AD-A105 788 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 16/4
COMPUTER PROGRAM APPLICATIONS TO TACTICAL MISSILE CONCEPTUAL DE--ETC(U)
JUN 81 M D SULLIVAN

UNCLASSIFIED NL

1 of 3
AD A
105788

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

COMPUTER PROGRAM APPLICATIONS TO
TACTICAL MISSILE CONCEPTUAL DESIGN

by

Martin David Sullivan

June 1981

Thesis Advisor: Gerald H. Lindsey

Approved for public release, distribution unlimited

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO. AD-A105 788	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Computer Program Applications to Tactical Missile Conceptual Design		5. TYPE OF REPORT & PERIOD COVERED Master's Thesis June 1981
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Martin David Sullivan		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 12) 200
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940		12. REPORT DATE June 1981
		13. NUMBER OF PAGES 199
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited

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

18. SUPPLEMENTARY NOTES

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

Computer programs for warhead design, missile guidance, propulsion and aerodynamic coefficients.

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

This thesis is comprised of four independent computer programs and their related operating instructions. Each of these programs focuses on a particular facet of tactical missile design. The topics covered include guidance and trajectory calculations, rocket motor propulsion sizing, warhead design, and aerodynamic coefficient determination. The programs are developed from accepted mathematical procedures and are tailored to optimize operator interaction for educational purposes. This thesis is intended to be utilized

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
(Page 1) S/N 0102-014-6601

UNCLASSIFIED 251450

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Block 20 continued.
as a supplement to the thesis Tactical Missile Conceptual Design by D. R. Redmon, Naval Postgraduate School, September 1980.

Approved for public release, distribution unlimited

Computer Program Applications to Tactical Missile Conceptual Design

by

Martin David Sullivan
Lieutenant, United States Navy
B.S., Georgia Institute of Technology, 1975

Submitted in partial fullfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
June 1981

Author: Martin David Sullivan

Approved by: D. Lindsey
Thesis Advisor

D Lindsey for M. F. Platzer
Chairman, Department of Aeronautics

William M. Tolles
Dean of Science and Engineering

ABSTRACT

This thesis is comprised of four independent computer programs and their related operating instructions. Each of these programs focuses on a particular facet of tactical missile design. The topics covered include guidance and trajectory calculations, rocket motor propulsion sizing, warhead design, and aerodynamic coefficient determination. The programs are developed from accepted mathematical procedures and are tailored to optimize operator interaction for educational purposes. This thesis is intended to be utilized as a supplement to the thesis Tactical Missile Conceptual Design by D.R.Redmon, Naval Postgraduate School, September 1980.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGEMENTS

I am deeply obligated to Professor Gerald H. Lindsey both for his faith in my abilities and for his guidance in the preparation of this thesis. He allowed me to work freely in an area of great interest to me and provided the proper direction and assistance to help me understand the obvious.

The greatest appreciation must be given to my wife, Kathleen, who has learned to live without her husband quite bravely during the preparation of this thesis. Her support, understanding, and love have been incomparable and indispensible during this trying period.

Finally, a certain acknowledgement must be provided to the staff of the Naval Postgraduate School Computer Center who, in the process of installing and debugging a whole new computer system, have made this thesis a most exciting and memorable undertaking.

I. INTRODUCTION

The programs contained in this thesis were assembled expressly to supplement the work of Dan Redmon in his thesis, Tactical Missile Conceptual Design. Two of the programs, LPATH, the trajectory model, and LAERO1, the aerodynamic coefficients program, originated in Redmon's thesis and were expanded/modified for use on the Naval Postgraduate School's new IBM 370 computer system. The other two programs were generated for this thesis and utilize the procedures and principles outlined by Redmon.

The specific intention of these programs is to provide students of tactical missile engineering and design with a method of solving complex mathematical routines rapidly and interactively. Each of the programs request data which are likely to be used as design parameters for the topics concerned. The programs also allow repeated operation with input alteration capability, allowing the user to personally optimize his design. This approach was chosen to allow students to understand the relationships various input parameters have with the final solutions.

II. TRAJECTORY MODELS

A. DESCRIPTION AND ORIGIN OF PROGRAM

This program applies the principles of missile guidance laws to the terminal phase (the last 5 to 10 seconds) of a missile trajectory in order to determine the maximum normal acceleration on the missile for a given scenario. Of the three general guidance law categories, pursuit guidance is not included in the program capability. It has been found that pursuit guidance invariably produces a tail-chase situation, greatly reducing an anti-air missile's effectiveness against maneuvering targets of similar speed characteristics. Line-of-sight guidance and proportional navigation guidance are both options of the program.

Figure (II-1) shows a typical encounter geometry as required for this program. The encounter is considered to occur entirely within a two- dimensional plane. No differentiation is required or assumed concerning the orientation of the encounter plane. The plane may be at any angle to the horizontal as desired by the program user. The reference direction is an arbitrary choice by the program user. The angles shown are positive in value, however the program does not require positive angles. If TAL were 150 degrees, it could also be entered as -210 degrees. The IRA term represents the initial range to the target.

Tangential velocities (air speeds) of the missile and the target are considered by the program to be constant throughout the problem. Since the program concerns itself with only the final moments of a trajectory, this is a reasonable consideration. Target normal accelerations, when specified by the user, are also held constant throughout the problem for the same reason. The missile normal

Program Variables

Ø=LOS (line of sight angle)
αt=TAL (target alpha)
αm=MAL (missile alpha)
Vt=TSP (target speed)
Vm=MSP (missile speed)
Rt=IRT (initial range to target from missile launch point)
Rm=IRM (initial range to missile from launch point)

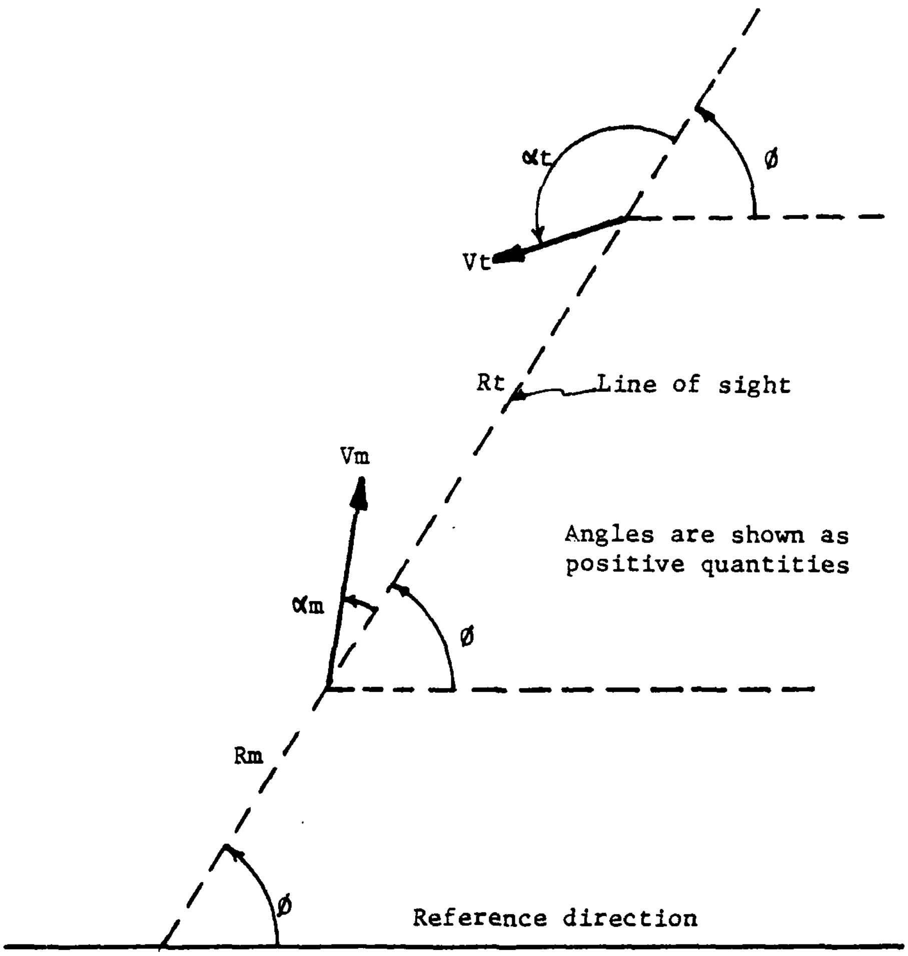


Figure (II-1). Encounter geometry

acceleration, however, is a function of the guidance laws and will vary according to the resulting flight path requirements. A constraint on the encounter is that the missile normal acceleration must be zero at the start of the problem.

The program "flies" both the missile and the target as simple points in space with no consideration given to aerodynamics. The missile will always strike the target dead center when it is given the proper speed advantage for the encounter since there is no provision for statistical miss analysis. The unbounded nature of the missile normal acceleration allows the missile to turn as rapidly as necessary to hit the target.

This program analyzes the given encounter by time increment calculations. As is the case with all integrations conducted by incremental steps, the accuracy of the results is a function of the increment size. The results will become increasingly accurate as the time increment is made smaller. However, as the time increment decreases in size, the length of the output becomes increasingly longer. The user must balance the desire for accuracy against the amount of time he wishes to spend on the computer terminal.

The primary output is a tabular listing of the missile and target coordinates at each time increment. The coordinate frame is established within the encounter plane with the abscissa oriented along the reference direction. The problem stops once the missile has passed its closest point of approach to the target. Output then provides the time to intercept from time of problem initiation and the maximum acceleration the missile was required to endure. A Versatec plot of overlaying successive encounters is an optional output.

This program originated as two seperate BASIC programs written by Redmon [Ref. 1] for use on the HP 9830 desktop calculator. It was subsequently translated into FORTRAN IV for use on the IBM 370 computer system.

B. USER INSTRUCTIONS

If it is desired to abort the operation of this program prematurely, two methods are available. When the program is waiting for data entry, press $\boxed{\text{ENTER}}$. When the program is not waiting for data entry but is processing, type "HX" and press $\boxed{\text{ENTER}}$. Both actions will return the terminal to CMS mode.

When the screen becomes full, or "MORE...." appears in the status area, clear the screen by pressing $\boxed{\text{ALT}}$ and $\boxed{\text{CLEAR}}$ simultaneously. At several points in the program, the user will be directed to "CLEAR SCREEN AND INPUT 0". This is to provide proper positioning of the output on the screen for ease of reading. If any other symbol than "$\boxed{4}$ A ///" should appear in the lower left of the screen, press $\boxed{\text{RESET}}$ and continue.

1. Turn the terminal on with the red $\boxed{\text{I O}}$ switch.

2. When the large "NPS" appears after about 30 seconds, press $\boxed{\text{RESET}}$, then press $\boxed{\text{ENTER}}$.

3. When "CP READ" appears in the status area on the lower right of the screen, type "L nnnnP", where nnnn is your 4-digit user number, then press $\boxed{\text{ENTER}}$.

4. You will now be asked for your password. Type it in (the characters will not appear on the screen), then press $\boxed{\text{ENTER}}$.

5. Your personal file must contain a PROFILE EXEC routine with the appropriate Fortran GLOBAL statement. If it does not, type "GLOBAL TXTLIB FORTMOD2 MOD2EEH", then press $\boxed{\text{ENTER}}$.

6. Type "LINK TO xxxxP 191 AS 192 RR", where xxxx is the 4-digit user number for the project file, then press $\boxed{\text{ENTER}}$.

7. You will now be asked for the project file password. Type it in (the characters will not appear on the screen), then press $\boxed{\text{ENTER}}$.

8. Type "ACCESS 192 B" and press $\boxed{\text{ENTER}}$.

9. Press $\boxed{\text{ALT}}$ and $\boxed{\text{CLEAR}}$ simultaneously to clear screen.

10. Type "LPATH" and press $\boxed{\text{ENTER}}$.

11. Input the following data as it is requested on the screen by typing the data and then pressing $\boxed{\text{ENTER}}$. Ensure that the data is input as either decimal or integer as specified. The terms in parenthesis below are the program variable names.

Parameter	Units	Value range
Trajectory option (TITLE)	none	0=Line-of-sight 1=Proportional navigation 2=Both
Time increment (DEL)	seconds	Larger than .0001 times the problem time
Navigation constant (NAV)	none	2.5 to 6.5
LOS Angle (LOS)	degrees	0.0 to 360.0
Initial target range from the launch site (IRT)	meters	Larger than the missile range
Target speed (TSP)	m/sec	Larger than 0.0
Target normal acceleration (TAC)	m/sec/sec	Positive is to target's left
Initial target angle to line of sight (TAL)	degrees	0.0 to 360.0 the
Missile speed (MSP)	m/sec	Larger than 0.0
Initial missile range from launch site (IRM)	meters	Such that impact occurs in less than 10 seconds

12. If you desire to rerun the problem, or want to run a new problem, answer the questions appropriately when asked by the terminal after the execution of the current problem.

13. To receive the printout and plot of your encounters, answer "no" to rerunning or restarting the problem when asked by the terminal and follow the directions presented on the screen.

14. Upon completion of the program, type "LOGOFF" and press $\boxed{\text{ENTER}}$.

15. Turn the terminal off with the red $\boxed{\begin{matrix}\text{I}\\ \text{O}\end{matrix}}$ switch.

C. EXAMPLE PROBLEMS

1. <u>Example II-A</u>. <u>Line-of-sight Non-maneuvering Crossing Target</u>

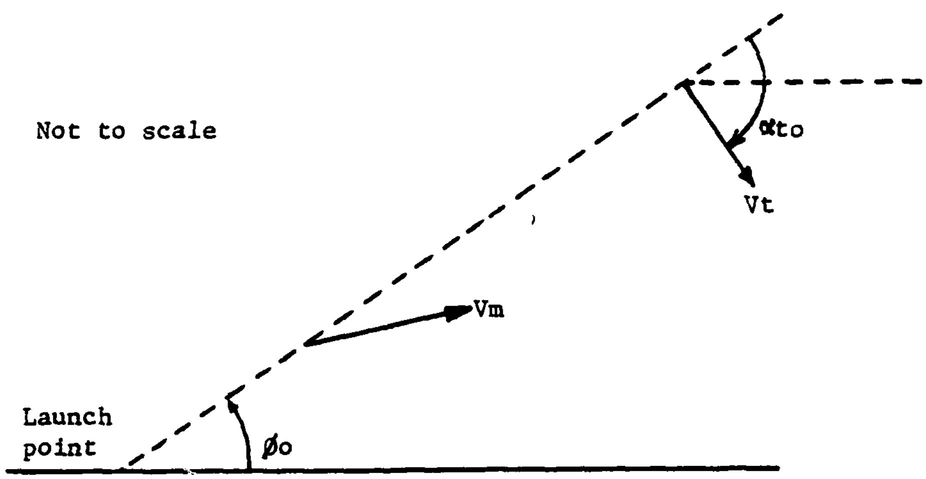


Figure (II-2). Non-maneuvering crossing target

Rt=10000 meters

Rm=9000 meters

Øo=30.0 degrees

αto=-90.0 degrees

at=0

Vt=200 meters/second

Vm=800 meters/second

TABLE (II-1). OUTPUT OF EXAMPLE II-A

```
$$$$ RUN NUMBER  1
                    PROBLEM PARAMETERS
         01) TIME INCREMENT               0.0050 SECONDS
         02) NAVIGATION CONSTANT          0.0
         03) LINE-OF-SIGHT ANGLE         30.000 DEGREES
         04) INITIAL SEPARATION        1000.000 METERS
         05) TARGET SPEED               200.000 METERS/SEC
         06) TA GET ACCELERATION          0.0   M/SEC/SEC
         07) TARGET ALPHA               -90.000 DEGREES
         08) MISSILE SPEED              800.000 METERS/SEC
         09) MISSILE INITIAL ALPHA      -13.003 DEGREES

        LINE OF SIGHT GUIDANCE INTERCEPT TRAJECTORY
PROBLEM  --POSITION COORDINATES (METERS)--       RANGE      ACCEL.
TIME(S)      XM        YM        XT        YT   (METERS)   (M/S/S)
 0.0         0.0       0.0     866.0     500.0      0.0       0.0
 0.125      95.7      29.0     878.5     478.3    902.6    -31.982
 0.250     191.6      57.5     891.0     456.7    805.4    -31.962
 0.375     287.5      85.5     903.5     435.0    708.2    -31.942
 0.500     383.7     113.1     916.0     413.4    611.2    -31.921
 0.625     479.9     140.1     928.5     391.7    514.3    -31.899
 0.750     576.3     166.7     941.0     370.1    417.5    -31.876
 0.875     672.9     192.8     953.5     348.4    320.9    -31.852
 1.000     769.5     218.4     966.0     326.8    224.3    -31.828
 1.125     866.3     243.6     978.5     305.1    127.9    -31.803
 1.250     963.2     268.3     991.0     283.4     31.7    -31.777
 1.290     994.3     276.0     995.0     276.5      0.9    -31.768
          MAXIMUM LATERAL ACCELERATION ON THE MISSILE
          IS    -32.000 M/SEC/SEC, OR     -3.26 G"S.
          THE TIME TO INTERCEPT IS  1.295 SECONDS.
```

