. When the initial y-displacement error exceeded 1800 feet, the missile did not acquire the target. All errors smaller than this resulted in miss distances within 0.5 feet of the target. The midcourse guidance reference altitude was changed to reflect a sea-skimming missile. This simulation ran and impact was recorded. An attempt at adding random noise to the homing seeker was tried, but revealed that more information is required on this topic. The MOD6DF computer program was successfully converted and altered to run using the simplified ramjet model.

TABLE OF CONTENTS

LIST OF TABLES	6
LIST OF FIGURES	7
TABLE OF SYMBOLS AND ABBREVIATIONS	8
ACKNOWLEDGEMENT	12
I. INTRODUCTION	13
II. MISSION REQUIREMENTS	14
A. SEPARATION	15
B. MIDCOURSE GUIDANCE	18
C. TERMINAL DIVE	18
D. TERMINAL GUIDANCE	19
III. COMPUTER SIMULATIONS	20
A. EXECUTIVE PROGRAMS	21
B. OPERATIONAL SUBROUTINES	23
C. MODULES	24
D. INPUT DATA DECK	29
E. AUXILIARY DECK	34
F. SEQUENCING	36
IV. COMPUTER PROGRAM CHECKOUT	40
A. MIDCOURSE GUIDANCE FLIGHT	41
B. TERMINAL GUIDANCE FLIGHT	47
V. TERMINAL FLIGHT DISTURBANCES	56
VI. CONCLUSION	64
COMPUTER OUTPUT	65
COMPUTER PROGRAM	68
LIST OF REFERENCES	171
INITIAL DISTRIBUTION LIST	173

LIST OF TABLES

I. TXBA(073) Comparison (MIDCOURSE) -----	48
II. VXTP(286) Comparison (MIDCOURSE) -----	48
III. YTP(298) Comparison (MIDCOURSE) -----	49
IV. TXBA(073) Comparison (TERMINAL) -----	49
V. VXTP(286) Comparison (TERMINAL) -----	50
VI. YTP(298) Comparison (TERMINAL) -----	50
VII. Initial Conditions -----	57
VIII. Impact Results -----	58

LIST OF FIGURES

1. Control Phases/Mission Phases Relationship -----	16
2. Flight Mode Sequencing -----	17
3. Typical Type 2 Card -----	32
4. Typical Type 3 Card -----	32
5. Module Diagram -----	38
6. Processing Sequence -----	39
7. Altitude vs. Time (MIDCOURSE) -----	42
8. Thrust vs. Time (MIDCOURSE) -----	44
9. VXTP vs. Time (MIDCOURSE) -----	45
10. YTP vs. Time (MIDCOURSE) -----	46
11. Altitude vs. Time (TERMINAL) -----	51
12. Thrust vs. Time (TERMINAL) -----	53
13. VXTP vs. Time (TERMINAL) -----	54
14. YTP vs. Time (TERMINAL) -----	55
15. Altitude vs. Time (TERMINAL-ZTP) -----	60
16. Impact Results -----	61
17. Altitude vs. Time (SEA-SKIMMER) -----	62

TABLE OF SYMBOLS AND ABBREVIATIONS

The following is a list of the abbreviations and some of the more common fortran symbols used in the MOD6DF computer program. Each symbol is defined in two ways. The primary source of identification is by its COMMON(3415) location and the second is by its fortran symbol. The fortran symbol is not always a good identifier since it will change from subroutine to subroutine (i.e. W and WE both are used for missile weight).

A. ABBREVIATIONS

<u>Symbol</u>	<u>Definition</u>
TP	tangent plane axes
BA	body axes
SA	stability axes

B. FORTRAN SYMBOLS

<u>Symbol</u>	<u>Definition</u>
TXBA(073)	missile thrust in the x-direction in BA (lb)
TYBA(074)	missile thrust in the y-direction in BA (lb)
TZBA(075)	missile thrust in the z-direction in BA (lb)
W(086)	missile weight (lb)
S(110)	missile reference area (ft ²)
CA(111)	drag coefficient
CY(113)	side-force coefficient
CZ(115)	normal force coefficient
CBAR(116)	mean aerodynamic chord (ft)
CMQ(118)	damping in pitch coefficient
CNR(119)	damping in yaw coefficient

<u>Symbol</u>	<u>Definition</u>
CLP(120)	damping in roll coefficient
CM(121)	pitching moment coefficient
CN(122)	yawing moment coefficient
CL(123)	rolling moment coefficient
CGI(136)	center of gravity (ft)
A(201)	moment-of-inertia about missile roll axis (x-body axis) (slug-ft ²)
B(202)	moment-of-inertia about missile pitch axis (y-body axis) (slug-ft ²)
CC(203)	moment-of-inertia about missile yaw axis (z-body axis) (slug-ft ²)
TSA(208)	angle between SA and BA (rad)
P(212)	missile angular velocity about x-BA (rad/s)
Q(216)	missile angular velocity about y-BA (rad/s)
R(220)	missile angular velocity about z-BA (rad/s)
AG(282)	unit conversion lb-slug
VXTP(286)	missile velocity in x-TP (ft/s)
XTP(290)	missile displacement in x-TP (ft)
VYTP(294)	missile velocity in y-TP (ft/s)
YTP(298)	missile displacement in y-TP (ft)
VZTP(302)	missile velocity in z-TP (ft/s)
ZTP(306)	missile displacement in z-TP (ft)
GZRO(404)	constant set to zero for flat earth gravitational field and set to one for a spherical gravitational field
ER(405)	angular velocity of earth (rad/s)
ALPO(406)	angle between north and x-TP (rad)

<u>Symbol</u>	<u>Definition</u>
OLAMO(407)	latitude origin of tangent plane
HO(414)	distance tangent plane is from earth (ft)
GO(415)	gravitational acceleration (ft/s ²)
HREF(501)	reference altitude for midcourse guidance (ft)
RE(503)	earth's radius (ft)
H(507)	altitude normal to earth (ft)
AMACH(520)	missile Mach number
THET(521)	missile pitch angle, TP (rad)
PSI(522)	missile yaw angle, TP (rad)
PHI(523)	missile roll angle, TP (rad)
GAMMAV(527)	vertical flight path angle, TP (rad)
VEL(528)	magnitude of missile velocity (ft/s)
VAT(529)	missile velocity (ft/s)
TF(550)	program termination time (s)
VAH(561)	computed speed of sound (ft/s)
ALAT(576)	latitude of target position (deg)
AZ(578)	azimuth of target position (deg)
DYNP(581)	dynamic pressure (psi)
FLGRJ(606)	constant set to zero indicates run will use ENGINE subroutine and when set to one run will use RAMJET subroutine (simplified ramjet model)
T(932)	actual time (s)
T1(933)	boost engine ignition (s)
T2(934)	commence acceleration command mode (s)
T3(935)	boost engine burn-out/port cover blow-in (s)
T4(936)	start ramjet engine (cruise) (s)
T5(937)	commence heading and altitude guidance control (s)

<u>Symbol</u>	<u>Definition</u>
T6(938)	commence terminal dive (s)
T7(939)	commence terminal guidance - search (s)
T8(940)	commence terminal guidance - track (s)
DMAX(1740)	maximum fin deflection (rad)
CPP(2669)	time between printouts (s)
ROLLO(2901)	initial roll angle (rad)
PITCHO(2902)	initial pitch angle (rad)
YAWO(2903)	initial yaw angle (rad)
STEP(2905)	determines executive program flow after staging
DOC(2909)	defines number of times COMMON will be printed
HMIN(2911)	minimum integration step size
HMAX(2912)	maximum integration step size
DER(1)(2913)	integration step size (s)

ACKNOWLEDGEMENT

This author is grateful to Professor D. L. Collins, NAVAL POSTGRADUATE SCHOOL, for his helpful discussions and suggestions. I also acknowledge NWC CHINA LAKE for their cooperation in obtaining the MOD6DF computer program.

I. INTRODUCTION

Research was undertaken to convert the MOD6DF computer program received from NWC China Lake for use on the IBM-360 computer at the Naval Postgraduate School. Once a functioning program was obtained, initial displacement error versus impact miss distance was investigated for the supersonic tactical missile. Additionally, the midcourse guidance reference altitude was changed to reflect a sea-skimming scenario and this was examined for its effect on the terminal guidance problem. This report not only discusses the aforementioned topics, but also describes the missile mission requirements and the MOD6DF computer program.

II. MISSION REQUIREMENTS

The Supersonic Tactical Missile (STM) mission is divided into six phases (Ref. 1).

- * initial conditions
- * separation
- * boost
- * transition
- * cruise
- * terminal

These divisions are based on the missile aerodynamics. The initial condition phase establishes the starting conditions for each launch. This is done while the missile is still attached to the launch platform. The separation phase starts when the missile is launched. The missile falls for approximately five seconds until the boost engine ignites. This initiates the boost phase which continues until the missile achieves Mach two. As the missile passes through Mach one, plume effects are encountered which the aerodynamics account for. At the end of the boost phase, the port covers blow in. This initiates the transition phase. This phase is very short and allows the debris to be ejected from inlet ports. The cruise phase commences when the ramjet engine ignites. This engine propels the missile until target impact. The terminal phase of the flight begins when the missile is commanded to dive from the cruise altitude. This phase concludes when the flight is terminated at target impact.

The six STM mission phases require four phases of control. These control phases are:

- * separation
- * midcourse guidance
- * terminal dive
- * terminal guidance

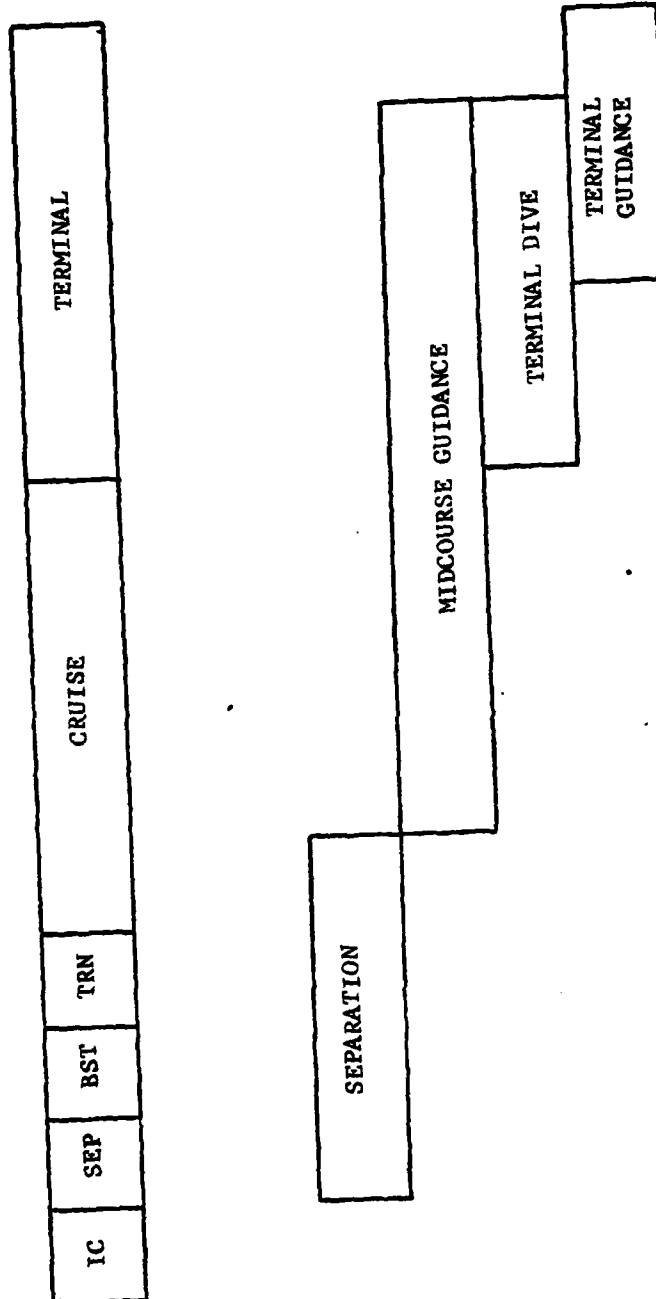
The four Guidance, Navigation, and Control (GN&C) system control phases are directly related to the mission phases. Figure 1 displays this relationship (Ref. 2). To obtain a successful flight, various types of control and guidance functions are required. Figure 2 (Ref. 3) illustrates the guidance control mode sequencing in relation to the mission phases and also indicates the critical missile switching times. The control phases are discussed below, along with the appropriate guidance modes.

A. SEPARATION

The separation phase commences upon launch. The missile is ejected downward from the launch platform. Additionally, the pitch attitude of the missile is commanded down. Since this portion of the flight is unguided, the ejection force and gravity are the only forces acting on the missile. At launch, the missile is required to be in the attitude command mode.

Five seconds into the flight, the missile pitch attitude is commanded up and the boost engine is ignited. The booster continues until the missile attains Mach two, at which time control is shifted from attitude to acceleration control mode. The acceleration control then requires the missile to maintain the normal and lateral accelerations at zero.

The last event to occur in the separation phase is inlet port cover blow-in. The time delay that allows the port covers to clear is a function of altitude.

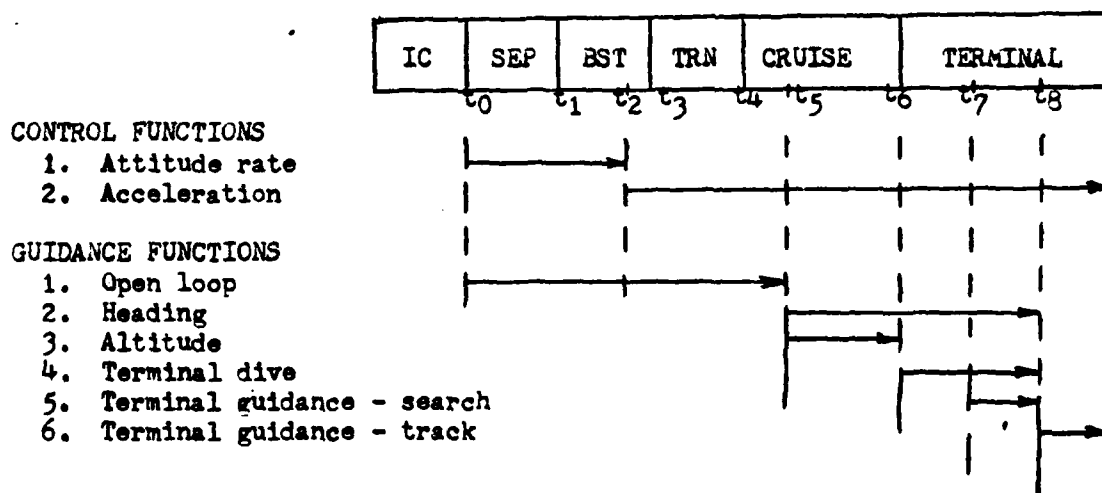


IC - INITIAL CONDITIONS
 SEP- SEPARATION
 BST- BOOST
 TRN- TRANSITION

CONTROL PHASES/MISSION PHASES
 RELATIONSHIP

FIGURE 1

- t₀ - LAUNCH: START SIMULATION
- t₁ - BOOSTER IGNITION
- t₂ - CONTROL SYSTEM MODE CHANGE FROM ATTITUDE TO ACCELERATION CONDITION AT M=2.0
- t₃ - PORT COVER BLOW-IN, BOOSTER BURNOUT
- t₄ - RAMJET IGNITION, t₃ + .3 sec
- t₅ - ENGAGE GUIDANCE MODES
- t₆ - DIVE COMMAND
- t₇ - SEEKER SEARCH MODE IS ACTIVATED
- t₈ - TERMINAL TRACKING



FLIGHT MODE SEQUENCING

FIGURE 2

B. MIDCOURSE GUIDANCE

Midcourse guidance begins upon completion of the separation phase. The moment the port covers are clear, the phases shift. During this phase two things are required. The GN&C must guide the missile to the established cruise altitude and then guide the missile (in the horizontal plane) to the predetermined target location.

To accomplish these requirements the GN&C system employs both altitude and heading control. The altitude control acts upon the pitch axis to drive the missile to the cruise altitude and then to maintain that altitude until the terminal dive phase commences.

The actual guidance work is performed by the yaw axis. The GN&C system uses the heading control to steer the missile to the target position. To generate the steering commands a guidance law is necessary. The guidance law should minimize the cross-track error and the final lateral acceleration. A minimum cross-track error allows for minimum flight time and minimum displacement error when target search is initiated. Minimizing the final lateral acceleration ensures that the seeker will continuously be pointing toward the target area. To insure accuracy, the guidance law must be able to perform these functions when subjected to disturbances such as wind and thrust misalignment.

C. TERMINAL DIVE

The terminal dive phase commences when the missile is commanded to dive. The actual time that this command is given is a function of the predetermined target coordinates. Upon diving, the missile must accelerate downward along a 60-degree (from horizontal) dive angle to the target position. Once the missile has steadied up in the proper dive aspect, terminal guidance commences.

D. TERMINAL GUIDANCE

During the terminal guidance phase the seeker is required to search, acquire, and track the target. The GN&C system then uses these tracking signals to direct the missile to the target. The seeker uses a preprogrammed search pattern to locate the target. When the target has been acquired, the seeker then commences tracking it and also notifies the GN&C system that the target is being tracked.

Target acquisition takes place when the target falls within the instantaneous four-degree beamwidth pattern as it traverses the scan pattern on the earth's surface. The computer program uses this assumption since the exact missile target acquisition mechanism has not yet been defined.

Upon acquisition, the GN&C system closes the seeker tracking loop. The radiometer error signal is defined as the difference between the look vector and the line-of-sight. This signal is used to reposition the antenna gimbals so the look vector and the line-of-sight coincide. Under the closed loop operation the radiometer output is proportional to the line-of-sight rate and this is the signal used to guide the missile to target impact.

III. COMPUTER SIMULATION

The modularized six-degree-of-freedom (MOD6DF) computer program was developed by the Litton Systems, Inc., Guidance and Control Systems Division (Ref. 4) to be used in analyzing missile guidance and control. The program uses a building block approach, where each module corresponds either to a missile subsystem or an environmental system. In its original form, the program is used primarily for terminal guidance of air-to-surface missiles. The Naval Weapons Center, China Lake modified the original program to specifically apply to the Supersonic Tactical Missile (STM). The modification also allows the user the capability of simulating any portion of the missile flight trajectory.

The MOD6DF computer program consists of four main decks and one auxiliary deck. The main decks include the executive programs, operational subroutines, modules, and input data deck. All the subprograms utilized in the integration and the sequencing of the modules are contained in the executive programs. The operational subroutines are used by the user to control the program while it is running. The physical system and the environmental subroutines are included in the modules. The input data deck contains all the information the executive programs need to execute the desired run. Lastly, the auxiliary deck consists of all those subprograms (subroutines and functions) that are required by the modules.

Once the user starts making simulations, he must concern himself with program sequencing. Proper sequencing is required to ensure a valid run is achieved. Since it is of such importance, sequencing will be discussed as the final part of this section.

A. EXECUTIVE PROGRAMS

The executive programs are the main core of the MOD6DF program. These subroutines have various functions in setting up, sequencing, integrating, and resetting the program. Since these functions are required by any type of analysis done, the user should never have to change any of these subroutines. Reference 5 should be consulted if the user desires more information about the content of any of the subroutines described herein.

1. Zero

Subroutine ZERO is used to set all the COMMON(3415) locations to the value zero. This is done to ensure that no erroneous information is used in the simulation.

2. Oinpt1

Subroutine OINPT1 is the basic input routine for the MOD6DF program. The normal input is from punched cards. However, inputs may also be read from tapes. The basic input cards will be discussed in the section covering the input data deck.

3. Auxi

Subroutine AUXI is used to call the initialization modules for each run. Additionally, it sets up the list of state variables which are used in AMRK.

4. Auxsub

Subroutine AUXSUB is used to call the dynamic modules. In calling the dynamic modules, AUXSUB sets and resets the lists needed for AMRK and the COMMON(3415) storage cells with the most recent values of the state variables and their derivatives.

5. Amrk

Subroutine AMRK is the integration subroutine. It uses a point-wise first order Runge-Kutta method. All the state variables to be integrated must be listed in COMMON(3415) and also must appear in a processing list.

6. Reset

Subroutine RESET is used to reinitialize up to fifty input parameters. This is done prior to the start of any repeated runs. To indicate which parameters are to be reset, the number one is punched in columns 46-60 on the respective type three card.

7. Return Group

The return group is a collection of all the unused modules. Since all the modules and their initialization modules are called by AUXI and AUXSUB, the unused ones must still remain in the deck. These are required to ensure proper linking when the computer attempts to link all the subroutines. All the subroutines in this grouping contain three cards. The three cards are the subroutine title, and a return and an end card.

8. Sub11, Sub12, Sub13

These subroutines are used to call the operational subroutines that are required. They call the routines in the order prescribed by the input data deck. The number at the end of each subroutine title indicates which operational subroutines it can call. For example, SUB11 can call STGE1.

B. OPERATIONAL SUBROUTINES

The operational subroutines provide the user with control of the program while it is running. The order in which the operational subroutines are called is specified by the input data deck. Since these routines assist the user in controlling the simulation, they can be reprogrammed. However, it is advised that they not be changed until the user has become quite familiar with the overall operation and sequencing of the MOD6DF program. For more in depth knowledge of the operational subroutines the user should consult Ref. 6, and the computer listing, which is contained at the end of this report.

1. Inpt1, Inpt2, Inpt3

These subroutines are available for new inputs during the simulated flight. The only one presently used in the program is INPT1. It is utilized to input a namelist file which is used by ENGINE and is described there.

2. Oupt1, Oupt2, Oupt3

These subroutines allow for the print-out of up to fifty different variables during the flight simulation. OUPT1 is not utilized in the deck. OUPT2 is used when the desired output is to be put on tape. OUPT3 is the basic output routine. It prints the desired output on regular computer paper.

3. Stge1, Stge2, Stge3

These subroutines allow for proper staging, run termination, etc.. Presently STGE1 is not being used. STGE2 is being used as the staging initialization subroutine. STGE3 is the primary subroutine of this group. It stages when impact with the earth is made, when the final time, TF(550), is reached, or when LCONV(2672) is set equal to two. All the tolerances for staging are listed in STGE3.

4. Cntr1, Cntr2, Cntr3

These subroutines allow the external dynamic control inputs to the modules. In the MOD6DF program these are not used.

5. Rndm1, Rndm2, Rndm3

These subroutines allow random noise to be added to the state variables generated in the modules. RNDM1 is not used in the program. RNDM2 is used as the initialization subroutine, while RNDM3 provides continuous noise values. These subroutines produce correlated noise values for as many modules as required. The noise values remain fixed during each individual integration cycle.

6. Auxa1, Auxa2, Auxa3 Auxb1, Auxb2, Auxb3 Auxc1, Auxc2, Auxc3

These subroutines are auxiliary routines that allow for external input, output, control, etc. of the modules. At the present time, none are utilized in the MOD6DF program.

C. MODULES

The modules are of prime importance to the user since they represent the 'model' of the dynamic system. In general, the model is described by ordinary non-linear time-varying differential equations with both random and deterministic forcing functions. The user must first reduce these equations to an equivalent system of first order equations, which can then be described by each module. Generally speaking, the physical system is so complex that this would be impossible to do. However, due to the modularity of the MOD6DF program, the user can think of each module as a completely independent system described by the equations within that module.

There are thirty-six possible modules divided into five functional

categories. Each group is identified by a letter which pertains to that groups function; A (airframe), C (computers), D (dynamics), G (geophysical), S (sensors). A complete printing of each module is contained in the computer program listing.

1. Airframe

a. Subroutine A1

This is the aerodynamic forces and moments modules. It calculates all the necessary forces and moments in body axes. These values are then used in the computation of the dynamics.

b. Subroutine A2

This is the missile aerodynamic coefficient module. It calculates the required coefficients using the information stored in the BLOCK DATA. Using the timing inputs, this routine computes the coefficients for the different effects. Some of the effects accounted for are; plume, separation, and control surfaces effectiveness. With this done the total coefficients are determined.

c. Subroutine A3

This is the missile propulsion module. The timing inputs are used to determine whether the missile is in free fall, boost, transition, or cruise phase. With this determined the correct engine subroutine (BOOST, RAMJET, ENGINE) can be called. Three variables are calculated using the body axes as the frame of reference. They are the missile thrust in all coordinate directions, the principal moments of inertia, and the missile weight.

d. Subroutine A4

This is the fin actuator module. The four control surfaces commands are calculated as either ideal actuators or as second-order

ones. In addition to control surfaces commands, the rate of change of yaw, and roll are computed.

e. Subroutine A5

This module is part of the return group.

2. Computers

a. Subroutine C1

This is the autopilot module for the STV-G. It uses the cruise engine ignition time to divide the routine into boost/separation and cruise phases. These two phases use different algorithms to calculate the turning moments in pitch/yaw and roll.

b. Subroutine C2

This is the guidance command module. Using the timing inputs, this routine is divided into separation/boost, dive/climb, cruise, terminal dive, and terminal homing sections. Each section uses slightly different algorithms to calculate the guidance commands to maintain the proper flight profile.

c. Subroutines C3 - C10

These modules are part of the return group.

3. Dynamics

a. Subroutine D1

This is the translational dynamics module. It computes the total acceleration in body axes and then converts them to the tangent plane reference. Then, accounting for aerodynamics, thrust, gravity, and coriolis, the velocity and acceleration are calculated.

b. Subroutine D2

This is the rotational dynamics module. With the principal axes as a reference, this subroutine computes the body angular rates and the attitude direction cosines.

c. Subroutines D3 - D5

These modules are part of the return group.

4. Geophysical

a. Subroutine G1

This is the gravitational and coriolis acceleration module.

It calculates the gravitational acceleration using one of two fields.

The user specifies the field to be used by an input card. To use a flat-earth gravitational field, GZRO(404) must equal 0.0, and to use a spherical gravitational field, GZRO(404) must equal 1.0.

b. Subroutine G2

This module is part of the return group.

c. Subroutine G3

This is the air data module. It computes the velocity, in all three coordinate directions, with respect to the air mass. These values are then resolved into body and stability axes. This module also computes all the properties of air by calling subroutine AIR. These values are stored in their COMMON(3415) locations for use in the other modules.

d. Subroutine G4

This module is part of the return group.

e. Subroutine G5

This is the coordinate conversion module. It takes the missile position, does a coordinate conversion and then it determines the position, velocity, and acceleration in the ECI system.

f. Subroutine G6

This module is part of the return group.

5. Sensors

a. Subroutine S1

This is the homing seeker module. It simulates the missile seeker and computes the seeker dynamics and Euler angle rates. Several flags are used to control the seeker sequencing:

- (1) FLAGS(335). FLAGS signals the start of the seeker search.
- (2) FLGT(317). FLGT signals when the seeker is locked-on.
- (3) FLGTS(347). FLGTS signals the end of search.
- (4) FLGD(336). FLGD signals when the seeker has detected the target.
- (5) FLAGLT. FLAGLT signals when the target is outside the seeker field of vision.

b. Subroutine S2

This is the radiometer module. It takes the target position and the ATIGS target position and converts them from body axes to the seeker axes. Using target position, the module then calculates the azimuth and elevation error signals.

c. Subroutines S3, S4

These modules are part of the return group.

d. Subroutine S5

This is the accelerometers and gyros modules. The user has the option of using ideal or digital accelerometers. To specify the type of accelerometer, the user must include the appropriate data statement in the subroutine. If ideal accelerometers are desired, FLGA must equal 0.0 and for digital accelerometers, FLGA must equal 1.0.

e. Subroutines S6 - S10

These modules are part of the return group.

D. INPUT DATA DECK

The input data deck provides the user with the means of specifying which operational subroutines and modules are to be utilized for the desired run. It also allows the user to set the starting conditions. In general, only the constants and the state variables must be given initial values. All quantities in COMMON not given initial values will be set to zero by ZERO. In addition to the state variables, the upper and lower error bounds must be initialized. There are seven types of input cards, each indicating a certain function.

<u>Type</u>	<u>Function</u>
0	read/write tape
1	operational subroutine to be called
2	module to be called
3	numerical input
4	printed output
5	parameter square and sum
6	termination and random noise generator input

A separate card is required for each subroutine, module, input, and output quantity. A sample computer printout of the input data deck is contained in the computer output section.

1. Type 0 Card

Type 0 cards are used to indicate if the type 3 inputs are to be read from or written onto an auxiliary tape. These procedures can be used rather than reading the inputs from a deck of cards. A typical card is defined by punching a zero in column 2. The field that covers columns 5 - 20 is used by the user for any descriptive statements with which he wishes to identify the input. Column 21 - 25 contains the right-

justified integer number of the tape transport to be used. The number of the first record to be read is punched in column 31 - 45. The last field is column 46 - 60 which contains the number of records to be read. The last two fields may use either fixed or floating-point notation.

2. Type 1 Card

Type 1 cards are used to specify which operational subroutines are called during the flight simulation. This type card is identified by the number one in column 2. The second field is column 5 - 20 which contains any identifying information. This information is printed out when the data deck is read and allows the user to read exactly which subroutines were called. Column 21 - 25 contains the right-justified integer number which is the subroutine identifying number. The operational subroutine numbers are;

<u>Subroutine</u>	<u>Subroutine Number</u>
INPT1, INPT2, INPT3	2
OUPT1, OUPT2, OUPT3	3
STGE1, STGE2, STGE3	4
CNTR1, CNTR2, CNTR3	5
RNDM1, RNDM2, RNDM3	6
AUXA1, AUXA2, AUXA3	7
AUXB1, AUXB2, AUXB3	8
AUXC1, AUXC2, AUXC3	9

It should be noted that all or any of the subroutines listed under one number can be called by including only one card. The cards are placed in the data deck in the order in which they will be called. This order or sequencing will be explained further in section F, Sequencing. A typical type 2 card is identified by the number two in

column 2. In general, column 5 - 20 should contain the module title, but any pertinent information is allowed. The module number is punched in column 21 - 25 and it must be right-justified. A listing of the module numbers follows:

<u>Module</u>	<u>Module Number</u>	<u>Module</u>	<u>Module Number</u>
A1,A1I	2	D4,D4I	20
A2,A2I	3	D5,D5I	21
A3,A3I	4	G1,G1I	22
A4,A4I	5	G2,G2I	23
A5,A5I	6	G3,G3I	24
C1,C1I	7	G4,G4I	25
C2,C2I	8	G5,G5I	26
C3,C3I	9	G6,G6I	27
C4,C4I	10	S1,S1I	28
C5,C5I	11	S2,S2I	29
C6,C6I	12	S3,S3I	30
C7,C7I	13	S4,S4I	31
C8,C8I	14	S5,S5I	32
C9,C9I	15	S6,S6I	33
C10,C10I	16	S7,S7I	34
D1,D1I	17	S8,S8I	35
D2,D2I	18	S9,S9I	36
D3,D3I	19	S10,S10I	37

Note that either the module, the initialization module, or both may be called by including only one card in the deck. A sample of a typical type 2 card is shown in Figure 3 (Ref. 7).

TYPICAL TYPE 2 CARD

2	5	2021	25	31	45	46	60
3	WEIGHT		86	$\frac{1152.0}{1.152} - - \bar{E}+03$	$\frac{1.0}{1.0} - - - \bar{E}+00$		

TYPICAL TYPE 3 CARD

FIGURE 4

4. Type 3 Card

Type 3 cards are used to set any COMMON(3415) location to any value other than zero. In general, four items must be initialized. The state variable initial values and any constants used in the flight simulation are the most obvious. Additionally, there are some constants associated with the executive programs and operational subroutines and the state variable upper and lower bounds which must be initialized. As with all cards, column 2 defines the type card and it must contain a three. Column 5 - 20 holds the statement describing the input. The user should be specific here since it will save him having to remember every COMMON(3415) location. The only other means of input identification is by column 21 - 25. These columns contain the right-justified COMMON location of the input. Columns 31 - 45 hold the actual numerical value of the input. The last field, column 46 - 60, contains the reset flag. If the reset flag equals one, the COMMON(3415) location and the numerical value are placed in the reset list. This list may contain up to fifty different variables. This, in the case of multiple runs, allows the variables to be reset to its initial value prior to each run without additional input cards. The sample card in Figure 4 (Ref. 8) shows that either fixed or floating-point notation may be used to input the numerical value and the reset flag. These cards need not be inputted in any specific order, but for ease of checkout, it is advised to place them sequentially by their COMMON location.