Figure (II-3). Versatec plot of example II-A

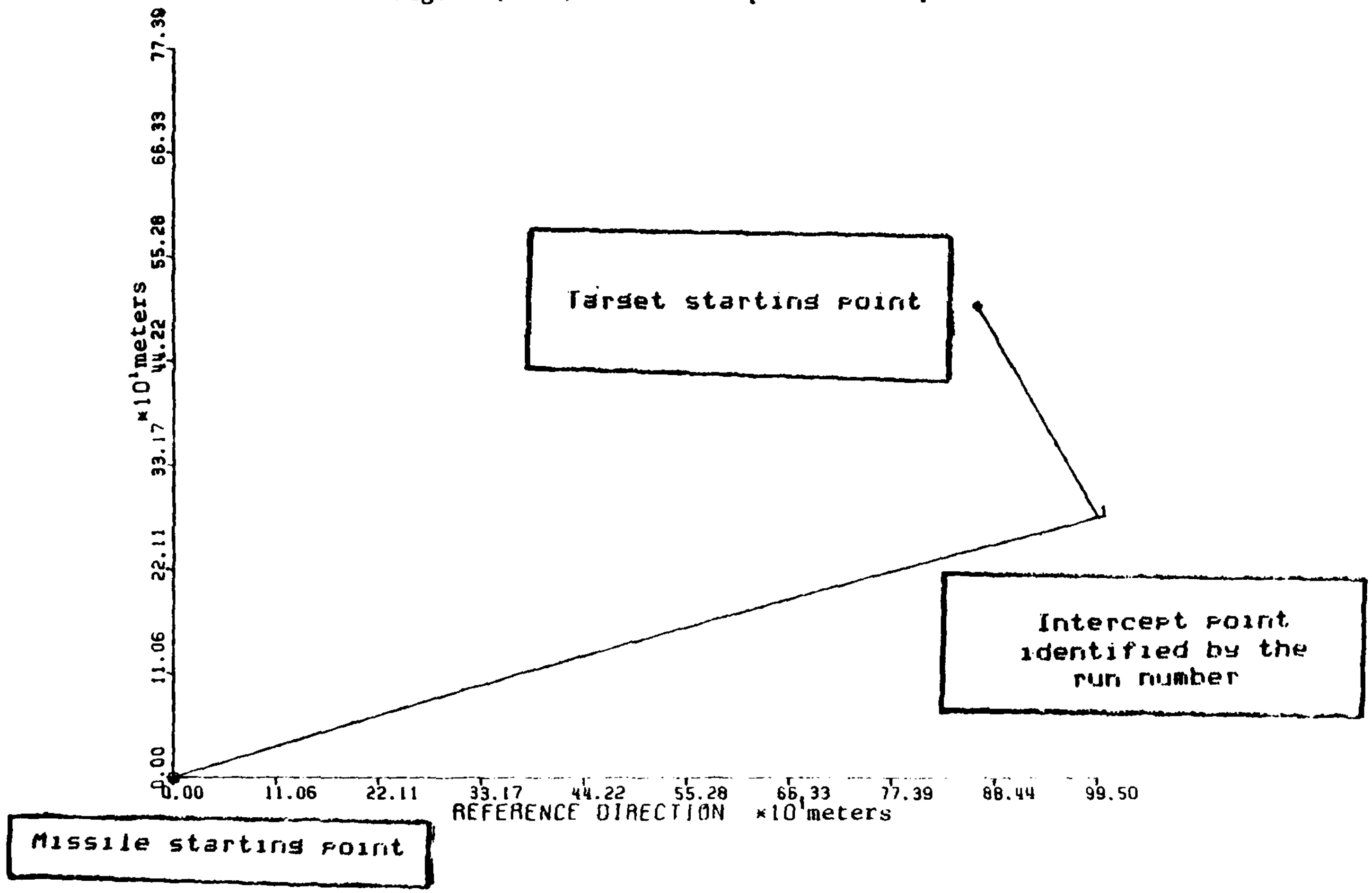


Table (II-1) is the corresponding computer output for this encounter. As indicated, the missile maximum normal acceleration is

$$a_m = -32.00 \text{ m/sec/sec} \quad \text{or } -3.26 \text{ g's.}$$

Figure (II-3) is the Versatec plot of the engagement.

2. <u>Example II-B</u>. <u>Proportional Navigation Maneuvering Crossing Target</u>

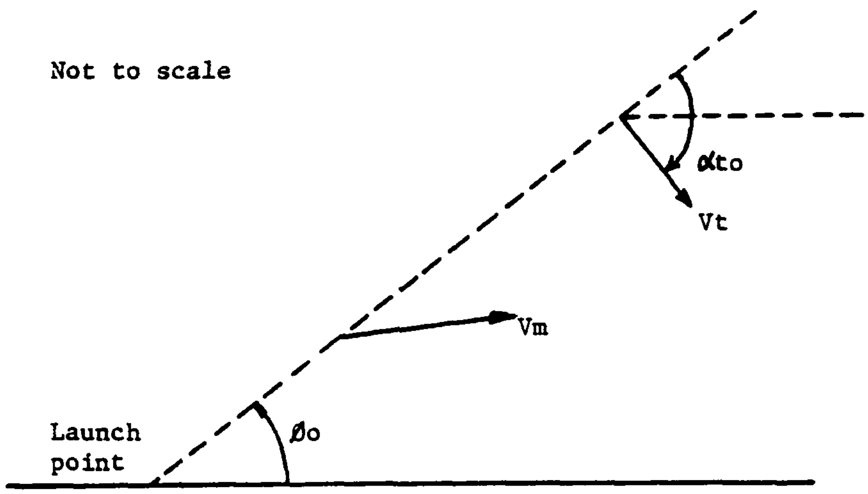


Figure (II-4). Maneuvering crossing target

Rt=10000 meters

Rm=9000 meters

Øo=30.0 degrees

αto=-90.0 degrees

at=156.8 m/sec/sec (16 g's)

Vt=200 meters/second

Vm=800 meters/second

Navigation constant=3.06

TABLE (II-2). OUTPUT OF EXAMPLE II-B

```
$$$$ RUN NUMBER  1
                    PROBLEM PARAMETERS
          01) TIME INCREMENT                 0.0050 SECONDS
          02) NAVIGATION CONSTANT            3.060
          03) LINE-OF-SIGHT ANGLE           30.000 DEGREES
          04) INITIAL SEPARATION          1000.000 METERS
          05) TARGET SPEED                 200.000 METERS/SEC
          06) TA GET ACCELERATION          156.800 M/SEC/SEC
          07) TARGET ALPHA                 -90.000 DEGREES
          08) MISSILE SPEED                800.000 METERS/SEC
          09) MISSILE INITIAL ALPHA        -14.478 DEGREES
```

PROPORTIONAL NAVIGATION GUIDANCE INTERCEPT TRAJECTORY

PROBLEM TIME(S)	--POSITION COORDINATES (METERS)-- XM	YM	XT	YT	RANGE (METERS)	ACCEL. (M/S/S)
0.0	0.0	0.0	866.0	500.0	1000.0	0.0
0.125	96.4	26.8	879.5	479.0	904.3	2.326
0.250	192.7	53.6	895.0	459.4	811.1	9.733
0.375	289.0	80.5	912.3	441.4	720.3	22.483
0.500	385.2	107.8	931.4	425.1	631.7	40.670
0.625	481.2	135.7	951.9	410.9	545.2	64.170
0.750	577.0	164.6	973.7	398.6	460.6	92.587
0.875	672.3	194.8	996.6	388.6	377.8	125.188
1.000	767.0	226.9	1020.3	380.9	296.5	160.838
1.125	860.8	261.4	1044.7	375.5	216.4	197.903
1.250	953.6	298.8	1069.5	372.6	137.5	234.093
1.375	1044.9	339.5	1094.5	372.1	59.4	266.065
1.470	1113.2	372.9	1113.5	373.3	0.5	283.935

MAXIMUM LATERAL ACCELERATION ON THE MISSILE
IS 283.935 M/SEC/SEC, OR 28.95 G"S.
THE TIME TO INTERCEPT IS 1.475 SECONDS.

Figure (II-5). Versatec plot of example II-B

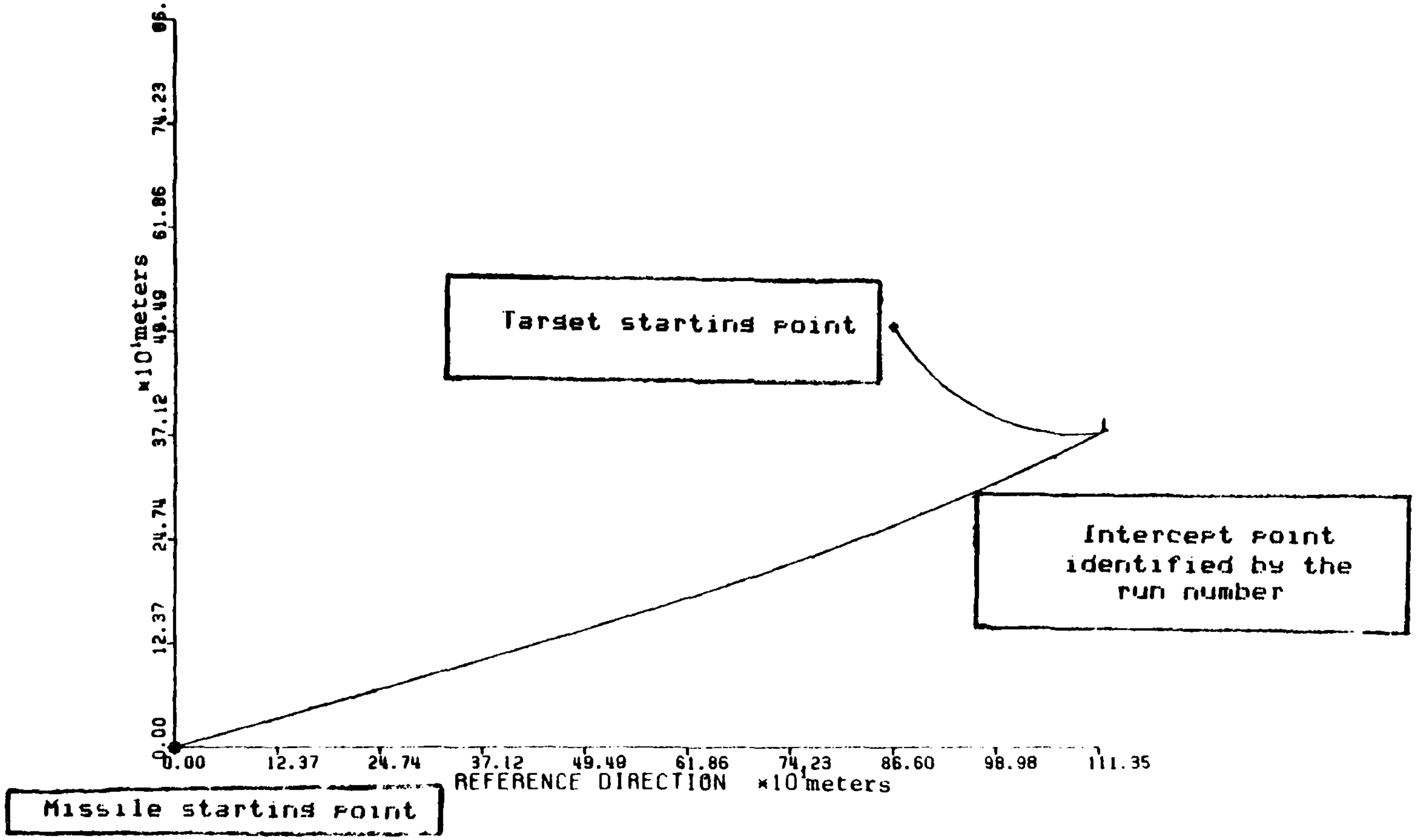
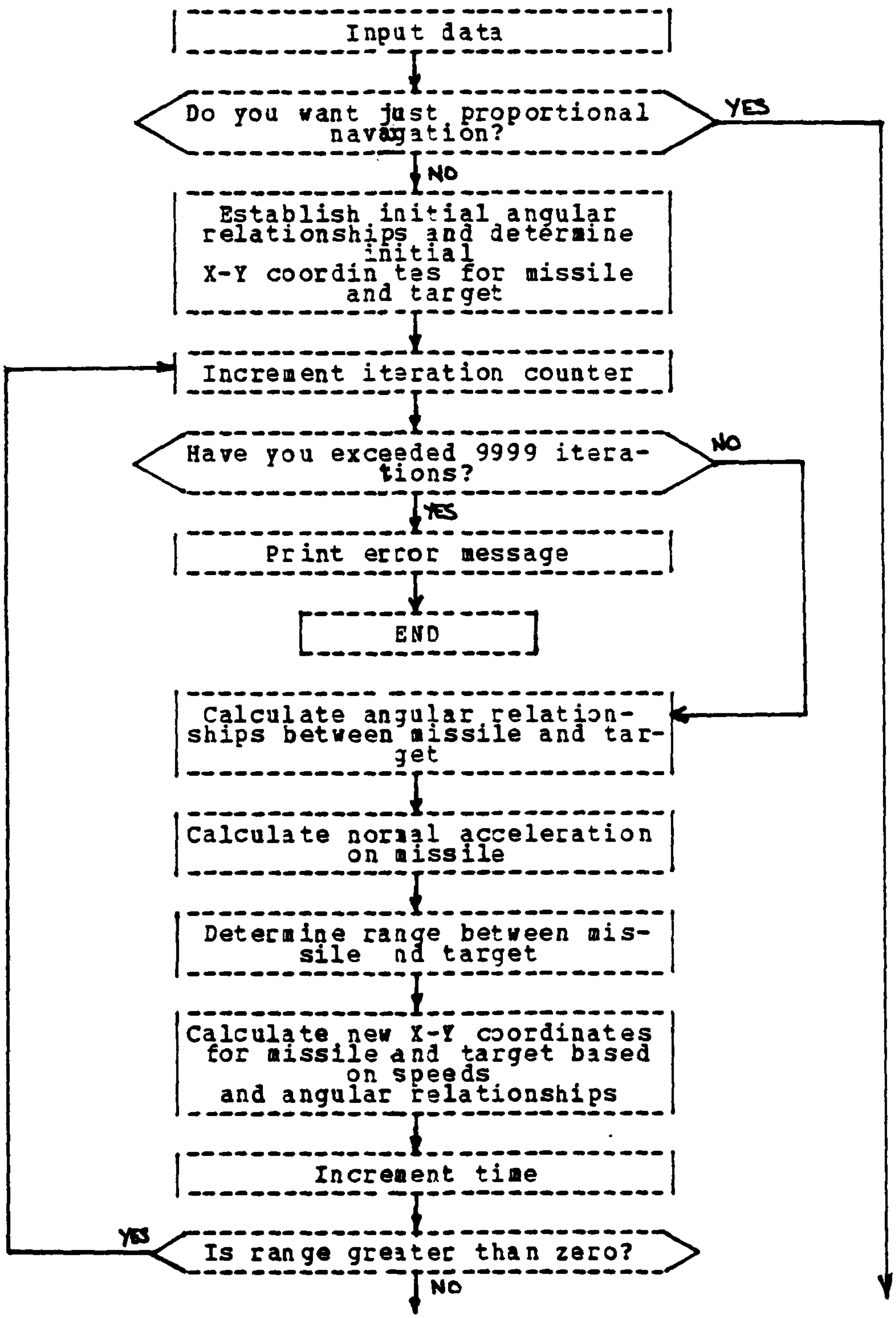