5. Type 4 Card

The MOD6DF program can printout a maximum of fifty variables for each simulation. Type 4 cards are used to specify which variables are to be printed. Column 2 must contain the number four to indicate a type

4 card. When the results are printed out headings are included. These headings are designated in column 9 - 20. The exact alphanumeric title punched will be printed at the top of each page, this need not be the fortran symbol used within the program. The COMMON(3415) location of the output variable is contained in column 21 - 25 and must be right-justified.

6. Type 5 Card

Type 5 cards are used to indicate which variables are to be root-mean-squared. These cards are similar to type 1 and type 2 cards. Column 2 must contain the number five. Any pertinent information about the variable to be operated on is punched in column 5 - 20. The COMMON(3415) location of the variable must be right-justified in columns 21 - 25. The last field, column 31 - 45, indicates whether the root-mean-square operation will occur along the trajectory or at the end.

7. Type 6 Card

The type 6 card has two purposes. Its first function is trivial, but required. The number six is typed in column 2 and the rest is blank. This indicates to the computer program that there are no more input cards. Its second function involves random inputs. This card is used to indicate the number of random process (noise) generator cards that are to be read before the input process is terminated. Column 2 has the same information as before. Any pertinent information is contained in column 5 - 20. Columns 21 - 25 must be right-justified and they contain the number of random generator cards to be read.

E. AUXILIARY DECK

The auxiliary deck is a collection of subroutines and functions required by the modules. These routines are general in nature since

they can be called by several modules. They are used to calculate such things as the properties of air, engine performance, various ratios, and to locate values in the many data tables. A brief discussion of the most common subroutines is presented.

1. Boost

This subroutine calculates the thrust coefficient (CT) and the fuel flow rate (FF) during the boost phase. These values are returned to module A3 and used to calculate the missile thrust in the body axes and the missile weight.

2. Ramjet

This subroutine is used during the midcourse, cruise, phase of the missile flight. It uses a simplified ramjet model to calculate the thrust coefficient and fuel flow rate. This routine is not automatically called. The user must designate that he wishes to use it by inputting a type 3 card setting FLGRJ(606) equal to one. This then sets up the proper stepping in module A3.

3. Engine

This is the primary routine during the cruise phase. It is one of many routines within the NWC air-breathing propulsion package. This entire package is utilized to calculate the engine parameters from inlet to exhaust. Again, thrust coefficient and fuel flow rate are eventually computed and returned to module A3. The user does not need to supply any special input cards to use this subroutine. If no initial value is inputted for FLGRJ(606) it is automatically set to zero, which indicates this routine is to be used.

4. Air

All the properties associated with the air are calculated in

this subroutine. This includes the computation of the speed of sound and the dynamic pressure. Since the missile does vary in altitude, the routine takes this into account as well as the latitude. Once these values are calculated, they are returned to module G3.

5. Block Data

This routine contains data tables. These tables cover parameters from aerodynamic coefficients to thermal properties. The total data package covers specific bands within the missile operating envelope. Data is stored in matrices which includes one, two, and three dimensional ones. These data tables are readily available and there are routines designed to retrieve this information quickly.

6. Serch

This subroutine, along with several functions, is used to retrieve information from BLOCK DATA. If the present operating point of the missile is not within one of the bands of information, then the tables are interpolated. The functions THREDL, STDLI, STDLIA, and TAB do the interpolation of the tables. Since the tables are of various lengths, these functions are very general.

F. SEQUENCING

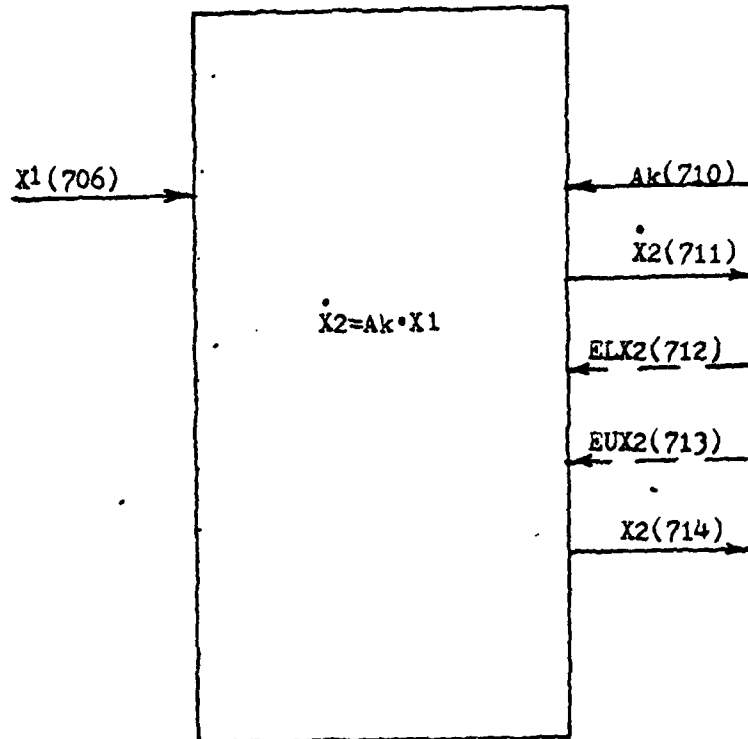
Sequencing is very important in the running of the MOD6DF program. Care must be taken to ensure that the modules are processed in the correct order at each step. This is essential to eliminate the use of obsolete values from the last cycle. An exception to this problem is the state variables. These are updated simultaneously by the integration algorithm. Any module is capable of using the most current value of these, no matter what the order of processing.

To help remedy this problem, module diagrams were devised. Module

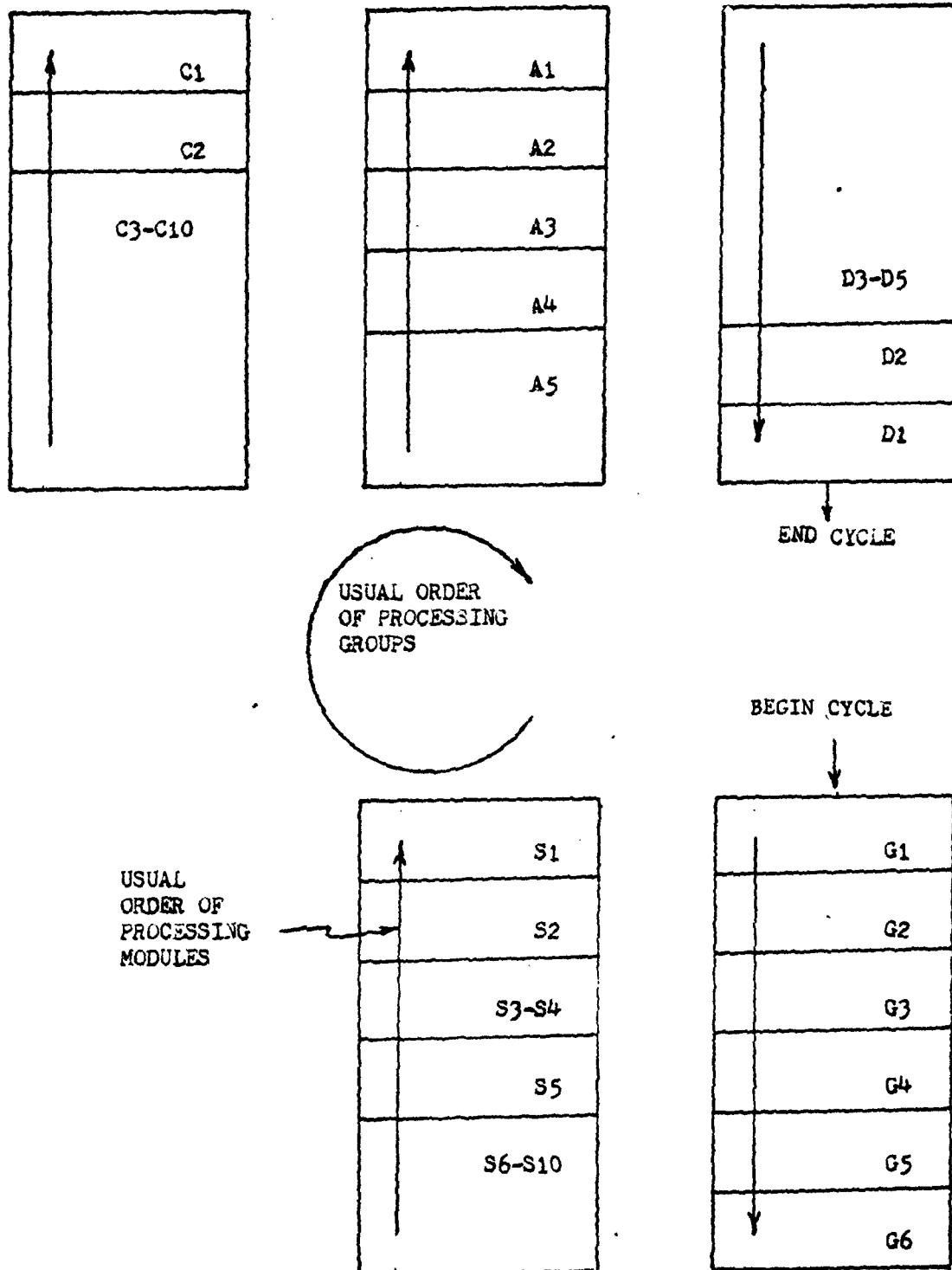
diagrams aid the user in maintaining the proper flow of variables into and out of the modules. To design a module diagram start with a box. This box will contain all the equations for a specific module. The standard procedure for showing inputs and outputs is to use arrows pointing in or out from either the left or the right side. The example in Figure 5 (Ref. 9) shows this technique. The arrows pointing in from the left indicate variable inputs from other modules. Arrows pointing out to the right indicate output going to either other modules or program output. The last set of arrows points in from the right. These show the constants brought in directly or indirectly from the initialization module. Since each arrow represents a variable, they must be defined. The usual means of labeling the arrows is to use the variable fortran symbol and in parenthesis its COMMON(3415) location.

Each variable usually has only one COMMON location associated with it. In the case of 'state' variables this is not true. State variables are defined by four consecutive COMMON locations. The first is for the derivative of the variable. The second and the third locations hold the lower and upper bounds, respectively, of the integration error. The last one contains the variable itself.

Once all the required module diagrams have been completed, they can be combined to get the overall processing order. The usual processing order, shown in Figure 6 (Ref. 10) is to start with the geophysical group, then proceed to sensors, computers, airframe, and finally dynamics. Within each group is a usual processing order and this too is shown in Figure 6. This process for determining the program sequence will eliminate the use of any obsolete values in the computer run.



MODULE DIAGRAM
FIGURE 5



PROCESSING SEQUENCE

FIGURE 6

IV. COMPUTER PROGRAM CHECKOUT

The basis for the research was the MOD6DF computer program from NWC China Lake. The total package received from them consisted of a listing of the program and an uninterpreted deck of cards. The initial step was to input the cards in small groups into the computer and then to examine the source listing. This listing revealed that the original deck was punched in BCD. This fact was easy to determine since several characters were changed (Ref. 11). The library routine NEWDEK was used to translate the cards from BCD to EBCDIC.

Once the translation was completed, an attempt was made to compile the new deck. This produced an output which contained many syntax errors. These errors were divided into two major groups. One effected the use of quotation marks in FORMAT and comment statements. The other one, the more difficult, effected the DATA statements in the BLOCK DATA subroutine.

The problem with the DATA statements was due primarily to the difference in the compilers used. The compiler at NWC was much newer and allowed for the use of more sophisticated inputs. The compiler at NPS only allows a data set to start with the first element. This required the rewriting of many data groups. To complicate these revisions, a limit of nineteen continuation cards is also imposed. These restrictions demanded not only the rewriting of many data sets, but also the formation of two new ones.

With the corrections finally completed, the computer would then compile the program. The next step was to link all the subroutines together. The first attempt was unsuccessful. Inadvertently, the subroutine INTR20 had been omitted from the original deck. Using the program listing, the contents of INTR20 were typed and included in the

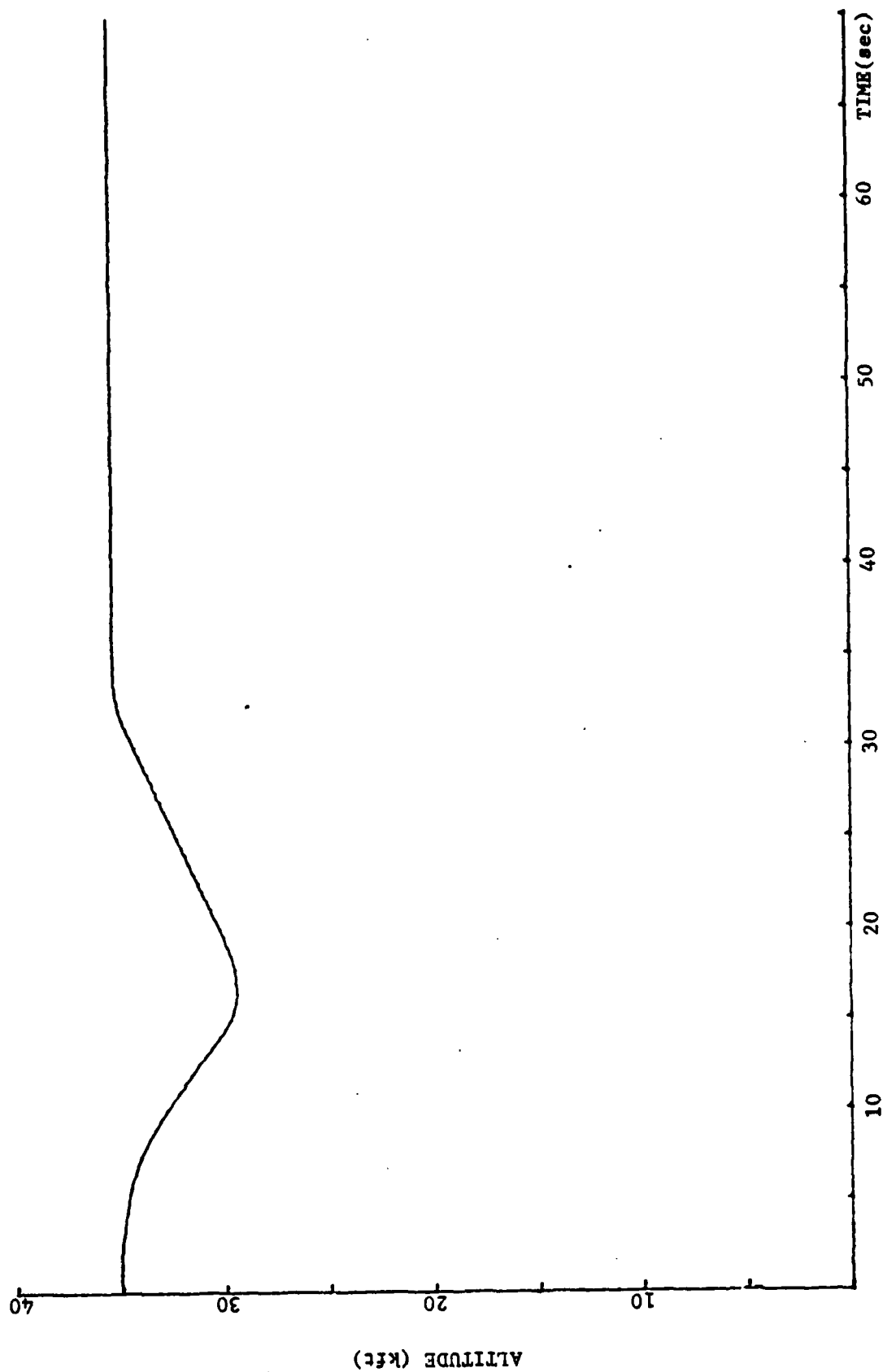
main deck. With this addition the program would now compile and link.

Now that the program would compile and link, attention was turned to getting a good simulation. To facilitate this process, two sample outputs were obtained from NWC. These outputs cover one missile flight which is broken down into a midcourse guidance and a terminal guidance simulation. Hereafter these will be referred to as midcourse baseline and terminal baseline, respectively. Using the initial conditions from the baseline models, it was hoped that the outputs could be duplicated.

A. MIDCOURSE GUIDANCE FLIGHT

The initial run, using the midcourse baseline inputs, revealed an overflow problem with the dynamic pressure (QD(508)). Using the traceback procedures outlined in the Users Manual from the W. R. Church Computer Center, the problem was confined to subroutine AIR. The problem turned out to be a translation error. The symbol $P\emptyset$ ($\emptyset$ - zero) had translated to $P\emptyset$ in one place and P0 in another. This error caused the program to use a value left in that memory location from a previous run to calculate the dynamic pressure. With this problem remedied, the program could progress a little farther. The next stumbling block appeared as a divide check. These errors were resolved by introducing patches that would bypass a statement that tried to divide by zero. Once bypassed, that quantity would be set equal to zero. This was the normal procedure of the computer, but it would stop the run after ten such errors. Having corrected all these errors, output was obtained which covered the desired seventy seconds of flight.

When the output was examined a major switching problem within subroutine A3 (missile propulsion module) was found. The midcourse baseline utilized a simplified ramjet model, but the output was not.



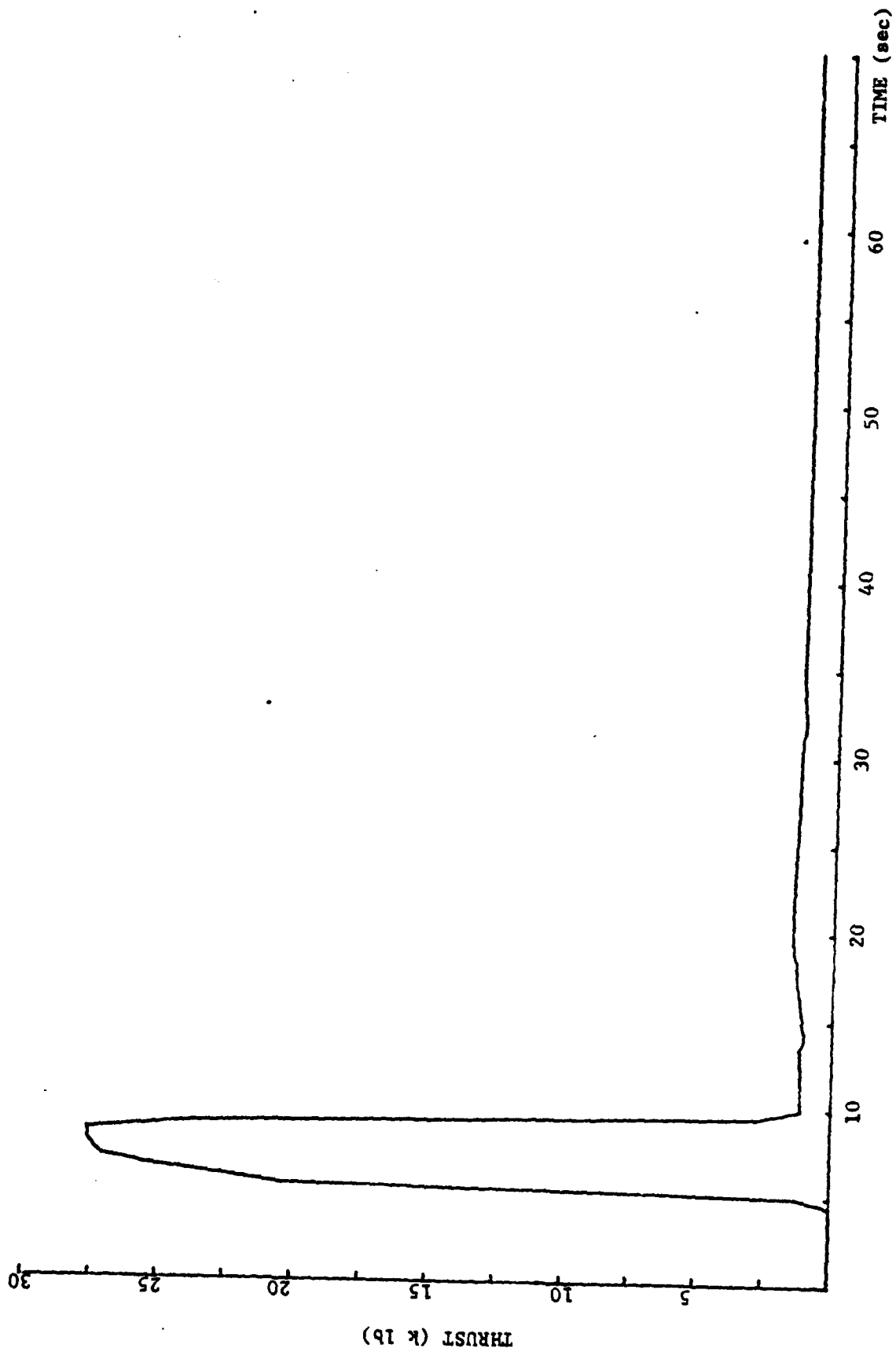
ALTITUDE VS. TIME (MIDCOURSE)

FIGURE 7

Rather than use RAMJET, the output disclosed that ENGINE had been used. The problem was in the logic statement, FLGRJ 0.0. The program was not making the desired switch to the simplified ramjet model. To remedy the problem, the data card FLGRJ(606) 1.0, was added to the input data deck. Another problem was discovered which occurred between 14.0 and 14.5 seconds. During that time period the missile angle of attack (ALPHA (330)) exceeded ten degrees. When ALPHA exceeds this angle the engine is turned off. With the engine off, no thrust is produced causing the forward velocity (VXTP(286)) to decrease. This limitation was removed from the program. After eliminating these problems a flight trajectory, Figure 7, was obtained which closely resembled the output of the mid-course baseline. A random sampling of the outputed variables were compared for exactness. The differences noted were due primarily to computer round-off error.

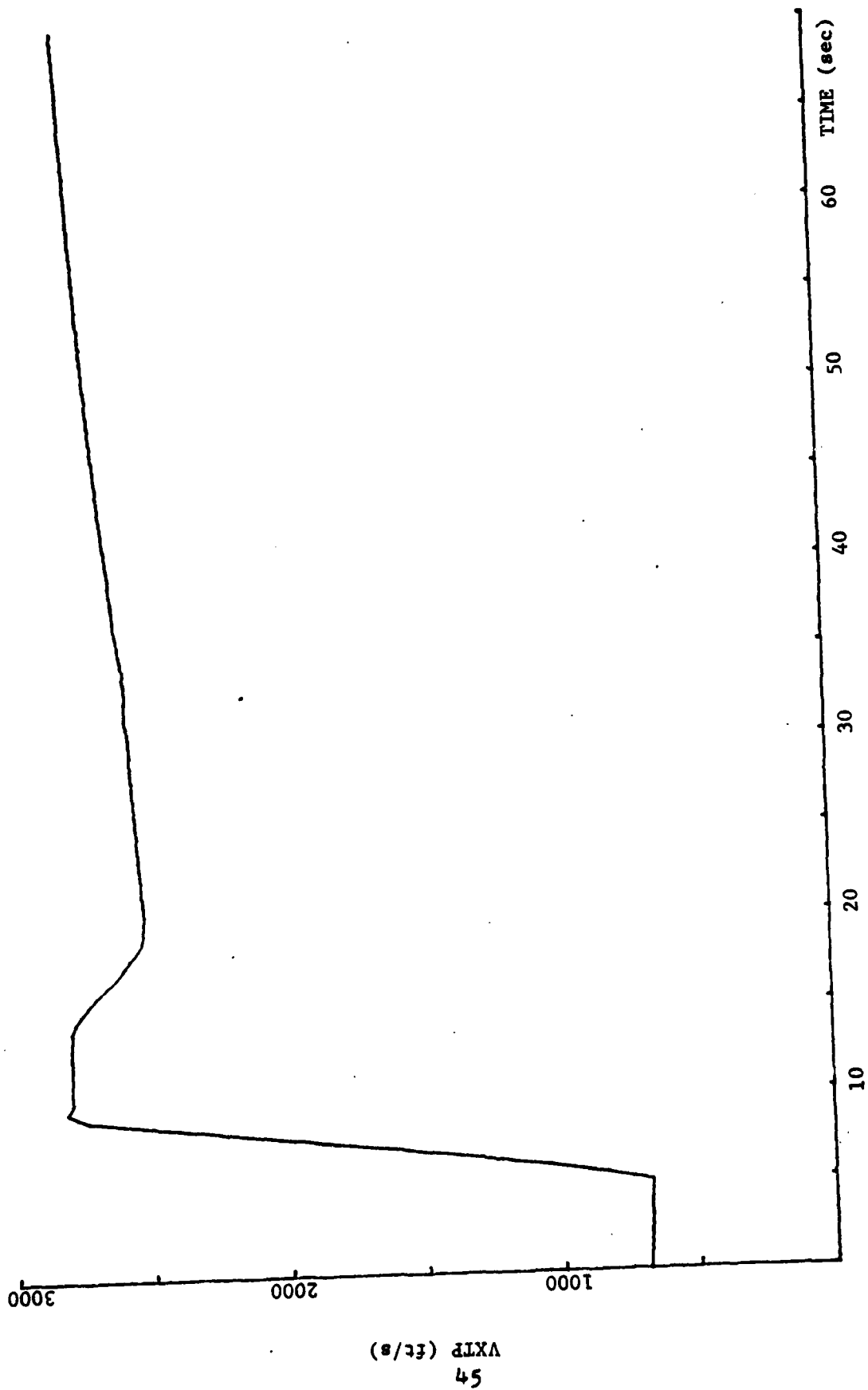
To further verify the accuracy of the two simulations, three parameter (TXBA(073), VXTP(286), and YTP(298)) were chosen as representative values for the runs. To obtain a feel for the run, these parameters are plotted in Figures 8, 9, 10. Additionally, random time samples are tabulated in Tables I, II, III.

The thrust (TXBA(073)) plot, Figure 8, can easily be divided into four mission phases. During the separation phase (0.0 - 4.5 seconds) the thrust is zero. This is expected since neither of the engines have ignited. This is followed by a rapid increase in the thrust. The maximum thrust, 27800 lbs, happens during the boost phase (4.5 - 9.5 seconds). At 9.5 seconds the boost motor stops and the transition phase occurs for the next 0.3 seconds. During this time the thrust decreases rapidly since neither engine is on. Once the engine port covers are



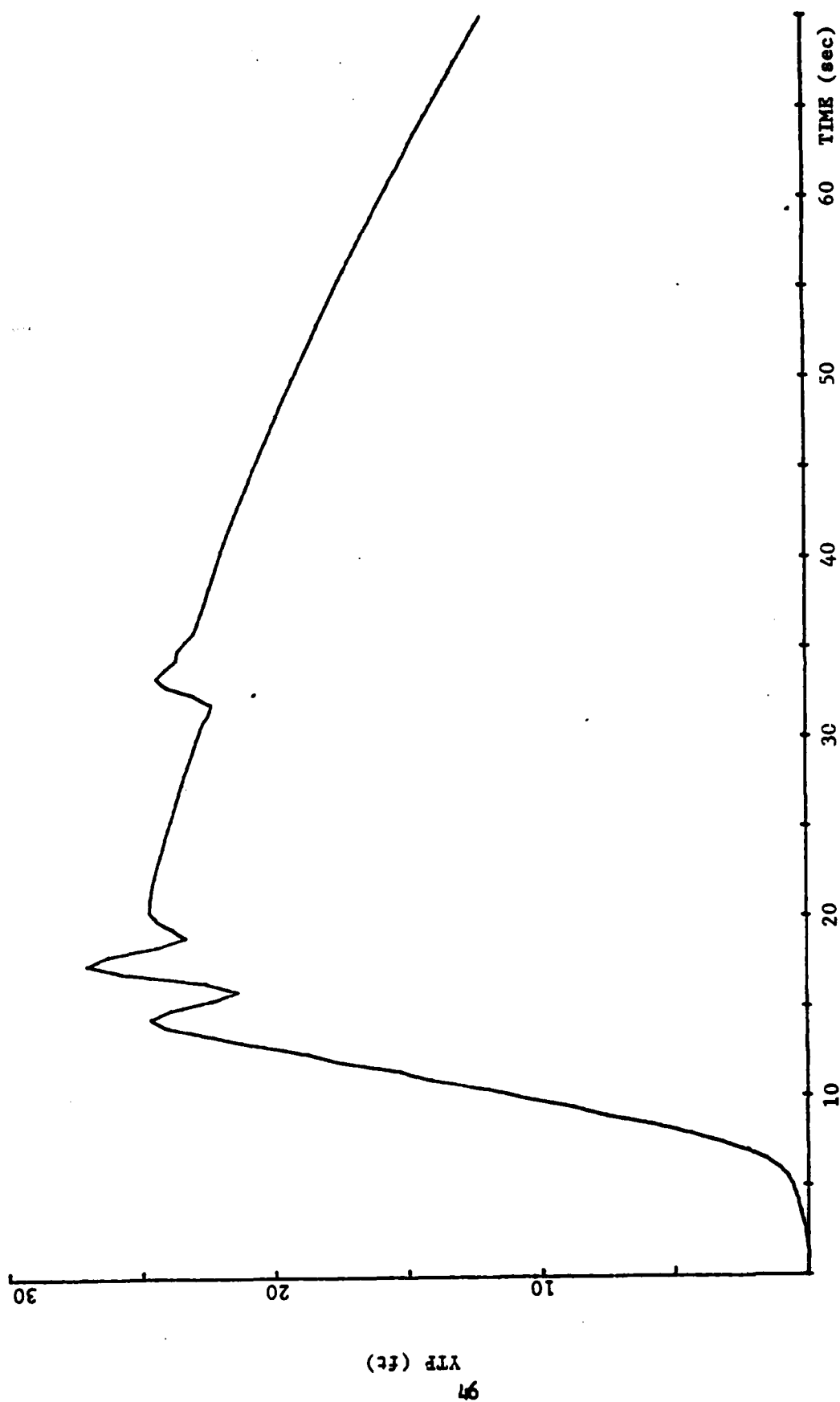
THRUST VS. TIME (MIDCOURSE)

FIGURE 8



VXTP VS. TIME (MIDCOURSE)

FIGURE 9



YTP VS. TIME (MIDCOURSE)

FIGURE 10

clear, the ramjet sustainer ignites. This action initiates the midcourse phase (9.8 - 70 seconds) and the thrust remains relatively constant at 1200 lbs.

The missile forward velocity (VOTP(286)) can be directly related to the thrust. During separation no thrust is produced, therefore the velocity is zero. As the missile is boosted to Mach two, the velocity increases and approaches its maximum value (2800 ft/s) just prior to boost engine shut-down. As the sustainer engine port covers clear, the velocity decreases slightly. Once the ramjet engine ignites the velocity profile is displayed in Figure 9 and it closely resembles that for the midcourse baseline.

The last parameter, y-displacement (YTP(298)), was included to demonstrate the accuracy of the guidance system. Figure 10 shows that prior to attaining the cruise altitude, the missile wanders off track. Once the guidance system is activated, it makes the necessary corrections to return the missile to the planned flight path. Table III shows that the y-displacement corresponds to the baseline case and at the conclusion of the run the off-track error is down to 12.03 feet.

All the information obtained from the new run was checked against the midcourse baseline. The results showed that the two simulations are within acceptable limits. With this milestone completed, the program checkout could proceed to the terminal guidance flight.