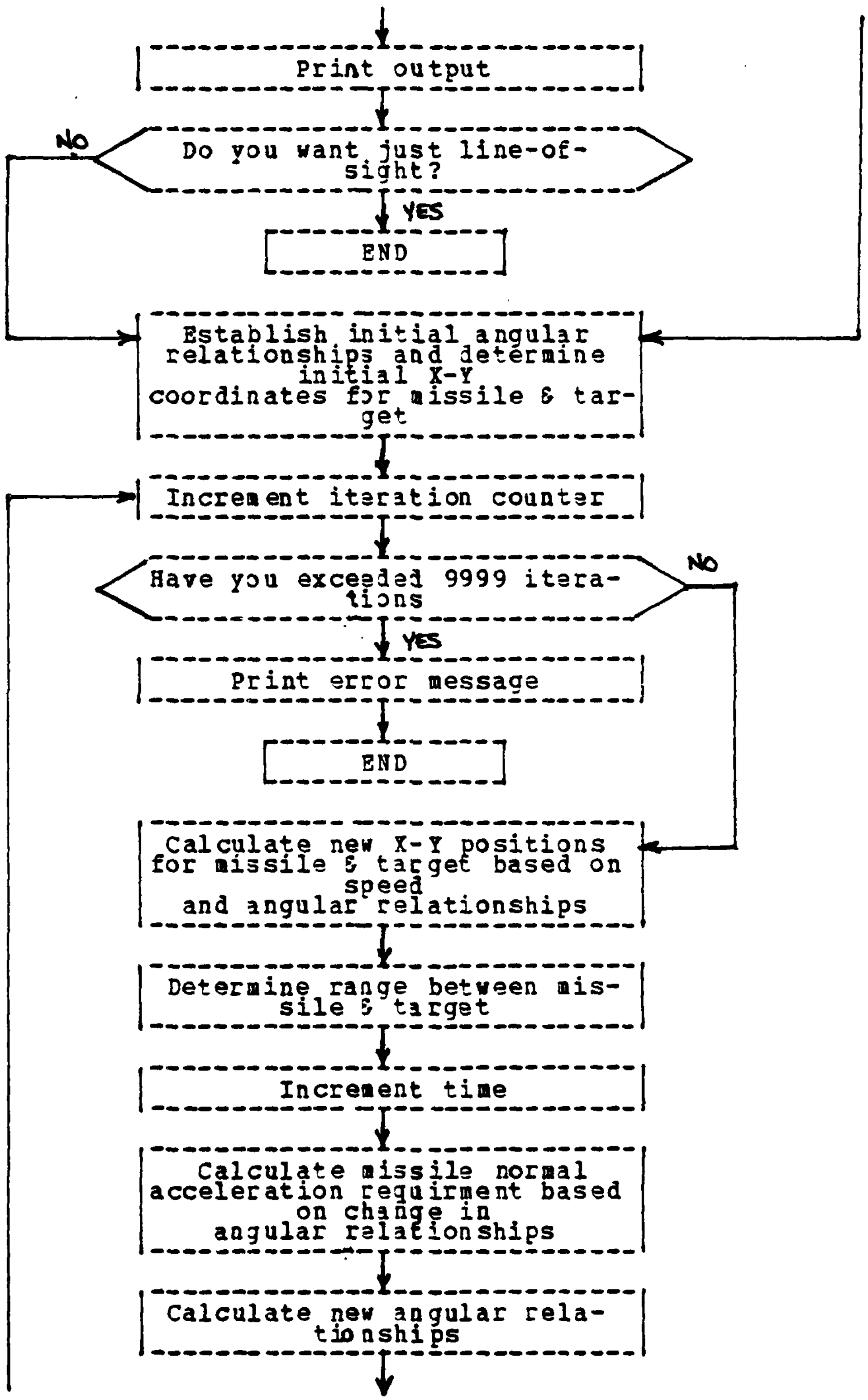
Table (II-2) is the computer output for this encounter. The missile maximum normal acceleration is

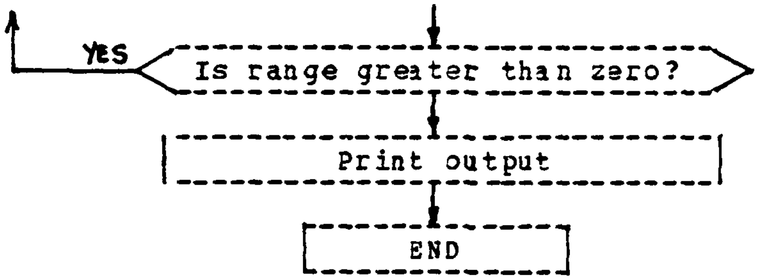
$$a_m = 283.94 \text{ m/sec/sec} \quad \text{or} \quad 28.95 \text{ g's.}$$

Figure (II-5) is the Versatec plot of the engagement.

D. PROCEDURAL FLOWCHART




Print output
Do you want just line-of-sight?
NO
YES
END
Establish initial angular relationships and determine initial X-Y coordinates for missile & target
Increment iteration counter
Have you exceeded 9999 iterations
NO
YES
Print error message
END
Calculate new X-Y positions for missile & target based on speed and angular relationships
Determine range between missile & target
Increment time
Calculate missile normal acceleration requirment based on change in angular relationships
Calculate new angular relationships



E. PROGRAM CHANGES

1. Language Translation.

The two BASIC programs contained in Reference 1 were translated into standard FORTRAN IV.

2. Program Condensation.

The two individual programs were combined to form a single integrated routine which allows the user to choose either of the two guidance laws or both for a given encounter. The two original programs were meshed such that only the actual guidance law algorithms are separate routines, all input and output routines are now common.

3. Input and Output Facility.

The data input instructions were modified to maximize user facility on the IBM 370 System 3278 terminals. Completion of data input is now followed by data feedback for user verification prior to actual program execution.

Data output has been expanded to provide data delivery to three destinations; to the user's terminal for observation, to a printer file for hardcopy duplication of the terminal display, and to a plot data file for subsequent use by the plot program. The destinations are options chosen by the user for each execution of the program. Up to nine different problems can be printed and plotted each time the program is entered.

The program can now be rerun multiple times without exiting and re-entering each time. The user has a choice of either rerunning the same problem or initiating a new problem completely.

4. Plot Program.

PATHPLOT FORTRAN was developed to produce a Versatec plot picture of the encounter. It will produce a single plot each time the program is entered and will plot up to 9 engagements in an overlaying manner. This format was adopted to allow comparisons of successive input data modifications.

5. Data Overcapacity Check.

If the user initiates a problem requiring more than 9999 iterations, the program will stop. The user will be notified of the error and given the opportunity to rerun the problem.

6. Initial Missile Acceleration.

Initial missile acceleration was removed as a user input variable and established as zero. Due to the mathematical nature (i.e., no physical constraints) of the program, any "wrong" accelerations of the missile in the initial state were immediately "corrected" by the algorithm. The model is better served by providing no initial accelerations.

7. Theta Angles.

Both the target and the missile theta angles (the angles between the velocity vectors and the reference direction) were removed as user input variables. The program now calculates the theta angles from other input data, reducing redundancy and possible contradiction of input data.

8. Initial Conditions Perspective.

Originally, the missile guidance calculations started at t=0. Specifically, the anchor point for the line of sight solution was the point where the missile commenced the problem, whether or not that was on the launch site. This produced a situation removed from reality where the LOS anchor point should be at the fire control location, usually at or near the launch site. A similar, though less pronounced, condition existed for the proportional navigation solution. The program was modified to provide proper positioning of the external guidance reference. As a result of the modification, an additional output is the correct lead angle for the missile at the start of the problem. This angle is based on the assumption that neither the target nor the missile have maneuvered prior to time t=0.

III. WARHEAD DESIGN

A. DESCRIPTION AND ORIGIN

This program develops a warhead using the same methods as presented by D Redmon [Ref. 1]. However, its capability is somewhat greater and it applies the relationships in a slightly different manner. This program was written for this thesis in FORTRAN IV for use on the IBM 370 computer system 3278 terminals. It is configured expressly for operator-computer interaction.

It starts with an initial input of data from which a table of fragment initial velocities is generated and presented to the user. From this, the user selects values for fragment mass and impact velocity and another table is generated and displayed. From this one, the desired probability of hit given a detonation is selected and the final solution is produced. At various points during the operation of the program, the user will have opportunities to alter or revise selected parameters.

The program is limited to a cylindrical warhead with either a solid or hollow core. The fragments are required to be square in shape and are sized by the program. Figure (III-1) illustrates the shape of the warhead and the location of various input and output quantities.

Initially, the target altitude is used to determine the atmospheric density, temperature and speed of sound. These values are, in turn, used to calculate required initial velocities for the fragments. The program is preloaded with various values for the fragment mass and impact velocity, which are used to generate the table of initial velocities. These velocities come from the following relationship:

$$V(hit)=Vi[exp(-ks)]$$

$$k=\frac{1}{2m}\rho_a A C_d$$

where V(hit) is the impact velocity, Vi is the initial velocity, s is the kill radius, m is the fragment mass, ρ_a is the atmospheric density, A is the plan area of the fragment, and Cd is the drag coefficient of the fragment.

The ballistic limit velocity is calculated for the various presized fragments and is provided as a reference when choosing an appropriate impact velocity. The ballistic limit velocity is that velocity at which one half of the fragments will penetrate the target's skin and the other half will not. The empirical relationship, developed by A. E. Fuhs [Ref. 6], presents the ballistic limit velocity as a function of the fragment size to skin thickness ratio. His function dealt with steel fragments impacting an aluminum plate. His results were modified to qualitatively account for different skin materials and fragment densities.

Next computed is the fragment spray angle and the critical miss distance. The spray angle is a function of the initial velocity, the detonation velocity and the warhead length. The critical miss distance is defined as the range where the fragment spray exactly covers the entire target. The critical miss distance is used by the program to separate the two functions which determine the average number of hits received by the target. The program assumes the target is always centered within the fragment spray and aligned perpendicular to the central ray of the spray.

A selection of warheads is then produced, one for each of a preloaded set of Pd's (probability of a hit given a detonation) to provide the user with a Pd versus warhead weight/size trade off comparison. This sizing process is based entirely on the following relationship:

$$Pd=1-\exp(-a)$$

where a is the average number of hits. The variable a, as shown by Redmon [Ref. 1], is a function of the cube of the warhead radius. The user then chooses a desired Pd which, in turn, produces the final warhead sizing.

Values that are shown in tabular form for user selection and input into the program are not limited to those displayed. Any value in between the displayed values or completely outside of the value range may be used. The one exception is that Pd can never be selected to be larger than .999 and may even, if forced by the program limitations, be required to be lower. Since ultimately in this algorithm, Pd is a function of the warhead radius, the maximum value for Pd may be reduced in order to keep the radius within the original input parameters. The user is notified if this condition occurs.

Useful reference information for some common explosives and case materials is contained in the following tables.

Table (III-1). Characteristics of common explosives

Explosive	Density (lb/cu.in)	2E(ft/sec)	Vd (ft/sec)
TNT	.0574	7600.	21785.
RDX	.0596	9300.	26837.
HMX	.0665	10230.	29934.
PETN	.0625	9300.	27231.
TETRYL	.0585	8200.	25755.
COMP B	.0607	8800.	25722.
OCTOL	.0650	9500.	28356.

Table (III-2). Densities of common case materials

Case material	Density (lb/cu.in)
Steel	.283
Aluminum	.100
Uranium	.688
Titanium	.163
Lead	.410

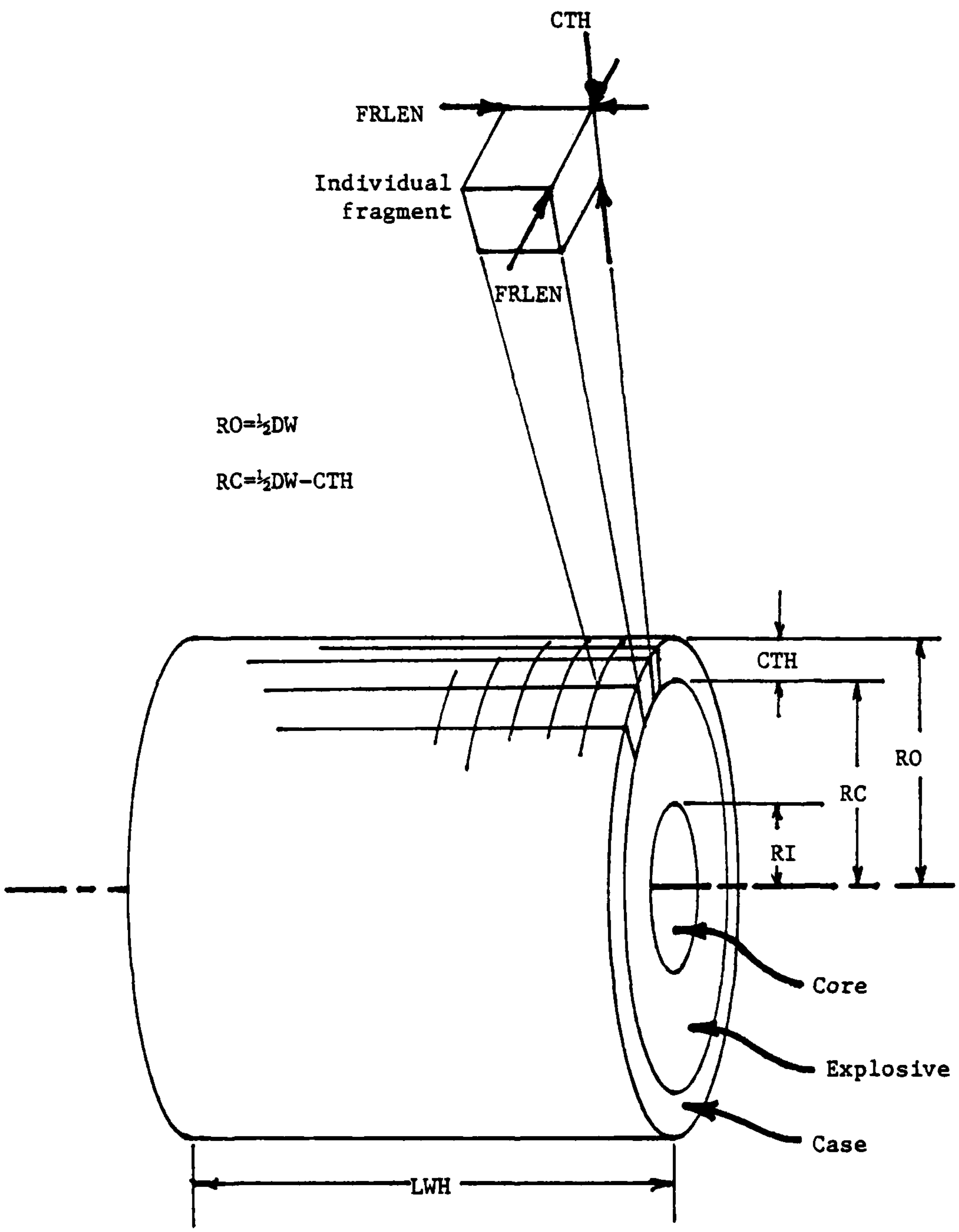


Figure (III-1). Form of warhead as used by the program

B. USER INSTRUCTIONS

If it is desired to abort the operation of this program prematurely, two methods are available. When the program is waiting for data entry, press $\boxed{\text{ENTER}}$. When the program is not waiting for data entry but is processing, type "HX" and press $\boxed{\text{ENTER}}$. Both actions will return the terminal to CMS mode.