B. TERMINAL GUIDANCE FLIGHT

To checkout the terminal guidance portion of the MOD6DF program, the initial conditions from the terminal baseline were used. This required changing about a dozen input cards in the midcourse input data deck. The first run attempted turned out successful. This was due to the

TIME	0.0	7.5	10.0	20.0	25.5	30.0	40.0	50.0	60.0	70.0
BASELINE	0.0	27395	1140.6	1403.0	1377.4	1281.0	1211.3	1178.0	1148.1	1124.8
NEW	0.0	27391	1138.9	1514.3	1418.4	1314.3	1249.9	1207.0	1175.5	1149.7

TXBA (073) COMPARISON (MIDCOURSE)

TABLE I

TIME	0.0	7.5	10.0	20.0	25.0	30.0	40.0	50.0	60.0	70.0
BASELINE	681.99	1858.3	2790.2	2606.7	2604.9	2613.4	2652.4	2705.0	2743.8	2771.5
NEW	681.99	1851.7	2786.3	2506.4	2525.9	2544.5	2605.0	2665.5	2710.8	2743.9

VXTP (286) COMPARISON (MIDCOURSE)

TABLE II

TIME	0.0	7.5	10.0	20.0	25.0	30.0	40.0	50.0	60.0	70.0
BASELINE	-5E-5	3.46	11.37	27.21	25.92	24.34	20.45	16.93	13.24	9.39
NEW	-5E-5	3.19	10.68	24.43	23.93	22.82	21.96	19.14	15.81	12.03

YTP (298) COMPARISON (MIDCOURSE)

TABLE III

TIME	0	5.0	10.0	12.6	15.0	17.6	20.0	22.6	25.0	TF
BASELINE	1141.9	1131.0	1077.8	1170.9	1309.4	1501.1	1730.0	2245.3	2565.8	2691.3
NEW	1141.6	1134.8	1099.4	1212.8	1359.3	1561.7	1800.0	2305.9	2584.3	3273.6

TXBA (073) COMPARISON (TERMINAL)

TABLE IV

TIME	0.0	5.0	10.0	12.6	15.0	17.6	20.0	22.6	25.0	TF
BASELINE	2750.0	2764.2	2654.6	2471.4	2225.7	1854.5	1391.4	1376.9	1413.1	1425.0
NEW	2750.0	2761.5	2631.5	2438.7	2192.1	1828.2	1389.3	1366.7	1406.1	1465.2

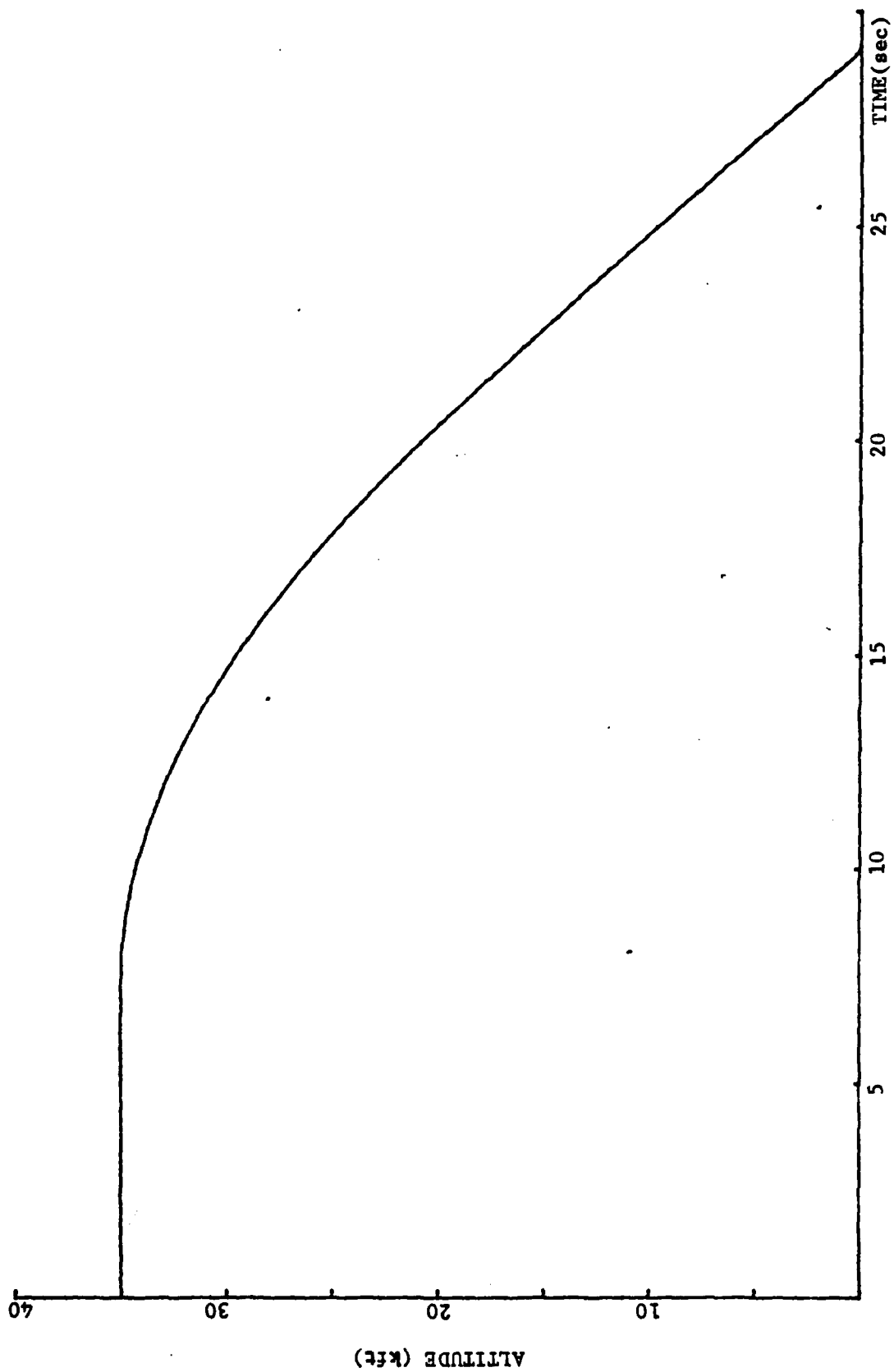
VXTP (286) COMPARISON (TERMINAL)

TABLE V

TIME	0.0	5.0	10.0	12.6	15.0	17.6	20.0	22.6	25.0	TF
BASELINE	0.0	0.0	$\sim 4E-6$	$\sim 5E-6$	$.7E-6$	$.9E-6$	$.1E-5$	$.2E-6$	$.1E-6$	$.1E-6$
NEW	0.0	$\sim .0074$	$.019$	-1.38	-.095	-.868	-1.77	.071	-.151	-.05

YTP (298) COMPARISON (TERMINAL)

TABLE VI



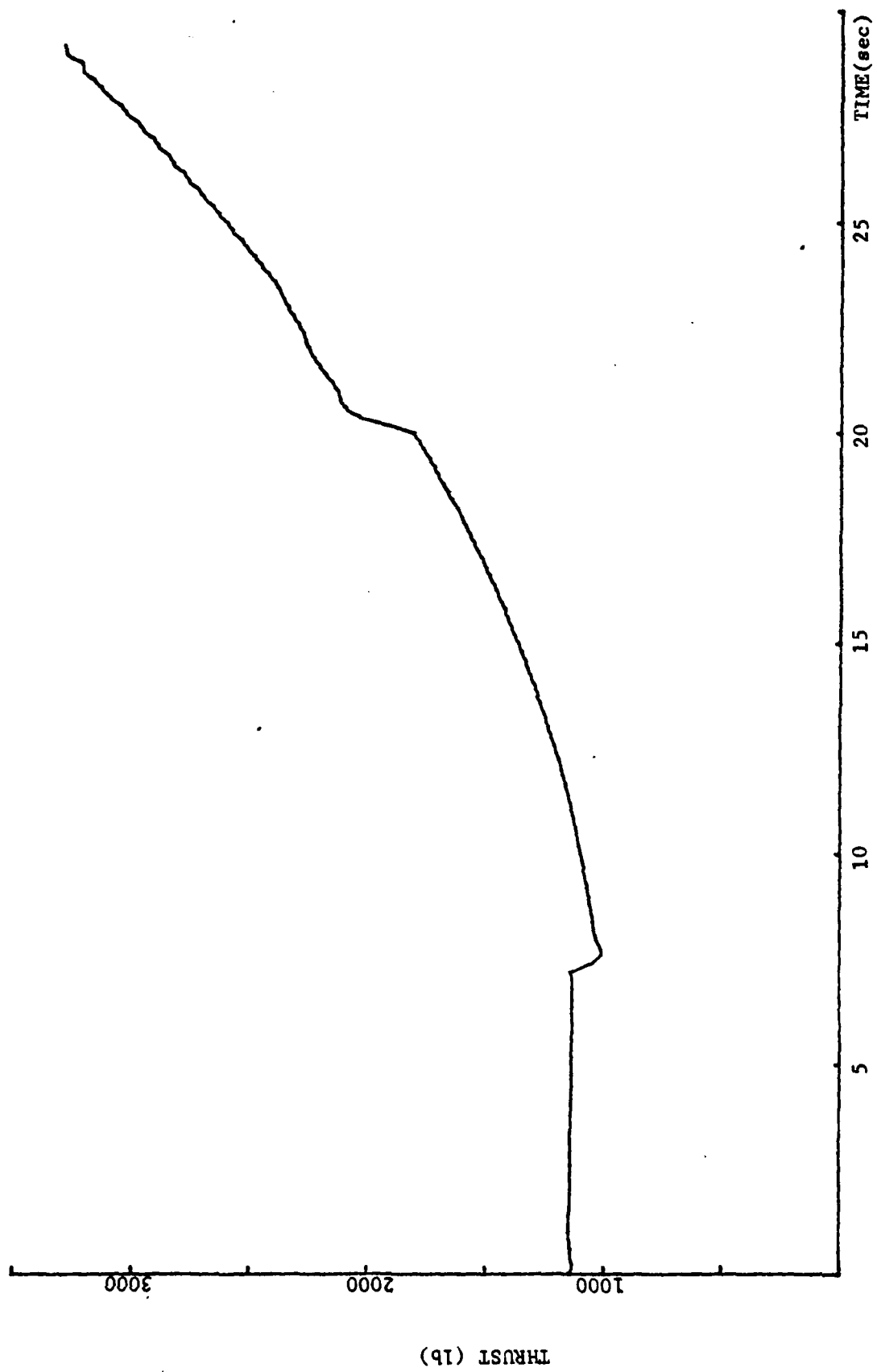
ALTITUDE VS. TIME (TERMINAL)

FIGURE 11

elimination of all the errors during the midcourse checkout. The resulting flight trajectory, Figure 11, is a very good illustration of the desired profile. To check the accuracy of the simulation, the same three parameters were chosen. Tables IV and V show that this run is almost identical to the terminal baseline. The difference is primarily due to computer round-off error. Figures 12 and 13 show the overall flight performance of the thrust and forward velocity, respectively.

For this simulation, the y-displacement (YTP(298)) does not compare favorably with the terminal baseline. Figure 14 shows that between ten and twenty seconds in the flight, YTP develops rapid oscillations. By examining Table VI, it is noted that during this interval the two simulations differ the most. No reason for this discrepancy could be found. These oscillations do effect the flight by increasing the time of flight from 25.825 to 29.09 seconds and by increasing the miss distance. The imposed time constraints force this problem to be overlooked and to direct attention to the accuracy of the impact. Since the impact is within 0.5 feet of the target location, it is concluded that the terminal guidance simulation works properly.

With both segments of the MOD6DF computer program working correctly, thoughts could now turn towards the actual experimental runs. The results of these runs are discussed in the next section.



THRUST VS. TIME (TERMINAL)

FIGURE 12

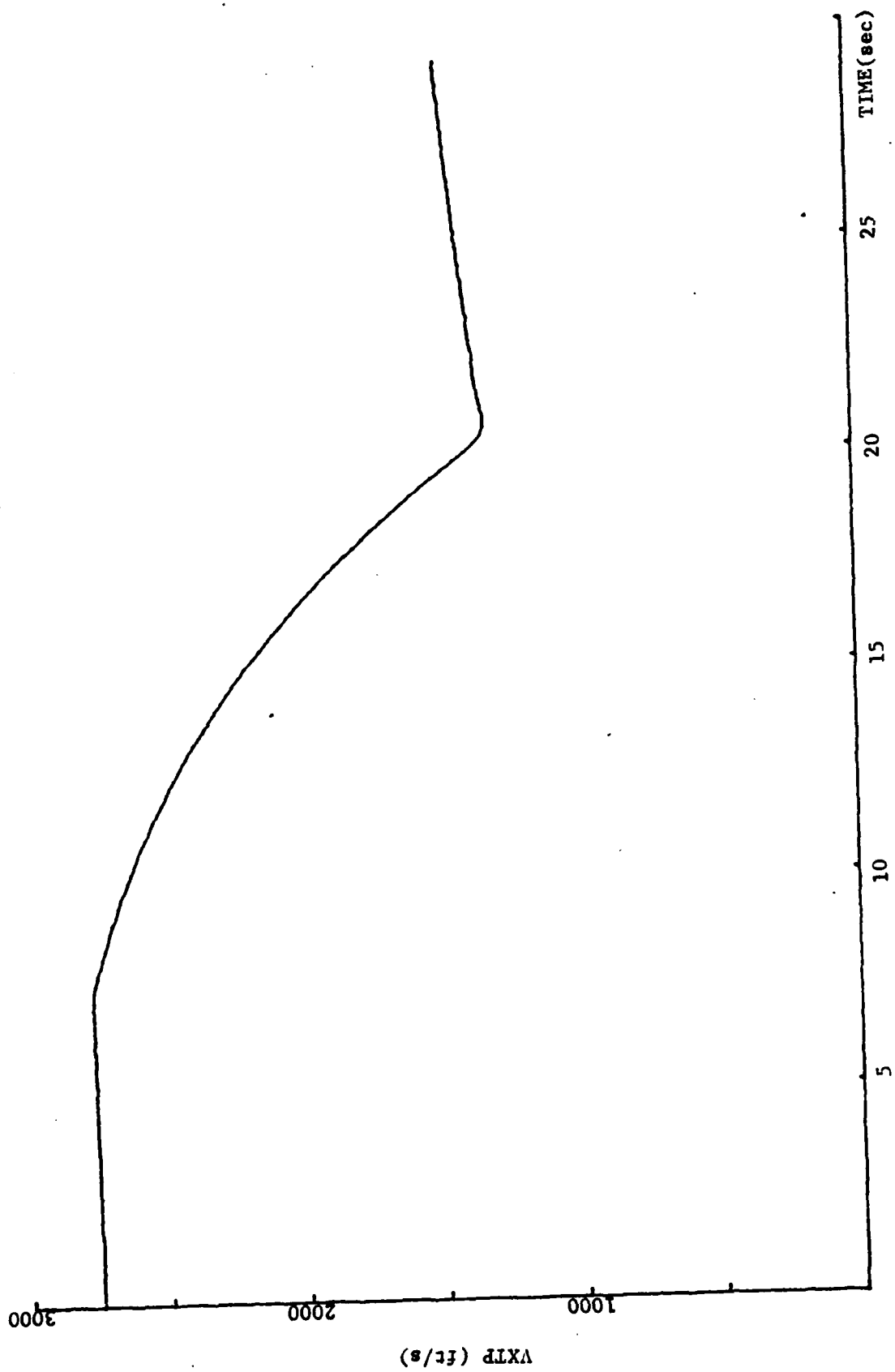
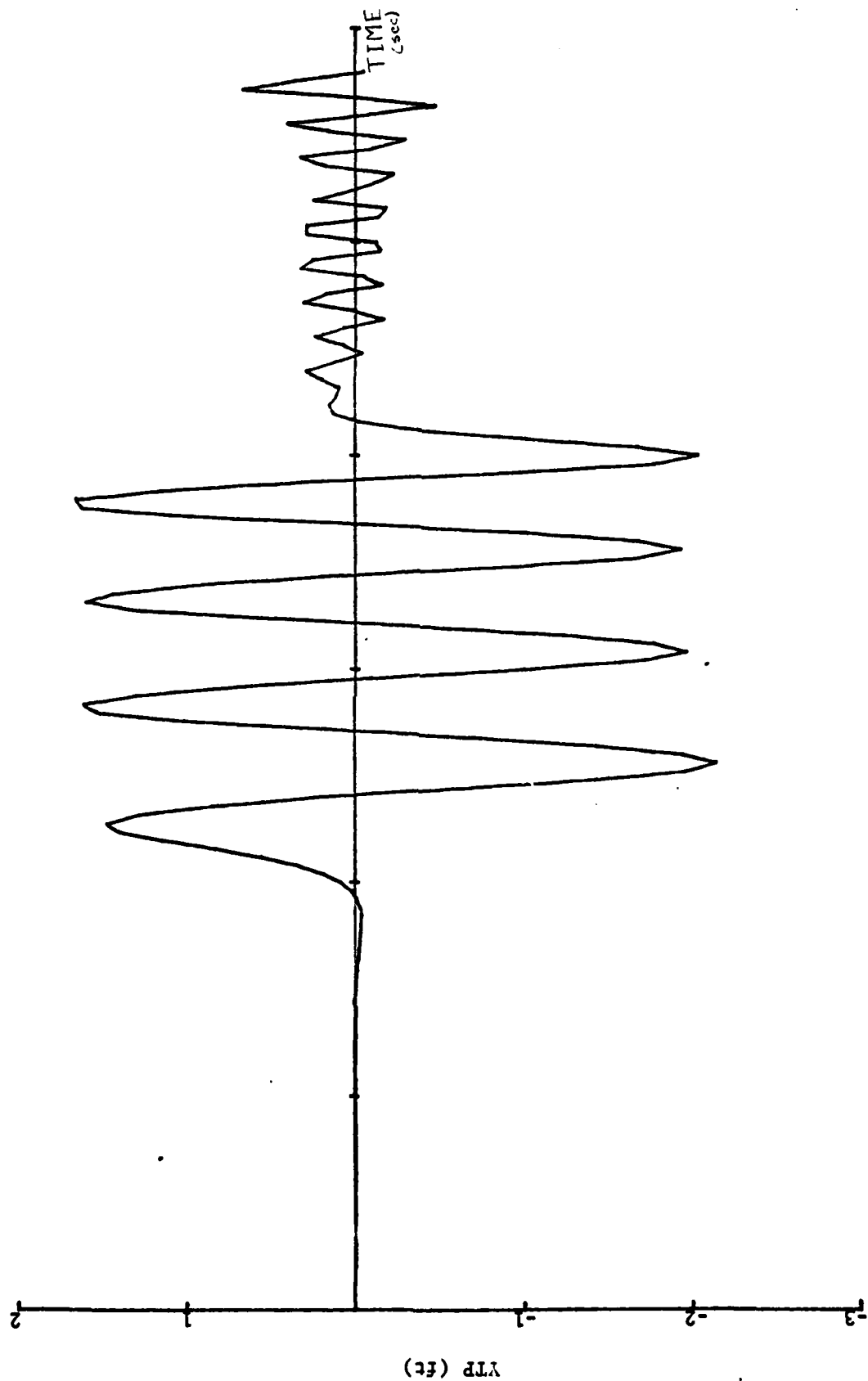


FIGURE 13



YTP VS. TIME (TERMINAL)

FIGURE 14

V. TERMINAL FLIGHT DISTURBANCES

With the checkout of the MOD6DF computer program completed, many ideas were discussed concerning alterations to the simulations. The three modifications decided upon were:

- * examine target miss distance when changes were made to the initial x-y-z conditions,
- * examine the terminal flight profile, range, and miss distance when the cruise altitude was reduced to approximate a sea skimming mode,
- * examine target miss distance when random noise is applied to the missile homing seeker.

To understand the changes and results, one must be familiar with the frame of reference. The origin of the reference frame travels from the launch platform to the target location at the cruise altitude. This is called the tangent plane reference system. Displacements in the x-direction (XTP(290)) are measured from the launch platform in the direction of the target. Y-displacements (YTP(298)) are measured left or right of the ideal flight path, in the tangent plane. Any vertical displacement (ZTP(306)) is measured normal to the ideal flight path, with down being the positive direction. Using this frame of reference the information in Table VII is easier to understand.

The first modifications demonstrate the effect of changing YTP. YTP was increased until a result was reached that was unsatisfactory. From the results in Table VIII, the maximum value of YTP was determined to be 1800 feet. This conclusion corresponds with the results in reference 13. Since the missile is symmetric, it was also concluded that moving YTP either right or left would give the same results.

The fifth run simulated a drop in the missile's altitude. ZTP was

RUN	XTP	YTP	ZTP
BASELINE	173552.5	0.0	0.0
1	173552.5	1000.0	0.0
2	173552.5	1500.0	0.0
3	173552.5	1800.0	0.0
4	173552.5	2000.0	0.0
5	173552.5	0.0	2000.0

INITIAL CONDITIONS

TABLE VII

RUN	XTP	YTP	ZTP	IMPACT TIME
BASELINE	234999.56	-.0506363	34999.34	28.825
1	234999.81	.1259258	34999.836	29.108
2	235000.19	.11479032	35000.32	29.124
3	235000.25	.15263242	35000.391	29.134
4	269802.0	2000.0	19705.00	35.0
5	234999.63	.45691	35000.426	30.365

IMPACT RESULTS

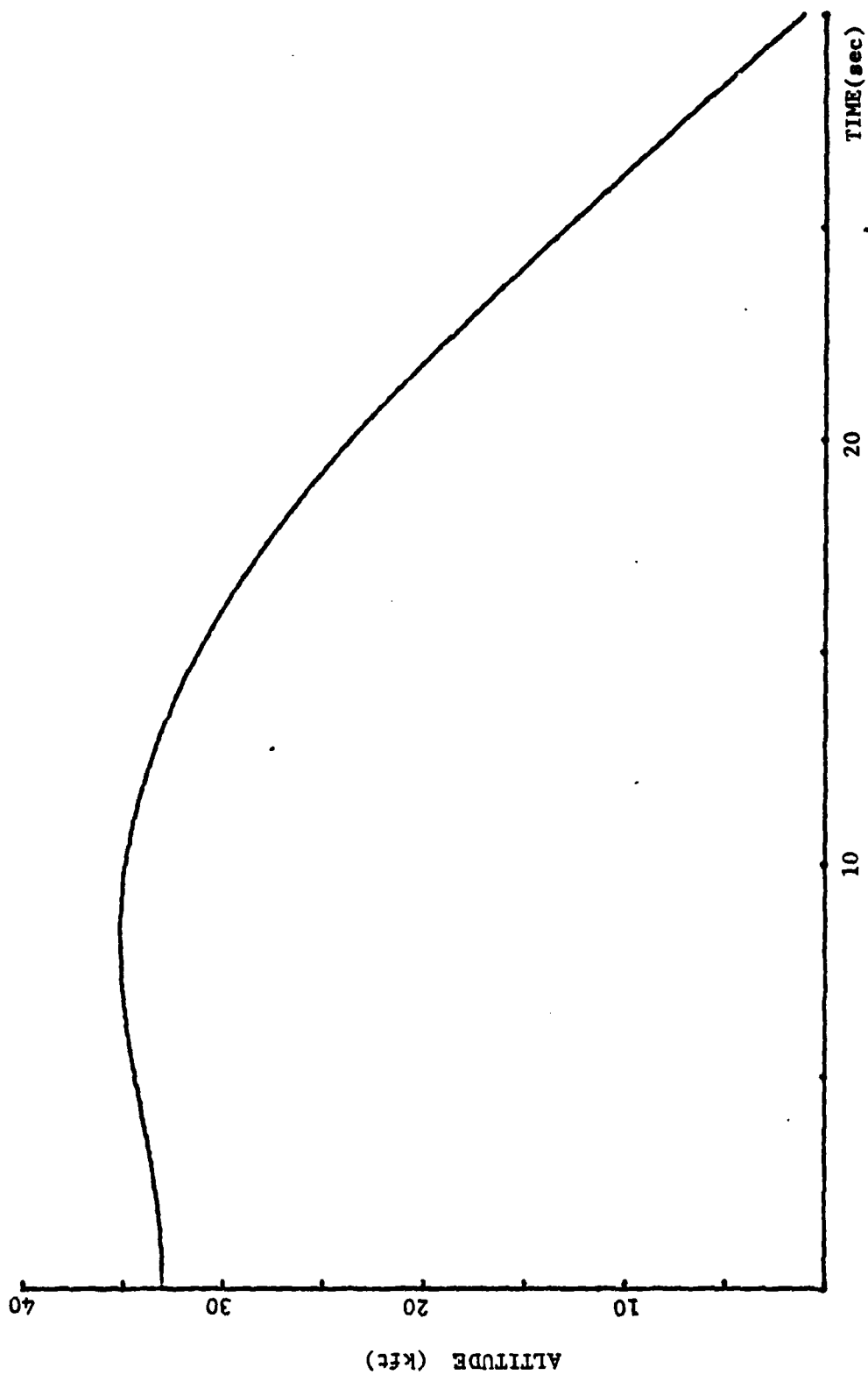
TABLE VIII

inputed as 2000.0 feet. As the simulation progressed, the missile developed the necessary commands to climb and regain the desired cruise altitude (Figure 15). It then commenced its terminal dive. The target was detected, acquired, and impact followed. This initial condition modification produced the largest miss distance (0.457 ft.).

Figure 16 was presented to show the relative locations of the miss distances for each change. Table VIII reveals that up until 1800 feet, changes to YTP created very little variation in the miss distance. It should be noted that the larger the displacement became, the greater the time until impact. The increase in time was necessary to allow the missile to acquire the target and then compute the required actuator commands to impact the target. This concluded the investigation of the effects of initial displacement error on target miss distance.

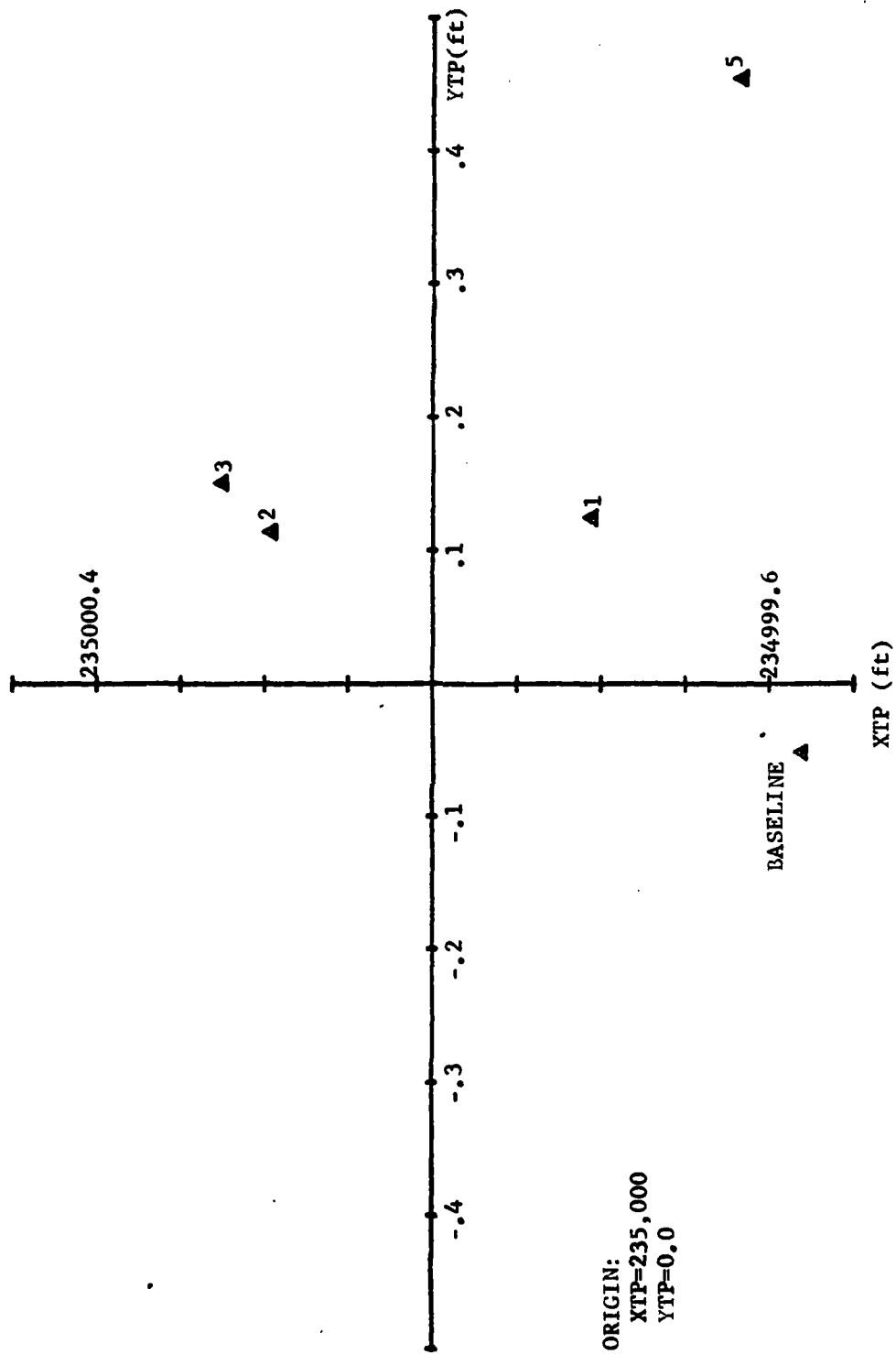
The idea behind changing the cruise altitude was to try and simulate a sea skimming profile. RAMJET contains tables that allow for four cruise altitudes. Of these four altitudes, the lowest (500 ft) was chosen even though it is high for a sea-skimmer. To run this simulation only two input data card changes were required. HO(414) and HREF(501) had to be set equal to the desired altitude. The resultant flight path is shown in Figure 17.

The simulation produced an error-free output, but at first glance the results appeared unacceptable. Still trying to compare miss distances, the downrange distance (XTP) was found to be 197916 feet. Comparing this to the terminal baseline (234999) resulted in an extreme error. It was then realized that the missile must expend more fuel at this altitude to attain the same speed. Therefore, the results could be correct. As further verification of the correctness of the run, it was assumed that the missile weight at impact should be fairly equal. The



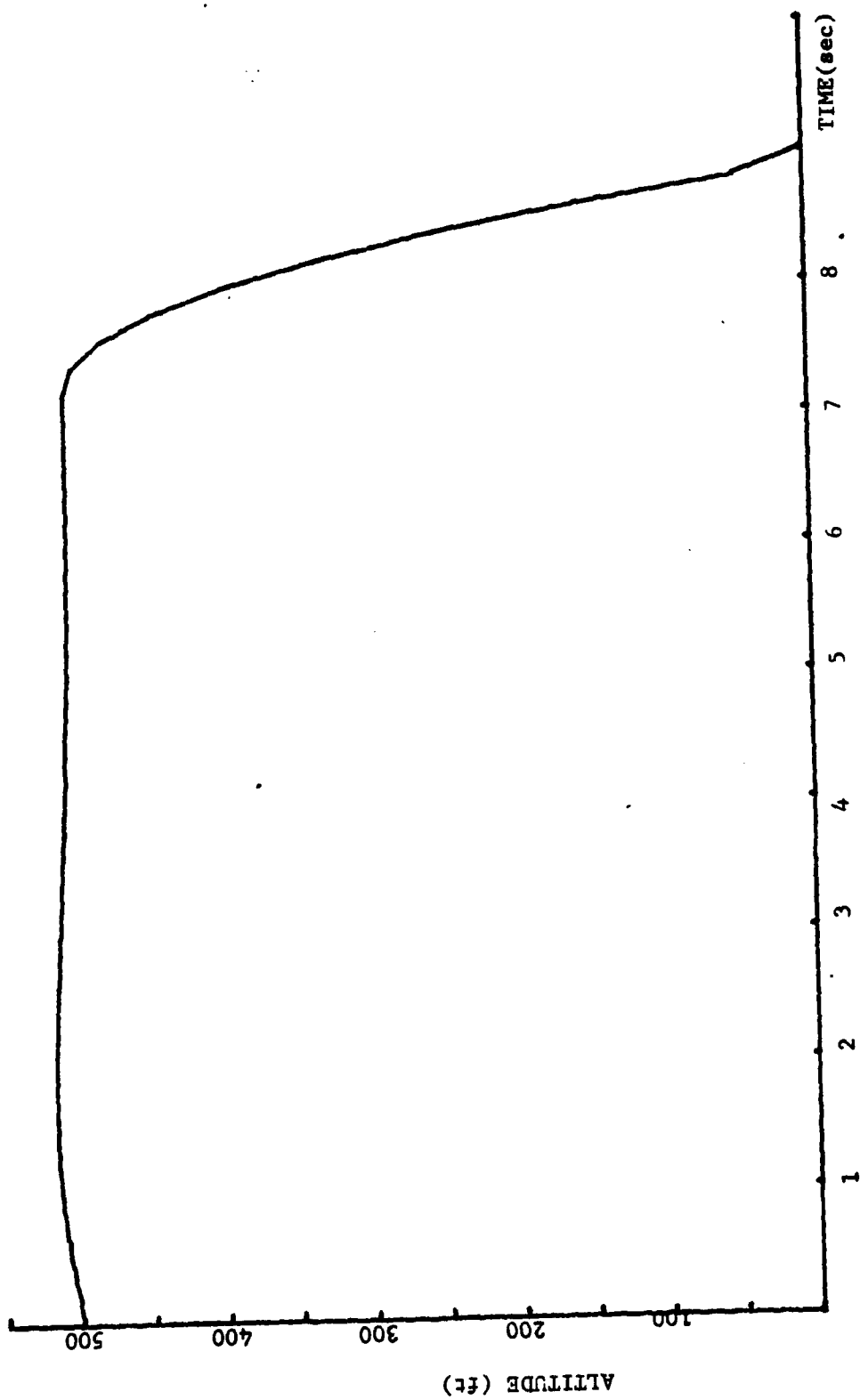
ALTITUDE VS. TIME (TERMINAL - ZTP)

FIGURE 15



IMPACT RESULTS

FIGURE 16



ALTITUDE VS. TIME (SEA-SKIMMER)

FIGURE 17

simulation impact weight was 1126.34 lbs while the terminal baseline weighed 1113.12 lbs. Since no conflicting information could be found in reference 12, it was concluded that this simulation was correct.