When the screen becomes full, or "MORE...." appears in the status area, clear the screen by pressing $\boxed{\text{ALT}}$ and $\boxed{\text{CLEAR}}$ simultaneously. At several points in the program, the user will be directed to "CLEAR SCREEN AND INPUT 0". This is to provide proper positioning of the output on the screen for ease of reading. If any other symbol than "$\boxed{4}$ A" should appear in the lower left of the screen, press $\boxed{\text{RESET}}$ and continue.

1. Turn the terminal on with the red $\boxed{\text{I O}}$ switch.

2. When the large "NPS" appears after about 30 seconds, press $\boxed{\text{RESET}}$, then press $\boxed{\text{ENTER}}$.

3. When "CP READ" appears in the status area on the lower right of the screen, type "L nnnnP", where nnnn is your 4-digit user number, then press $\boxed{\text{ENTER}}$.

4. You will now be asked for your password. Type it in (the characters will not appear on the screen), then press $\boxed{\text{ENTER}}$.

5. Your personal file must contain a PROFILE EXEC routine with the appropriate Fortran GLOBAL statement. If it does not, type "GLOBAL TXTLIB FORTMOD2 MOD2EEH", then press $\boxed{\text{ENTER}}$.

6. Type "LINK TO xxxxP 191 AS 192 RR", where xxxx is the 4-digit user number for the project file, then press $\boxed{\text{ENTER}}$.

7. You will now be asked for the project file password. Type it in (the characters will not appear on the screen), then press $\boxed{\text{ENTER}}$.

8. Type "ACCESS 192 B" and press [ENTER].

9. Press [ALT] and [CLEAR] simultaneously to clear screen.

10. Type "LBOMB" and press [ENTER].

11. Input the following data as it is requested on the screen by typing the data and then pressing [ENTER]. Ensure that the data is input as either decimal or integer as specified. The terms in parenthesis are the program variable names. Input the following variables as decimal values:

Explosive density (XDEN)	lb/cu.in
Explosive Gurney constant (GC)	ft/sec
Explosive detonation velocity (VD)	ft/sec
Case material density (CDEN)	lb/cu.ft
Desired kill radius (RKILL)	feet
Warhead diameter (DW)	inches
Warhead length-to-diameter ratio (LTD)	
Target length (LT)	feet
Target mean width (WT)	feet
Target vulnerability, P(k/h) (PKH)	
Target altitude (ALT)	feet
Target skin thickness (TTH)	inches
Target skin material (MAT)	1.0=Aluminum 2.0=Fiberglass/Kevlar 3.0=Steel

12. After entering the above data, you will be presented with an initial velocity table built around your desired kill radius. The initial velocities will be listed as a function of fragment mass and impact velocity. Also provided will be the ballistic limit velocities for each of the fragment masses. Input the following parameters as decimal values:

Fragment mass (IFMLB(1))	grains
Impact velocity (VHIT(1))	ft/sec

13. You will now be presented with a shopping list of warheads developed for a range of Pd's. The warheads are described by the following quantities:

Warhead weight in pounds (total weight)
Case thickness in inches
Core diameter in inches
Number of fragments from the warhead
Number of fragments hitting the target
Edge length of the fragments in inches

Input the following parameter as a decimal value:

Desired probability of a hit given a detonation (PDF)

14. If you desire to rerun the problem, or want to run a new problem, answer the questions appropriately when asked by the terminal after the execution of the current problem.

15. To receive the printout of your encounters, answer "nc" to rerunning or restarting the problem when asked by the terminal and follow the directions presented on the screen.

16. Upon completion of the program, type "LOGOFF" and press $\boxed{\text{ENTER}}$.

17. Turn the terminal off with the red $\boxed{\substack{\text{I}\\ \text{O}}}$ switch.

C. EXAMPLE PROBLEMS

1. <u>Example III-A</u>

It was desired to build a warhead which would kill a typical cruise missile flying at 100 feet altitude. The warhead was required to have a lethal radius of 50 feet with a Pd of 0.98. TNT was selected for the explosive and steel was chosen for the case material. The diameter of the missile was 13.5 inches.

Table (III-3) outlines the input parameters. Table (III-4) shows the program output. After the initial velocity table was displayed at the terminal, the fragment mass was chosen to be 130 grains and the impact velocity was selected to be 2500 feet per second. After the Pd table was displayed, 0.98 was entered as the kill probability.

2. Example III-B

A warhead was required which would kill a typical manned aircraft at 30000 feet. A lethal radius of 75 feet was specified. The warhead was limited in weight to 50 pounds and needed to have a core diameter of at least 5 inches. The diameter of the missile was 10.0 inches. HMX was chosen as the explosive and depleted uranium as the case material.

Table (III-5) outlines the input parameters. Table (III-6) shows the program output. After the initial velocity table was displayed at the terminal, the fragment mass was chosen to be 210 grains and the impact velocity was selected to be 5000 feet per second. After the Pd table was displayed, it was determined that the fragments were too large and the fragment mass was then reduced to 100 grains. The impact velocity was also reduced to 3000 feet per second. When the Pd table was redisplayed, 0.995 was chosen as the desired kill probability.

TABLE (III-3). INPUT DATA FOR EXAMPLE III-A

THE FOLLOWING IS A SUMMARY OF THE INPUT DATA:

01)	EXPLOSIVE DENSITY	0.05740	LB/CU.IN
02)	EXPLOSIVE GURNEY CONSTANT	7600.00	FT/SEC
03)	EXPLOSIVE DETONATION VELOCITY	21785.00	FT/SEC
04)	CASE MATERIAL DENSITY	0.2830	LB/CU.IN
05)	DESIRED KILL RADIUS	50.0	FEET
06)	WARHEAD DIAMETE	13.50	INCHES
07)	WARHEAD LENGTH-TO-DIAMETER RATIO	2.50	
08)	TARGET LENGTH	36.00	FEET
09)	TARGET WIDTH	3.50	FEET
10)	TARGET VULNERABILITY, P(K/H)	0.150	
11)	TARGET ALTITUDE	100.	FEET
12)	TARGET SKIN THICKNESS	0.060	INCHES
13)	TARGET SKIN MATERIAL	STEEL	

TABLE (III-4). OUTPUT DATA FOR EXAMPLE III-A

INITIAL VELOCITY TABLE FOR 50.0 FT KILL RADIUS

IMPACT VELOCITY	50 GR.	100 GR.	150 GR.	200 GR.	250 GR.	300 GR.
	FRAGMENT MASSES					
1000.	2513.	2078.	1895.	1787.	1714.	1661.
2000.	5027.	4156.	3789.	3574.	3429.	3321.
3000.	7540.	6235.	5684.	5361.	5143.	4982.
4000.	10054.	8313.	7579.	7148.	6857.	6643.
5000.	12567.	10391.	9473.	8936.	8571.	8303.
6000.	15081.	12469.	11368.	10723.	10286.	9964.
BALLISTIC LIMIT	641.	533.	476.	439.	412.	391.

FRAGMENT MASS........ 130.0 GRAINS
IMPACT VELOCITY...... 2500. FT/SEC

PD	WARHEAD WEIGHT	CASE THICKNESS	CORE DIAMETER	FRAGMENTS NUMBER	FRAGMENTS ON TARGET	FRAGMENTS LENGTH
0.999	183.17	0.55	10.51	3587.64	71.94	0.35
0.990	139.85	0.48	11.14	2392.40	47.98	0.37
0.980	125.45	0.45	11.35	2032.49	40.76	0.38
0.950	105.03	0.41	11.65	1556.62	31.22	0.40
0.900	88.15	0.38	11.89	1196.59	24.00	0.42

KILL PROBABILITY..... 0.980

WARHEAD DESCRIPTION--------------------------------------

WARHEAD WEIGHT	125.45 POUNDS
EXPLOSIVE WEIGHT	45.39 POUNDS
CASE WEIGHT	80.07 POUNDS
CASE THICKNESS	0.4513 INCHES
CORE DIAMETER	11.35 INCHES
FRAGMENT WEIGHT	130.00 GRAINS
FRAGMENT DIMENSIONS	0.381 X 0.381 X 0.451 INCHES
NUMBER OF FRAGMENTS	2032.
NUMBER OF HITS ON TARGET	41.
PROBABILITY OF KILL (PD)	0.980
INITIAL FRAGMENT VELOCITY	4886.8 FT/SEC
CRITICAL MISS DISTANCE	1912.1 FEET

TABLE (III-5). INPUT DATA FOR EXAMPLE III-B

THE FOLLOWING IS A SUMMARY OF THE INPUT DATA:

01)	EXPLOSIVE DENSITY	0.06650	LB/CU.IN
02)	EXPLOSIVE GURNEY CONSTANT	10230.00	FT/SEC
03)	EXPLOSIVE DETONATION VELOCITY	29934.00	FT/SEC
04)	CASE MATERIAL DENSITY	0.6880	LB/CU.IN
05)	DESIRED KILL RADIUS	75.0	FEET
06)	WARHEAD DIAMETE	10.00	INCHES
07)	WARHEAD LENGTH-TO-DIAMETER RATIO	2.00	
08)	TARGET LENGTH	60.00	FEET
09)	TARGET WIDTH	20.00	FEET
10)	TARGET VULNERABILITY, P(K/H)	0.100	
11)	TARGET ALTITUDE	30000.	FEET
12)	TARGET SKIN THICKNESS	0.180	INCHES
13)	TARGET SKIN MATERIAL	ALUMINUM	

TABLE (III-6). OUTPUT DATA FOR EXAMPLE III-B

INITIAL VELOCITY TABLE FOR 75.0 FT KILL RADIUS

IMPACT VELOCITY	FRAGMENT MASSES 50 GR.	100 GR.	150 GR.	200 GR.	250 GR.	300 GR.
1000.	1329.	1253.	1218.	1196.	1181.	1170.
2000.	2658.	2507.	2436.	2393.	2362.	2339.
3000.	3988.	3760.	3654.	3589.	3543.	3509.
4000.	5317.	5014.	4873.	4785.	4724.	4678.
5000.	6646.	6267.	6091.	5982.	5905.	5848.
6000.	7975.	7521.	7309.	7178.	7087.	7017.
BALLISTIC LIMIT	329.	277.	249.	231.	217.	207.

FRAGMENT MASS......... 210.0 GRAINS
IMPACT VELOCITY...... 5000. FT/SEC

PD	WARHEAD WEIGHT	CASE THICKNESS	CORE DIAMETER	FRAGMENTS NUMBER	FRAGMENTS ON TARGET	FRAGMENTS LENGTH
0.999	90.17	0.27	7.95	1480.22	75.39	0.40
0.990	68.82	0.23	8.41	986.93	50.26	0.43
0.980	61.73	0.22	8.56	838.42	42.70	0.44
0.950	51.67	0.20	8.77	642.08	32.70	0.46
0.900	43.3	0.19	8.95	493.54	25.14	0.48

INITIAL VELOCITY TABLE FOR 75.0 FT KILL RADIUS

IMPACT VELOCITY	FRAGMENT MASSES 50 GR.	100 GR.	150 GR.	200 GR.	250 GR.	300 GR.
1000.	1329.	1253.	1218.	1196.	1181.	1170.
2000.	2658.	2507.	2436.	2393.	2362.	2339.
3000.	3988.	3760.	3654.	3589.	3543.	3509.
4000.	5317.	5014.	4873.	4785.	4724.	4678.
5000.	6646.	6267.	6091.	5982.	5905.	5848.
6000.	7975.	7521.	7309.	7178.	7087.	7017.
BALLISTIC LIMIT	329.	277.	249.	231.	217.	207.

FRAGMENT MASS......... 100.0 GRAINS
IMPACT VELOCITY...... 3000. FT/SEC

TABLE (III-6). CONTINUED

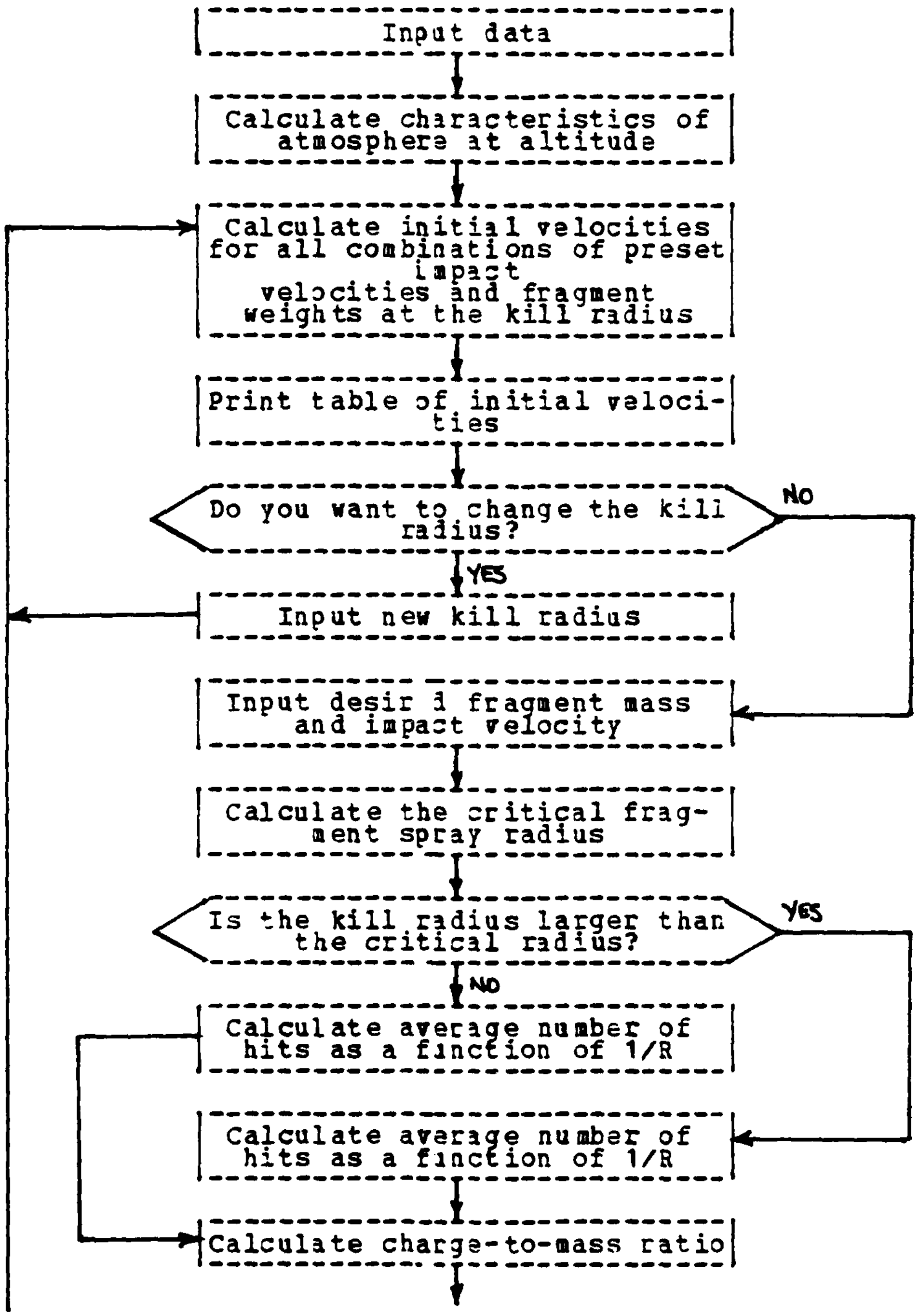
PD	WARHEAD WEIGHT	CASE THICKNESS	CORE DIAMETER	FRAGMENTS NUMBER	FRAGMENTS ON TARGET	FRAGMENTS LENGTH
0.999	52.41	0.36	8.92	1306.57	66.54	0.24
0.990	40.00	0.32	9.10	871.07	44.36	0.26
C.980	35.87	0.30	9.16	739.97	37.69	0.26
0.950	30.03	0.27	9.25	566.66	28.86	0.27
0.900	25.20	0.25	9.33	435.55	22.18	0.29