The final flight disturbance investigated was the addition of random noise to the missile homing seeker. This simulation proved to be unsuccessful due to invalid input format. The information describing the process is brief and difficult to understand. Further information from NWC China Lake is required to be able to successfully run simulations with random noise.

With the completion of these modifications, many unanswered problems and questions still exist. The two most critical problems concern the input of random noise generators and the missile exceeding the angle of attack limitation when using ENGINE for cruise propulsion. However, the simulations did show that the MOD6DF program runs correctly at alternate cruise altitudes and with initial displacement errors.

VI. CONCLUSION

The MOD6DF computer program from NWC China Lake was converted to operate on the IBM-360 computer at the Naval Postgraduate School. The program would only function properly when using the simplified ramjet model. When this model was not used, the missile angle of attack exceeded the maximum limit of ten degrees. This error caused the ramjet engine to flame out.

Target impact errors for the terminal guidance problem were investigated when the initial displacement was modified. These modifications demonstrate the missile's accuracy when removed from the ideal flight path. The results also point out that if the target falls within the seeker search pattern, target impact will inevitably happen.

Random noise generation is possible with the MOD6DF program. However, more information discussing the parameters required to develop the noise is necessary. Additionally, sample noise inputs should be obtained that reflect the alteration to the noise subroutines.

This research involved the preliminary investigation of the MOD6DF program. The program has many possible areas, concerning guidance and control of tactical missiles, which could be developed for future study. These areas not only include the unresolved problems encountered during this research. Additionally, reference 14 and 15 contain many examples of possible advanced guidance concepts.

COMPUTER OUTPUT

[illegible]

W. FLOST
S
CRA
CGI
EPSTH
NLT

ARC
C
T
S
F
F
P
L
C
C
L
L
L

LA11
LA11
LA12
LA12
LA13
LA21
LA21
LA22
LA22
LA23
LA23
LA31
LA31
LA32
LA32
LA33
LA33
GK14
AKS
GNGYD
FLCVCY
AC

GNGY
GNGZD
ELCNGZ
ELCNGZ
GNCZ
THTSO
THTST
THTST
THSIS
THSIS
FLPSIS
FLPSIS
HPSIS
HPSIS
X
FLVY
FLVY
VY

THE UNIVERSITY OF CHICAGO LIBRARY

[illegible]

THIS PAGE IS BEST QUALITY PRACTICABLE
FROM COPY FURNISHED TO DDC

TIME	TP-CST	WIGHT	X(TP)	W(TP)	Y(TP)
0.40555950F-02	C.1141750F-04 C.2504340F-01 C.0 C.0 C.0	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04
0.40555952F-00	C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04
0.40555951F-00	C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04
0.40555954F-00	C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04
0.40555953F-00	C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04 C.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04	0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04 0.1152550F-04

THIS PAGE IS BEST QUALITY PRACTICE
FROM COPY FURNISHED TO BDC

COMPUTER PROGRAM

THIS IS THE MAIN MCD6CF PROGRAM

WCD60010
 WCD6002C
 WCD60030
 WCD6004C
 WCD6005C
 WCD60060
 WCD6007C
 WCD60100
 WCD60110
 WCD60120
 WCD60130
 WCD60140
 WCD60150
 WCD60160
 WCD6017C
 WCD60200
 WCD60210

M C606220
 M C606230
 M C606240
 M C606250
 M C606260
 M C606270
 M C606280
 M C606310
 M C606320
 M C606330
 M C606340
 M C606350
 M C606360
 M C606370
 M C606400
 M C606410
 M C606420
 M C606430
 M C606440
 M C606450
 M C606460
 M C606470
 M C606500
 M C606520

[illegible]

```

100C CALL LSTEP
1001 CALL CINSTP
1002 CALL SUELI
1003 CALL AUXI
1004 CALL SUBL2,N
1005 DO J = IFL(I-1),N
      EU(I-1) = C(J+1)
      VAR(I) = C(J+3)
      CER(I) = C(J)
      VAR(I) = T
60   CALL AMK,AUXSUB
      CALL DO,I=2,N
1007 DO J = IFL(I-1),N
      CJ+3 = VAR(I)
      T = VAR(I)
1009 CALL KSTEP,EQ,I,T
      IF(KSTEP-I) 70
      IF(KFCEES
CALL PFCESET
GO TC(1000,1001

```

CALL RESE!
GO TC (1000,1001,10C2,10C3,1004,1005,1006,1C07,1008,1009,1Q10),

MCD60530
 MCD60540
 MCD60545
 MCD60550

1 LSTEP
 1010 CALL EXIT
 STCP
 END

C C C C

SUBROUTINE ZERO

COMMON C(2415)
 DO 1 KLEAF = 1, 4915
 1 C(KLEAF) = 0.0
 RETURN
 END

C C C C C C C

SUBROUTINE QINPT1

BASIC INFLT SUBROUTINE CINT1
 SUBROUTINE QINPT1 IS THE BASIC INPUT SUBROUTINE

DIMENSION LISTNO(50), VALUE(50)
 DIMENSION SUBNO(99), IR(2), VR(2)
 DIMENSION RNMNO(50)
 DIMENSION ALPHA(4), CNAME1(50), CNAME2(50), CNAME3(50), OUTNC(50)
 DIMENSION MCCNO(99)
 DIMENSION STATNO(100)
 COMMON C(2415)

REAL MCDNC
 INTEGER CLTNC
 INTEGER RADMNO
 INTEGER STATNG
 EQUIVALENCE (C(2442), LOSTAT)
 EQUIVALENCE (C(2200), STATNO(1)), (C(2441), NOSTAT)
 EQUIVALENCE (C(2801), NOSUB), (C(2802), SUBNO(1)), (C(2663), IR(1)),
 1 (C(2661), VR(1))
 EQUIVALENCE (C(2701), NCMOD)
 EQUIVALENCE (C(2661), NCMOUT)
 EQUIVALENCE (C(2448), NORNDM)
 EQUIVALENCE (C(2300), NOLIST), (C(2301), LISTNO(1)), (C(2351),
 1 VALUE(1))
 EQUIVALENCE (C(2500), CUTNO(1))
 EQUIVALENCE (C(2550), CNAME1(1))
 EQUIVALENCE (C(2600), CNAME2(1))
 EQUIVALENCE (C(2150), CNAME3(1))
 EQUIVALENCE (C(2450), RADMNO(1))
 EQUIVALENCE (C(2702), MCDNO(1))

JAR = C

C INP0030
 C INP001C
 C INP0020
 C INP0230
 C INP0240
 C INP0250
 C INP030C
 C INP0040
 C INP031C
 C INP032C
 C INP0330
 C INP034C
 C INP005C
 C INP011C
 C INP012C
 C INP0130
 C INP035C

```

1 READ(5,2) IR(1),ALPHA(1),ALPHA(2),ALPHA(3),ALPHA(4),IR(2),VR(1),
1 VR(2)
1 WRITE(6,2) IR(1),ALPHA(1),ALPHA(2),ALPHA(3),ALPHA(4),IR(2),VR(1),
1 VR(2)
2 FORMAT(12,2X,4A4,15,5X,2E15.8)
3 IF (IR(1) .NE. 0) GC TO 7
4 IF (IR(1)) 7,18,7
18 REWIND 11
13 = VR(1) - 1.
13 = VR(2)
IF (I2) 11,15,14
14 DO 8 I = 1,12
15 READ (11,10) J,X
15 CO 9 I = 1,13
5 C(J) = X
10 FORMAT (15,E15.9)
11 JAR = 1
7 GO TO 11
7 IF (IR(1) .NE. 1) GC TO 3
7 IF (IR(1) - 1) 3,19,3
15 NOSUB = NCSUB + 1
15 SUBNC(NCSUB) = IR(2)
5 GO TO 11
5 IF (IR(1) .NE. 2) GC TO 4
5 IF (IR(1) - 2) 4,20,4
20 NOMOD = NCMOD + 1
20 MODNC(NCMOD) = IR(2)
5 GO TO 11
5 IF (IR(1) .NE. 3) GC TO 5
5 IF (IR(1) - 3) 5,21,5
21 L = IR(2)
C(L) = VR(1)
IF (JAR .EQ. 1) WRITE (11,10)L,VR(1)
IF (JAR - 1) 23,22,23
22 WRITE (11,10)L,VR(1)
22 IF (VR(2) .EQ. 0.1) GC TO 1
23 IF (VR(2)) 24,1,24
24 NOLIST = NOLIST + 1
LISTNC(NCLIST) = L
VALUE(NCLIST) = VR(1)
5 GO TO 11
5 IF (IR(1) .NE. 4) GC TO 6
5 IF (IR(1) - 4) 6,25,6
25 ACCUT = ACCUT + 1
CNAME1(NCCLT) = ALPHA(2)

```

```

CINP0410
CINP042C
CINP043C
CINP0440
CINP045C
CINP0460
CINP0470
CINP050C
CINP0510
CINP052C
CINP053C
CINP0540
CINP055C
CINP056C
CINP0570
CINP060C
CINP0610
CINP062C
CINP063C
CINP064C
CINP065C
CINP0660
CINP0670
CINP070C
CINP071C
CINP072C
CINP073C
CINP0740
CINP075C
CINP076C
CINP0770
CINP100C
CINP1010
CINP102C
CINP103C
CINP1040
CINP105C
CINP1060
CINP1070
CINP110C
CINP111C
CINP112C

```



```

6 GO TC 1
  CALL RNCM1
7 GO TC 1
  CALL ALXA1
8 GO TC 1
  CALL ALXB1
9 GO TC 1
  CALL ALXC1
  CONTINUE
  RETURN
  END

```

CC C

SUBROUTINE SUBL2

```

DIMENSION SUBNO(99)
COMMON C(3415)
EQUIVALENCE (C(2801),NCSLB)
DO 1 I = 1, NCSUB
  J = SUBNO(I)
  GO TO (1, 2, 3, 4, 5, 6, 7, 8, 9), J
2 CALL INFT2
3 CALL CLFT2
4 CALL STGE2
5 CALL CNTR2
6 CALL RNCM2
7 CALL ALXA2
8 CALL ALXB2
9 CALL ALXC2
  CONTINUE
  RETURN
  END

```

CC C

SUBROUTINE SUBL3

```

DIMENSION SUBNO(99)
COMMON C(3415)

```

SUBL0200
SUBL0210
SUBL0220
SUBL0230
SUBL0240
SUBL0250
SUBL0260
SUBL0270
SUBL0300
SUBL0310
SUBL0320

SUBL0010
SUBL0050
SUBL0020
SUBL0030
SUBL0060
SUBL0070
SUBL0100
SUBL0110
SUBL0120
SUBL0130
SUBL0140
SUBL0150
SUBL0160
SUBL0170
SUBL0200
SUBL0210
SUBL0220
SUBL0230
SUBL0240
SUBL0250
SUBL0260
SUBL0270
SUBL0300
SUBL0310
SUBL0320

SUBL0010
SUBL0050
SUBL0020

SURL0030
SURL006C
SURL0070
SURL0100
SURL0110
SURL0120
SURL0130
SURL0140
SURL0150
SURL0160
SURL0170
SURL0200
SURL0210
SURL0220
SURL0230
SURL0240

SURL0250
SURL0260
SURL0270
SURL0300
SURL0310
SURL0320

ALXI0010
ALXI0060
ALXI0020
ALXI0070
ALXI0030

ALXI0050
ALXI0100
ALXI0110
ALXI0120
ALXI0130
ALXI0140
ALXI0150
ALXI0160
ALXI0170
ALXI0200
ALXI0210
ALXI0220

```

EQUIVALENCE (C(2801),NOSUB)
EQUIVALENCE (C(2802),SUENO(1))
DO I=1,NOSUB
  J=SUENO(I)
  GO TC (1,2,3,4,5,6,7,8,9),J
2 CALL INPT3
3 GO TC 1
4 CALL CLFT3
5 GO TC 1
6 CALL STGE3
7 GO TC 1
8 CALL CNTR3
9 GO TC 1
10 CALL RNCM3
11 GO TC 1
12 CALL ALX3
13 GO TC 1
14 CALL ALX83
15 GO TC 1
16 CALL ALXC3
17 CONTINUE
18 RETURN
19 END

```

```

SUBROUTINE ALXI
DIMENSION MCDNO(99)
COMMON C(3415)
REAL MCDNO
EQUIVALENCE (C(2701),ACMOD)
EQUIVALENCE (C(2702),MCDNO(1))
EQUIVALENCE (C(2659),CNCE)
EQUIVALENCE (C(3315),N)
N=1
CNCE=0
DO I=1,N
  GO TC (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37),I
2 CALL AI
3 GO TC 1
4 CALL AI
5 GO TC 1
6 CALL AI
7 GO TC 1
8 CALL AI
9 GO TC 1
10 GO TC 1
11 GO TC 1
12 GO TC 1
13 GO TC 1
14 GO TC 1
15 GO TC 1
16 GO TC 1
17 GO TC 1
18 GO TC 1
19 GO TC 1
20 GO TC 1
21 GO TC 1
22 GO TC 1
23 GO TC 1
24 GO TC 1
25 GO TC 1
26 GO TC 1
27 GO TC 1
28 GO TC 1
29 GO TC 1
30 GO TC 1
31 GO TC 1
32 GO TC 1
33 GO TC 1
34 GO TC 1
35 GO TC 1
36 GO TC 1
37 GO TC 1

```

CC C

AIXI023C
 AIXI024C
 AIXI025C
 AIXI026C
 AIXI027C
 AIXI028C
 AIXI029C
 AIXI030C
 AIXI031C
 AIXI032C
 AIXI033C
 AIXI034C
 AIXI035C
 AIXI036C
 AIXI037C
 AIXI038C
 AIXI039C
 AIXI040C
 AIXI041C
 AIXI042C
 AIXI043C
 AIXI044C
 AIXI045C
 AIXI046C
 AIXI047C
 AIXI048C
 AIXI049C
 AIXI050C
 AIXI051C
 AIXI052C
 AIXI053C
 AIXI054C
 AIXI055C
 AIXI056C
 AIXI057C
 AIXI058C
 AIXI059C
 AIXI060C
 AIXI061C
 AIXI062C
 AIXI063C
 AIXI064C
 AIXI065C
 AIXI066C
 AIXI067C
 AIXI068C
 AIXI069C
 AIXI070C
 AIXI071C
 AIXI072C
 AIXI073C
 AIXI074C
 AIXI075C
 AIXI076C
 AIXI077C
 AIXI078C
 AIXI079C
 AIXI080C
 AIXI081C
 AIXI082C
 AIXI083C
 AIXI084C
 AIXI085C
 AIXI086C
 AIXI087C
 AIXI088C
 AIXI089C
 AIXI090C
 AIXI091C
 AIXI092C
 AIXI093C
 AIXI094C
 AIXI095C
 AIXI096C
 AIXI097C
 AIXI098C
 AIXI099C
 AIXI100C
 AIXI101C
 AIXI102C

5 CALL A4I
 6 GO TC I
 7 CALL A5I
 8 GO TC I
 9 CALL C1I
 10 GO TC I
 11 CALL C2I
 12 GO TC I
 13 CALL C3I
 14 GO TC I
 15 CALL C4I
 16 GO TC I
 17 CALL C5I
 18 GO TC I
 19 CALL C6I
 20 GO TC I
 21 CALL C7I
 22 GO TC I
 23 CALL C8I
 24 GO TC I
 25 CALL C9I
 26 GO TC I
 27 CALL D1I
 28 GO TC I
 29 CALL D2I
 30 GO TC I
 31 CALL D3I
 32 GO TC I
 33 CALL D4I
 34 GO TC I
 35 CALL D5I
 36 GO TC I
 37 CALL D6I
 38 GO TC I
 39 CALL D7I
 40 GO TC I
 41 CALL D8I
 42 GO TC I
 43 CALL D9I
 44 GO TC I
 45 CALL E1I
 46 GO TC I
 47 CALL E2I
 48 GO TC I
 49 CALL E3I
 50 GO TC I
 51 CALL E4I
 52 GO TC I
 53 CALL E5I
 54 GO TC I
 55 CALL E6I
 56 GO TC I
 57 CALL E7I
 58 GO TC I
 59 CALL E8I
 60 GO TC I
 61 CALL E9I
 62 GO TC I
 63 CALL F1I
 64 GO TC I
 65 CALL F2I
 66 GO TC I
 67 CALL F3I
 68 GO TC I
 69 CALL F4I
 70 GO TC I
 71 CALL F5I
 72 GO TC I
 73 CALL F6I
 74 GO TC I
 75 CALL F7I
 76 GO TC I
 77 CALL F8I
 78 GO TC I
 79 CALL F9I
 80 GO TC I
 81 CALL G1I
 82 GO TC I
 83 CALL G2I
 84 GO TC I
 85 CALL G3I
 86 GO TC I
 87 CALL G4I
 88 GO TC I
 89 CALL G5I
 90 GO TC I
 91 CALL G6I
 92 GO TC I
 93 CALL G7I
 94 GO TC I
 95 CALL G8I
 96 GO TC I
 97 CALL G9I
 98 GO TC I
 99 CALL H1I
 100 GO TC I
 101 CALL H2I
 102 GO TC I
 103 CALL H3I
 104 GO TC I
 105 CALL H4I
 106 GO TC I
 107 CALL H5I
 108 GO TC I
 109 CALL H6I
 110 GO TC I
 111 CALL H7I
 112 GO TC I
 113 CALL H8I
 114 GO TC I
 115 CALL H9I
 116 GO TC I
 117 CALL I1I
 118 GO TC I
 119 CALL I2I
 120 GO TC I
 121 CALL I3I
 122 GO TC I
 123 CALL I4I
 124 GO TC I
 125 CALL I5I
 126 GO TC I
 127 CALL I6I
 128 GO TC I
 129 CALL I7I
 130 GO TC I
 131 CALL I8I
 132 GO TC I
 133 CALL I9I
 134 GO TC I
 135 CALL J1I
 136 GO TC I
 137 CALL J2I
 138 GO TC I
 139 CALL J3I
 140 GO TC I
 141 CALL J4I
 142 GO TC I
 143 CALL J5I
 144 GO TC I
 145 CALL J6I
 146 GO TC I
 147 CALL J7I
 148 GO TC I
 149 CALL J8I
 150 GO TC I
 151 CALL J9I
 152 GO TC I
 153 CALL K1I
 154 GO TC I
 155 CALL K2I
 156 GO TC I
 157 CALL K3I
 158 GO TC I
 159 CALL K4I
 160 GO TC I
 161 CALL K5I
 162 GO TC I
 163 CALL K6I
 164 GO TC I
 165 CALL K7I
 166 GO TC I
 167 CALL K8I
 168 GO TC I
 169 CALL K9I
 170 GO TC I
 171 CALL L1I
 172 GO TC I
 173 CALL L2I
 174 GO TC I
 175 CALL L3I
 176 GO TC I
 177 CALL L4I
 178 GO TC I
 179 CALL L5I
 180 GO TC I
 181 CALL L6I
 182 GO TC I
 183 CALL L7I
 184 GO TC I
 185 CALL L8I
 186 GO TC I
 187 CALL L9I
 188 GO TC I
 189 CALL M1I
 190 GO TC I
 191 CALL M2I
 192 GO TC I
 193 CALL M3I
 194 GO TC I
 195 CALL M4I
 196 GO TC I
 197 CALL M5I
 198 GO TC I
 199 CALL M6I
 200 GO TC I
 201 CALL M7I
 202 GO TC I
 203 CALL M8I
 204 GO TC I
 205 CALL M9I
 206 GO TC I
 207 CALL N1I
 208 GO TC I
 209 CALL N2I
 210 GO TC I
 211 CALL N3I
 212 GO TC I
 213 CALL N4I
 214 GO TC I
 215 CALL N5I
 216 GO TC I
 217 CALL N6I
 218 GO TC I
 219 CALL N7I
 220 GO TC I
 221 CALL N8I
 222 GO TC I
 223 CALL N9I
 224 GO TC I
 225 CALL O1I
 226 GO TC I
 227 CALL O2I
 228 GO TC I
 229 CALL O3I
 230 GO TC I
 231 CALL O4I
 232 GO TC I
 233 CALL O5I
 234 GO TC I
 235 CALL O6I
 236 GO TC I
 237 CALL O7I
 238 GO TC I
 239 CALL O8I
 240 GO TC I
 241 CALL O9I
 242 GO TC I
 243 CALL P1I
 244 GO TC I
 245 CALL P2I
 246 GO TC I
 247 CALL P3I
 248 GO TC I
 249 CALL P4I
 250 GO TC I
 251 CALL P5I
 252 GO TC I
 253 CALL P6I
 254 GO TC I
 255 CALL P7I
 256 GO TC I
 257 CALL P8I
 258 GO TC I
 259 CALL P9I
 260 GO TC I
 261 CALL Q1I
 262 GO TC I
 263 CALL Q2I
 264 GO TC I
 265 CALL Q3I
 266 GO TC I
 267 CALL Q4I
 268 GO TC I
 269 CALL Q5I
 270 GO TC I
 271 CALL Q6I
 272 GO TC I
 273 CALL Q7I
 274 GO TC I
 275 CALL Q8I
 276 GO TC I
 277 CALL Q9I
 278 GO TC I
 279 CALL R1I
 280 GO TC I
 281 CALL R2I
 282 GO TC I
 283 CALL R3I
 284 GO TC I
 285 CALL R4I
 286 GO TC I
 287 CALL R5I
 288 GO TC I
 289 CALL R6I
 290 GO TC I
 291 CALL R7I
 292 GO TC I
 293 CALL R8I
 294 GO TC I
 295 CALL R9I
 296 GO TC I
 297 CALL S1I
 298 GO TC I
 299 CALL S2I
 300 GO TC I
 301 CALL S3I
 302 GO TC I
 303 CALL S4I
 304 GO TC I
 305 CALL S5I
 306 GO TC I
 307 CALL S6I
 308 GO TC I
 309 CALL S7I
 310 GO TC I
 311 CALL S8I
 312 GO TC I
 313 CALL S9I
 314 GO TC I
 315 CALL T1I
 316 GO TC I
 317 CALL T2I
 318 GO TC I
 319 CALL T3I
 320 GO TC I
 321 CALL T4I
 322 GO TC I
 323 CALL T5I
 324 GO TC I
 325 CALL T6I
 326 GO TC I
 327 CALL T7I
 328 GO TC I
 329 CALL T8I
 330 GO TC I
 331 CALL T9I
 332 GO TC I
 333 CALL U1I
 334 GO TC I
 335 CALL U2I
 336 GO TC I
 337 CALL U3I
 338 GO TC I
 339 CALL U4I
 340 GO TC I
 341 CALL U5I
 342 GO TC I
 343 CALL U6I
 344 GO TC I
 345 CALL U7I
 346 GO TC I
 347 CALL U8I
 348 GO TC I
 349 CALL U9I
 350 GO TC I
 351 CALL V1I
 352 GO TC I
 353 CALL V2I
 354 GO TC I
 355 CALL V3I
 356 GO TC I
 357 CALL V4I
 358 GO TC I
 359 CALL V5I
 360 GO TC I
 361 CALL V6I
 362 GO TC I
 363 CALL V7I
 364 GO TC I
 365 CALL V8I
 366 GO TC I
 367 CALL V9I
 368 GO TC I
 369 CALL W1I
 370 GO TC I
 371 CALL W2I
 372 GO TC I
 373 CALL W3I
 374 GO TC I
 375 CALL W4I
 376 GO TC I
 377 CALL W5I
 378 GO TC I
 379 CALL W6I
 380 GO TC I
 381 CALL W7I
 382 GO TC I
 383 CALL W8I
 384 GO TC I
 385 CALL W9I
 386 GO TC I
 387 CALL X1I
 388 GO TC I
 389 CALL X2I
 390 GO TC I
 391 CALL X3I
 392 GO TC I
 393 CALL X4I
 394 GO TC I
 395 CALL X5I
 396 GO TC I
 397 CALL X6I
 398 GO TC I
 399 CALL X7I
 400 GO TC I
 401 CALL X8I
 402 GO TC I
 403 CALL X9I
 404 GO TC I
 405 CALL Y1I
 406 GO TC I
 407 CALL Y2I
 408 GO TC I
 409 CALL Y3I
 410 GO TC I
 411 CALL Y4I
 412 GO TC I
 413 CALL Y5I
 414 GO TC I
 415 CALL Y6I
 416 GO TC I
 417 CALL Y7I
 418 GO TC I
 419 CALL Y8I
 420 GO TC I
 421 CALL Y9I
 422 GO TC I
 423 CALL Z1I
 424 GO TC I
 425 CALL Z2I
 426 GO TC I
 427 CALL Z3I
 428 GO TC I
 429 CALL Z4I
 430 GO TC I
 431 CALL Z5I
 432 GO TC I
 433 CALL Z6I
 434 GO TC I
 435 CALL Z7I
 436 GO TC I
 437 CALL Z8I
 438 GO TC I
 439 CALL Z9I
 440 GO TC I
 441 CALL AA1I
 442 GO TC I
 443 CALL AA2I
 444 GO TC I
 445 CALL AA3I
 446 GO TC I
 447 CALL AA4I
 448 GO TC I
 449 CALL AA5I
 450 GO TC I
 451 CALL AA6I
 452 GO TC I
 453 CALL AA7I
 454 GO TC I
 455 CALL AA8I
 456 GO TC I
 457 CALL AA9I
 458 GO TC I
 459 CALL AB1I
 460 GO TC I
 461 CALL AB2I
 462 GO TC I
 463 CALL AB3I
 464 GO TC I
 465 CALL AB4I
 466 GO TC I
 467 CALL AB5I
 468 GO TC I
 469 CALL AB6I
 470 GO TC I
 471 CALL AB7I
 472 GO TC I
 473 CALL AB8I
 474 GO TC I
 475 CALL AB9I
 476 GO TC I
 477 CALL AC1I
 478 GO TC I
 479 CALL AC2I
 480 GO TC I
 481 CALL AC3I
 482 GO TC I
 483 CALL AC4I
 484 GO TC I
 485 CALL AC5I
 486 GO TC I
 487 CALL AC6I
 488 GO TC I
 489 CALL AC7I
 490 GO TC I
 491 CALL AC8I
 492 GO TC I
 493 CALL AC9I
 494 GO TC I
 495 CALL AD1I
 496 GO TC I
 497 CALL AD2I
 498 GO TC I
 499 CALL AD3I
 500 GO TC I
 501 CALL AD4I
 502 GO TC I
 503 CALL AD5I
 504 GO TC I
 505 CALL AD6I
 506 GO TC I
 507 CALL AD7I
 508 GO TC I
 509 CALL AD8I
 510 GO TC I
 511 CALL AD9I
 512 GO TC I
 513 CALL AE1I
 514 GO TC I
 515 CALL AE2I
 516 GO TC I
 517 CALL AE3I
 518 GO TC I
 519 CALL AE4I
 520 GO TC I
 521 CALL AE5I
 522 GO TC I
 523 CALL AE6I
 524 GO TC I
 525 CALL AE7I
 526 GO TC I
 527 CALL AE8I
 528 GO TC I
 529 CALL AE9I
 530 GO TC I
 531 CALL AF1I
 532 GO TC I
 533 CALL AF2I
 534 GO TC I
 535 CALL AF3I
 536 GO TC I
 537 CALL AF4I
 538 GO TC I
 539 CALL AF5I
 540 GO TC I
 541 CALL AF6I
 542 GO TC I
 543 CALL AF7I
 544 GO TC I
 545 CALL AF8I
 546 GO TC I
 547 CALL AF9I
 548 GO TC I
 549 CALL AG1I
 550 GO TC I
 551 CALL AG2I
 552 GO TC I
 553 CALL AG3I
 554 GO TC I
 555 CALL AG4I
 556 GO TC I
 557 CALL AG5I
 558 GO TC I
 559 CALL AG6I
 560 GO TC I
 561 CALL AG7I
 562 GO TC I
 563 CALL AG8I
 564 GO TC I
 565 CALL AG9I
 566 GO TC I
 567 CALL AH1I
 568 GO TC I
 569 CALL AH2I
 570 GO TC I
 571 CALL AH3I
 572 GO TC I
 573 CALL AH4I
 574 GO TC I
 575 CALL AH5I
 576 GO TC I
 577 CALL AH6I
 578 GO TC I
 579 CALL AH7I
 580 GO TC I
 581 CALL AH8I
 582 GO TC I
 583 CALL AH9I
 584 GO TC I
 585 CALL AI1I
 586 GO TC I
 587 CALL AI2I
 588 GO TC I
 589 CALL AI3I
 590 GO TC I
 591 CALL AI4I
 592 GO TC I
 593 CALL AI5I
 594 GO TC I
 595 CALL AI6I
 596 GO TC I
 597 CALL AI7I
 598 GO TC I
 599 CALL AI8I
 600 GO TC I
 601 CALL AI9I
 602 GO TC I
 603 CALL AJ1I
 604 GO TC I
 605 CALL AJ2I
 606 GO TC I
 607 CALL AJ3I
 608 GO TC I
 609 CALL AJ4I
 610 GO TC I
 611 CALL AJ5I
 612 GO TC I
 613 CALL AJ6I
 614 GO TC I
 615 CALL AJ7I
 616 GO TC I
 617 CALL AJ8I
 618 GO TC I
 619 CALL AJ9I
 620 GO TC I
 621 CALL AK1I
 622 GO TC I
 623 CALL AK2I
 624 GO TC I
 625 CALL AK3I
 626 GO TC I
 627 CALL AK4I
 628 GO TC I
 629 CALL AK5I
 630 GO TC I
 631 CALL AK6I
 632 GO TC I
 633 CALL AK7I
 634 GO TC I
 635 CALL AK8I
 636 GO TC I
 637 CALL AK9I
 638 GO TC I
 639 CALL AL1I
 640 GO TC I
 641 CALL AL2I
 642 GO TC I
 643 CALL AL3I
 644 GO TC I
 645 CALL AL4I
 646 GO TC I
 647 CALL AL5I
 648 GO TC I
 649 CALL AL6I
 650 GO TC I
 651 CALL AL7I
 652 GO TC I
 653 CALL AL8I
 654 GO TC I
 655 CALL AL9I
 656 GO TC I
 657 CALL AM1I
 658 GO TC I
 659 CALL AM2I
 660 GO TC I
 661 CALL AM3I
 662 GO TC I
 663 CALL AM4I
 664 GO TC I
 665 CALL AM5I
 666 GO TC I
 667 CALL AM6I
 668 GO TC I
 669 CALL AM7I
 670 GO TC I
 671 CALL AM8I
 672 GO TC I
 673 CALL AM9I
 674 GO TC I
 675 CALL AN1I
 676 GO TC I
 677 CALL AN2I
 678 GO TC I
 679 CALL AN3I
 680 GO TC I
 681 CALL AN4I
 682 GO TC I
 683 CALL AN5I
 684 GO TC I
 685 CALL AN6I
 686 GO TC I
 687 CALL AN7I
 688 GO TC I
 689 CALL AN8I
 690 GO TC I
 691 CALL AN9I
 692 GO TC I
 693 CALL AO1I
 694 GO TC I
 695 CALL AO2I
 696 GO TC I
 697 CALL AO3I
 698 GO TC I
 699 CALL AO4I
 700 GO TC I
 701 CALL AO5I
 702 GO TC I
 703 CALL AO6I
 704 GO TC I
 705 CALL AO7I
 706 GO TC I
 707 CALL AO8I
 708 GO TC I
 709 CALL AO9I
 710 GO TC I
 711 CALL AP1I
 712 GO TC I
 713 CALL AP2I
 714 GO TC I
 715 CALL AP3I
 716 GO TC I
 717 CALL AP4I
 718 GO TC I
 719 CALL AP5I
 720 GO TC I
 721 CALL AP6I
 722 GO TC I
 723 CALL AP7I
 724 GO TC I
 725 CALL AP8I
 726 GO TC I
 727 CALL AP9I
 728 GO TC I
 729 CALL AQ1I
 730 GO TC I
 731 CALL AQ2I
 732 GO TC I
 733 CALL AQ3I
 734 GO TC I
 735 CALL AQ4I
 736 GO TC I
 737 CALL AQ5I
 738 GO TC I
 739 CALL AQ6I
 740 GO TC I
 741 CALL AQ7I
 742 GO TC I
 743 CALL AQ8I
 744 GO TC I
 745 CALL AQ9I
 746 GO TC I
 747 CALL AR1I
 748 GO TC I
 749 CALL AR2I
 750 GO TC I
 751 CALL AR3I
 752 GO TC I
 753 CALL AR4I
 754 GO TC I
 755 CALL AR5I
 756 GO TC I
 757 CALL AR6I
 758 GO TC I
 759 CALL AR7I
 760 GO TC I
 761 CALL AR8I
 762 GO TC I
 763 CALL AR9I
 764 GO TC I
 765 CALL AS1I
 766 GO TC I
 767 CALL AS2I
 768 GO TC I
 769 CALL AS3I
 770 GO TC I
 771 CALL AS4I
 772 GO TC I
 773 CALL AS5I
 774 GO TC I
 775 CALL AS6I
 776 GO TC I
 777 CALL AS7I
 778 GO TC I
 779 CALL AS8I
 780 GO TC I
 781 CALL AS9I
 782 GO TC I
 783 CALL AT1I
 784 GO TC I
 785 CALL AT2I
 786 GO TC I
 787 CALL AT3I
 788 GO TC I
 789 CALL AT4I
 790 GO TC I
 791 CALL AT5I
 792 GO TC I
 793 CALL AT6I
 794 GO TC I
 795 CALL AT7I
 796 GO TC I
 797 CALL AT8I
 798 GO TC I
 799 CALL AT9I
 800 GO TC I
 801 CALL AU1I
 802 GO TC I
 803 CALL AU2I
 804 GO TC I
 805 CALL AU3I
 806 GO TC I
 807 CALL AU4I
 808 GO TC I
 809 CALL AU5I
 810 GO TC I
 811 CALL AU6I
 812 GO TC I
 813 CALL AU7I
 814 GO TC I
 815 CALL AU8I
 816 GO TC I
 817 CALL AU9I
 818 GO TC I
 819 CALL AV1I
 820 GO TC I
 821 CALL AV2I
 822 GO TC I
 823 CALL AV3I
 824 GO TC I
 825 CALL AV4I
 826 GO TC I
 827 CALL AV5I
 828 GO TC I
 829 CALL AV6I
 830 GO TC I
 831 CALL AV7I
 832 GO TC I
 833 CALL AV8I
 834 GO TC I
 835 CALL AV9I
 836 GO TC I
 837 CALL AW1I
 838 GO TC I
 839 CALL AW2I
 840 GO TC I
 841 CALL AW3I
 842 GO TC I
 843 CALL AW4I
 844 GO TC I
 845 CALL AW5I
 846 GO TC I
 847 CALL AW6I
 848 GO TC I
 849 CALL AW7I
 850 GO TC I
 851 CALL AW8I
 852 GO TC I
 853 CALL AW9I
 854 GO TC I
 855 CALL AX1I
 856 GO TC I
 857 CALL AX2I
 858 GO TC I
 859 CALL AX3I
 860 GO TC I
 861 CALL AX4I
 862 GO TC I
 863 CALL AX5I
 864 GO TC I
 865 CALL AX6I
 866 GO TC I
 867 CALL AX7I
 868 GO TC I
 869 CALL AX8I
 870 GO TC I
 871 CALL AX9I
 872 GO TC I
 873 CALL AY1I
 874 GO TC I
 875 CALL AY2I
 876 GO TC I
 877 CALL AY3I
 878 GO TC I
 879 CALL AY4I
 880 GO TC I
 881 CALL AY5I
 882 GO TC I
 883 CALL AY6I
 884 GO TC I
 885 CALL AY7I
 886 GO TC I
 887 CALL AY8I
 888 GO TC I
 889 CALL AY9I
 890 GO TC I
 891 CALL AZ1I
 892 GO TC I
 893 CALL AZ2I
 894 GO TC I
 895 CALL AZ3I
 896 GO TC I
 897 CALL AZ4I
 898 GO TC I
 899 CALL AZ5I
 900 GO TC I
 901 CALL AZ6I
 902 GO TC I
 903 CALL AZ7I
 904 GO TC I
 905 CALL AZ8I
 906 GO TC I
 907 CALL AZ9I
 908 GO TC I
 909 CALL BA1I
 910 GO TC I
 911 CALL BA2I
 912 GO TC I
 913 CALL BA3I
 914 GO TC I
 915 CALL BA4I
 916 GO TC I
 917 CALL BA5I
 918 GO TC I
 919 CALL BA6I
 920 GO TC I
 921 CALL BA7I
 922 GO TC I
 923 CALL BA8I
 924 GO TC I
 925 CALL BA9I
 926 GO TC I
 927 CALL BB1I
 928 GO TC I
 929 CALL BB2I
 930 GO TC I
 931 CALL BB3I
 932 GO TC I
 933 CALL BB4I
 934 GO TC I
 935 CALL BB5I
 936 GO TC I
 937 CALL BB6I
 938 GO TC I
 939 CALL BB7I
 940 GO TC I
 941 CALL BB8I
 942 GO TC I
 943 CALL BB9I
 944 GO TC I
 945 CALL BC1I
 946 GO TC I
 947 CALL BC2I
 948 GO TC I
 949 CALL BC3I
 950 GO TC I
 951 CALL BC4I
 952 GO TC I
 953 CALL BC5I
 954 GO TC I
 955 CALL BC6I
 956 GO TC I
 957 CALL BC7I
 958 GO TC I
 959 CALL BC8I
 960 GO TC I
 961 CALL BC9I
 962 GO TC I
 963 CALL BD1I
 964 GO TC I
 965 CALL BD2I
 966 GO TC I
 967 CALL BD3I
 968 GO TC I
 969 CALL BD4I
 970 GO TC I
 971 CALL BD5I
 972 GO TC I
 973 CALL BD6I
 974 GO TC I
 975 CALL BD7I
 976 GO TC I
 977 CALL BD8I
 978 GO TC I
 979 CALL BD9I
 980 GO TC I
 981 CALL BE1I
 982 GO TC I
 983 CALL BE2I
 984 GO TC I
 985 CALL BE3I
 986 GO TC I
 987 CALL BE4I
 988 GO TC I
 989 CALL BE5I
 990 GO TC I
 991 CALL BE6I
 992 GO TC I
 993 CALL BE7I
 994 GO TC I
 995 CALL BE8I
 996 GO TC I
 997 CALL BE9I
 998 GO TC I
 999 CALL BF1I
 1000 GO TC I