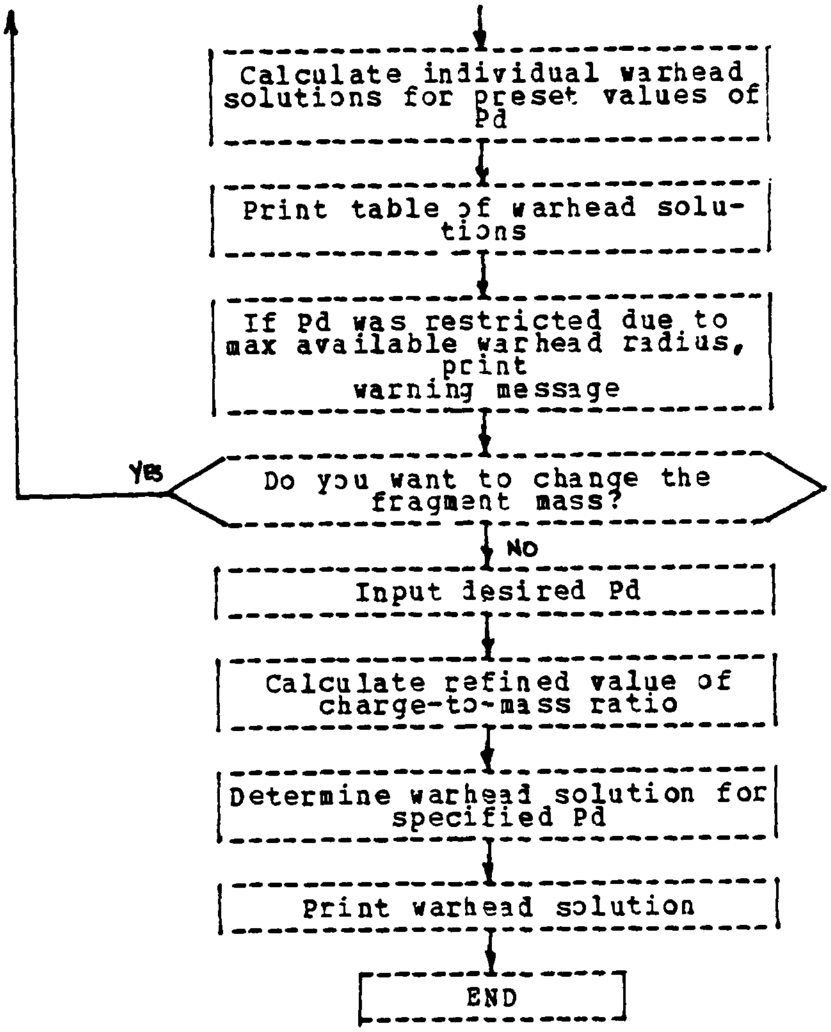
KILL PROBABILITY..... 0.995

WARHEAD DESCRIPTION--------------------------------

WARHEAD WEIGHT	43.91	POUNDS
EXPLOSIVE WEIGHT	5.67	POUNDS
CASE WEIGHT	38.24	POUNDS
CASE THICKNESS	0.3322	INCHES
CORE DIAMETER	9.04	INCHES
FRAGMENT WEIGHT	100.00	GRAINS
FRAGMENT DIMENIONS	0.250 X 0.250 X 0.332	INCHES
NUMBER OF FRAGMENTS	1002.	
NUMBER OF HITS ON TARGET	51.	
PROBABILITY OF KILL (PD)	0.995	
INITIAL FRAGMENT VELOCITY	3760.3	FT/SEC
CRITICAL MISS DISTANCE	6230.0	FEET

D. PROCEDURAL FLOWCHART




Calculate individual warhead solutions for preset values of Pd
Print table of warhead solutions
If Pd was restricted due to max available warhead radius, print warning message
Do you want to change the fragment mass?
YES
NO
Input desired Pd
Calculate refined value of charge-to-mass ratio
Determine warhead solution for specified Pd
Print warhead solution
END

IV. PROPULSION MOTOR SIZING

A. DESCRIPTION AND ORIGIN OF PROGRAM

This program provides a method for the preliminary sizing of a solid propellant rocket motor for a boost-sustain trajectory of a tactical missile. The analytical method was developed by Redmon [Ref. 1] and was expanded with the addition of material from Platzek [Ref. 2] and Hill [Ref. 3]. The program was written for this thesis in FORTRAN IV for use on the IBM 370 computer system. Essentially, the analysis consists of sizing the booster motor from differential velocity and acceleration requirements with limitations imposed by the physical dimensions of the missile. The booster is at all times considered to be a core-burning motor. The sustainer motor calculations are controlled by the maximum range specified by the user and by the solution of the booster motor. The sustainer can be either a core-burning or an end-burning motor.

The rocket motor configuration is assumed to be either an integral booster-sustainer motor as shown in Figure (IV-1) or a staged motor as in Figure (IV-2). The booster and the sustainer always burn exclusively, or, in other words, one is not burning while the other one is. The nozzle half angle is specified by the user consistant with space available in the missile. If a staged motor is utilized, both nozzles will have the same half angle.

The booster calculations start by determining the amount of thrust needed to boost the total weight of the missile to its cruise velocity at the prescribed acceleration. From this, the necessary amount of propellant is estimated and the process is iterated to account for the decreasing mass situation. The chamber pressure is initially estimated by

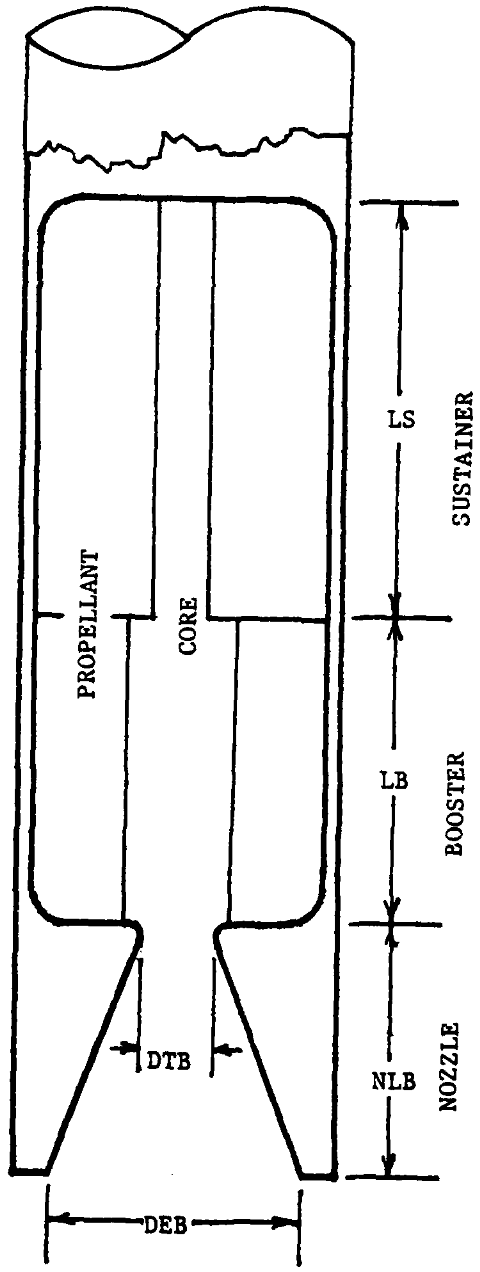


Figure (IV-1). Integral booster-sustainer motor

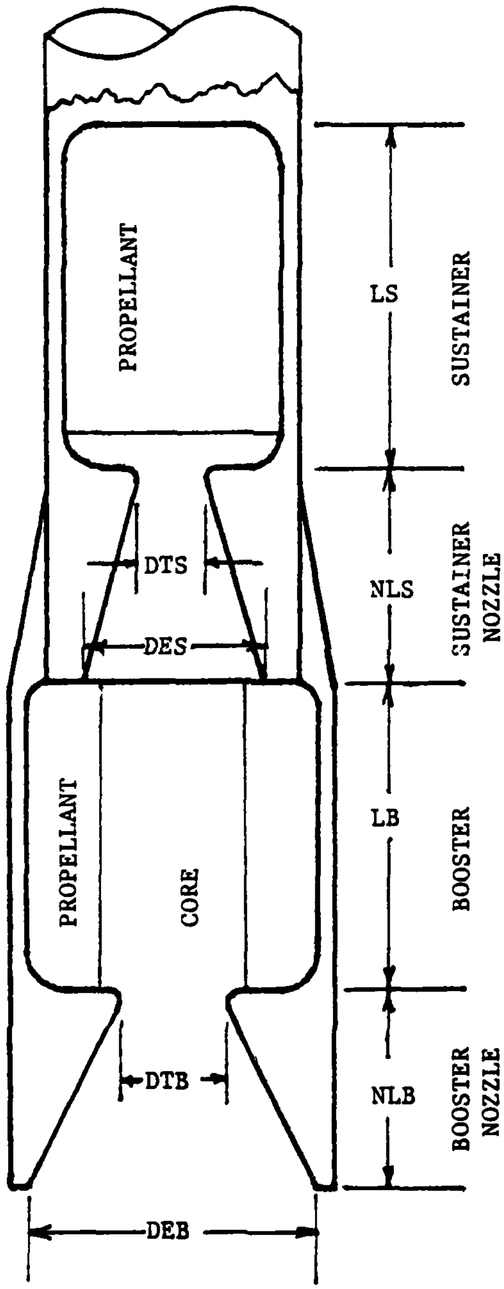


Figure (IV-2). Staged motor

minimizing the motor weight to propellant specific impulse ratio, as presented by Redmon [Ref. 1]. The next step of the program is to size the nozzle using the now known initial pressure ratio. Once the ideal nozzle is developed, the chamber pressure is raised or lowered as necessary in an attempt to cause the nozzle exit diameter to exactly match the booster diameter. However, if the integral motor option is used, the program will drive the nozzle as small as it can without violating one of the following limits in order to increase the probability that the sustainer will operate properly. The iterative process has two limits: an absolute maximum of 2000 PSI chamber pressure [Ref. 2] and a minimum of 1000 PSI if the pressure had previously been higher. The solution can be less than 1000 PSI if the pressure remained below that level throughout the problem. Also limited is the exit pressure to ambient pressure ratio. At the low end, it is limited to 0.4 to prevent flow separation in the nozzle. At the high end, it is limited to 2.5 to prevent excessive underexpansion and loss of physical reality in the program results [Ref. 4]. The burn rate is initialized at a potential maximum of 1.25 inches/second [Ref. 2] and is allowed to decrease to arrive at a compatible burn area and web thickness combination.

The sustainer motor, in the integral motor case, is virtually a continuation of the booster solution. The initial thrust requirement is determined by increasing the cruise speed drag to account for speed loss through maneuvers. It is also then refined for weight change if any climbing or diving is required to reach the target altitude. The nozzle is the same one as developed for the booster except that the throat area will be expanded as a result of the erosion effect. From the new area ratio, a pressure ratio is determined. The chamber pressure and thrust coefficient are then iterated until a steady chamber pressure evolves to provide the required thrust. If at any time it drops below 125 PSI, the program will stop since this is considered a minimum chamber pressure for proper propellant combustion. The exit pressure to ambient pressure ratio remains subject to the same restrictions. The burn rate starts at 0.45 inches/second [Ref. 2] and is decreased to provide an acceptable web thickness and burn area. The solution can be either an end burner or a core burner, depending on the burn area required.

The sustainer for the staged motor is solved in essentially the same manner as the booster. The two major exceptions are that it can produce an end burner if the burn area is small enough and the thrust required is based on the cruise drag and not the velocity to be gained. The chamber pressure is limited to an absolute maximum of 800 PSI and a minimum of 250 PSI if the pressure had previously been higher [Ref. 2]. The same pressure ratio restrictions apply and the burn rate starts at 0.45 inches/second.

Other motor-nozzle combinations can be created from the results of this program on a first-order approximation basis. Figures (IV-3) and (IV-4) illustrate two possible methods for separating the motor nozzles without resorting to staging. Although these two methods will probably have the same weight disadvantage that staging does, they can be packaged in to a smaller volume of space. A hint for "creative construction" is to rerun the problem after sizing the motors using a smaller missile diameter to force the nozzles to a smaller size.

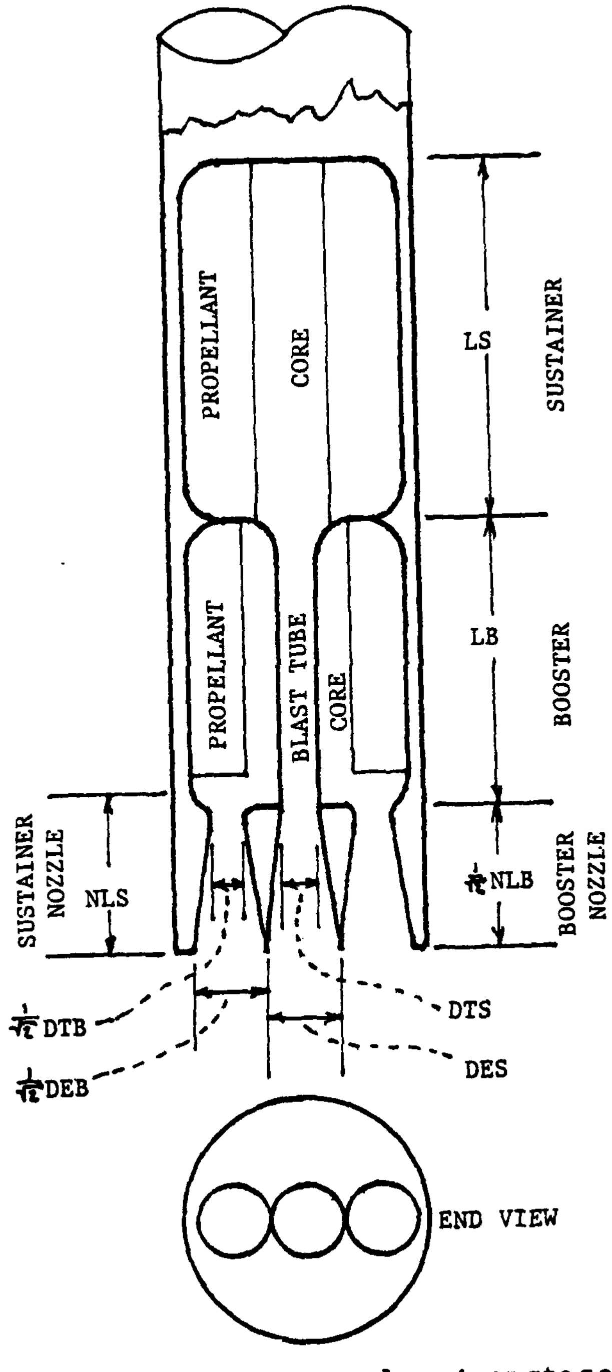


Figure (IV-3). Separate nozzles (nonstaged motors)
The values are obtained from the staged motor option.

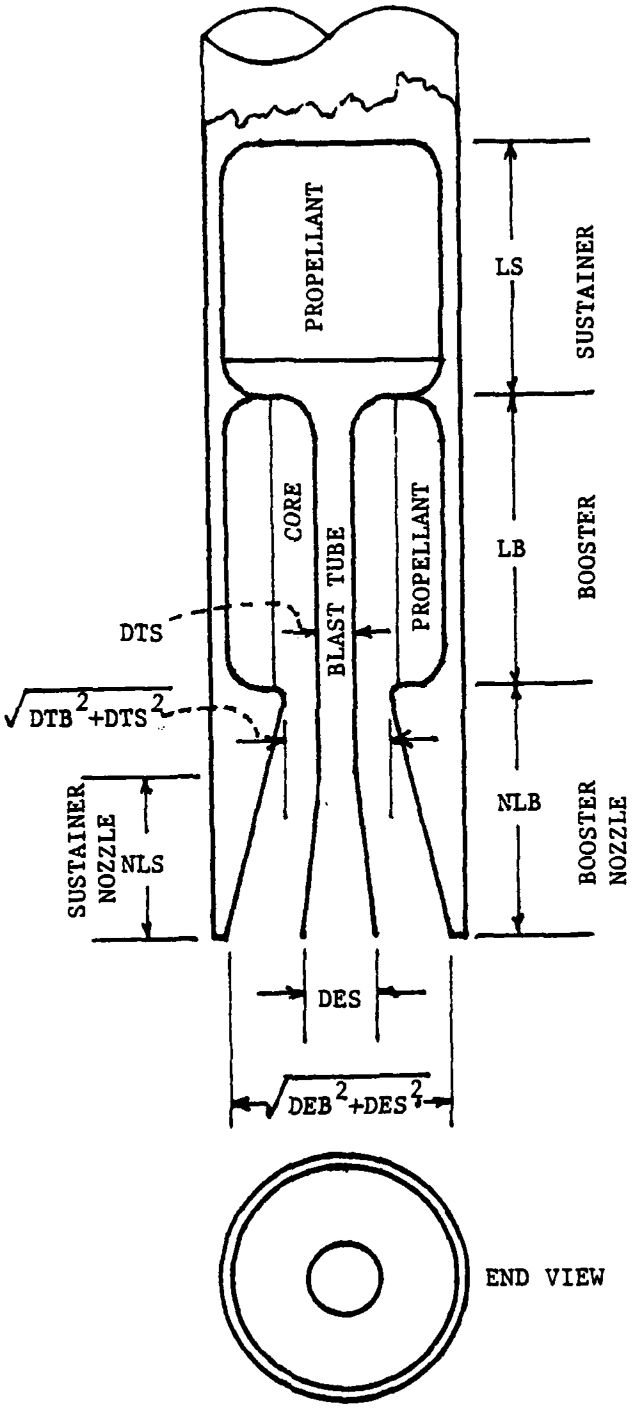


Figure (IV-4). Concentric nozzles (nonstaged motors) The values are obtained from the staged motor option.