[illegible]

```

50 C(J+I) = VAR(I)
   DO 1 I=1,NCMCC
      L = MCMCC(I)
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,
123,24,45,26,27,28,29,30,31,32,33,34,35,36,37),L
      CALL A1
   GO TO 1

```

AXS0310
 AXS0320
 AXS0330
 AXS0340
 AXS0350
 AXS0360
 AXS0370
 AXS0400
 AXS0410
 AXS0420
 AXS0430
 AXS0440
 AXS0450
 AXS0460
 AXS0470
 AXS0500
 AXS0510
 AXS0520
 AXS0530
 AXS0540
 AXS0550
 AXS0560
 AXS0570
 AXS0600
 AXS0610
 AXS0620
 AXS0630
 AXS0640
 AXS0650
 AXS0660
 AXS0670
 AXS0700
 AXS0710
 AXS0720
 AXS0730
 AXS0740
 AXS0750
 AXS0760
 AXS0770
 AXS1000
 AXS1010
 AXS1020
 AXS1030
 AXS1040
 AXS1050
 AXS1060
 AXS1070
 AXS1100

3 CALL A1
 4 GO CALL A1
 5 GO CALL A1
 6 GO CALL A1
 7 GO CALL C1
 8 GO CALL C1
 9 GO CALL C1
 10 GO CALL C1
 11 GO CALL C1
 12 GO CALL C1
 13 GO CALL C1
 14 GO CALL C1
 15 GO CALL C1
 16 GO CALL C1
 17 GO CALL C1
 18 GO CALL C1
 19 GO CALL C1
 20 GO CALL C1
 21 GO CALL C1
 22 GO CALL C1
 23 GO CALL C1
 24 GO CALL C1
 25 GO CALL C1
 26 GO CALL C1

APRK0223C
 APRK0224C
 APRK0225C
 APRK0226C
 APRK0227C
 APRK0228C
 APRK0229C
 APRK0230C
 APRK0231C
 APRK0232C
 APRK0233C
 APRK0234C
 APRK0235C
 APRK0236C
 APRK0237C
 APRK0238C
 APRK0239C
 APRK0240C
 APRK0241C
 APRK0242C
 APRK0243C
 APRK0244C
 APRK0245C
 APRK0246C
 APRK0247C
 APRK0248C
 APRK0249C
 APRK0250C
 APRK0251C
 APRK0252C
 APRK0253C
 APRK0254C
 APRK0255C
 APRK0256C
 APRK0257C
 APRK0258C
 APRK0259C
 APRK0260C
 APRK0261C
 APRK0262C
 APRK0263C
 APRK0264C
 APRK0265C
 APRK0266C
 APRK0267C
 APRK0268C
 APRK0269C
 APRK0270C
 APRK0271C
 APRK0272C
 APRK0273C
 APRK0274C
 APRK0275C
 APRK100C
 APRK101C
 APRK102C

```

1 IF (CNCE) 2,1,2
  CNCE = 1
  N = NI - 1
  CPT = 0
  P1 = 0.2516666666666667
  P2 = 0.4583333333333333
  P3 = 0.5416666666666667
  P4 = 0.7516666666666667
  C2 = 0.2083333333333333
  C3 = 0.4166666666666667
  KCUNT = 0
  N1 = N
  J2 = J1 + N1
  J3 = J2 + N1
  J4 = J3 + N1
  J5 = J4 + N1
  J6 = J5 + N1
  J7 = J6 + N1
  J8 = J7 + N1
  J9 = J8 + N1
  J10 = J9 + N1
  J11 = J10 + N1
  J12 = J11 + N1
  J13 = J12 + N1
  J14 = J13 + N1
  J15 = J14 + N1
  IF (CPT) 60,49,60
  IF (OPT) 60,49,60
  IF (DELT) 50,200,50
  IF (DELT) 50,200,50
  KOUNT=C
  STAFF FLNGE-KUTTA INTEGRATION.
  COMPUTE KC
  CONTINUE
  CO 25 I=1,N1
  K5=J15+1
  IF (SAGL(T(K5)) - NE. V(I+1)) T(K5) = V(I+1)
  IF (SAGL(T(K5)) - V(I+1)) T(K5) = V(I+1)
  24 T(K5) = V(I+1)
  25 CONTINUE
  IF (SAGL(TME) - NE. V(1)) TME = V(1)
  TME = V(1)
  26 KOUNT = KCUNT + 1
  27 CC 80 I=1,N1
  70 KO=J7+1
  
```

AMRK1030
 AMRK1040
 AMRK1050
 AMRK1060
 AMRK1070
 AMRK1100
 AMRK1110
 AMRK1120
 AMRK1130
 AMRK1140
 AMRK1150
 AMRK1160
 AMRK1170
 AMRK1200
 AMRK1210
 AMRK1220
 AMRK1230
 AMRK1240
 AMRK1250
 AMRK1260
 AMRK1270
 AMRK1300
 AMRK1310
 AMRK1320
 AMRK1330
 AMRK1340
 AMRK1350
 AMRK1360
 AMRK1370
 AMRK1400
 AMRK1410
 AMRK1420
 AMRK1430
 AMRK1440
 AMRK1450
 AMRK1460
 AMRK1470
 AMRK1500
 AMRK1510
 AMRK1520
 AMRK1530
 AMRK1540
 AMRK1550
 AMRK1560
 AMRK1570
 AMRK1600
 AMRK1610
 AMRK1620

```

      80  T(K0)=C(1)*D(I+1)
C
C
      90  COMPUTE K1
      TIME=0.5*C(1)
      V(1)=TIME
      DO 90 I=1,N1
      K0=J7+I
      K1=J11+I
      K2=J12+I
      K3=J13+I
      K5=J15+I
      T(1)=C(I+1)
      T(K1)=T(K2)
      T(K2)=T(K3)
      T(K3)=T(K5)
      T(K5)=T(K5) +0.5*T(K0)
      V(I+1)=T(K5)
      CALL AUXSLE
      DO 100 I=1,N1
      K1=J8+I
      100  T(K1)=C(1)*D(I+1)
C
C
      110  COMPUTE K2
      DO 110 I=1,N1
      K5=J15+I
      K1=J8+I
      K0=J7+I
      T(K5)=T(K5) +0.5*(T(K1)-T(K0))
      V(I+1)=T(K5)
      CALL AUXSLE
      DO 120 I=1,N1
      K2=J6+I
      120  T(K2)=C(1)*C(I+1)
C
C
      130  COMPUTE K3
      TIME=DELTT+TIME
      V(1)=TIME
      DO 130 I=1,N1
      K5=J15+I
      K2=J6+I
      K1=J8+I
      T(K5)=T(K5) +T(K2)-0.5*T(K1)
      V(I+1)=T(K5)
      CALL AUXSLE

```

```

DO 140 I=1,N1
K3=J10+I
T(K3)=C(I)*D(I+1)
CC
C
COMPLETE VALUE OF FUNCTION.
CC
DO 150 I=1,N1
K0=J7+I
K1=J8+I
K2=J9+I
K3=J10+I
K5=J15+I
T(K5)=T(K5)+O.16666666666666667*
1(T(KC))+T(K1)+T(K2)+T(K3)+T(K5)
V(I+1)=T(K5)
CC
CONTINUE
17C
CALL ACXSUB
CO J180 I=1,N1
K5=J1+I
K0=J2+I
K1=J3+I
K2=J4+I
K3=J5+I
K4=J6+I
T(K4)=T(K4)+(K3)
T(K3)=T(K3)+(K2)
T(K2)=T(K2)+(K1)
T(K1)=T(K1)+(K0)
T(KC)=T(KC)+(K5)
T(I)=T(I)+(I+1)
CONTINUE
180 CONTINUE
190 RETURN
CC
CC
DO ACAMS-MCULTON INTEGRATION
CC
CONTINUE .LT. 3) GO TO 60
IF (KCLNT - 3) 60,201,201
IF (KCLNT = KCLNT + 1
201 KOUNT = C(1)*.5
DO 210 I=1,N1
K0=J7+I
K1=J8+I
K2=J9+I
K3=J10+I

```



```

APRK317C
APRK320C
APRK321C
APRK322C
APRK323C
APRK324C
APRK325C
APRK326C
APRK327C
APRK330C
APRK331C
APRK332C
APRK333C
APRK334C
APRK335C
APRK336C
APRK337C
APRK340C
APRK341C
APRK342C
APRK343C
APRK344C
APRK345C
APRK346C
APRK347C
APRK350C
APRK351C
APRK352C
APRK353C
APRK354C
APRK355C
APRK356C
APRK357C
APRK360C
APRK361C
APRK362C
APRK363C
APRK364C
APRK365C
APRK366C
APRK367C
APRK370C
APRK371C
APRK372C
APRK373C
APRK374C
APRK375C
APRK376C

```

```

240      K5=J5+I
        T(K1)=T(K1)
        T(K2)=T(K2)
        C(1)=C(1)+C(1)
        KOUNT=4
        DELT=.5*C(1)
        GO TC 200

C      SET-UP FOR HALVING STEP SIZE.
C
C      CONTINUE
C      IF(KOUNT .LE. 4) GO TO 250
C      IF(KOUNT .GT. 4) 350,350,261
C      TIME=TIME-C(1)
C      V(1)=TAE
C      D(1)=DELT
C      DO 265 I=1,N1
C      K0=J7+I
C      K1=J1+I
C      K2=J2+I
C      K3=J3+I
C      K5=J15+I
C      T(K5)=T(KC)
C      V(I+1)=T(KC)
C      T(K3)=T(K2)+0.5*(T(K1)-T(K2))
C      T(K2)=T(K1)
C      T(K1)=T(K1)+0.5*(T(I)-T(K1))
C      KOUNT=4
C      GO TC 200

C      INTEGRATION IS FINISHED. SET UP DERIVATIVES AND EXIT.
C
C      DO 310 I=1,N1
C      K5=J15+I
C      K0=J8+I
C      T(K5)=T(K0)
C      V(I+1)=T(KC)
C      GO TC 170
C      CONTINUE

C      RETURN TO 3RD RK INTEGRATION AND RESTART
C
C      DO 360 I=1,N1
C      K5=J15+I
C      K1=J1+I
C      T(K5)=T(K1)

```

APRK277C
APRK4000
APRK401C
APRK402C
APRK403C
APRK404C
APRK405C

36C V(I+1)=T(K1)
TME=TME-C(1)*4.
V(1)=TME
V(1)=CELT
CALL AUXSUE
GO TC 5C
END

CC

SUBROUTINE PROCES
RETURN
END

CC

SUBROUTINE RESFT

COMMON C(2415)
EQUIVALENCE (C(2300),NOLIST),(C(2301),LISTNC(1)),(C(2351),
1 VALUE(1))
EQUIVALENCE (C(2675),KRUN)
DIMENSION LISTNG(50),VALUE(50)
IF(NOLIST.EQ.0) RETURN
DO 1 I=1,NCLIST
J=LISTAC(I)
1 C(I)=VALUE(I)
KRUN=KRUN+1
RETURN
END

CC

SUBROUTINE INPT1

COMMON C(2415)
COMMON/CFIS/ ENG(30),IENG(15)
NAMELIST /MCNT/ ENG,IENG
EQUIVALENCE (C(2675),KRUN)
FLG1=0.
IF (KRUN.GT.0) GO TC 100
READ (5,MCNT)
WRITE (6,MCNT)
RETURN
10C END

CC

SUBROUTINE INPT2

INPT001C

F 96

F 106
F 108

INPT001C

CLPT002C
CLPT001C
CLPT017C
CLPT003C
CLPT020C

CLPT0070
CLPT010C
CLPT011C
CLPT0120
CLPT013C
CLPT014C
CLPT0150
CLPT016C

CLPT0210
CLPT0220
CLPT023C
CLPT024C
CLPT025C
CLPT026C
CLPT027C

CLPT0340
CLPT0350
CLPT036C
CLPT040C
CLPT041C

```

SUBROUTINE CLPT3
  CUTPLT SUBROUTINE CLPT3
  DIMENSION B(50), OUTNO(50), ONAME1(50), ONAME2(50)
  DIMENSION C(2415), NAME3(150)
  COMMON C(2415)
  INTEGER CLTCNT, PGCNT, CUTNO(11),
  EQUIVALENCE (C(2500), CUTNO(11))
  EQUIVALENCE (C(2550), ONAME1(11))
  EQUIVALENCE (C(2600), ONAME2(11))
  EQUIVALENCE (C(2150), ONAME3(11))
  EQUIVALENCE (C(2660), DTCNT)
  EQUIVALENCE (C(2661), PGCNT)
  EQUIVALENCE (C(2662), PCNT)
  EQUIVALENCE (C(2667), ITCNT), (C(2668), PCNT), (C(2669), CPP)
  EQUIVALENCE (C(2932), T)
  EQUIVALENCE (C(2913), DER)
  EQUIVALENCE (C(2670), TAPE), (C(2671), TAPEND)
  EQUIVALENCE (C(10290), XTP), (C(10298), YTP), (C(10306), ZTP)
  EQUIVALENCE (C(3400), PUNCH)
  EQUIVALENCE (C(10521), TSET)
  EQUIVALENCE (C(10522), PST)
  EQUIVALENCE (C(10523), PHI)
  EQUIVALENCE (C(10554), AXECI)
  EQUIVALENCE (C(10555), AYECI)
  EQUIVALENCE (C(10556), AZECI)
  EQUIVALENCE (C(10577), VXECI)
  EQUIVALENCE (C(10552), VZECI)
  EQUIVALENCE (C(10547), XECI)
  EQUIVALENCE (C(10548), YECI)
  EQUIVALENCE (C(10549), ZECI)
  IF (ITCNT - GT. 6) GC TC 7
  IF (ITCNT - 6) 10, 10, 7
  1C ITCNT = ITCNT + 1
  WRITE (6, 6) (1, C(I), C(I+1), C(I+2), C(I+3), C(I+4), C(I+5), C(I+6),
  1 C(I+7), I=1, 2415, 8)
  6 FORMAT (1F11/15, 2X, RE14.7)
  PGCNT = 1
  7 CCNT INCL
  8 IF (IT - LT. PCNT) RETURN
  9 IF (IT - PCNT) 12, 9, 9
  12 REILRN
  15 PCNT = PCNT + CPP
  IF (PGCNT - NE. 1) GC TC 3
  IF (PGCNT - 1) 3, 1, 3
  1 WRITE (6, 2) (ONAME1(I), ONAME2(I), ONAME3(I), I = 1, NOOUT)

```


STGE064C
STGE065C
STGE066C
STGE067C
STGE070C
STGE071C
STGE072C
STGE073C
STGE074C
STGE075C
STGE076C
STGE077C
STGE100C
STGE101C

CNTR001C
CNTR002C
CNTR003C

CNTR001C
CNTR002C
CNTR003C

CNTR001C
CNTR002C
CNTR003C

RNDM001C
RNDM002C
RNDM003C

IF (K .NE. 0) END FILE K
IF (K .NE. 0) REWIND K
IF (K) 16,17,18
END FILE K
16 REWIND K
17 KSTEP = 2
IF (LCSTAT .EQ. NCSTAT) GO TO 4
IF (LCSTAT - NCSTAT) 18,4,18
18 DO 3 I = 1, NCSTAT
L = LCSTAT + 1
= STATNC(I)
= RMS(I) = C(N)
4 RETURN
4 END

SUBROUTINE CNTR1
RETURN
END

SUBROUTINE CNTR2
RETURN
END

SUBROUTINE CNTR3
RETURN
END

SUBROUTINE RADM1
RETURN
END

SUBROUTINE RADM2

COMMON C(3415)
EQUIVALENCE (C(2443), RNFLG)
EQUIVALENCE (C(2446), ZN)
EQUIVALENCE (C(2448), NOFNDM)
EQUIVALENCE (C(2449), DELT)
EQUIVALENCE (C(2450), RNCMNO)
EQUIVALENCE (C(2913), DEF)
DIMENSION RNDMND(50)
INTEGER RNDMND

```

3 IF (ACENDM) 2,2,3
  RNFLG=CE
  CELT=CE
  DO 1 I=1,ACRNDM
    J=RNPMAC(I)
    C(J+5)=2.718281828**(-DER*C(J+4))
    C(J+6)=C(J+5)*SQRT(1.0-C(J+5)*C(J+5))
    ZN=C(J+1)
    Y=C(J)
    N=INT(Y)
    CALL RFG(N,X)
    C(J+7)=C(J+3)*X
    C(J+1)=ZN
  2 RETURN
  3 END

```

CC C

SUBROUTINE RNDM3

```

COMMON C(3415)
EQUIVALENCE (C(2443), RNFLG)
EQUIVALENCE (C(2444), RN)
EQUIVALENCE (C(2446), ZN)
EQUIVALENCE (C(2448), NCRNDM)
EQUIVALENCE (C(2449), DELT)
EQUIVALENCE (C(2450), RNDMAC)
EQUIVALENCE (C(2913), DER)
DIMENSION RNDMND(50)
INTEGER RNDMND
IF (ACENDM) 20,20,10 GC TC 8
IF (RNFLG=CE.1.0) GO TC 6
IF (RNFLG=CE.0.0) GO TC 6
J=RNPMAC(I)
X=(C(J+7)-C(J+5)*C(J+8))/C(J+6)
IF (RNFLG=CE.1.0) GC TC 5
RNFLG=-1-ER
GO TC 7
DELT=DELT+DER
C(J+5)=2.718281828**(-DELT*C(J+4))
C(J+6)=C(J+5)*SQRT(1.0-C(J+5)*C(J+5))
C(J+7)=C(J+6)*X+C(J+8)
RETURN
IF (CELT=EC.DER) GC TC 4
DO 3 I=1,ACRNDM
  J=RNPMAC(I)

```

10 8

7 5 4 2

```

3 C(J+5)=2.718281828*(-DER*C(J+4))
C(J+6)=C(J+3)*SQRT(1.0-C(J+5)*C(J+5))
DEL=DER
4 DO 1 I=1,NCRNDM
J=RNCRND(I)
Y=C(J)
ZN=C(J+1)
N=INT(V)
CALL RFG(N,X)
C(J+8)=C(J+7)
C(J+7)=C(J+6)*X+C(J+5)*C(J+7)
1 C(J+1)=ZN
20 RETURN
END

```

CC

```

SUBROUTINE AUXA1
RETURN
END

```

ALXA0010
ALXA0020
ALXA0030

CC

```

SUBROUTINE AUXA2
RETURN
END

```

CC

```

SUBROUTINE AUXA3
RETURN
END

```

CC

```

SUBROUTINE AUXB1
RETURN
END

```

ALXB0010
ALXB0020
ALXB0030

CC

```

SUBROUTINE AUXB2
RETURN
END

```

ALXB0010
ALXB0020
ALXB0030

CC

```

SUBROUTINE AUXB3
RETURN
END

```

ALXB0010
ALXB0020
ALXB0030

CC

```

SUBROUTINE AUXC1
RETURN

```

ALXC0010
ALXC0020

ALXC003C
ALXC001C
ALXC002C
ALXC003C
ALXC001C
ALXC002C
ALXC003C

```

END
SUBROUTINE ALXC2
RETURN
END
SUBROUTINE AUXC3
RETURN
END
SUBROUTINE RFG(N,X)
COMMON C(2415)
EQUIVALENCE (C(2446), ZN)
Y=FLCAT(N)
X=0.0
GENERATE UNIFORM RANCOV NO.(0 TO 1)
SUM FCR RAND NG(SIGMA=1,MEAN=0)
LN:FCR+ DIST FOR N=1
NORMAL DIST FOR LARGE N (12 OR GREATER)
DO 1 I=1,N
X=X+ZN-0.5
RNC=557.*ZN
ZN=RND-AINT(RNC)
1 X=SCRT(12./Y)*X
RETURN
END

```

```

SUBROUTINE AII
AERODYNAMIC FORCES AND MOMENTS INITIALIZATION MODULE AII
PCY AXES
COMMON C(2415)
EQUIVALENCE (C(02CE), TSA )
EQUIVALENCE (C(0257), CTS )
EQUIVALENCE (C(025E), STS )
CTS = (CTS(TSA)
STS = SIN(TSA)
RETURN
END

```


END

SUBROUTINE A2

CAB - ZERC-LIFT BODY DRAG
CAC - DRAG DUE TO DEFLECTED CONTROL SURFACES
CAE+CAC
CZB - NORMAL FORCE DUE TO ANGLE OF ATTACK
CZC - NORMAL FORCE DUE TO FIN DEFLECTION
CYB - SIDE FORCE DUE TO SIDESLIP
CYC - SIDE FORCE DUE TO FIN DEFLECTION
CY - SIDE FORCE
CNG - PITCH LAMPING MOMENT
CAR - YAW LAMPING MOMENT
CLP - ROLL LAMPING MOMENT
CMB - PITCH LAMPING DUE TO ANGLE OF ATTACK
CNC - CONTROL SURFACE PITCH MOMENT
CMA+CNC
CAB - YAW LAMPING DUE TO SIDESLIP
CNC - CONTROL SURFACE YAW MOMENT
CNA+CNC
CLB - INDUCED ROLL MOMENT
CNC - CONTROL SURFACE ROLL MOMENT
CLB+CLC
CLC

COMMCN/APPLE/TM1(7), TM2(15), TM3(14), TM4(5), TETA1(7), TETA2(7),
1TETA3(6), TX11(2), TX12(10), TX13(5), TDELTA(4), TH1(3), TH2(2),
2TCZBC(1,2), TDCZP(14,7), TCYBC(7,10,6), TCMBC(7,5,7), TDCMF(14,7),
3TCZBC(15,2), TDC(15,2), TAD2(15), TCN8C(7,10,6), TCM01(15,4),
4TCMG2(15,4), TDCM0(15), TDCMCA(15), TPCMD(14,3), TCMQ(15), TPQ(14,3),
5TCLBC(7,10,6), TPLD(15), TPLD(14,3), TCLP(15), TPC(14,3), TDCYSP(3),
LTCCZSF(3), TDCMSP(3), TCCNSP(3), TH3(3), TFEJ(13), TFEJ(13)
COMMCN C(1,2415)
EQUIVALENCE (C(0016), DP)
EQUIVALENCE (C(0020), CC)
EQUIVALENCE (C(0024), CR)
EQUIVALENCE (C(0111), CA)
EQUIVALENCE (C(0113), CY)
EQUIVALENCE (C(0114), XI)
EQUIVALENCE (C(0115), CZ)
EQUIVALENCE (C(0116), CBAF)
EQUIVALENCE (C(0117), ETA)
EQUIVALENCE (C(0118), CMG)
EQUIVALENCE (C(0119), CNR)
EQUIVALENCE (C(0120), CLP)

[illegible]

```

VN=SQRT(VEA**2+WBA**2)
IF (VA-LT.C.03001) GT TC 10
XIR=ATAN2(VEA,WBA)
GO TC 20
1C XIR=C.C
20 XI=XIR/RAD
1C XI=XI
IF (XI.LT.C.C) XI1=-XI
XI2=XI1
IF (XI1.GE.90.) XI2=180.-XI1
C TABLE LOCK-UP,ALL PHASES
LCZP=0.0
CCMP=C.C
PMC=1.0
FNC=1.0
PLC=1.0
PQ=1.0
PR=1.0
PP=1.0
CCASEF=C.
CCLSEF=C.
CCYSEF=C.
CCZSEF=C.
CCLSP=C.
CCASP=C.
CCYSP=C.
CCMENT CFFICIENTS
FITTING MENT
CMRCC=TRFEDL(AM,XI2,ETA,TM1,TX13,TETAL,TCMBC,7,5,7)
CCMUL=CLLI(AM,TM2,TCMD,15)
CCMUL=-CCMUL
CCMLR=-CCMUL
CCMLA=CLLI(AM,TM2,TCMDA,15)
CCMURA=-CCMULA
CCMLLA=-CCMULA
CCMLRA=CCMULA
CMG=CLLI(AM,TM2,TCMG,15)
YARING MENT
CNBC=TRFEDL(AM,XI2,ETA,TM1,TX12,TETA3,TCNBC,7,10,6)
IF (XI1.GT.90.) CNBC=-CNBC
IF (XI1.LT.0.0) CNBC=-CNBC
CCNUL=-CCMUL
CCNULF=CCMUL
CCNLL=-CCMUL
CCNLR=CCMUL
CCNULB=CCMLA