B. USER INSTRUCTIONS

If it is desired to abort the operation of this program prematurely, two methods are available. When the program is waiting for data entry, press ENTER. When the program is not waiting for data entry but is processing, type "HX" and press ENTER. Both actions will return the terminal to CMS mode.

When the screen becomes full, or "MORE...." appears in the status area, clear the screen by pressing ALT and CLEAR simultaneously. At several points in the program, the user will be directed to "CLEAR SCREEN AND INPUT 0". This is to provide proper positioning of the output on the screen for ease of reading. If any other symbol than "4 A" should appear in the lower left of the screen, press RESET and continue.

1. Turn the terminal on with the red I O switch.

2. When the large "NPS" appears after about 30 seconds, press RESET, then press ENTER.

3. When "CP READ" appears in the status area on the lower right of the screen, type "L nnnnP", where nnnn is your 4-digit user number, then press ENTER.

4. You will now be asked for your password. Type it in (the characters will not appear on the screen), then press ENTER.

5. Your personal file must contain a PROFILE EXEC routine with the appropriate Fortran GLOBAL statement. If it does not, type "GLOBAL TXTLIB FORTMOD2 MOD2EEH", then press ENTER.

6. Type "LINK TO xxxxP 191 AS 192 RR", where xxxx is the 4-digit user number for the project file, then press ENTER.

7. You will now be asked for the project file password. Type it in (the characters will not appear on the screen), then press ENTER.

8. Type "ACCESS 192 B" and press $\boxed{\text{ENTER}}$.

9. Press $\boxed{\text{ALT}}$ and $\boxed{\text{CLEAR}}$ simultaneously to clear screen.

10. Type "LPROP" and press $\boxed{\text{ENTER}}$.

11. Input the following data as it is requested on the screen by typing the data and then pressing $\boxed{\text{ENTER}}$. Ensure that the data is input as either decimal or integer as specified. The terms in parenthesis are the program variable names.

The following variables are required inputs for both motor option problems.

Motor option	(IMOTOR)	0=integral motors 1=staged motors
Launch altitude	(LALT)	feet
Launch weight	(WL)	pounds
Launch velocity	(VBI)	feet/second
Launch elevation angle	(ELB)	degrees
Boost acceleration	(A)	gravities
Cruise velocity	(VBF)	feet/second
Cruise velocity drag	(DRAGS)	pounds
Maximum range to target	(R)	nautical miles
Maximum target altitude	(TALT)	feet
Booster propellant specific impulse	(ISPB)	seconds
Booster propellant density	(DENSB)	pounds/cu. inch
Booster exhaust specific heat ratio	(GB)	
Sustainer propellant specific impulse	(ISPS)	seconds
Sustainer propellant density	(DENSS)	pounds/cu. inch
Sustainer exhaust specific heat ratio	(GS)	
Nozzle half angle	(ALN)	degrees

The following variables are required inputs for the integral motors option only.

Nozzle design altitude	(ALTBN)	feet
Nozzle erosion rate	(ER)	inches/second
Missile diameter	(DB)	inches
Case yield strength	(YIELD)	PSI
Case density	(DENSC)	pounds/cu. inch

The following variables are required inputs for the staged motors option only.

Booster design altitude	(ALTBN)	feet
Booster diameter	(DB)	inches
Booster case yield strength	(YIELD)	PSI
Booster case density	(DENSC)	pounds/cu. inch
Sustainer design altitude	(ALTSN)	feet
Sustainer diameter	(D)	inches
Sustainer case yield strength	(YIELD)	PSI
Sustainer case density	(DENSCS)	pounds/cu. inch

12. This program will cue the user when the input parameters have dictated a scenario which either cannot be achieved within reality or produce less than optimum requirements on the propulsion system of the missile. They are not definitive and are only intended to make the user aware of a situation which may need correction. The following is a list of available cue messages with short definitions.

"SUSTAINER NOT CALCULATED SINCE THE BOOSTER BURNOUT RANGE EXCEEDS THE DESIGN RANGE." This can result from entering an extremely short range for the missile, or it can be caused by a very low acceleration requirement.

"SUSTAINER NOT CALCULATED: THE BOOSTER NOZZLE DESIGN PREVENTS SUSTAINER OPERATION. RECOMMEND STAGING OR INDEPENDENT NOZZLES." This occurs only when using the integral motors option. Then scenario described to the program can cause the booster nozzle to be too large to maintain the proper chamber pressures when the motor has shifted to sustainer operation. This usually occurs when a large acceleration is demanded but the thrust required for the cruise trajectory is small.

"BOOSTER BURN RATE WAS ADJUSTED TO CORRESPOND WITH THE WEB THICKNESS." This occurs quite often and simply indicates that the burn rate was decreased from its potential physical maximum of 1.25 inches/second.

"THE BOOSTER NOZZLE DESIGN WAS RESTRICTED DUE TO THE MISSILE DIAMETER," and "THE SUSTAINER NOZZLE DESIGN WAS RESTRICTED DUE TO THE MISSILE DIAMETER." The nozzle was not able to be designed for optimum pressure ratios at the mid point of the boost trajectory. Usually, the exit diameter is solved larger than the missile diameter and is subsequently reduced to fit.

"BOOSTER NOZZLE DESIGN IS NOT OPTIMUM DUE TO EXCESSIVE BOOSTER CHAMBER PRESSURES," and "SUSTAINER NOZZLE DESIGN IS NOT OPTIMUM DUE TO EXCESSIVE SUSTAINER CHAMBER PRESSURES." If the nozzle cannot be downsized without exceeding pressure thresholds (2000 PSI for the booster and 800 PSI for the sustainer), the chamber pressure is held just below the pressure threshold and the nozzle area ratio will be adjusted to allow the nozzle to fit in the missile.

"THE SUSTAINER MOTOR HAS A CORE-BURNING GRAIN." The required burn area for the sustainer was too large too permit an end burning grain with a properly realistic burn rate.

"SUSTAINER BURN RATE WAS ADJUSTED TO CORRESPOND WITH THE WEB THICKNESS." This indicates the sustainer burn rate was lowered from a potential maximum of 0.45 inches/second to provide a proper web thickness.

"THE SUSTAINER MOTOR HAS AN END BURNING GRAIN." The required burn area for the sustainer was small enough to permit an end burning grain. The burn rate is then adjusted to correspond with the nonreduceable burn area.

"REESTIMATION OF LAUNCH WEIGHT IS REQUIRED FOR THESE MISSILE PERFORMANCE PARAMETERS." The scenario described to the program produced a motor whose weight is either larger than 75% of the total or less than 25% of the total.

"ENLARGEMENT OF DIAMETER IS RECOMMENDED DUE TO A VERY HIGH LENGTH-TO-DIAMETER RATIO FOR THE MOTOR." This cue indicates the length-to-diameter ratio is greater than 15. Other components of the missile will make the missile's overall length-to-diameter ratio even larger.

13. Immediately after completion of the solution, the program will ask if you want to receive a hardcopy printout of that particular solution. A "yes" answer stores that solution in a file for retrieval by the user when he finishes with the program.

14. If you desire to rerun the problem, or want to run a new problem, answer the questions appropriately when asked by the terminal after the execution of the current problem.

15. To receive the printout of your encounters, answer "no" to rerunning or restarting the problem when asked by the terminal and follow the directions presented on the screen.

16. Upon completion of the program, type "LOGOFF" and press ENTER.

17. Turn the terminal off with the red switch.

C. EXAMPLE PROBLEMS

1. Example IV-A. Integral motors, common nozzle

The following parameters are input for the integral motor example:

Launch altitude	35.0	feet
Launch weight	1000.0	pounds
Launch velocity	0.0	feet/second
Launch angle	60.0	degrees
Average acceleration	30.0	g's
Cruise velocity	4000.0	feet/second
Drag at cruise velocity	1500.0	pounds
Maximum range	20.0	miles
Final (target) altitude	50000	feet
Booster propellant specific impulse	260.0	seconds
Booster propellant density	0.075	lbs/cu.inch
Booster exhaust specific heat ratio	1.244	
Sustainer propellant specific impulse	210.0	seconds
Sustainer propellant density	0.065	lbs/cu.inch
Sustainer exhaust specific heat ratio	1.270	
Nozzle half angle	20.0	degrees
Nozzle design altitude	0.0	feet

TABLE (IV-1). OUTPUT OF EXAMPLE IV-A

```
INTEGRAL MOTORS (COMMON NOZZLE)
SUMMARY OF INPUT  ARAMETERS=========================
     1) LAUNCH ALTITUDE                                 35.0 FEET
     2) LAUNCH WEIGHT                                 1000.00 POUNDS
     3) LAUNCH VELOCITY                                  0.0 FT/SEC
     4) LAUNCH ANGLE                                    60.0 DEGREES
     5) AVERAGE ACCELERATION                            30.00 G"S
     6) CRUISE VE OCITY                               4000.0 FT/SEC
     7) DRAG AT CRUISE VELOCITY                       1500.0 POUNDS
     8) MAXIMUM RANGE                                   20. MILES
     9) FINAL (TARGET) ALTITUDE                      50000.0 FEET
    10) BOOSTER PROPELLANT SPECIFIC IMPULSE            260.0 SEC
    11) BOOSTER PROPELLANT DENSITY                   0.0750 LBS/CU.IN
    12) BOOSTER EXHAUST SPECIFIC HEAT RATIO         1.24400
    13) SUSTAINER PROPELLANT SPECIFIC IMPULSE          210.0 SEC
    14) SUSTAINER PROPELLANT DENSITY                 0.0650 LBS/CU.IN
    15) SUSTAINER EXHAUST SPECIFIC HEAT RATIO       1.27000
    16) NOZZLE HALF ANGLE                             20.00 DEGREES
    17) NOZZLE DESIGN ALTITUDE                          0.0 FEET
    18) NOZZLE EROSION RATE                         0.00100 IN/SEC
    19) MISSILE DIAMETER                               10.0 INCHES
    20) YIELD STRENGTH OF CASE MATERIAL            180000.0 PSI
    21) DENSITY O  CASE MATERIAL                     0.2662 LBS/CU.IN
                                   BOOSTER                  SUSTAINER
PROPELLANT WEIGHT                  393.21 LBS               253.67 LBS
CASING WEIGHT                       36.95 LBS                24.92 LBS
TOTAL WEIGHT                       430.16 LBS               278.59 LBS
THRUST COEFFICIENT                 1.6032                   1.2748
CHARACTERISTIC VELOCITY            5217.7 FT/SEC            5300.1 FT/SEC
THRUST                            24669.5 LBS               1814.2 LBS
BURN TIME                           4.144 SEC               29.364 SEC
HORIZONTAL BURNOUT RANGE            4144. FEET
CHAMBER PRESSURE                  1999.66 PSI               196.09 PSI
GRAIN BURN AREA                  1590.873 SQ.IN            910.321 SQ.IN
GRAIN WEB THICKNESS                 3.296 IN                 4.287 IN
GRAIN PORT AREA                     9.053 SQ.IN              0.982 SQ.IN
REQUIRED BURN RATE                  0.795 IN/SEC             0.146 IN/SEC
MOTOR CASE LENGTH                   79.55 IN                 53.64 IN
MOTOR CASE VOLUME                 6046.29 CU.IN            4056.20 CU.IN
NOZZLE THROAT AREA                  7.695 SQ.IN            ******* SQ.IN
NOZZLE EXIT AREA                   58.510 SQ.IN            ******* SQ.IN
NOZZLE LENGTH                       7.557 IN               ******* IN
NOZZLE WEIGHT                       38.88 LBS              ******* LBS
CASE THICKNESS                    0.05555 IN               0.05555 IN
TOTAL CASE WEIGHT                                61.87 LBS
TOTAL PROPELLANT WEIGHT                         646.88 LBS
```

TABLE (IV-1). (CONTINUED)

TOTAL ROCKET MOTOR WEIGHT	747.62 LBS
TOTAL ROCKET MOTOR LENGTH	140.74 IN

BOOSTER BURN RATE WAS ADJUSTED TO CORRESPOND WITH THE WEB THICKNESS.

THE BOOSTER NOZZLE DESIGN WAS RESTRICTED DUE TO THE MISSILE DIAMETER.

BOOSTER NOZZLE DESIGN IS NOT OPTIMUM DUE TO EXCESSIVE BOOSTER CHAMBER PRESSURES.

THE SUSTAINER MOTOR HAS A CORE-BURNING GRAIN.

SUSTAINER BURN RATE WAS ADJUSTED TO CORRESPOND WITH THE WEB THICKNESS.

Nozzle erosion rate	0.001	inches/second
Missile diameter	10.0	inches
Yield strength of case material	180000.0	PSI
Density of case material	0.2662	lbs/cu.inch

The solution for this problem is presented in Table (IV-1).

2. Example IV-B. Staged motors, separate nozzles

The following are input for the staged motor prblem:

Launch altitude	35.0	feet
Launch weight	2200.0	pounds
Launch velocity	0.0	feet/second
Launch angle	30.0	degrees
Average acceleration	25.0	g's
Cruise velocity	2200.0	feet/second
Drag at cruise velocity	1000.0	pounds
Maximum range	50.0	miles
Final (target) altitude	75000	feet
Booster propellant specific impulse	250.0	seconds
Booster propellant density	.0647	lbs/cu.inch
Booster exhaust specific heat ratio	1.225	
Sustainer propellant specific impulse	205.0	seconds
Sustainer propellant density	.0625	lbs/cu.inch
Sustainer exhaust specific heat ratio	1.257	
Nozzle half angle	15.0	degrees
Booster design altitude	0.0	feet
Booster diameter	14.5	inches
Yield strength of booster case	180000.0	PSI
Density of booster case material	0.2662	lbs/cu.inch
Sustainer design altitude	0.0	feet

TABLE (IV-2). OUTPUT OF EXAMPLE IV-B

STAGED MOTORS (INDEPENDENT NOZZLES)

SUMMARY OF INPUT PARAMETERS=======================

1) LAUNCH ALTITUDE	35.0	FEET
2) LAUNCH WEIGHT	2200.00	POUNDS
3) LAUNCH VELOCITY	0.0	FT/SEC
4) LAUNCH ANGLE	30.0	DEGREES
5) AVERAGE ACCELERATION	25.00	G"S
6) CRUISE VELOCITY	2200.0	FT/SEC
7) DRAG AT CRUISE VELOCITY	1000.0	POUNDS
8) MAXIMUM RANGE	50.	MILES
9) FINAL (TARGET) ALTITUDE	75000.0	FEET
10) BOOSTER PROPELLANT SPECIFIC IMPULSE	250.0	SEC
11) BOOSTER PROPELLANT DENSITY	0.0647	LBS/CU.IN
12) BOOSTER EXHAUST SPECIFIC HEAT RATIO	1.22500	
13) SUSTAINER PROPELLANT SPECIFIC IMPULSE	205.0	SEC
14) SUSTAINER PROPELLANT DENSITY	0.0625	LBS/CU.IN
15) SUSTAINER EXHAUST SPECIFIC HEAT RATIO	1.25700	
16) NOZZLE HALF ANGLE	15.00	DEGREES
17) BOOSTER DESIGN ALTITUDE	0.0	FEET
18) BOOSTER DIAMETER	14.50	INCHES
19) YIELD STRENGTH OF BOOSTER CASE	180000.0	PSI
20) DENSITY OF BOOSTER CASE MATERIAL	0.2662	LBS/CU.IN
21) SUSTAINER DESIGN ALTITUDE	0.0	FEET
22) SUSTAINER DIAMETER	14.50	INCHES
23) YIELD STRENGTH OF SUSTAINER CASE	180000.0	PSI
24) DENSITY OF SUSTAINER CASE MATERIAL	0.2662	LBS/CU.IN

	BOOSTER		SUSTAINER	
PROPELLANT WEIGHT	537.95	LBS	916.16	LBS
CASING WEIGHT	59.42	LBS	11.82	LBS
TOTAL WEIGHT	597.37	LBS	927.98	LBS
THRUST COEFFICIENT	1.6035		1.2900	
CHARACTERISTIC VELOCITY	5016.1	FT/SEC	5113.1	FT/SEC
THRUST	49171.0	LBS	1370.9	LBS
BURN TIME	2.735	SEC	136.997	SEC
HORIZONTAL BURNOUT RANGE	2606.	FEET		
CHAMBER PRESSURE	1998.07	PSI	249.17	PSI
GRAIN BURN AREA	2431.947	SQ.IN	2472.758	SQ.IN
GRAIN WEB THICKNESS	3.419	IN	5.928	IN
GRAIN PORT AREA	18.055	SQ.IN	5.018	SQ.IN
REQUIRED BURN RATE	1.250	IN/SEC	0.043	IN/SEC
MOTOR CASE LENGTH	63.30	IN	97.15	IN
MOTOR CASE VOLUME	9641.00	CU.IN	15804.59	CU.IN
NOZZLE THROAT AREA	15.347	SQ.IN	4.265	SQ.IN
NOZZLE EXIT AREA	165.068	SQ.IN	12.894	SQ.IN
NOZZLE LENGTH	18.804	IN	3.212	IN

TABLE (IV-2). (CONTINUED)

NOZZLE WEIGHT	33.62 LBS		46.95 LBS
CASE THICKNESS	0.08048 IN		0.01004 IN
TOTAL CASE WEIGHT		71.24 LBS	
TOTAL PROPELLANT WEIGHT		1454.11 LBS	
TOTAL ROCKET MOTOR WEIGHT		1605.93 LBS	
TOTAL ROCKET MOTOR LENGTH		182.47 IN	

THE BOOSTER NOZZLE DESIGN WAS RESTRICTED DUE TO THE MISSILE DIAMETER.

BOOSTER NOZZLE DESIGN IS NOT OPTIMUM DUE TO EXCESSIVE BOOSTER CHAMBER PRESSURES.

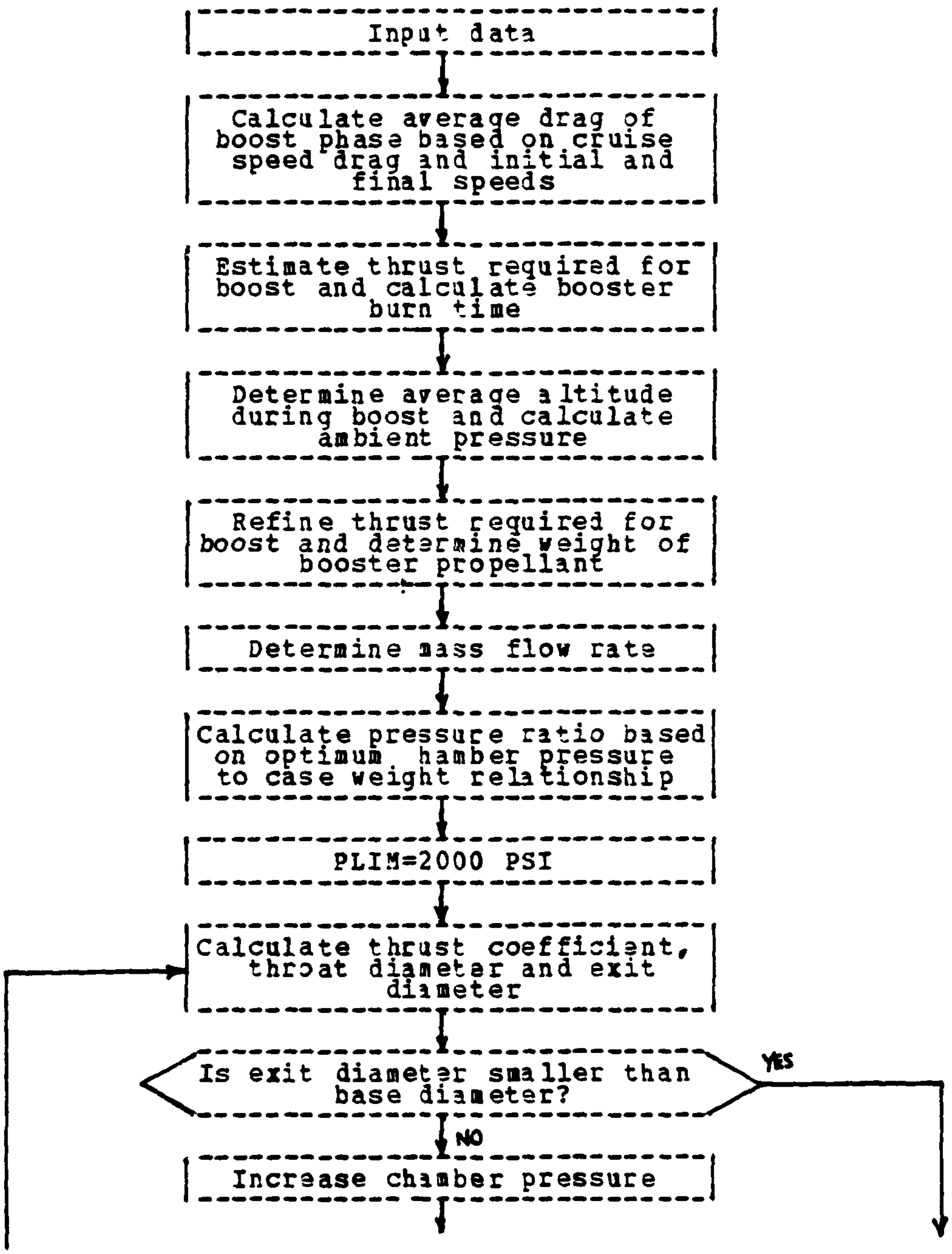
THE SUSTAINER MOTOR HAS A CORE-BURNING GRAIN.

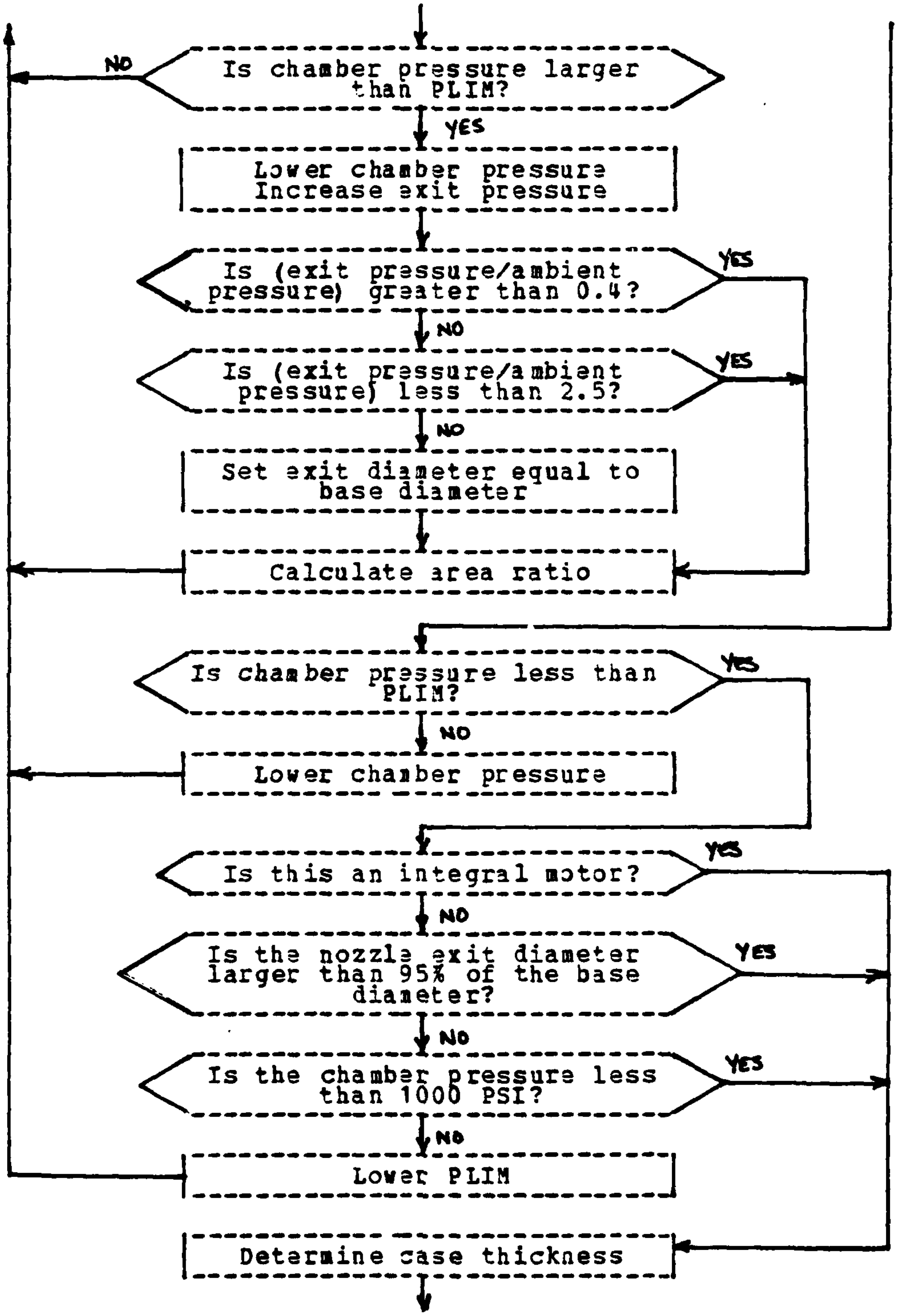
SUSTAINER BURN RATE WAS ADJUSTED TO CORRESPOND WITH THE WEB THICKNESS.

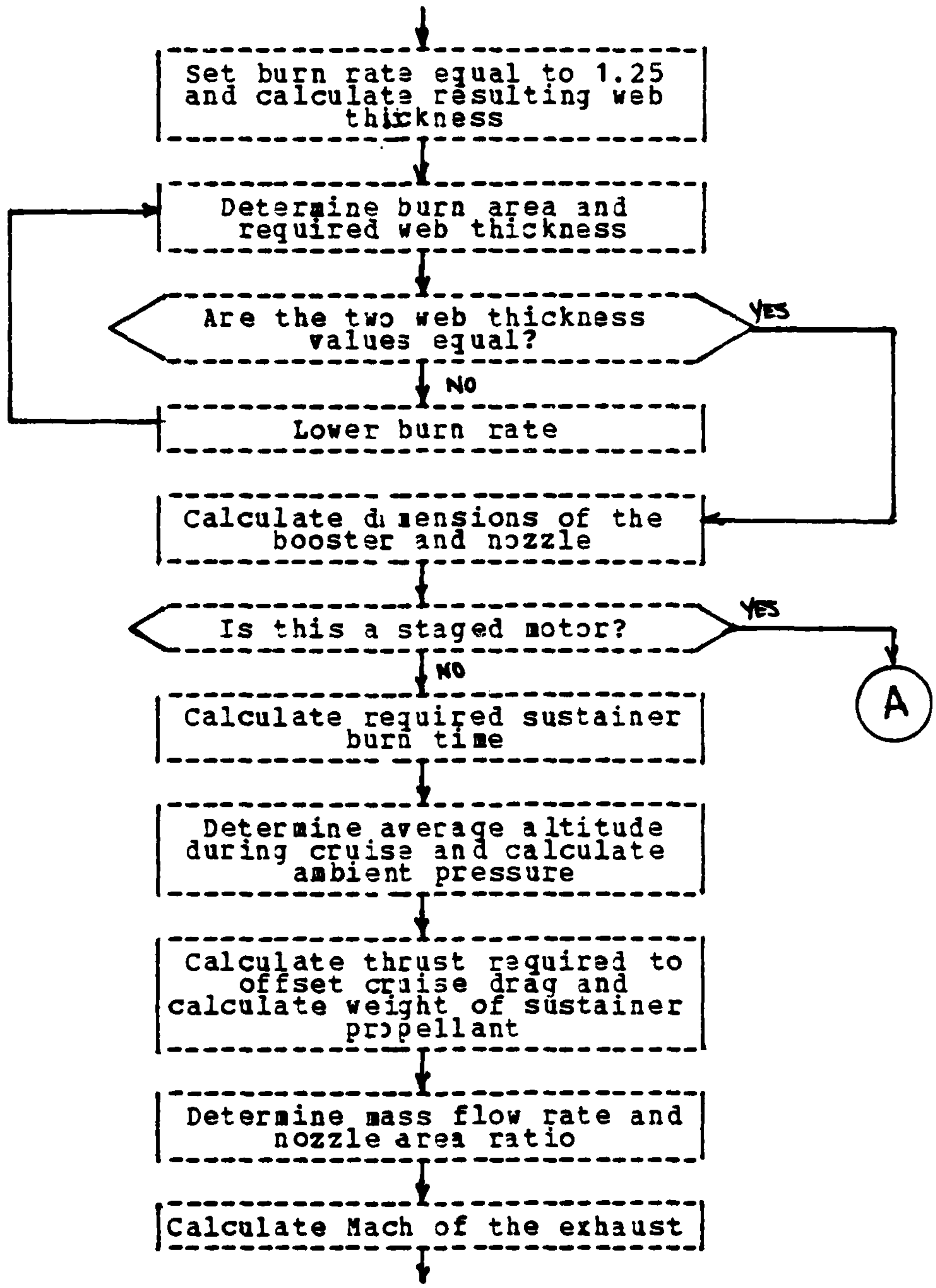
Sustainer diameter	14.5 inches
Yield strength of sustainer case	180000.0 PSI
Density of sustainer case material	0.2662 lbs/cu.inch

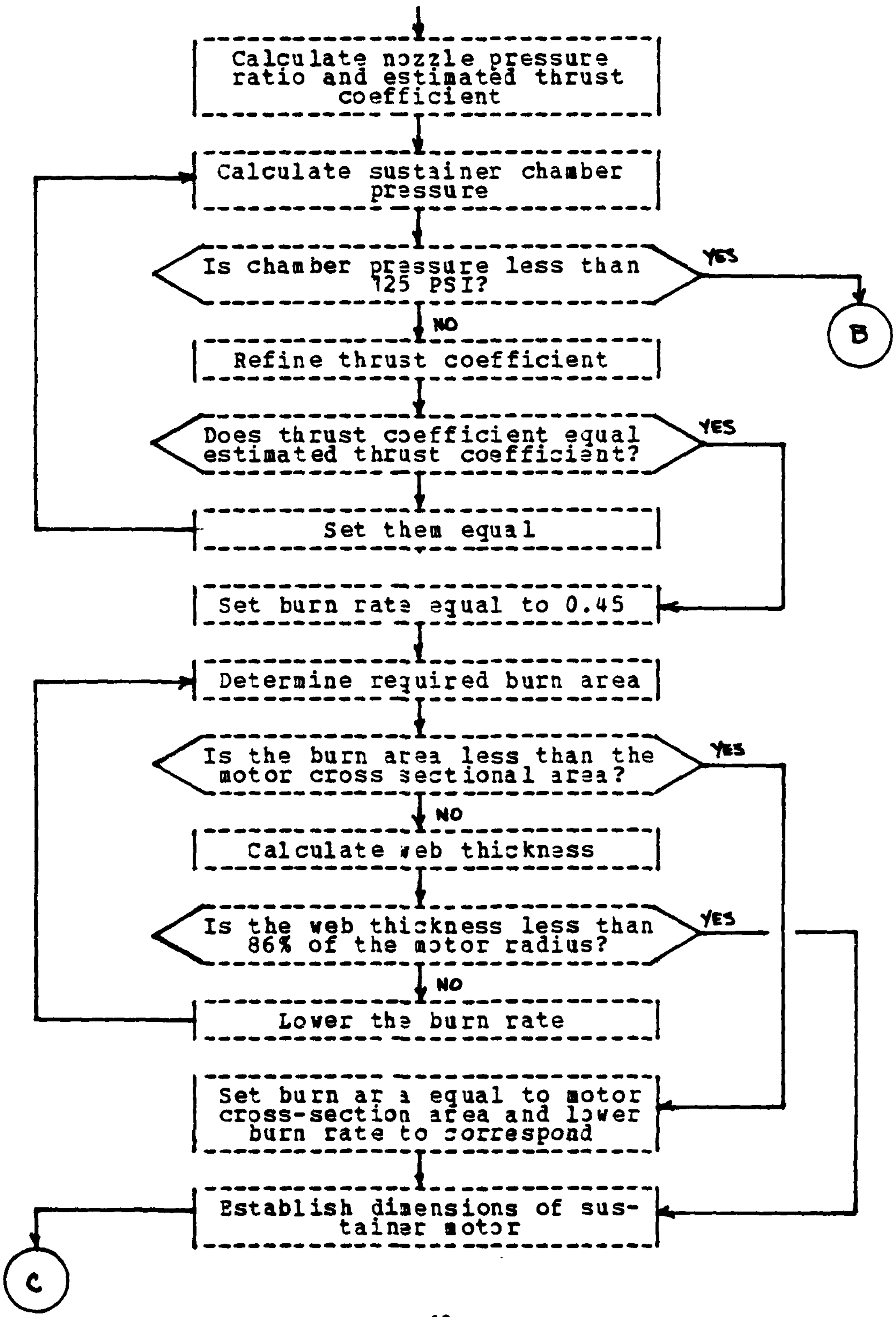
The output for this example is provided in Table (IV-2).