```

AD-A085 147

NAVAL POSTGRADUATE SCHOOL MONTEREY CA
GUIDANCE AND CONTROL OF TACTICAL MISSILES.(U)
DEC 79 T A GROTE

F/G 16/2

UNCLASSIFIED

NL

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

2-2

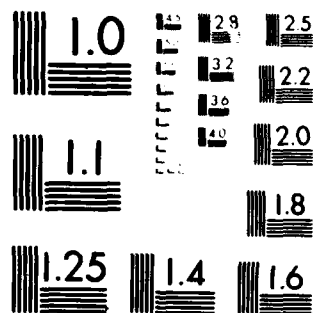
2-2

2-2

2-2

2-2

END
DATE
FILMED
7-80
DTIC



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

```

CCNUFB=-CCMULA
CCNLFB=-CCMULA
CCNLRB=CCMULA
C
RCLLIAG MCMENT
CLBC=TFRECL(AM,XI2,ETA,TM1,TXI2,TETA3,TCLBC,7,10,6)
IF (XI1.GT.90.) CLBC=-CLBC
IF (XI1.LT.0.) CLBC=-CLBC
CLC=COLI(AM,TM2,TCLD,15)
CLF=CLLI(AM,TM2,TCLF,15)
FCRCE CCEFFICIENTS
CZBCC=TFRECL(AM,XI2,ETA,TM1,TXI2,TETA3,TCLBC,7,2,7)
CYBCC=TFRECL(AM,XI2,ETA,TM1,TXI2,TETA3,TCLBC,7,10,6)
IF (XI1.GT.50.) CYBCC=-CYBCC
IF (XI1.LT.0.) CYBCC=-CYBCC
CAC2=CLLI(AM,TM2,TCLD,15)
IF (T.GE.T3) GO TO 100
TAELE LICK-LF SEP AND BOOST PHASE
C
ZERC-LIFT CRAG
CDO=STLLIA(TM2,TH2,TCD8,AM,H,15,2)
IF (T.LT.T1) CDO=CCC+.08
C
CENTRCL SLRFACE EFFECTIVENESS
IF (DUL.LT.0.0) CMLIC=-CMLD
CMURC=STDLIA(TM2,TDEL,TCDI,AM,ADUR,15,4)
IF (CUR.GT.0.0) CMURC=-CMURC
CMLLC=STDLIA(TM2,TDEL,TCDI,AM,ADLL,15,4)
IF (CLL.LT.0.0) CMLLC=-CMLLC
CMLRC=STDLIA(TM2,TDEL,TCDI,AM,ADLR,15,4)
IF (CLR.GT.C.0) CMLRC=-CMLRC
FLUME EFFECTS
IF (AM.GE.1.6) GO TC 200
POK=F/1000
FM=2.1212E-06*HOK**3-5.2017E-04*HOK**2+.044193*HOK-9.0905E-04
DCMP=STCLIA(TM3,TETA2,TDCMP,AM,ETA,14,7)
IF (DCMP.LT.C.0) DCMP=C.0
LC2P=STCLIA(TM3,TETA2,TDC2P,AM,ETA,14,7)
IF (DC2P.GT.0.0) DC2P=C.0
DCMP=DC2P*FM
DCZF=STLLIA(TM3,TH1,TFMC,AM,H,14,3)
IF (FMC.GT.1.0) PMC=1.0
FNC=FMC
PLD=STLLIA(TM3,TH1,TFLC,AM,H,14,3)
IF (FLL.GT.1.0) PLL=1.0
FQ=STCLIA(TM3,TH1,TFC,AM,H,14,3)
IF (FQ.GT.1.0) PQ=1.0
FR=PC

```

```

C
PP=STGLIA(TM3,TH1,TFF,AM,H,14,3)
IF (PP.GT.1) EFFECTS
SEPARATESEP=LE.O.) GC TC 200
IF (T.GE.1.0) GC TC 200
IF (T.GE.1.0) GC TC 200
FS2=(1.-ZTF/15.) *EXP(-ZTF/15.)
DCMSFF=CCLH(H,TH3,TECMSP,3)
DCZSFF=CCLH(H,TH3,TECZSP,3)
DCZSFF=FS2+ECZSPP
IF (ZTF.GE.4.0) GC TC 30
FS1=(1.-ZTF/4.) *EXP(-ZTF/4.)
DCVSFF=CCLH(H,TH3,TECYSP,3)
DCNSFF=ODLI(F,TH3,TECNSP,3)
DCVSFF=FS1+DCYSPP
DCNSFF=FS1+DCNSPP
CCLSF=-13*FS1
CPFI=CCS(FPI)
ECZSEP=ECZSP*CPHI+DCYSP*SPHI
DCYSEP=DCYSP*CPHI+DCNSP*SPHI
DCNSSEP=DCNSP*CPHI+DCMSF*SPHI
GO TO 100
TABLE CRUISE PHASE
ZERR=LEFT CRAG
100 CCONTR=STGLIA(TM2,TH2,TCC,AM,H,15,2)
CMUL=STGLIA(TM4,TDELT,TCMD2,AM,ADUL,5,4)
IF (CMUL.LT.0.0) CMUL=-CMUL
CMURC=STGLIA(TM4,TDELT,TCMD2,AM,ADUR,5,4)
IF (CMURC.GT.0.0) CMURC=-CMURC
IF (CCL.LT.0.0) CMLLC=-CMLLC
IF (CCLR.GT.0.0) CMLRC=-CMLRC
IF (CCLC.GT.0.0) CMLFC=-CMLFC
CCLC=CCLC+0.017
CCLC=CMULC+LCKK-UP
200 CMLLC=CMULC
CMURC=-CMURC
CMLLC=-CMLLC
CMURC=CMURC+DCZP
CMEC=CMURC+ECMP
CABC=CCLC+CCSETA
C TRANSFORM FROM CROSS-FLCW TO BODY AXIS COMPONENTS

```

```

CXI=C(XIR)
SXI=S(XIR)
CAE=CAEC
CYE=CYEC
CLE=CLBC
CMB=CNBC
CNTRCL=CNTRCL
F=(15.25-CC)/15.25
CMUL=CMULC+DCMURA*ALF*CLUL+DCMUL*ADUL
CMUR=CMURC+DCMURA*ALF*CLUR+DCMUR*ADUR
CMLL=CMLLC+DCMLLA*ALP*CLL+DCMLL*ADLL
CMLR=CMLRC+DCMLRA*ALP*CLLR+DCMLR*ADLR
CMC=(CMUL+CMUR+CMLL+CLLR)*F*PMD
CNUL=CNULC+DCNULB*BET*CLUL+DCNUL*ADUL
CNUR=CNURC+DCNURB*BET*CLUR+DCNUR*ADUR
CNLL=CNLLC+DCNLLB*BET*CLL+DCNLL*ADLL
CNLR=CNLRC+DCNLRB*BET*CLLR+DCNLR*ADLR
CNC=(CNCUL+CNUR+CNLL+CLLR)*F*PND
CLUL=CLULC+DUL
CLUR=CLURC+DUR
CLL=CLLC+DLL
CLLR=CLLRC+DLR
CLC=(CLUL+CLUR+CLL+CLLR)*PLD
CZC=-CNC*CBAR/(15.25-CC)
CYC=-CNC*CEAR/(15.25-CC)
CAG=CAEZ*(CUL**2+DUR**2+CLL**2+DLR**2+4.*1.3*(ALP*DELO-BET*DELR))
C TOTAL AERODYNAMIC FORCE COEFFICIENTS
CA=CAE+CAC
CY=CYE+CYC
CZ=CZE+CZC
CZ=-CZ
C TOTAL AERODYNAMIC MOMENT COEFFICIENTS
CL=CLC+CLC+CLLSEP
CLF=CLF+CLF+CLF+CLF
CM=CMC+CMC+CMC+CMC*(CG-CCR)/CBAR+CCMSEP
CMC=CMC+CMC+CMC+CMC
CNR=CMR+CMR+CMR+CMR*(CG-CCR)/CBAR+CCNSEP
RETURN
END

```

CC

C

C

CCCC

```

SUBROUTINE A31
COMMON C(3415)
COMMON /FLAG/ FLG1
EQUIVALENCE (C(0601), FF1)
EQUIVALENCE (C(0602), WEIG)
EQUIVALENCE (C(0603), IA)
EQUIVALENCE (C(0604), FLG2)
FF1=C.C
WEIG=0.C
IA=0
FLG1=C.C
FLG2=C.C
RETURN
END

```

CC C C C C C C C

SUBROUTINE A3

THIS IS THE MISSILE PREPULSION MODULE

```

T1=BOOST ENGINE IGNITION TIME
BT=BOOST ENGINE BURNCUT TIME(=T3)
BE=CRUISE ENGINE IGNITION TIME(=T4)

```

```

COMMON /FLAG/ FLG1
COMMON C(3415)
COMMON C(0086) 2) WEIGHT )
EQUIVALENCE (C(0110), T1)
EQUIVALENCE (C(0932), BT)
EQUIVALENCE (C(0933), T1)
EQUIVALENCE (C(0935), BT)
EQUIVALENCE (C(0936), RE)
EQUIVALENCE (C(0136), CG)
EQUIVALENCE (C(0150), CLT)
EQUIVALENCE (C(0151), CMT)
EQUIVALENCE (C(0152), CNT)
EQUIVALENCE (C(0153), EPSTH)
EQUIVALENCE (C(0154), PHITH)
EQUIVALENCE (C(0201), A)
EQUIVALENCE (C(0202), PC)
EQUIVALENCE (C(0203), TXEA)
EQUIVALENCE (C(0073), TYEA)
EQUIVALENCE (C(0074), TZE)
EQUIVALENCE (C(0075), F)
EQUIVALENCE (C(0507), AMACH)
EQUIVALENCE (C(0520), ETA)
EQUIVALENCE (C(0117),

```

```

EQUIVALENCE (C(0601), FF1)
EQUIVALENCE (C(0602), WFIG)
EQUIVALENCE (C(0603), IA)
EQUIVALENCE (C(0604), FLG2)
EQUIVALENCE (C(0605), DIFFM)
EQUIVALENCE (C(0606), FLGRJ)
EQUIVALENCE (C(0508), CD)
EQUIVALENCE (C(2913), CEF)
DATA WFLLEL/189./

C 80 FORMAT(1H,10X,'TOTAL ANGLE OF ATTACK EXCEEDED, ETA=',F8.3)
C
IF (FLG1.GT.0.0) GO TO 1C
CT = 0.0
C DIFFM = C.0
FF = 0.0
TXBA = C.0
TYBA = C.0
TZBA = 0.0
IF (T.LI.11) RETURN
PIN = P*12.4.0254
TIN = T
AIN = ETA
IF (T.CE.BE) GO TO 1
C BCCST P-ASE
IT = T
IF (T.GT.ET) GO TO 10
CALL BCCST (T, THRU, FF)
TXBA = THRU*CS(EPSTF)
TNEA = THRU*CS(EPSTF)
TYBA = -TNEA*SIN(PHITF)
TZBA = TNEA*CS(PHITF)
ID = 0
WEIGHT = WEIGHT-FF+FF1
CG = CG - (FF-FF1)/405.24
A = A - (FF-FF1)/256.25
B = B - (FF-FF1)/1.881
CC = B
CLT = C
CMT = TZBA*(15.525-CG)
CNT = -TYEA*(15.525-CG)
FF1 = FF
TW = T+DER
IF (T.LI.11) GO TO 2
IF (FLG2.GT.0.0) GO TO 2
WEIGHT = WEIGHT-31.
CG = CG -.15
FLG2 = 1.0

```

```

10 CONTINUE
   IF (FLGRJ.GT.0.0) GC TC 21
   CRUISE FFASE, RAMJET CFF
   IF (IB.GE.1) GO TO 11
   CALL ENGINE (PIN, AMACH, AIN, 0., TIN, 1.0, 0.0, 0., CT, SMARG, -1, 1)
   IB=1
   IF (AIN.GT.9.3) AIN=9.3
   CALL ENGINE (HIN, AMACH, AIN, 0., TIN, 1.0, 0.0, 0., CT, SMARG, 1, 1)
   GO TC 4
   C SIMPLIFIED RAMJET MODEL
   CRUISE FFASE, RAMJET CFF
   21 CALL RAMJET (HIN, AMACH, AIN, FF, 1.0, CT)
   GO TC 4
   1 CONTINUE
   IF (FLGRJ.GT.0.0) GC TC 22
   CRUISE FFASE, RAMJET CN
   IB=0
   IF (AIN.GT.10.0) GO TC 5
   FF=FF1
   IF (IA.GE.1) GO TC 2
   CALL ENGINE (HIN, AMACH, AIN, FF, TIN, 0.0, 0.0, 0.0, CT, SMARG, -1, 1)
   TT=1
   FF1=FF
   IA=1
   XIA=0
   IF (WEIG.GT.WFUEL) XIN=1.0
   CALL ENGINE (HIN, AMACH, AIN, FF, TIN, XIN, 0.0, 0.0, CT, SMARG, 1, 1)
   GO TC 4
   C SIMPLIFIED RAMJET MODEL
   CRUISE FFASE, RAMJET CN
   22 IF (AIN.GT.10.0) GC TC 5
   XIN=0
   IF (WEIG.GT.WFUEL) XIN=1.0
   CALL RAMJET (HIN, AMACH, AIN, FF, XIN, CT)
   GO TC 4
   5 IF (ELG1.GT.0.0) GC TC 10
   WRITE (C, 2) ETA
   GO TC 10
   4 TXPA=CT*CD+S
   TYEA=0.
   TZBA=C
   CT=T-T1
   IF (FLGRJ.GT.0.0) GC TC 6
   FF=FF/.45259237
   C DIFFUSER (INLET) MARGIN IS NOT COMPUTED IN SIMPLIFIED RAMJET MODEL
   C DIFFN=SMARG/100.

```

```

6  WFIG=WEIG+(ABS(FF+FF1))*CT/2.0
  IF (WEIG.GT.WFUEL) FLG1=1.0
  WEIGHT = WEIGHT - ABS(FF + FF1)*DT/2.
  TT=T
  FFI = FF
  CG=CG-.0026455*ABS(FF+FF1)*DT/2.
  A=A-.0031746*ABS(FF+FF1)*DT/2.
  B=B-.003452*ABS(FF+FF1)*CT/2.
  CC = B
  RETURN
2  ENC

```

CC C C C

SUBROUTINE A4I ACTUATOR INITIALIZATION MODULE A4I

```

COMMON C(2415)
DIMENSION IPL(1071)
EQUIVALENCE (C(1735),WFIN)
EQUIVALENCE (C(3315),N)
EQUIVALENCE (C(3316),IPL(1))
IF (WFIN.GT.C.0) GO TO 100
GO TO 200
100 IPL(N+1)=1653
    IPL(N+2)=1657
    IPL(N+3)=1701
    IPL(N+4)=1705
    IPL(N+5)=1709
    IPL(N+6)=1713
    IPL(N+7)=1717
    IPL(N+8)=1721
    A=N+8
    RETURN
200 ENC

```

100

200

CC C C C

SUBROUTINE A4 FIN ACTUATORS MODULE A4

```

COMMON C(2415)
EQUIVALENCE (C(1016),CPI)
EQUIVALENCE (C(1020),CC)
EQUIVALENCE (C(1024),DRI)
EQUIVALENCE (C(1693),CULC)

```

103

```

IF (ABS(C2).GE.RTLM) GC TO 130
GO TC 140
13C PRCD2=C2D*C2
IF (FRCD2.GE.0.0) C2D=0.0
C 140 LOWER=LEFT FIN RESPONSE
C3C=WFIN*(DLLC-CLL)-2.*DFIN*WFIN*D3
IF (ABS(C3).GE.RTLM) GC TC 150
GO TC 160
150 FRCC3=C3E*C3
IF (FRCC3.GE.0.0) C3C=0.0
C 160 LOWER=LEFT FIN RESPONSE
D4D=WFIN*(DLRC-CLR)-2.*DFIN*WFIN*D4
IF (ABS(C4).GE.RTLM) GC TO 170
GO TC 180
17C PRCD4=C4C*C4
IF (FRCD4.GE.0.0) C4D=0.0
18C GO TO 220
C *** IDEAL ACTUATORS ***
20C CUL=DULC
CUH=CUFC
CLL=CLLC
CLR=DLFC
C FIN ANGLE LIMITERS
IF (ABS(CLL).GT.DMAX) CUL=SIGN(DMAX,DUL)
IF (ABS(CLR).GT.DMAX) CLR=SIGN(DMAX,CLR)
IF (ABS(CLL).GT.DMAX) CLL=SIGN(DMAX,DLL)
IF (ABS(CLR).GT.DMAX) CLR=SIGN(DMAX,CLR)
C PITCH YAW ROLL CHANNEL INPUTS
220 CQ=0.25*(CUL-CLR-CLL)
DR=0.25*(CUL+CLR-CLL)
CP=0.25*(CUL+CLR+CLL)
CULCEG=CUL*57.29578
CLRCEG=CLR*57.29578
RETURN
ENG
C
C SUBROUTINE A51
RETURN
END
C
C SUBROUTINE A5
RETURN
END

```

```

SUBROUTINE C11
INITIALIZATION MODULE FOR LVRJ AUTOPILOT

COMMON C(3415)
EQUIVALENCE (C(3315),N)
DIMENSION IPL(100)
IPL(N)=1677
IPL(N+1)=1689
IPL(N+2)=1725
IPL(N+3)=1725
N=N+4
RETURN
END

```

```

SUBROUTINE C1
SUBROUTINE C1-LVRJ
THIS AUTOFILCT MODULE IS FOR STV G

COMMON C(3415)
EQUIVALENCE (C(3315),N)
EQUIVALENCE (C(3316),IPL(1))
EQUIVALENCE (C(3317),IPL(2))
EQUIVALENCE (C(3318),IPL(3))
EQUIVALENCE (C(3319),IPL(4))
EQUIVALENCE (C(3320),IPL(5))
EQUIVALENCE (C(3321),IPL(6))
EQUIVALENCE (C(3322),IPL(7))
EQUIVALENCE (C(3323),IPL(8))
EQUIVALENCE (C(3324),IPL(9))
EQUIVALENCE (C(3325),IPL(10))
EQUIVALENCE (C(3326),IPL(11))
EQUIVALENCE (C(3327),IPL(12))
EQUIVALENCE (C(3328),IPL(13))
EQUIVALENCE (C(3329),IPL(14))
EQUIVALENCE (C(3330),IPL(15))
EQUIVALENCE (C(3331),IPL(16))
EQUIVALENCE (C(3332),IPL(17))
EQUIVALENCE (C(3333),IPL(18))
EQUIVALENCE (C(3334),IPL(19))
EQUIVALENCE (C(3335),IPL(20))
EQUIVALENCE (C(3336),IPL(21))
EQUIVALENCE (C(3337),IPL(22))
EQUIVALENCE (C(3338),IPL(23))
EQUIVALENCE (C(3339),IPL(24))
EQUIVALENCE (C(3340),IPL(25))
EQUIVALENCE (C(3341),IPL(26))
EQUIVALENCE (C(3342),IPL(27))
EQUIVALENCE (C(3343),IPL(28))
EQUIVALENCE (C(3344),IPL(29))
EQUIVALENCE (C(3345),IPL(30))
EQUIVALENCE (C(3346),IPL(31))
EQUIVALENCE (C(3347),IPL(32))
EQUIVALENCE (C(3348),IPL(33))
EQUIVALENCE (C(3349),IPL(34))
EQUIVALENCE (C(3350),IPL(35))
EQUIVALENCE (C(3351),IPL(36))
EQUIVALENCE (C(3352),IPL(37))
EQUIVALENCE (C(3353),IPL(38))
EQUIVALENCE (C(3354),IPL(39))
EQUIVALENCE (C(3355),IPL(40))
EQUIVALENCE (C(3356),IPL(41))
EQUIVALENCE (C(3357),IPL(42))
EQUIVALENCE (C(3358),IPL(43))
EQUIVALENCE (C(3359),IPL(44))
EQUIVALENCE (C(3360),IPL(45))
EQUIVALENCE (C(3361),IPL(46))
EQUIVALENCE (C(3362),IPL(47))
EQUIVALENCE (C(3363),IPL(48))
EQUIVALENCE (C(3364),IPL(49))
EQUIVALENCE (C(3365),IPL(50))
EQUIVALENCE (C(3366),IPL(51))
EQUIVALENCE (C(3367),IPL(52))
EQUIVALENCE (C(3368),IPL(53))
EQUIVALENCE (C(3369),IPL(54))
EQUIVALENCE (C(3370),IPL(55))
EQUIVALENCE (C(3371),IPL(56))
EQUIVALENCE (C(3372),IPL(57))
EQUIVALENCE (C(3373),IPL(58))
EQUIVALENCE (C(3374),IPL(59))
EQUIVALENCE (C(3375),IPL(60))
EQUIVALENCE (C(3376),IPL(61))
EQUIVALENCE (C(3377),IPL(62))
EQUIVALENCE (C(3378),IPL(63))
EQUIVALENCE (C(3379),IPL(64))
EQUIVALENCE (C(3380),IPL(65))
EQUIVALENCE (C(3381),IPL(66))
EQUIVALENCE (C(3382),IPL(67))
EQUIVALENCE (C(3383),IPL(68))
EQUIVALENCE (C(3384),IPL(69))
EQUIVALENCE (C(3385),IPL(70))
EQUIVALENCE (C(3386),IPL(71))
EQUIVALENCE (C(3387),IPL(72))
EQUIVALENCE (C(3388),IPL(73))
EQUIVALENCE (C(3389),IPL(74))
EQUIVALENCE (C(3390),IPL(75))
EQUIVALENCE (C(3391),IPL(76))
EQUIVALENCE (C(3392),IPL(77))
EQUIVALENCE (C(3393),IPL(78))
EQUIVALENCE (C(3394),IPL(79))
EQUIVALENCE (C(3395),IPL(80))
EQUIVALENCE (C(3396),IPL(81))
EQUIVALENCE (C(3397),IPL(82))
EQUIVALENCE (C(3398),IPL(83))
EQUIVALENCE (C(3399),IPL(84))
EQUIVALENCE (C(3400),IPL(85))
EQUIVALENCE (C(3401),IPL(86))
EQUIVALENCE (C(3402),IPL(87))
EQUIVALENCE (C(3403),IPL(88))
EQUIVALENCE (C(3404),IPL(89))
EQUIVALENCE (C(3405),IPL(90))
EQUIVALENCE (C(3406),IPL(91))
EQUIVALENCE (C(3407),IPL(92))
EQUIVALENCE (C(3408),IPL(93))
EQUIVALENCE (C(3409),IPL(94))
EQUIVALENCE (C(3410),IPL(95))
EQUIVALENCE (C(3411),IPL(96))
EQUIVALENCE (C(3412),IPL(97))
EQUIVALENCE (C(3413),IPL(98))
EQUIVALENCE (C(3414),IPL(99))
EQUIVALENCE (C(3415),IPL(100))

```



```

S2=0.0
AKCI=AKCI1
AKFI=AKFI1
AKPI=AKPI1
GO TC 16
C CRUISE ALTICFILCT
1 CONTINUE
S1=0.0
S2=1.0
AKCI=AKCI2
AKFI=AKFI2
AKPI=AKPI2
16 CONTINUE
C FITCH AND YAW CHANNELS
GI=AKG*(CN-S1*QC)
RI=AKR*RN
ANZM=-A2N/G
ANYM=AYN/G
ANZ=ANZC-ANZM
ANY=ANYC-ANYM
POS DCC AND DRC COMMAND NEG TURNING MOMENTS
DCCI=GI-S2*AKNZ*ANZ
ZID=CCC1
IF (ABS(CCC1-GE.DMAX) ZID=0.0
LQC=CCC1+AKCI*Z1
IF (ABS(CCC1-GE.DMAX) DCC=SIGN(DMAX,DCC)
CRC1=RI-S2*AKNY*ANY
YID=CCC1
IF (ABS(CRC1-GE.DMAX) YID=0.0
LRC=CCC1+AKRI*Y1
IF (ABS(CRC1-GE.DMAX) DRC=SIGN(DMAX,DRC)
2 CONTINUE
C ROLL CHANNEL
PLD=AKFI*FN
LAG CCPENSATOR
CMG1=.1
CMG2=.1
LPCD=-AKFI*(FN+P1)*QMG2/CMG1
LPCIC=(CMG1-CMG2)*CFCE-CMG2*DPCI
DPC=CFCI+DPCD
IF (ABS(DPC)-GE.DMAX) DPC=SIGN(DMAX,DPC)
RETURN
END
C C C
SUBROUTINE C21
C

```

```

DIMFASICA IPL(100)
COMMON C(3415)
EQUIVALENCE (C(0507),F)
EQUIVALENCE (C(3315),N)
EQUIVALENCE (C(3316),IPL(1))
IPL(N)=1633
IPL(N+1)=1637
N=N+2
HNCK=H
RETURN
END

```

CC C

SUERCLTINE C2
GLIDANCE CCMAND MODULE

```

COMMON C(3415)
EQUIVALENCE (C(0256),A33)
EQUIVALENCE (C(0286),VXTP)
EQUIVALENCE (C(0294),VYTP)
EQUIVALENCE (C(0302),VZTP)
EQUIVALENCE (C(0337),CMGYC)
EQUIVALENCE (C(0338),CMGZC)
EQUIVALENCE (C(0317),FLGT)
EQUIVALENCE (C(0335),XMA)
EQUIVALENCE (C(0340),YMA)
EQUIVALENCE (C(0341),ZMA)
EQUIVALENCE (C(0520),AMACH)
EQUIVALENCE (C(0500),ACC)
EQUIVALENCE (C(0501),PEF)
EQUIVALENCE (C(0502),HD)
EQUIVALENCE (C(0507),F)
EQUIVALENCE (C(0508),CF)
EQUIVALENCE (C(0515),PSF)
EQUIVALENCE (C(0527),GAT)
EQUIVALENCE (C(0528),VT)
EQUIVALENCE (C(0531),VF)
EQUIVALENCE (C(0581),CYNPI)
EQUIVALENCE (C(0582),AKG)
EQUIVALENCE (C(0932),T)
EQUIVALENCE (C(0933),T1)
EQUIVALENCE (C(0933),T2)
EQUIVALENCE (C(0933),T3)
EQUIVALENCE (C(0937),T5)
EQUIVALENCE (C(0942),TLC)
EQUIVALENCE (C(0944),TLC)
EQUIVALENCE (C(0945),TLC)
EQUIVALENCE (C(0945),TLC)

```

```

EQUIVALENCE (C11633), ANZCD)
EQUIVALENCE (C11636), ANZCI)
EQUIVALENCE (C11637), ANYCD)
EQUIVALENCE (C11640), ANYC)
EQUIVALENCE (C11750), WFI)
DATA RCCN/36500./
CYNP=CC/144.
C ZERC MODE CONTROL SWITCHES
S4=0.0
S5=C.C
S6=0.0
S8=0.0
IF (T.GE.T3) GO TC 1
IF (A.ACF.GE.2.0) GC TC 1
GO TC 100
C ZERC TRAJECTRY
1 IF (T.GE.T5) GO TO 2
GO TC 100
C LEVEL FLIGHT
2 IF (T.GE.TCC) GO TO 5
S4=1.0
S5=1.0
GO TC 100
C DIVE CLIMB
5 IF (T.GE.TLC) GO TO 4
7 PDCL=HEC
S4=1.0
S5=1.0
S8=1.0
GO TC 100
C ALTITUDE TO HREF
4 FORI2=SCRT(XMA**2+YMA**2)
DI=1.575*VF+RCCN-PCRIZ
IF (DI.GE.C.C) GO TC 6
S4=1.0
S5=1.0
S6=1.0
GO TC 100
C CIVE
CIVE (XMA*VZTP)/(XMA*VXTP+ZMA*VZTP)
W3=(ZMA*V)TP-2900.*W3/32.172+A33
ANZCI=-2.
IF (FLGT.GT.0.0) GC TC 8
GO TC 100
C TERMINAL PCPING
TERMINAL PCPING
ANZCI=ANGC*VT*CMGYC/22.174
GO TC 100
C CONTINUE
100 CONTINUE

```

```

C **** VERTICAL CHANNEL ****
C FERR LIMITER
  FERR1=56*(1-FREF)
  FERR2=C.5*FERR1
  IF (ABS(FERR1).GT.8CC.) HERR2=SIGN(400.,HERR1)
  ANZC1=.0233*(S8*HCCI-S4*(HD+HERR2))+S5*A33
C NZ LIMITER
  IF (ANZC1) 108,107,106
108 AZMAX=.35*(CYNP-5.714)
  IF (CYNP.LE.20.0) AZMAX=E.5*DYNP/30.
  AZMAX=AMIN1(AZMAX,12.0)
  IF (ABS(ANZC1).GE.AZMAX) ANZC1=SIGN(AZMAX,ANZC1)
C 107 CONTINUE
C **** LATERAL CHANNEL ****
  ANYC1=C
  IF (1.175) GO TO 210
  IF (FLEGT.GT.0.0) GC TC 200
  CROSS=PRCCCT MIDCOURSE GUIDANCE
  CPROD=(YMA*VXTP-XMA*VYTP)/(XMA*VXTP+YMA*VYTP)
  ANYC1=2900.*CPROD/32.172
  GO TO 205
C 200 ANYC1=AKG*VT*GMGZC/32.172
  C 205 NY LIMITER
  IF (ANYC1) 208,207,208
  208 AYPAX=E.C*(DYNP-11.25)/22.5
  IF (CYNP.LE.22.5) AYPAX=4.0*DYNP/22.5
  AYPAX=AMIN1(AYMAX,12.0)
  IF (ABS(ANYC1).GT.AYPAX) ANYC1=SIGN(AYMAX,ANYC1)
C 207 CONTINUE
  LIMITING
  DENOM=SCRT((AZMAX*ANYC1)**2+(AYMAX*ANZC1)**2)
  IF (DENOM.LE.0.0) GC TO 210
  AZLIM=AZMAX*ANYC1/DENOM
  IF (ABS(ANZC1).GE.ABS(AZLIM)) ANZC1=AZLIM
  AYLIM=AYMAX*ANYC1/DENOM
  IF (ABS(ANYC1).GE.ABS(AYLIM)) ANYC1=AYLIM
C LATERAL COMMAND FILTER
  210 ANYC1=WF1*(ANYC1-ANYC)
C VERTICAL COMMAND FILTER
  ANZC1=WF1*(ANZC1-ANZC)
  RETCFN
  END
C
C
  SUBROUTINE C31
  RETURN
  END

```

CC	SUBROUTINE C3 RETURN END
CC	SUBROUTINE C41 RETURN END
CC	SUBROUTINE C4 RETURN END
CC	SUBROUTINE C51 RETURN END
CC	SUBROUTINE C5 RETURN END
CC	SUBROUTINE C61 RETURN END
CC	SUBROUTINE C6 RETURN END
CC	SUBROUTINE C71 RETURN END
CC	SUBROUTINE C7 RETURN END
CC	SUBROUTINE C81

RETURN
END

CC

SUBRCUTINE C8
RETURN
END

CC

SUBRCUTINE C9
RETURN
END

CC

SUBRCUTINE C5
RETURN
END

CC

SUBRCUTINE C101
RETURN
END

CC

SUBRCUTINE C10
RETURN
END

CCC

SUBRCUTINE C11

CCC

TRANSLATIONAL DYNAMICS INITIALIZATION MODULE D1A8
FOR USE WITH MODULE C1A CR D1B

DIMENSION IPL (100)
COMMON C(2415)
EQUIVALENCE (C(3315), A,)
EQUIVALENCE (C(3316), IPL(1))
IPL(N) = 283
IPL(N+1) = 251
IPL(N+2) = 259
IPL(N+3) = 287
IPL(N+4) = 255
IPL(N+5) = 303
A = A+6
RETURN
END

C11A003C

C11A001C
C11A0020

C11A007C
C11A0040
C11A005C

C11A0100
C11A011C
C11A012C
C11A013C
C11A014C
C11A0150
C11A016C
C11A0170
C11A0200

[illegible]


```

AD22=A32*PBA-A12*RRBA
AD21=A31*PBA-A11*RRBA
AD13=A33*PBA-A33*ORBA
AD12=A32*PBA-A32*ORBA
AD33=A13*PBA-A23*PBA
AD32=A12*PBA-A22*PBA
AD31=A11*PBA-A21*PBA
AD23=A33*PBA-A13*RRBA
RETURN
END

```

CC

```

SUBROUTINE D31
RETURN
END

```

CC

```

SUBROUTINE D3
RETURN
END

```

CC

```

SUBROUTINE D41
RETURN
END

```

CC

```

SUBROUTINE D4
RETURN
END

```

CC

```

SUBROUTINE D51
RETURN
END

```

CC

```

SUBROUTINE D5
RETURN
END

```

CCC

```

SUBROUTINE G11

```

CCC

GRAVITATIONAL AND CCFCILS ACCELERATION INITIALIZATION MODULE G11C

CCMCM (13415)

SUBROUTINE G1
GRAVITATIONAL AND CORIOLIS ACCELERATION MODULE G1C

SUBCUTANE

GRAVITATIONAL AND CIRCLIS ACCELERATION MODULE GIC

[illegible]


```

EQUIVALENCE (C(0515), FSF)
EQUIVALENCE (C(0525), VAT)
EQUIVALENCE (C(0551), RFCH)
EQUIVALENCE (C(0561), VAH)
F=FC-ZTF
ALT=F/3.28339895
VEL=VAT/3.28339895
CALL AIRF (ALT, VEL, -1, P, T, SON, DEN, Q, G, 1, 2)
CTS = CCS(TSA)
STS = SIN(TSA)
RETURN
END

```

000 000

SUBROUTINE G3

AIR DATA PCLE G3

```

CGMCA (3415)
EQUIVALENCE (C(0224), A11)
EQUIVALENCE (C(0228), A12)
EQUIVALENCE (C(0232), A13)
EQUIVALENCE (C(0236), A21)
EQUIVALENCE (C(0240), A22)
EQUIVALENCE (C(0244), A23)
EQUIVALENCE (C(0248), A31)
EQUIVALENCE (C(0252), A32)
EQUIVALENCE (C(0256), A33)
EQUIVALENCE (C(0258), CTS)
EQUIVALENCE (C(0262), STS)
EQUIVALENCE (C(0266), AGTP)
EQUIVALENCE (C(0270), VXTTP)
EQUIVALENCE (C(0274), VZTTP)
EQUIVALENCE (C(0278), THXTTP)
EQUIVALENCE (C(0282), TWXTTP)
EQUIVALENCE (C(0286), THXTTP)
EQUIVALENCE (C(0290), TWXTTP)
EQUIVALENCE (C(0294), VATX)
EQUIVALENCE (C(0298), VATY)
EQUIVALENCE (C(0302), VATZ)
EQUIVALENCE (C(0306), F)
EQUIVALENCE (C(0310), CD)
EQUIVALENCE (C(0314), LBA)
EQUIVALENCE (C(0318), VBA)
EQUIVALENCE (C(0322), VSA)
EQUIVALENCE (C(0326), USA)

```



```

A22=E221
A23=E231
A32=E232
A33=E233
VELCCITY IN BODY AXES
UBA=A11*VATX+A12*VATY+A13*VATZ
VBA=A21*VATX+A22*VATY+A23*VATZ
WBA=A31*VATX+A32*VATY+A33*VATZ
WELCCITY IN STABILITY AXES
LSA=VBA
VSA=VBA
WSA=-STS*UBA+CTS*WBA
ANGLE LF ATTACK AND SIDESLIP
ASA = ATAN(WSA/USA)
BSA = ARCSIN(VSA/SQRT(USA*USA + VSA*VSA + WSA*WSA))
RETURN
ENC

```

```

TXY0126C
TXY0127C
TXY0128C

```

```

SUBROUTINE G4I
RETURN
ENC

```

```

SUBROUTINE G4
RETURN
ENC

```

```

SUBROUTINE G5I

```

COORDINATE CCNVERSION INITIALIZATION MODULE

```

COMMON C(3415)
EQUIVALENCE (C(0557),CTPI11)
EQUIVALENCE (C(0558),CTPI12)
EQUIVALENCE (C(0559),CTPI13)
EQUIVALENCE (C(0560),CTPI21)
EQUIVALENCE (C(0561),CTPI22)
EQUIVALENCE (C(0562),CTPI23)
EQUIVALENCE (C(0563),CTPI31)
EQUIVALENCE (C(0564),CTPI32)
EQUIVALENCE (C(0565),CTPI33)
EQUIVALENCE (C(0566),CTPI1)
EQUIVALENCE (C(0576),ALZ)
EQUIVALENCE (C(0578),AZ)
AZ = A27.C1745329

```


129

```

200 PSISF=CAGZ-P*STHTS-R*CTHTS
130 THILIM=SIGN(GMBLIM,THTS)
140 PSILIM=SIGN(GMBLIM,PSIS)
150 IF(IABS(THTS).GT.GMBLIM) THTSD=50.*(THILIM-THTS)
    IF(IABS(PSIS).GT.GMBLIM) PSISD=50.*(PSILIM-PSIS)
    RETURN
    FORMAT (1H,10X,'END CF SEARCH,NO TARGET DETECTION')
    FORMAT (1H,10X,'SEEKER LCST TARGET,LOOK ANGLE=',F8.3)
    FORMAT (1H,10X,'STARTY SEARCH,ATIGS TGT RGE=',F10.3,3X,'T=',F8.3)
    END

```

CC C

SUBROUTINE S2I

```

COMMON C(3415)
EQUIVALENCE (C(03361),FLGD)
EQUIVALENCE (C(03477),FLGTS)
FLGD=C.C
FLGTS=C.C
RETURN
END

```

CC CC

SUBROUTINE S2

RADICMETER MODULE S2

```

COMMON C(3415)
EQUIVALENCE (C(01721),RTAQ2)
EQUIVALENCE (C(02224),A11)
EQUIVALENCE (C(02228),A12)
EQUIVALENCE (C(02322),A13)
EQUIVALENCE (C(02366),A21)
EQUIVALENCE (C(02440),A22)
EQUIVALENCE (C(02448),A23)
EQUIVALENCE (C(02522),A31)
EQUIVALENCE (C(02556),A32)
EQUIVALENCE (C(02773),A33)
EQUIVALENCE (C(02777),THTS)
EQUIVALENCE (C(02800),PSIS)
EQUIVALENCE (C(02900),ANGLT)
EQUIVALENCE (C(02908),XTP)
EQUIVALENCE (C(03000),YTP)
EQUIVALENCE (C(03006),ZTP)
EQUIVALENCE (C(03110),XTTP)
EQUIVALENCE (C(03111),YTTP)
EQUIVALENCE (C(03112),ZTTP)

```



```

      IN(THTS)
      CPST=CPSTIS)
      B112=CPSTIS*CTHTS
      B112=-CPSTIS*STHTS
      B211=-CPSTIS*CTHTS
      B221=-CPSTIS*STHTS
      B311=0
      B321=0
      RESOLVE TGT POSITION FROM BODY SYSTEM TO SEEKER AXES
      XOT=B111*YMT+B12*YMT+B13*ZMT
      YOT=B211*YMT+B22*YMT+B23*ZMT
      ZOT=B311*YMT+B32*YMT+B33*ZMT
      RXVT=SCRT((YCT**2+ZCT**2)**.5)
      RESOLVE AT TGT POSITION FROM BODY SYSTEM TO SEEKER AXES
      XCA=E111*XMA+B12*YMA+B13*ZMA
      YCA=E211*XMA+B22*YMA+B23*ZMA
      ZCA=E311*XMA+B32*YMA+B33*ZMA
      IF (FLGS.GT.0.0) GC TC 5
      FLG1=0
      GO TC 2
      5 ANGLT=ATAN(RYZT/XCT)
      IF (FLGTS.GT.0.0) GC TO 30
      BETA=EFFECTIVE SEARCH BEAM WIDTH (DEG)
      BTAC2=6.2-7.07E-09*SCRT(CMGC*RAD/20.)*RMA**2
      TEST FOR TARGET DETECTION
      IF (FLG1.GT.0.0) GO TC 10
      FLG1=SET WHEN TARGET ENTERS BEAM
      IF (ABS(ANGLT).LE.BTAC2) FLG1=1.0
      GO TC 20
      10 IF (FLGC.GE.1.0) GC TC 20
      TIME=1
      FLGC=SET WHEN TARGET LEAVES BEAM
      IF (ABS(ANGLT).GE.BTAC2) FLGC=1.0
      GO TC 20
      20 DETECTION TIME DECLARED 30 MSEC AFTER TGT LEAVES BEAM
      IF (TC.GE.C.03) GO TC 25
      GO TC 20
      25 IF (FLGTS.GE.1.0) GC TC 20
      WRITE(6,160) T
      FLGTS=1.0

```

```

C
30 GO TC 200
   IF (FLIGHT-LE-0.0) GC TC 200
   RADICMETER SIGNAL
   CHI1=2.0193*(2.*ANGL1/3.-1.0)
   CHI2=2.0193*(2.*ANGL1/3.+1.0)
   IF (CHI1-EC-0.0) GO TC 40
   ERSIG=(SIN(CHI1)/CHI1)**2-(SIN(CHI2)/CHI2)**2
   GO TC 50
40 ERSIG=1.0-(SIN(CHI2)/CHI2)**2
   ERSIG=ZERU IF SPEKER LCSES TPACK
50 IF (ANGL1-2.5) ERSIG=0.0
   RESCLVE INTC AZ,EL ERRCR SIGNALS (SMALL ANGLE APPROX)
   IF (RYZT-LE-0.0) GC TC 60
   ERSZ=ERSIG*YDT/RYZT
   ERSZ=ERSIG*(-ZDT)/RYZT
   GO TC 60
60 ERSZ=C.0
   ERSZ=C.0
200 RETURN
160 FORMAT (1F,10X,'END CF SEARCH,TARGET DETECTED.T=',F8.3)
END

C
SUBROUTINE S3I
RETURN
END

C
SUBROUTINE S3
RETURN
END

C
SUBROUTINE S4I
RETURN
END

C
SUBROUTINE S4
RETURN
END

C
SUBROUTINE S5I
RETURN
END

C

```

```

CC SUBROUTINE S5
CC ACCELEROMETERS AND GYRCS MCCULE
CCMCA (34151
EQUIVALENCE ((C(01361),CG)
EQUIVALENCE ((C(02121),P)
EQUIVALENCE ((C(02131),SC)
EQUIVALENCE ((C(02161),G)
EQUIVALENCE ((C(02171),RC)
EQUIVALENCE ((C(02201),F)
EQUIVALENCE ((C(07101),AX)
EQUIVALENCE ((C(07111),AY)
EQUIVALENCE ((C(07121),AZ)
EQUIVALENCE ((C(16441),AZM)
EQUIVALENCE ((C(16481),AYM)
EQUIVALENCE ((C(16521),QPM)
EQUIVALENCE ((C(16561),PM)
EQUIVALENCE ((C(09801),FM)
EQUIVALENCE ((C(17451),XAC)
DATA FLCA,0.07
CC ACCELEROMETERS CFFSET FROM CC
CC ACCELEROMETERS AT STA. 134.5
XACC=CG-4.5433
AXI=AX-XACC*(R**2+Q**2)
AZI=AZ+XACC*(R*P-QC)
AYI=AY+XACC*(G*P+RD)
IF (FLCA.GT.0.0) GO TO 1C
1CEAL ACCELEROMETERS
AXM=AXI
AYM=AYI
AZM=AZI
GO TO 20
CC DIGITAL ACCELEROMETERS
CC TRANSFORMATION TO SKEWED ACCELEROMETER AXES
1C EA11=EA11
EA12=EA11
EA13=EA11
EA21=-EA21
EA22=-EA21
EA23=EA21
EA31=EA31
EA32=-EA31
EA33=-EA31
AXA=EA11+BA12*AYI+BA13*AZI
AYA=EA21+BA22*AYI+BA23*AZI

```

```

C      AZA=EA31*AX1+BA32*AY1+BA23*AZ1
ACCELEROMETER QUANT. STEP=.194 G
AQS=6.22176
ANSX=AXA/ACS
AXMA=ACS*INT(ANSX)
ANSY=AYA/ACS
AYMA=ACS*INT(ANSY)
ANSZ=AZA/ACS
AZMA=ACS*INT(ANSZ)
TRANSFORMATION BACK TC ECDY AXES
AXM=BA11*AXMA+BA21*AYMA+BA31*AZMA
AYM=BA12*AXMA+BA22*AYMA+BA32*AZMA
AZM=BA13*AXMA+BA23*AYMA+BA33*AZMA
C      *** ICEAL CYRCS ***
20 CM=C
RM=R
FM=P
RETUFN
END
C      SUBROUTINE S6I
RETURN
END
C      SUBROUTINE S6
RETURN
END
C      SUBROUTINE S7I
RETURN
END
C      SUBROUTINE S7
RETURN
END
C      SUBROUTINE S8I
RETURN
END
C      SUBROUTINE S8
RETURN

```



```

CCCCCCCCCCCC
ENG(1)=A6/AR CR A6 (M SC) SEE 16
ENG(6)=AR (METRES SQUARED) EFFICIENT
ENG(7)=CCE BURNER CARG COEFFICIENT
ENG(8)=ETAN NOZZLE THRUST EFFICIENCY
ENG(9)=ETAC COMBUSTION RATE (KG/SEC) SEE 15
ENG(10)=WEP STAR FUEL GAS GENERATOR SYSTEMS)
ENG(11)=ISD CF PRIMARY GAS GENERATOR SYSTEMS)
ENG(12)=FLEET FLOW PERCENTAGE/100
ENG(13)=ELECT FUEL PERCENTAGE/100
ENG(14)=AIR FLOW COEFFICIENT (AFS)
ENG(15)=CAN NOZZLE MASS FLOW COEFFICIENT

REAL MCCSE MCCSE(176), EAGLE(176)
DIMENSION MCCSE(176), EAGLE(176)

*** THE INPLTS BELOW ARE SWITCHES ***
IENG(1)=11, ENGINE TYPE SWITCH
IF I1=0, RAMJET TYPE IF I1=1, GAS GEN TYPE
IENG(2)=12, CCMBUSTION EFF SWITCH
IF I2=0, ETAC IS SET BY INPUT
IF I2=1, ETAC IS SET IN A SUBROUTINE
IENG(3)=13, STANDARD DAY
I=2, 13, STANDARD DAY
IENG(4)=14, REAL AIR SWITCH SOLUTION
IF I4=0, USE THE GAMMA=1.4 SOLUTION
IF I4=1, USE THE GAMMA=1.4 SOLUTION
IENG(5)=15, USE ORCGRAM SWITCH IN THE SOLUTION
IF I5=1, THE PROGRAM USES WE IN THE SOLUTION
IF I5=2, 16, INLET CATS ARE ABSOLUTE VALUES
IENG(6)=16, THE INPLTS ARE ABSOLUTE VALUES
IF I6=0, DO NOT PRINT DEBUG DATA EACH ITERATION
IENG(7)=17, PRINT EACH ITERATION SWITCH
IF I7=0, DO NOT PRINT DEBUG DATA
IF I7=1, 18, COMBUSTION PRESSURE EFFECT SWITCH
IENG(8)=18, COMBUSTION PRESSURE EFFECT COMPUTED
IF I8=0, THERE IS NO PRESSURE EFFECT ON COMBUSTION TEMP IS COMPUTED
IF I8=1, THE PRESSURE EFFECT ON COMBUSTION TEMP IS COMPUTED

EQUIVALENCE(TABLE(206), MCCSE(1))
EQUIVALENCE(TABLE(382), EAGLE(1))

*** EXAMPLE OF NAMELIST INPUT ***
$MCNT
ENG=.5256,.05,.85,.44444,.56,.1140092,0,.98,.55,2,.50,1.2,.04,13.8
1.0,
IENG=C,0,2,C,1,0,0,0,
$EAD

```

```
C      IF (IREAL.EQ.1) GO TO 10
C      CMMENT=REAL IN DATA AND RETURN TO CALLING ROUTINE
C      CREAD(1,MPCNT)
C      WRITE(6,MCONT)
C      RETURN
C      FLT DATA IN FORM THAT THE ATTOCYC PROGRAM CAN USE
10    CCNTINLE
      PACCH=C.C
      P=XMIN
      ALPHA=Ain
      WF=XFIA
      TIME=TJA
      IF(WFINA.LT.0.0001.AND.TIME.GT.20.) IFUEL=4
      IF(IPST=IFI)
        X=XIN          @ GAMMA
        Y=YIN          @ NOT USEC
        Z=ZIN          @ NOT USEC
      Z=ZINANG(11)
      A1=(TIME.GT.17.10.0) A1=.43244
      IF(TIME.GT.1C.O.AND.TIME.LT.50.0) A1=.43162+.00008639*TIME
      IF(TIME.GE.50.0) A1=.43294
      A3=ENG(2)
      A4=ENG(3)
      IF(XIN=O GE.1.C) IFUEL=5
      IF(TIME.LE.12.5) A5=.50226
      IF(TIME.GT.12.5.AND.TIME.LE.40.) A5=.51410+.00022839*TIME
      IF(TIME.GT.40.0.AND.TIME.LE.66.) A5=.51776+.00014694*TIME
      IF(TIME.GT.66.) A5=.52746
      A6=ENG(5)
      AR=ENG(6)
      CDPR=ENG(7)
      CTAN=ENG(8)
      ETAC=ENG(9)
      IF(STAR=ENG(10).EQ.2) WF=ENG(10)
      IF(STAR=ENG(11))
      PHIE=ENG(12)
      PLFE=ENG(13)
      PLFS=ENG(14)
      CNM=ENG(15)
      IF(CNAP.LT.0.5) CNM=1.0
      I1=ENG(11)
      I2=ENG(12)
```

```

13=IENG(2)
14=IENG(3)
15=IENG(4)
16=IENG(5)
17=IENG(6)
18=IENG(7)
19=IENG(8)
20=IENG(9)
21=IENG(10)
22=IENG(11)
23=IENG(12)
24=IENG(13)
25=IENG(14)
26=IENG(15)
27=IENG(16)
28=IENG(17)
29=IENG(18)
30=IENG(19)
31=IENG(20)
32=IENG(21)
33=IENG(22)
34=IENG(23)
35=IENG(24)
36=IENG(25)
37=IENG(26)
38=IENG(27)
39=IENG(28)
40=IENG(29)
41=IENG(30)
42=IENG(31)
43=IENG(32)
44=IENG(33)
45=IENG(34)
46=IENG(35)
47=IENG(36)
48=IENG(37)
49=IENG(38)
50=IENG(39)
51=IENG(40)
52=IENG(41)
53=IENG(42)
54=IENG(43)
55=IENG(44)
56=IENG(45)
57=IENG(46)
58=IENG(47)
59=IENG(48)
60=IENG(49)
61=IENG(50)
62=IENG(51)
63=IENG(52)
64=IENG(53)
65=IENG(54)
66=IENG(55)
67=IENG(56)
68=IENG(57)
69=IENG(58)
70=IENG(59)
71=IENG(60)
72=IENG(61)
73=IENG(62)
74=IENG(63)
75=IENG(64)
76=IENG(65)
77=IENG(66)
78=IENG(67)
79=IENG(68)
80=IENG(69)
81=IENG(70)
82=IENG(71)
83=IENG(72)
84=IENG(73)
85=IENG(74)
86=IENG(75)
87=IENG(76)
88=IENG(77)
89=IENG(78)
90=IENG(79)
91=IENG(80)
92=IENG(81)
93=IENG(82)
94=IENG(83)
95=IENG(84)
96=IENG(85)
97=IENG(86)
98=IENG(87)
99=IENG(88)
100=IENG(89)
101=IENG(90)
102=IENG(91)
103=IENG(92)
104=IENG(93)
105=IENG(94)
106=IENG(95)
107=IENG(96)
108=IENG(97)
109=IENG(98)
110=IENG(99)
111=IENG(100)
112=IENG(101)
113=IENG(102)
114=IENG(103)
115=IENG(104)
116=IENG(105)
117=IENG(106)
118=IENG(107)
119=IENG(108)
120=IENG(109)
121=IENG(110)
122=IENG(111)
123=IENG(112)
124=IENG(113)
125=IENG(114)
126=IENG(115)
127=IENG(116)
128=IENG(117)
129=IENG(118)
130=IENG(119)
131=IENG(120)
132=IENG(121)
133=IENG(122)
134=IENG(123)
135=IENG(124)
136=IENG(125)
137=IENG(126)
138=IENG(127)
139=IENG(128)
140=IENG(129)
141=IENG(130)
142=IENG(131)
143=IENG(132)
144=IENG(133)
145=IENG(134)
146=IENG(135)
147=IENG(136)
148=IENG(137)
149=IENG(138)
150=IENG(139)
151=IENG(140)
152=IENG(141)
153=IENG(142)
154=IENG(143)
155=IENG(144)
156=IENG(145)
157=IENG(146)
158=IENG(147)
159=IENG(148)
160=IENG(149)
161=IENG(150)
162=IENG(151)
163=IENG(152)
164=IENG(153)
165=IENG(154)
166=IENG(155)
167=IENG(156)
168=IENG(157)
169=IENG(158)
170=IENG(159)
171=IENG(160)
172=IENG(161)
173=IENG(162)
174=IENG(163)
175=IENG(164)
176=IENG(165)
177=IENG(166)
178=IENG(167)
179=IENG(168)
180=IENG(169)
181=IENG(170)
182=IENG(171)
183=IENG(172)
184=IENG(173)
185=IENG(174)
186=IENG(175)
187=IENG(176)
188=IENG(177)
189=IENG(178)
190=IENG(179)
191=IENG(180)
192=IENG(181)
193=IENG(182)
194=IENG(183)
195=IENG(184)
196=IENG(185)
197=IENG(186)
198=IENG(187)
199=IENG(188)
200=IENG(189)
201=IENG(190)
202=IENG(191)
203=IENG(192)
204=IENG(193)
205=IENG(194)
206=IENG(195)
207=IENG(196)
208=IENG(197)
209=IENG(198)
210=IENG(199)
211=IENG(200)
212=IENG(201)
213=IENG(202)
214=IENG(203)
215=IENG(204)
216=IENG(205)
217=IENG(206)
218=IENG(207)
219=IENG(208)
220=IENG(209)
221=IENG(210)
222=IENG(211)
223=IENG(212)
224=IENG(213)
225=IENG(214)
226=IENG(215)
227=IENG(216)
228=IENG(217)
229=IENG(218)
230=IENG(219)
231=IENG(220)
232=IENG(221)
233=IENG(222)
234=IENG(223)
235=IENG(224)
236=IENG(225)
237=IENG(226)
238=IENG(227)
239=IENG(228)
240=IENG(229)
241=IENG(230)
242=IENG(231)
243=IENG(232)
244=IENG(233)
245=IENG(234)
246=IENG(235)
247=IENG(236)
248=IENG(237)
249=IENG(238)
250=IENG(239)
251=IENG(240)
252=IENG(241)
253=IENG(242)
254=IENG(243)
255=IENG(244)
256=IENG(245)
257=IENG(246)
258=IENG(247)
259=IENG(248)
260=IENG(249)
261=IENG(250)
262=IENG(251)
263=IENG(252)
264=IENG(253)
265=IENG(254)
266=IENG(255)
267=IENG(256)
268=IENG(257)
269=IENG(258)
270=IENG(259)
271=IENG(260)
272=IENG(261)
273=IENG(262)
274=IENG(263)
275=IENG(264)
276=IENG(265)
277=IENG(266)
278=IENG(267)
279=IENG(268)
280=IENG(269)
281=IENG(270)
282=IENG(271)
283=IENG(272)
284=IENG(273)
285=IENG(274)
286=IENG(275)
287=IENG(276)
288=IENG(277)
289=IENG(278)
290=IENG(279)
291=IENG(280)
292=IENG(281)
293=IENG(282)
294=IENG(283)
295=IENG(284)
296=IENG(285)
297=IENG(286)
298=IENG(287)
299=IENG(288)
300=IENG(289)
301=IENG(290)
302=IENG(291)
303=IENG(292)
304=IENG(293)
305=IENG(294)
306=IENG(295)
307=IENG(296)
308=IENG(297)
309=IENG(298)
310=IENG(299)
311=IENG(300)
312=IENG(301)
313=IENG(302)
314=IENG(303)
315=IENG(304)
316=IENG(305)
317=IENG(306)
318=IENG(307)
319=IENG(308)
320=IENG(309)
321=IENG(310)
322=IENG(311)
323=IENG(312)
324=IENG(313)
325=IENG(314)
326=IENG(315)
327=IENG(316)
328=IENG(317)
329=IENG(318)
330=IENG(319)
331=IENG(320)
332=IENG(321)
333=IENG(322)
334=IENG(323)
335=IENG(324)
336=IENG(325)
337=IENG(326)
338=IENG(327)
339=IENG(328)
340=IENG(329)
341=IENG(330)
342=IENG(331)
343=IENG(332)
344=IENG(333)
345=IENG(334)
346=IENG(335)
347=IENG(336)
348=IENG(337)
349=IENG(338)
350=IENG(339)
351=IENG(340)
352=IENG(341)
353=IENG(342)
354=IENG(343)
355=IENG(344)
356=IENG(345)
357=IENG(346)
358=IENG(347)
359=IENG(348)
360=IENG(349)
361=IENG(350)
362=IENG(351)
363=IENG(352)
364=IENG(353)
365=IENG(354)
366=IENG(355)
367=IENG(356)
368=IENG(357)
369=IENG(358)
37
```

```

C      RADEG=.572557779E+02
C      G=5.EC66E
C      START CALCULATIONS FOR EACH NEW POINT
C      COUNT=0
C      ICCNV=1
C      NEW ITERATIONS START HERE
C      4C ICCNV=C
C      CHECK TC SEE IF 100 ITERATIONS OF THE SAME PCINT HAVE BEEN MADE
C      IF (CCCNT-100) 60,60,50
C      IF NCT CCVERGED AFTER 100 ATTEMPTS, STOP
C      50 WRITE (6,530)
C      CALL EXIT
C      IF NCT CCVERGED WITH LESS THAN 100 ATTEMPTS, TRY AGAIN
C      6C CONTINUE
C      IF (CCCNT.GT.89) I7=1C
C      IF AECVE STATEMENT NCT CCVERG,PRINT LAST 10
C      IF (COUNT.NE.0) GO TO 50
C      IF (IFIRST.EC.-1) CALL AIR(F,M,-1,P,T,A,DEN,Q,G,1,1)
C      IF (IFIRST.EC.-1) GC TC 70
C      I3=1E+10C
C      CALL AIR(F,M,1,P,T,A,DEN,Q,G,1,1)
C      G=5.EC66E
C      7C IF (I4.EC.1) GO TO EC
C      ENTPR IN CAL/GM
C      ENTHR=46.427E+.2394E+(T-194.444)
C      V IN METRES PER SEC
C      V=M*A
C      ENERG=(V**2.)/8368.
C      ENTHAL=ENTHR+ENERG
C      ECGP=ENTHAL+ENERG
C      ECGM IN CAL/GM
C      TT3 IN DEGREE RANKINE
C      TT3=539.31+(17.5164*E(CGM))-(3.3559E-03*ECGM*ECGM)+(4.65976E-06*(ECGP
C      1*#3.1))-(1.22556E-09*(ECGM**4.1))
C      TT=TT3/1.8
C      GAMMA3=1.4048-(6.09654E-04*ECGM)+(2.12102E-06*ECGM*ECGM)-(5.56845E

```

```

1-C5*(ECGM**3.))+(6.45543E-12*(ECGM**4.))
GO TC 130
EC CONTINUE
GAMMA3=1.4
TT3=TT/FT(TT(1.4,M))
T3=TT3*FT(TT(1.4,M3))
SC CONTINUE
C
IF (IFIRST) 100,11C,11C
100 WRITE (6,540)
C
COMBLSTC
C
IF (12.5C.1) CALL ETA
CALL TAE (AFS)
V=-0.0
WA=1C.C
M3=
MACRS=1.
CALL FCRA
C
CALCULATION
C
WRITE (6,230)
C
DIFFUSER
C
CALL HCF
GO TC 130
11C CONTINUE
IF (A.GT.1.07) GO TC 130
IF (AF.LT..001) GO TC 120
PT=P/FFCPT(1.4,M)
PMOPT=FFCPT(1.4,M)
TT=TT/FT(TT(1.4,M))
ACR=.27661+2.4721*M-4.527*M*M+2.3036*M*M*M
AINF=AINF*FT*PMOPT)/SCRT(TT)
WA=(AINF*FT*PMOPT)/SCRT(TT)
CALL HCRA
120 CONTINUE
IF (AF.LT..001) ER=0.0
FAR=ER/AFS
CALL SLESCN (M,M,FAR,CF,AF,Q,AR)
ICCNV=C
COUNT=C
GO TC 225
C ***** END OF SUBSCNIC CALL SEQUENCE *****

```

```

13C CONTINUE
C
C ***** WE SET SA HERE *****
IF (15.EC.1) GO TO 140
IF (15.EC.2) GO TO 150
WRITE (6,240)
C GC THIS ROUTE IF ER IS INPUT OR CHANGED IN HCRA
140 CONTINUE
CALL FCRA
IF (FUEL.EC.5) ER=0.0
FAR=ER/AFS
WF=FAR*WA
AFR=1./FAR
GO TO 16C
C GC THIS ROUTE IS WF IS INPUT OR SET IN HCRA
15C CONTINUE
CALL FCRA
IF (FUEL.EC.5) WF=0.0
FAR=WF/WA
ER=FAR*AFS
IF (FAR.EC.0.0) GO TO 1400
AFR=1./FAR
GO TO 16C
1400 AFR=0.0
C
C 16C CONTINUE
C
TOTTE=FTCTT(GAMMA3,M3)
T3=TT3+TCTT3
SORTCT=SCRT(TT3)
C
C TO FIND AIR SPECIFIC IMPULSE
C
IF (110.EC.0) GO TO 161
CONTINUE
GO TO 16C
161 CONTINUE
CALL INTR20 (TT3,ER,TT41,TABLE,1)
CALL INTR20 (TT3,ER,GAMMA4,TABLE,2)
CALL INTR20 (TT3,ER,MW4,TABLE,3)
C
C COMPUTE THE EFFECT OF PRESSURE ON TEMPERATURE
IF (18.EC.0) GO TO 162
G1=GAMMA4+1.0
G2=GAMMA4-1.0
G3=G1/G2

```

```

1615 G4=2.*G1*(2./G1)**E3)
      G5=SCRT(G4)/(G5*A5)
      PT4=(S7*W4)/(G5*A5)
      PT4ATM=PT4/101360.
      IF(PT4ATM.LT.20.) PT4ATM=.2
      IF(PT4ATM.GT.20.) PT4ATM=20.
      IF(TT3.LT.255.) GC TC 1615
      IF(PT4ATM.LT.255.) GO TO 1615
      IF(PT4ATM.GT.255.) GO TO 1615
      CTEMP=TC(TT3,SR,PT4ATM)
      CCNTINLE
      TT4I=TT4I+CTEMP
162 CCNTINLE
      C
      C NEW FINISH CCMREUSTOR COMPUTATIONS
      C
      R=8214.24/MM4
      WAX=W4/.45E5524
      THROAT=AX/.(9290304
      TTT=((TT3*1.8)/1000.)*2.
      PSV=(WAX*TTT)/THROAT
      IF (I2.EC.1) CALL ETA
      TT4=ETAC*(TT4I+TT3)
      BBB=2.*C*P*(1.0+GAMMA4)*TT4/GAMMA4
      BBB=AMAX1(BBB,0)
      SA=(1.0+FAF)*SQRT(BBB)
      SART=SA/SQRTCT
      CCNTINLE
165
      C
      C COMMENT MAKE CALCULATIONS TAKING VARIABLE NOZZLE GAMMA INTO EFFECT
      C
      C
      A4CA5=A4/A5
      M4=FMARL(GAMMA4,A4CA5)
      PT14=PT1M(GAMMA4,M4)
      C
      C ***** WE COMPUTE CONDITIONS AT STATION 3 HERE *****
      C
      C
      STRCG=22.5613
      IF (I1.EQ.1) GO TO 170
      RANJET=CCFELTATIONS - FUEL ADDED IN THE INLETS
      SARTA=STFCG*SCRT((GAMMA3+1.)/GAMMA3)*(1.+FAF)
      PT13=(PT14*SART/SARTA)+((GAMMA3*CDB*M3*M3)/(2.*XMCIR3*SARTA))
      GO TC 180
      C
      C GAS GENERATOR SYSTEM COMPUTATIONS
      SART=STFCG*SCRT((GAMMA3+1.)/GAMMA3)
      PH13=(PT14*SART/SARTB)-((PHIPRI*ISTAR*FAF)/(SARTB*SQRTGT))+((GAMMA
170 13*CCB*M3*M3)/(2.*XMCIR3*SARTB))
      CCNTINLE
180 M3=FAFPHIN(GAMMA3,PH13)

```



```

C 200 CONTINUE, INCREASE CCUNT AND CALCULATE PARAMETERS NOT IN COMMON
C SET IF15,
C COUNT=CCUNT+1
C CHECK CONVERGENCE, ICCNVF IS USED TO MAKE SOLUTION CONVERGE TWICE
C IF CCNVERGED, PRINT. IF NOT, GO BACK AND TRY AGAIN
C
C IF (IFIRST.EC.-1) GO TO 225
C IF (ICCNV+ICCNVP) 210,220,210
C 210 ICCNVF=ICCNV
C GO TO 40
C 220 CONTINUE
C ***** WE NOW COMPUTE THRUST IF CCNVERGED *****
C
C A6CA5=A6/A5
C M6=FMARH(GAMMA4,A6CA5)
C PH16=PT1M(GAMMA4,M6)
C IF (110.(E.5) GO TO 221
C CFC6=(MA*(SA*PH16*ETAN-A6*F))/(AR*Q)
C 221 CCNTINLE
C CCINF=2.*C*ACR*MACRS*A1/AR
C CALPHA=CCS(ALPHA/RACEG)
C CF=CF6-(CFINF*CALPHA)+CFB+CFC
C F=CF+C*AR
C IF (MF.EC.0.0) GO TO 1300
C ISF=F/(WF*G)
C GO TO 1500
C 1300 ISF=C.C
C 1500 CCNTINLE
C G1=GAMMA4+1.0
C G2=GAMMA4-1.0
C G3=G1/G2
C G4=2.*G1*(2./G1)**G3)
C G5=SCRT(G4)
C PTE=(SA*A)/(G5*A5)
C
C PREPARE DATA FOR TRANSFER TO MAIN PROGRAM
C
C 225 CONTINUE
C FIN=F
C XMIN=M
C AIN=ALPHA
C WFIA=WF
C CDADC=CCASUB+CDASUP+CFT
C CT=CF-CCACC

```

```

SMARG=FM ST ANT+.01
IF1=IF1RST ANT+.01
FCCUNT=CCFT=CI
WRIT(1)=AA
WRIT(2)=TT4
WRIT(3)=MACR
WRIT(4)=MACR
WRIT(5)=ER
WRIT(6)=ES
WRIT(7)=ES
WRIT(8)=ES
WRIT(9)=ETAC
WRIT(10)=ETAC
WRIT(11)=CT
WRIT(12)=FCCUNT
WRIT(13)=FCCUNT
WRIT(14)=ESV
WRIT(15)=EC.C.C) GO TC 120C
IF(PT3.EC.C.C)=PT5/PT3
CONTINUE
WRIT(16)=PT5/1000.
FCCUNT=FCCUNT
RETURN
120C

C
230 FORMAT (1, F8.2, 8F7.4, 2F7.3, F7.2)
240 COMPUTE PT13.)
250 IS IS CUT (F RANGE-WHY.)
260
270
280
290
300
310
320
330
340
350
360
370
380
390
400
410
420
430
440
450
460

```

```

470 FCFMAT (14H      A 6  = F9.7,9H SQ METRE)
480 FCFMAT (14H      A R  = F9.7,9H SQ METRE)
490 FCFMAT (28H      PFI4  = F7.5)
500 FCFMAT (28H      PFI6  = F7.5)
510 FCFMAT (28H      BURNER DRAG COEFF. = F6.2)
520 FCFMAT (28H      NZZZLE MASS COEFF. = F6.2)
530 FCFMAT (36H      DID NOT CONVERGE AFTER 100 ATTEMPTS)
540 FCFMAT (14H      RAMJET PROPULSION
)

END

SUBROUTINE ETA

NWC COMPCN OFCK FCR AIRBREATHING PROPULSION PROGRAMS
CPIA NCMENCLATURE

REAL ISF,ISTAR,M,M1,M2,M3,M4,M5,M6,MW,MW1,MW2,MW3,MW4,MW5,MW6
REAL MACR,MACRS
INTEGER CCUNT

COMMON /RJ/ A,AINF,A1,A2,A3,A4,A5,A6,AR,AFR,AFS,ALPHA
COMMON /RJ/ ACR,BLFEEL
COMMON /RJ/ CDASUB,CDASUF,CDB,CF,CFINF,CF6,CFB,CFT,CFC,CNM
COMMON /RJ/ ER,ERRL,ETAC,ETAN,F
COMMON /RJ/ FAR,G,GAMMA,GAMMA1,GAMMA2,GAMMA3,GAMMA4,GAMMA5,GAMMA6
COMMON /RJ/ F,ISP,ISTAR,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
COMMON /RJ/ CCUNT,ICCNV,ICCNVP,IFIRST
COMMON /RJ/ M,M1,M2,M3,M4,M5,M6,MW,MW1,MW2,MW3,MW4,MW5,MW6
COMMON /RJ/ MACR,MACRS,PM
COMMON /RJ/ P,P1,P2,P3,P4,P5,P6,PT30I,PRAR
COMMON /RJ/ PT,PT1,PT2,PT3,PT4,PT5,PT6,PMOFT,PMOFT1,PMOFT2,PMOFT3
COMMON /RJ/ PHI1,PHI2,PHI3,PHI4,PHI5,PHI6,PHI7,PHI8,PHI9,PHI10
COMMON /RJ/ RADEGR,RA,SA,SA1,SA2,SA3,SA4,SA5,SA6,SA7,SA8,SA9,SA10
COMMON /RJ/ T,T1,T2,T3,T4,T5,T6,TT1,TT2,TT3,TT4,TT5,TT6
COMMON /RJ/ V,V1,V2,V3,V4,V5,V6
COMMON /RJ/ WA,WB,X,Y,Z

DATA A11,E11,C11,D11,E11,F11,G11
* /-1407.66, -95416, -8.55033, 43.7266, 1468.91, 80484.,
* -277056./

FAR2=FAR*FAR
FOVER=1./((1.+FAR)

```

```

TERM1=P11+C11*FAR+D11*FAF2
TERM2=E11+F11*FAR+G11*FAR2
TTR=1.E+11
TT41=A11+TTR*TERM1+FCVER*TERM2
STT41=SCRT(TT41)
ETAC=40.246*STT41-0.2513*TT41-1055.08
ETAC=ETAC/100
IF(ETAC.LT. .0001) ETAC=0.0
RETURN
END

SUBROUTINE FCRA
DIMENSION FS(66), PSTAT1(11), FUELF(11)
NWC COMMON DECK FOR AIRBREATHING PROPULSION PROGRAMS
CPIA NCMENCLATURE
REAL ISP,ISTAR,M,M1,M2,M3,M4,M5,M6,MW,MW1,MW2,MW3,MW4,MW5,MW6
REAL MACR,MACRS
INTEGER CCLAT
COMMON /RJ/ A,AINF,A1,A2,A3,A4,A5,A6,AR,AFR,AFS,ALPHA
COMMON /RJ/ ACR,BLEEC
COMMON /RJ/ CDASUB,CDASUF,CDB,CF,CFINF,CF6,CF8,CFT,CFC,CNM
COMMON /RJ/ EF,EPFL,ERLL,ETAC,ETAN,F
COMMON /RJ/ FAR,G,GAMMA,GAMMA1,GAMMA2,GAMMA3,GAMMA4,GAMMA5,GAMMA6
COMMON /RJ/ F,ISP,ISTAR,I1,I2,I3,I4,I5,I6,I7,I8,I9,I10
COMMON /RJ/ CCUNT,ICCNV,ICCNVP,IFIRST
COMMON /RJ/ M,M1,M2,M3,M4,M5,M6,MW,MW1,MW2,MW3,MW4,MW5,MW6
COMMON /RJ/ MACR,MACRS,PM
COMMON /RJ/ P1,P2,P3,P4,P5,P6,PT301,PRAR
COMMON /RJ/ PT,PT1,PT2,PT3,PT4,PT5,PT6,PMOPT,PMOPT1,PMOPT2,PMOPT3
COMMON /RJ/ PH1,PH12,PH13,PH14,PH15,PH16,PTPRI,Q
COMMON /RJ/ RADEG,T2,T3,T4,T5,T6,TIME
COMMON /RJ/ TT,TT1,TT2,TT3,TT4,TT5,TT6
COMMON /RJ/ V,V1,V2,V3,V4,V5,V6
COMMON /RJ/ WA,WFX,Y,Z
ANGLE CF ATTACK PS(1) I=3,14
PACH ALPHA PS(1) I=15,81
DATA PS/

```

CCCCC C

CCCCC


```

C
C
A .5825,.5819,.5812,.5800,.5700,.5650,.5050,
A .6725,.6712,.6700,.6675,.6520,.6400,.6275,
A .5010,.5000,.4990,.4970,.4825,.4700,.4550,
A .5425,.5400,.5375,.5325,.5200,.5075,.4950,
A .5420,.5425,.9410,.5375,.9300,.9225,.9125,
A .5450,.5437,.9425,.5400,.9370,.9325,.9250,
A .5450,.5437,.9425,.5400,.9370,.9325,.9250/
C
C
C DATA XM/
A .5,.5,1,.1,4,1,6,2,2,2,4,3,2,4,8,2,5,2,7,3,0,3,25/
C
C DATA CALC/
A .0213,.0212,.0132,.125,.0988,.0221,.0184,.0173,.0171
B .0123,.0145,.0176/
C
C DATA B*P/
A .0329,.0239,.0208,.0237,.0249,.0334,.0287,.0265,.0259,
B .0213,.0171,.0150/
C
C IF (IFIRST) 10,20,20
10 CONTINUE
PRAR=0.0
TCL=.0004
WRITE (6,100) TCL
RETURN
C
C
C 20 CONTINUE
C
C SET PRESENT PR/AR EQUAL TC PREVIOUS PR/AR
FRAR=FFAF
C
C CALCULATE NEW PR/AR
C
TT=1/FTC(TT(1,4,M))
TT*ATC=SCRT(TT3/TT)
PRAR=TT*ATC*PMOPT*MACRS*AI/(PMOPT3*A2)
IF (IBC.EQ.0) PRAR=TT*ATC*PMOPT*MACRS*AI*(1.0+FAI)/(PMOPT3*A3)
C
C CHECK FOR CONVERGENCE BY COMPARING PR/AR TO PREVIOUS PR/AR
C
IF (ABS(FRAR-PRAR)-TCL) 40,40,30
30 ICGN=1
40 CONTINUE
C
C CALFA=CCS(ALPHA/RAI*EG)
C

```

```

CC      COMPUTE CRITICAL PRESSURE RECOVERY AND MAXIMUM AIR CAPTURE RATIO
CC
      ALPHA=AES(ALPHA)
      CALL INTF20 (M,ALPHA,PACR,AA,1)
      ELCT=CCL1(M,XM,BWR,12)
      BLEED=.015
      BLEED=ELCT*8LDD8
      MACRS=MACR*(1.0-BLEED)
      CALL INTF20 (M,ALPHA,FRIAD,PR,1)
      CRITICAL PRESSURE RECOVERY
      PACR=FRIAD
      IF (FRIAD-PACR) 50,60,70
      PRESSURE RECOVERY .LT. CRITICAL
50      FT301=FRAR
      ACR=1.
      GO TC 80
CC
      PRESSURE RECOVERY .EQ. CRITICAL
60      PT301=FRIAD
      ACR=1.
      GC TC 80
CC
      PRESSURE RECOVERY .GT. CRITICAL --CHANGE AIR CAPTURE AND SET PRES
      RECOVERY EQUAL TO CRITICAL VALUE
CC
70      FT301=FRIAD
      ACR=FRIAD/FRAR
      POPT=FFCPT(1.4,M)
      PT=F/FCFT
      PT=PT*FT301
      PM=(FRIAD-PT301)/(FRIAD/100.)
      AINF=ACR*1*MACRS
      SQIAF=SQRT(TT)
      WA=(AINF*FT*FMOPT)/SQIAF
      IF (ICGNVF.NE.0 .OR. ICCNV.EC.1) RETURN
      CDSUB=C.0
      IF (FRAR.LE.FACR) GC TC 50
CC
      CALCULATE ADDITIVE CRAG CUE TO SUBSONIC SPILLAGE
      ETASUB=M/((M*10.2095)-11.5652)
      CDSUB=2.*(1.-ETASUB)*(1.-ACR)
      C

```

```

C          CALCULATE ADDITIVE DRAG DUE TO SUPERSONIC SPILLAGE
90  CAEC=CL1(P,XM,CACC,12)
   IF (PRAR.LE.PACR) CDSL8=C.0
   CCASUB=CCSUB*AL/AR
   CCASCP=CAED
   CFB=C.C
C          CFT IS RAT DRAG
   CFT=C.CC/2
   PESI=2.14678-1.11435*M+.22989*M*M
   FE=PESI*F
   AINFB=A1*BLOCB
   AE=9.33*.00064516
   TRM1=P*AINFB*1.4*M*M
   TRM2=P*AE
   XDCT=((FT*AINFB*PMOPT)/SQTNF
   XME=XCCT*SCRT(TT-222.)/(PE*AE)
   XMC=FRCM(1.4,XMCE)
   TRM3=PE*AE*(1.+1.4*XME*XME)
   CFC=-((TRM1+TRM2-TRM3)/(C*AR))
   RETURN
C100  FORMAT (T6,'ALVRJ INLET DATA DEC 1975 ,TOL=',F7.6)
C          END
C          FUNCTION INTFP (AH,AL,BF,BL,FH,FM)
C          REAL INTFF
C2 = AL + (AH - AL)*FM
C1 = BL + (BF - BL)*FM
   INTFF = C1 + (C2 - C1)*FF
   RETURN
   END
C          FUNCTION FATAS(GAM,XM)
C          FUNCTION SUBPRCGFAM TO CALCULATE A OVER ASTAR
C          FRCF GAMPA AND NACT NUMBER
C          Z=(GAM+1.C)/(2.0*(GAM-1.C))
A=2.C/(GAM+1.0)
B=(GAM-1.0)/2.0
C=1.C/XM

```

FATA0030
 FATA001C
 FATA0020
 FATA004C
 FATA0050
 FATA0060
 FATA0070

FATA008C
FATA009C
FATA010C
FATA011C

R 2
R 4
R 6
R 8
R 10
R 12
R 14
R 16
R 18
R 20-

C=A*(1-C*(B**X**XM))
FATA=C*(C**Z)
RETURN
END

FUNCTION FIXAC (A)

TFIX=AES(A)-1.0000005
IF (TFIX.LT.0.0) GO TC 1C
IF (A.LT.0.0) GO TC 2C
FIXAC=C.C
GO TC 3C
20 FIXAC=1.14159265
3C CONTINUE
RETURN
END

CC C

S 2
S 4
S 6
S 8
S 10
S 12
S 14
S 16
S 18
S 20-

FUNCTION FIXAS (A)

TFIX=AES(A)-1.0000005
IF (TFIX.LT.0.0) GO TC 1C
IF (A.LT.0.0) GO TC 2C
FIXAS=1.57079633
GO TC 3C
20 FIXAS=-1.57079633
3C CONTINUE
RETURN
END

CC C

FAR0030
FAR001C
FAR002C
FAR004C
FAR005C
FAR006C
FAR007C
FAR008C
FAR009C
FAR010C
FAR011C

FUNCTION FMARH (GAM,ATAS)

FUNCTION SUBPROGRAM TO CALCULATE MACH NUMBER
FROM ATAS AND GAMMA

Z = (GAM + 1.0) / (2.0*(GAM - 1.0))
XMG = 10.0
XMG = 10.0
500 XMG = (XMG / 2.0
XMG = APX1(1.0,AMIN1(100.0,XMG))
IF (XMG - 1.0/XMG*(2.0*(GAM - 1.0) *XMG **2)/(GAM + 1.0))**Z
200 IF (CATAS - ATAS) 700,100,800

CC C

FMAR0120
FMAR0130
FMAR0140
FMAR0150
FMAR0160
FMAR0170
FMAR0180

```

700 XMG = XPG + CXMG
    GO TO 500
800 XMG = XPG - DXMG
    GO TO 500
100 FMARL = XMG
    RETURN
    END

```

CCCC

```

FUNCTION FMARL(GAM,AR)
    FUNCTION ROUTINE TO FIND MACH FROM A/A* AND GAMMA
    USING NEWTON'S METHOD FOR FINDING ROOTS OF AN EQUATION
    SUBSCNIC SOLUTION
    IF(GAM.LT.1.) WRITE(6,20)
    IF(AR.LT.1.) WRITE(6,40)
    IF(GAM.LT.1. .OR. AR.LT.1.) CALL EXIT
    GM1=GAM-1.
    GPI=GAM+1.
    A=2./GFI
    B=GM1/(2.*GM1)
    Z=GFI/(2.*GM1)
    IF(AR.LT.1.3401) XMP=2.4706-(1.47059*AR)
    IF(AR.LT.1.34 .AND. AR.LT.3.001) XMP=.77443-(.2048*AR)
    IF(AR.GT.3.) XMP=(A**2)/AR
    TOL=.0001
    CCNT=1
    DO 10 INLE
    BRACK=A*(B*XMP*XMP)
    POWR=ERACK**Z
    TOF=(1./XMPF)-(AR/POWR)
    BOTTCM=(1./BRACK)-(1./XMPF*XMP)
    XM=XMP-(TCP/BOTTOM)
    CELTA=ABS(XM-XMP)
    XMP=XM
    IF(CELTA.GT.TOL) GO TO 10
    FMARL=XM
    RETURN
    FORMAT('GAMMA LT 1 IN FMARL')
    FORMAT('A/A* LT 1 IN FMARL')
    ENC

```

30
40
C
C
C
C
C

```

C      FUNCTION FPHIM (GAM , PHI )
C      CFMPHIM  FUNCTION SUBPRGAM TO CALC MACH NO  FROM PHIM
C
      TOL = .0001
      XMSN = 1.0 / ( PHI**2 )
      127 X = XMSN
      XMSN = (1.0 + (GAM*(XMSN**2))) / (PHI * SQRT(2.0 * (GAM+1.0))*(1.0
      1+(GAM-1.0)+.5*(XMSN**2)))
      XMSN = AMAX1(0. , AMIN1(1.0,XMSN))
      IF (ABS(X-XMSN)-TOL) 125, 128
      126 XMSN = (X + XMSN) / 2.0
      GO TO 127
      125 FMPHIM = XMSN
      RETURN
      END
C      FUNCTION FMPOPT (GAM,POPT)
C      CFMPOPT  FUNCTION SUBPRGAM TO CALC XMACF
C      FROM P/PT ANC GAMMA
C
      Z = (FCPT) ** ((1.0-GAM)/GAM)
      SMPPT = (Z - 1.0)/(GAM - 1.0) * 2.0
      FMPCFT = SQRT(SMPPT)
      RETURN
      END
C      FUNCTION PHIM (GAM, XMACF)
C      CPHIM  FUNCTION TO CALC PHI FROM GAMMA AND MACH
C
      PHIM = (1.0+GAM * (XMACF**2)) / (XMACF * SQRT(2.0*(GAM + 1.0))*(1.0+
      1.5*(GAM-1.0) * (XMACF**2)))
      RETURN
      END
C      FUNCTION FPOPT (GAM,XMACH)
C      CFPCPT  FUNCTION SUBPRGAM TO CALC TOTAL PRESSURE RATIO
C      FROM GAMMA ANC MACH NO.

```

FAPH0020

FAPH001C

FAPH003C

FAPH004C

FAPH005C

FAPH006C

FAPH007C

FAPH009C

FAPH010C

FAPH011C

FAPH012C

FAPH013C

FAPH014C

FAPH0030

FAPH001C

FAPH002C

FAPH004C

FAPH005C

FAPH006C

FAPH007C

FAPH008C

FPH002C

FPH001C

FPH003C

FPH004C

FPH005C

FPH006C

FPH003C

FPH001C

FPH002C

```

C      Z = -GAM / (GAM - 1.0)
C      FPOPT = (1.0+.5*(GAM - 1.0)*(XMACH**2))**Z
C      RETURN
C      ENC
C
C      FUNCTION FRCM(GAM,XMCIR)
C
C      FUNCTION ROUTINE TO COMPUTE MACH NUMBER FROM
C      MCIRCLE AND GAMMA FOR ALL MACH NUMBERS USING NEWTONS METHOD
C
C      IF (GAM.LT.1.) WRITE (6,100)
C      IF (GAM.LT.1.) CALL EXIT
C      A=GAM
C      P=0.5*(GAM*(GAM-1.))
C      C=XMCIR/.C5902063
C      XMP=C/(1.18+ (.0923*C))
C      TCLK=.0001
C      1C CONTINUE
C      PRACK=A+(F*XMP*XMP)
C      PONE=SCFT(BRACK)
C      TOP=(XPF*FCWER)-C
C      BOTTLCH=(A+(2.*B*XMP*XMP))/POWER
C      XM=XPF-(TOP/BOTTLCH)
C      CELTA=ABS(XM-XMP)
C      XMF=XM
C
C      IF (DELTA.GT.TOL) GO TO 1C
C
C      FRCM=XF
C
C      RETURN
C
C      100 FORMAT('GAMMA LT 1 IN FROM')
C      END
C
C      FUNCTION FTCTT (GAM,XMACH)
C
C      FUNCTION SUBPROGRAM TO CALC T/TT FROM GAMMA AND MACH NC
C
C      FTOTTT = (1.0+.5*( GAM-1.0)*(XMACH**2))**(-1.0)
C      RETURN
C      END
C

```

```

CC      SUBROUTINE TAB(FAS)
C
C      15      WRITE(6,15)
C              FORMAT(16,'NO.2 JFS AND JETA FUEL MAY 1,1976 ONE ATM')
C              FAS= 14.6354
C              RETURN
C              END
C
C      FUNCTION FMCIR (GAM,XM)
C
C      MCIRCLE      FUNCTION SUBPROGRAM TO CALC MCIR FROM GAMMA AND MACH NO
C                    R AIR= 314.34/28.96246 = 287.073 N M/ KG K
C                    M CIRCLE DIMENSION= SEC * SQRT(K) / METRE
C
C      FMCIR=.C55C2C6289*XM*SQRT(GAM*(1.0+((GAM-1.0)/2.0)*XM**2))
C      RETURN
C      END
C
C      SUBROUTINE AIR (Z,V,J,P,T,S,D,Q,GO,K,KK)
C
C      Z=ALTITUDE IN METERS
C      V=MACH NUMBER OR VELOCITY, SEE KK
C      J=FIRST PASS IDENTIFIER SWITCH, -=FIRST PASS AND +=NOT FIRST PASS
C      P=PRESSURE IN PASCALS OR PSI
C      T=TEMPERATURE IN K OR R
C      S=SPEED OF SOUND IN METERS/SEC OR FT/SEC
C      D=DENSITY IN KG/CUBIC METER OR LBM/CUBIC FCCT
C      GO=DYNAMIC PRESSURE IN PA OR PSI
C      KK=1, SI UNITS
C      KK=2, ENGLISH UNITS
C      KK=1, V IS VELOCITY, METERS/SEC OR FT/SEC
C      KK=2, V IS VELOCITY, METERS/SEC OR FT/SEC
C
C      DIMENSION A(2,16), C(15)
C
C      A(1,J),J=1,16
C      THESE ARE THE CHEBYSHEV COEFFICIENTS FOR P/PO
C
C      A(2,J),J=1,16
C      THESE ARE THE CHEBYSHEV COEFFICIENTS FOR D/DO

```

```

DATA A, Z10, -4.013517C, -2.2874890, -2.1765372,
A -4.0494300, -1.3585534, .0355105, .0498002,
B - .00552256, .0061561, .0030910, -.0121108,
C - .0026116, .0025277, .0004221, .0044268,
D - .0010261, .0039804, .0007523, -.0013126,
E - .00003145, .0033573, .0005513, -.0011477,
F - .0000031, .0021286, .0002918, .0008523,
H - .0001420, .0010109, .99998983, 1.0000000/

C ** SET CCNSTANTS
C
DATA PC, IC, RBAR, GC, Z1/
* 101225.0, 1.225, 287.07299, 9.80665, 30480.0/

C ** IF NECESSARY , CONVERT IC SI UNITS
C ** Z BECCMES METERS
C ** V BECCMES METERS/SEC
C
IF (K.EC.2) Z=Z*.3048
IF (K.EC.2.AND.KK.EC.2) V=V*.3048

C ** START HERE
C
IF (J) 10, 20, 20
1C WRITE (6,6C)
WRITE (6,7C) Z1

C ** CCMPUTE C(K) COEFFICIENTS
C
2C ETA=2.*(2.*Z/Z1-1.)
C(1)=ETA
C(2)=ETA**2-2.
DO 3C I=3,15
3C C(I)=ETA*C(I-1)-C(I-2)

C ** COMPLE CHEBYSHEV TRUNCATED EXPANSION
C
ALNP=1(1,1)
ALND=A(2,1)
DO 4C I=2,15
4C ALNF=ALNP+A(1,I)*C(I-1)
ALAC=ALND+A(2,I)*C(I-1)

C ** COMPLE ATMOSPHERIC PRESSURE
C ** P IN N/SQ METER
C
ALNF=.5*ALNF

```

```

CC CCFZ=EXP(ALNP)
CC F = FCFZ*FC*A(1,16)
CC ** COMPUTE GRAVITY AT ALTITUDE AND LATITUDE
CC RC IS EQUATORIAL RADIUS (M) PER CRC HANDBOOK
CC GC IN M/SEC SQ
CC
CC RC=6378377.45
CC GO=GC*((RC/(RO+Z))**2)
CC ** COMPUTE ATMOSPHERIC DENSITY
CC D IN KG PER CUBIC METRE
CC
CC ALND=.5*ALNC
CC DOZ=EXP(ALND)
CC L = CCC2*CC*A(2,16)
CC ** COMPUTE AIR TEMPERATURE
CC T IN DEGREE KELVIN
CC
CC T=P/(REAR*D)
CC ** COMPUTE SPEED OF SOUND IN M/SEC
CC
CC S=SQRT(401.50219*T)
CC ** COMPUTE DYNAMIC PRESSURE IN PA
CC
CC IF (KK.EC.1) Q=0.7*F*V**2
CC IF (KK.EC.2) Q=0.7*P*(V/S)**2
CC ** CONVERT TO ENGLISH UNITS IF NECESSARY
CC
CC IF (K.EC.1) GC TO 50
CC Z=Z/.3048
CC IF (K.EC.2) V=V/.3048
CC P=P/6894.757
CC T=T*1.8
CC S=S/.3048
CC C=C/16.01346
CC C=C/6894.757
CC GC=GC/.3048
CC RETURN
CC 50 FORMAT (TS, '*** US 1962 STANDARD ATMOSPHERE')
CC 60 FORMAT (TS, '*** ALTITUDE FROM 0 TO ', F8.0, ' METERS')
CC 7C ENC

```



```

C
C      FUNCTION VECMAG(X)
C
C      DIMENSION X(3)
C      X2=X(1)*X(1)+X(2)*X(2)+X(3)*X(3)
C      VECMAG=SQRT(X2)
C      RETURN
C      END

```

END

FUNCTION TC (T,FR,ATM)

THEORETICAL TEMPERATURE RISE CORRECTION FOR PRESSURES OTHER
THAN ONE ATMOSPHERE (HYDROCARBON FUELS)

DIMENSION	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510	511	512	513	514	515	516	517	518	519	520	521	522	523	52
-----------	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	-----	----

CC
CHECK FOR ENTRY OUTSIDE LATTICE
25270-0401127

IF	(T.L.T.	.258.)	G0	T.C	.5C
IF	(T.G.T.	.130.)	G0	T.C	.60
IF	(ER.G.T.	.1.5)	G0	T.C	.70
IF	(ATP.L.T.	.2)	G0	T.C	.50
IF	(ATP.L.T.	.20.)	G0	T.C	.80

```
C  
C      IF (AIR.EI.ZO.7) GO TO 80  
C      CCMPUTE FACTOR ANALYSIS FACTORS F1,F2,F3
```

```

C
F1=-((.227276E-3+.100807E-6*T)*T+.018785)
F2=1.00120097-.1254E-5*T+.0846043
F3=((.113312E-7-.244175E-11*T)*T-.141615E-4)*T+.00496295)*T+.0056
A0473

C
C
C LINEARLY INTERPOLATE FOR FACTORS G1,G2,G3
DO 10 I=1,3
IF (ATM.LE.P(I)) GO TO 20
10 CCATINLE
J=I-1
R=(E(E(I))-E(J))/(E(I)-E(J))
G1=G(J)+G(I,1)-G(J,1))*R
G2=G(J)+G(I,2)-G(J,2))*R
G3=G(J)+G(I,3)-G(J,3))*R
C LINEARLY INTERPOLATE FOR FACTORS H1,H2
DO 30 I=1,13
IF (ATM.LE.P(I)) GO TO 40
30 CCATINLE
J=I-1
H1=(A(I)-F(J))/(P(I)-P(J))
H2=H(J)+H(I,1)-H(J,1))*R
H2=H(J)+H(I,2)-H(J,2))*R
C COMPLETE TEMPERATURE CORRECTION (TC)
TC=C(1)*F1*G1*H1+C(2)*F2*G2*H1+C(3)*F1*G2*+2+C(4)*F2*G1*+2
D + C(5)*F3*G3*+F1
E + C(6)*F2*G3*+F2
F + C(7)*F3*G2*+F2
G + C(8)*F1*G3*+F2
C
RETURN (C,110)
50 GO TO 100
60 WRITE (C,120)
70 GO TO 100
80 WRITE (C,130)
90 GO TO 100
100 WRITE (C,150)
110 WRITE (C,140)
120 CALL EXIT
130 RETURN
C

```



```

CC 1C 11=4,IX1
IF(I)
  IF(X,LT,C(I)) GO TO 11
  CONTINUE
10 IO=1-1
  IX2=IX1+4
  IX3=IX1+N2
  DO 2C JJ=IX2,IX3
    J=JJ
    IF(Y,LT,C(J)) GO TO 21
    CONTINUE
20 JO=J-IX1
21 NA=I+3+N2*((I-1)*N1+IC-2)+JO-1
  F1=(X-C(I-1))/(O(I)-C(I-1))
  F2=(Y-C(J-1))/(O(J)-C(J-1))
  F3=F1+F2
  C= C(NA)*((1,-F1-F2+F3)
    1 + C(NA+N2))*(F1-F3)
    2 + C(NA+1)*(F2-F3)
    3 + C(NA+N2+1)*F3
  RETURN
END

```

CCC CCCC

FUNCTION TPREOL (X,Y,Z,AX,AY,AZ,XYZ,NX,NY,NZ)

THIS FUNCTION INTERPOLATES LINEARLY IN THREE DIMENSIONS. IT
 SELECTS THE TWO PLANES OF CONSTANT Z AND USES STDLI FOR TWO DIMEN-
 SIONAL INTERPOLATION. A LINEAR INTERPOLATION IS USED IN Z.
 PFCG FROM H. SOBEL DATE 22 AUGUST 1968
 DIMENSION AX(NX), AY(NY), AZ(NZ), XYZ(NX,NY,NZ)
 DATA N1,1/
 CALL SEFCT (Z,AZ,K,NZ,M)
 AL=STDLIA(AX,AY,XYZ(1,1,K),X,Y,NX,NY)
 AU=STDLIA(AX,AY,XYZ(1,K+1),X,Y,NX,NY)
 THREECL=AL+(AU-AL)*(Z-AZ(K))/(AZ(K+1)-AZ(K))
 RETURN
 END

2 4 6 8 10 12 14 16 18 20 22 24 26-
 F F F F F F F F F F F F F F F

LIST OF REFERENCES

1. Honeywell Defense System Division, System Design Tradeoff Study For Supersonic Tactical Missile (STM), Volume I, by Robert H. Josselson, p. 9-17, 24 March 1978.
2. Honeywell Defense System Division, System Design Tradeoff Study For Supersonic Tactical Missile (STM), Volume I, by Robert H. Josselson, p. 11, 24 March 1978.
3. Honeywell Defense System Division, System Design Tradeoff Study For Supersonic Tactical Missile (STM), Volume I, by Robert H. Josselson, p. 21, 24 March 1978.
4. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume I, by John W. Diesel and others, p. 1-1.
5. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume II, by John W. Diesel and others, p. C-1-C-37.
6. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume II, by John W. Diesel and others, p. C-38-C-51.
7. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume I, by John W. Diesel and others, p. 3-18.
8. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume I, by John W. Diesel and others, p. 3-16.
9. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume I, by John W. Diesel and others, p. 3-37.
10. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume I, by John W. Diesel and others, p. 3-10.
11. Naval Postgraduate School Technical Note 0141-25, User's Guide to Programming Aids Library, by Lloyd G. Nolan, p. 27-29, October 1978.
12. Honeywell Defense Systems Division, System Design Tradeoff Study For Supersonic Tactical Missile (STM), Volume I, by Robert H. Josselson, 24 March 1978.

13. Honeywell Defense Systems Division, Systems Design Tradeoff Study For Supersonic Tactical Missile (STM), Volume I, by Robert H. Josselson, p. 77-82, 24 March 1978.
14. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume I, by John W. Diesel and others, p. 4-41-4-83.
15. Litton Systems, Inc., Guidance and Control Systems Division, Modularized Six-Degree-Of-Freedom (MOD6DF) Computer Program, Volume II, by John W. Diesel and Others, p. B-1 - B-49.

INITIAL DISTRIBUTION LIST

	No. Copies
1. Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
2. Department Chairman, Code 67 Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93940	1
3. Professor D. J. Collins, Code 67CO (thesis advisor) Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93940	2
4. LT Thomas Alan Grote, USN (student) 1016 Orchid Avenue Modesto, California 95355	1
5. Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2