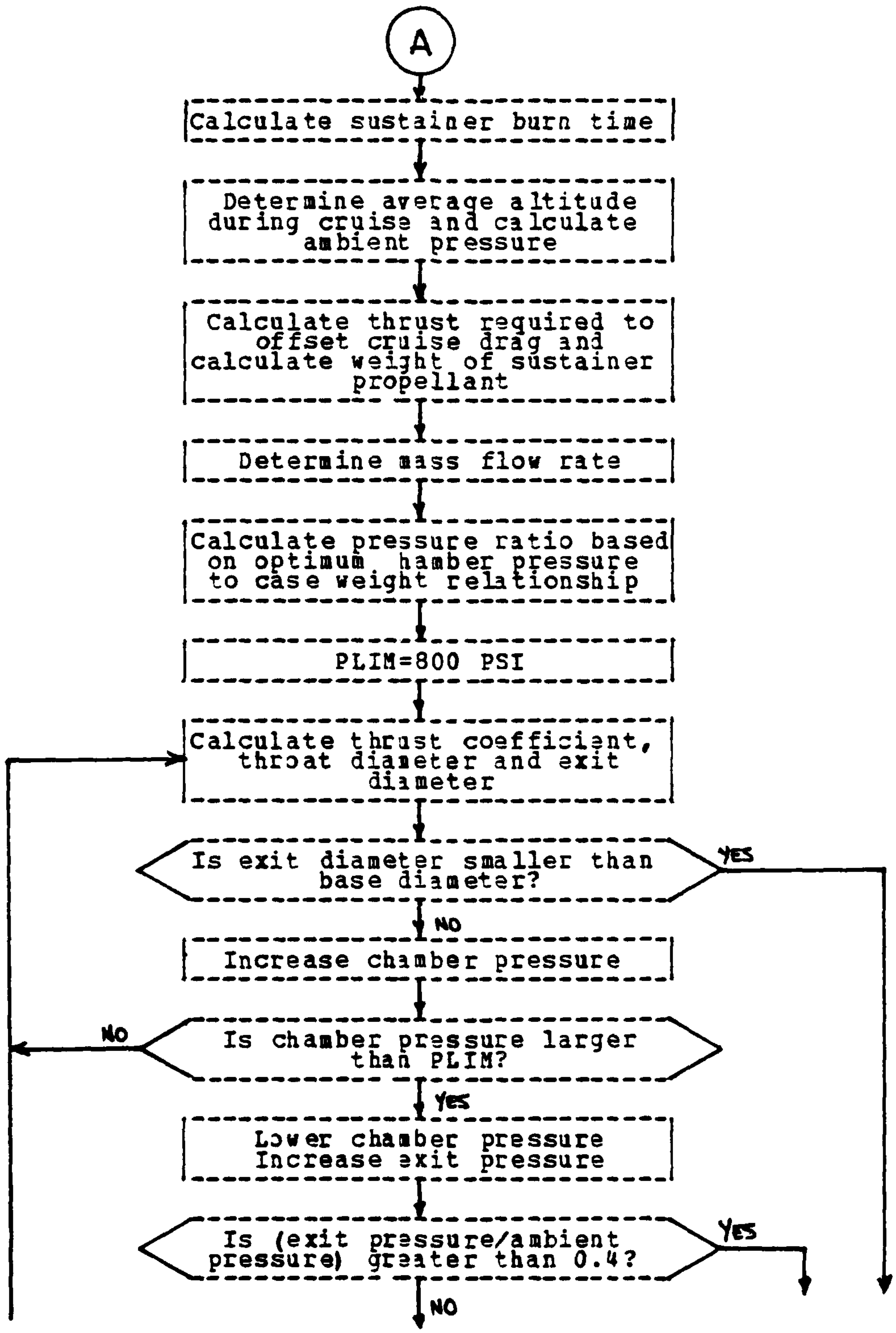
D. PROCEDURAL FLOWCHART

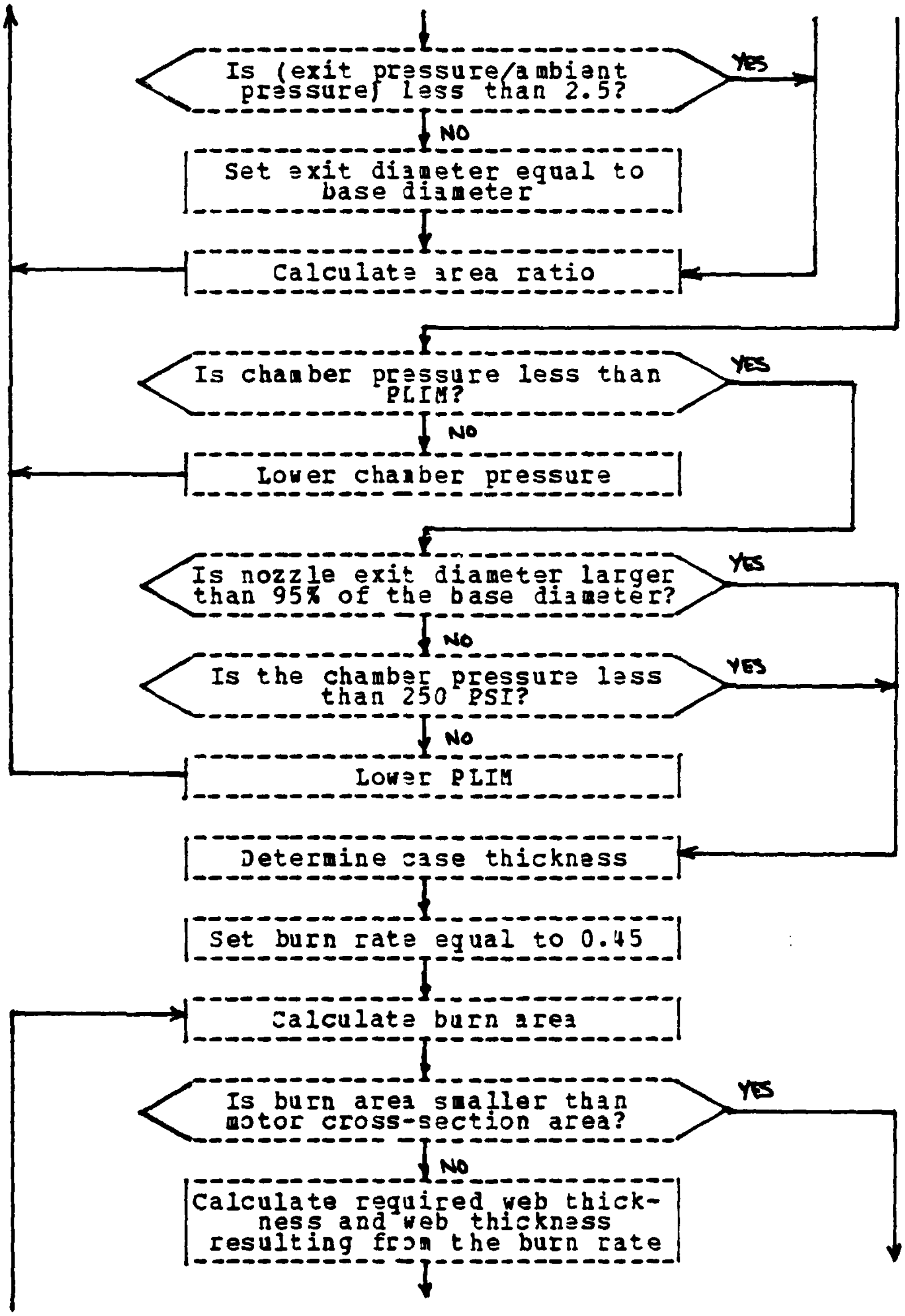


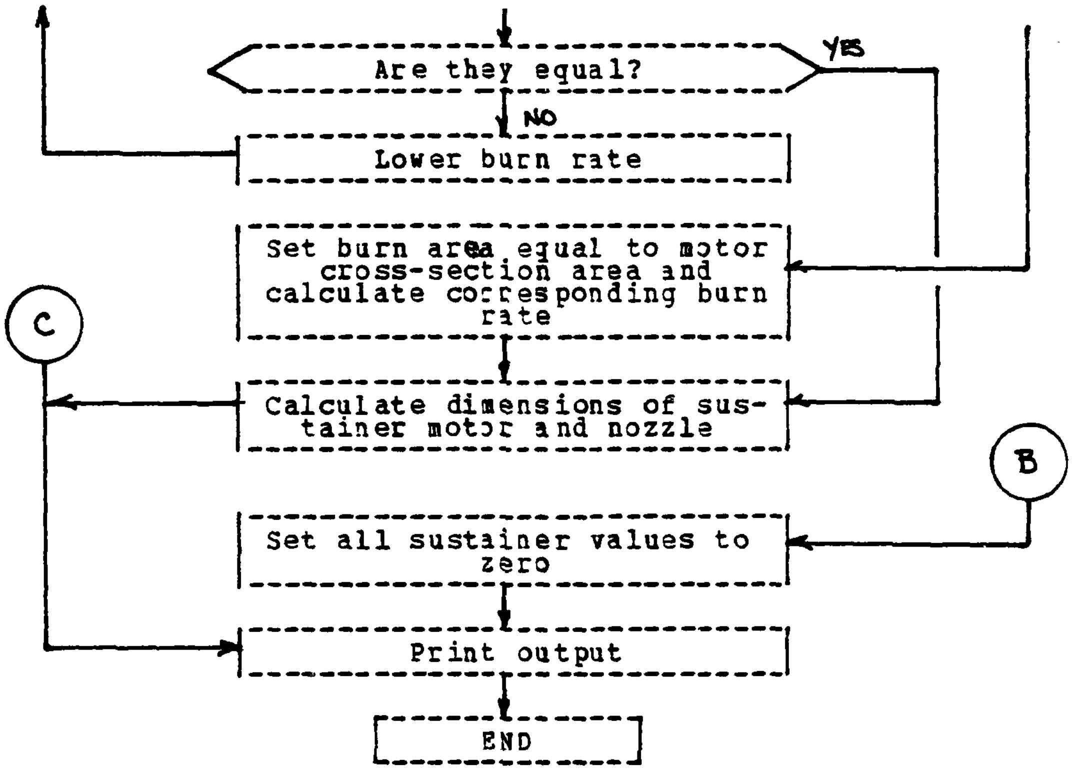

NO
Is chamber pressure larger than PLIM?
YES
Lower chamber pressure
Increase exit pressure
Is (exit pressure/ambient pressure) greater than 0.4?
YES
NO
Is (exit pressure/ambient pressure) less than 2.5?
YES
NO
Set exit diameter equal to base diameter
Calculate area ratio
Is chamber pressure less than PLIM?
YES
NO
Lower chamber pressure
Is this an integral motor?
YES
NO
Is the nozzle exit diameter larger than 95% of the base diameter?
YES
NO
Is the chamber pressure less than 1000 PSI?
YES
NO
Lower PLIM
Determine case thickness


Set burn rate equal to 1.25 and calculate resulting web thickness
Determine burn area and required web thickness
Are the two web thickness values equal?
YES
NO
Lower burn rate
Calculate dimensions of the booster and nozzle
Is this a staged motor?
YES
NO
A
Calculate required sustainer burn time
Determine average altitude during cruise and calculate ambient pressure
Calculate thrust required to offset cruise drag and calculate weight of sustainer propellant
Determine mass flow rate and nozzle area ratio
Calculate Mach of the exhaust


Calculate nozzle pressure ratio and estimated thrust coefficient
Calculate sustainer chamber pressure
Is chamber pressure less than 125 PSI?
YES
B
NO
Refine thrust coefficient
Does thrust coefficient equal estimated thrust coefficient?
YES
Set them equal
Set burn rate equal to 0.45
Determine required burn area
Is the burn area less than the motor cross sectional area?
YES
NO
Calculate web thickness
Is the web thickness less than 86% of the motor radius?
YES
NO
Lower the burn rate
Set burn area equal to motor cross-section area and lower burn rate to correspond
Establish dimensions of sustainer motor
C


A
Calculate sustainer burn time
Determine average altitude during cruise and calculate ambient pressure
Calculate thrust required to offset cruise drag and calculate weight of sustainer propellant
Determine mass flow rate
Calculate pressure ratio based on optimum hamber pressure to case weight relationship
PLIM=800 PSI
Calculate thrust coefficient, throat diameter and exit diameter
Is exit diameter smaller than base diameter?
YES
NO
Increase chamber pressure
Is chamber pressure larger than PLIM?
NO
YES
Lower chamber pressure
Increase exit pressure
Is (exit pressure/ambient pressure) greater than 0.4?
YES
NO


Is (exit pressure/ambient pressure) less than 2.5?
YES
NO
Set exit diameter equal to base diameter
Calculate area ratio
Is chamber pressure less than PLIM?
YES
NO
Lower chamber pressure
Is nozzle exit diameter larger than 95% of the base diameter?
YES
NO
Is the chamber pressure less than 250 PSI?
YES
NO
Lower PLIM
Determine case thickness
Set burn rate equal to 0.45
Calculate burn area
Is burn area smaller than motor cross-section area?
YES
NO
Calculate required web thickness and web thickness resulting from the burn rate


Are they equal?
YES
NO
Lower burn rate
Set burn area equal to motor cross-section area and calculate corresponding burn rate
C
Calculate dimensions of sustainer motor and nozzle
B
Set all sustainer values to zero
Print output
END

V. AERODYNAMIC COEFFICIENTS

A. DESCRIPTION AND ORIGIN

This program is the current edition of a program which originated at the Naval Ship Research and Development Center in 1971 [Ref. 5]. The program was written in FORTRAN II for use on the IBM 7090 digital computer. It was developed as a method for predicting the static, longitudenal aerodynamic characteristics of typical missile configurations with the control surfaces in a "plus" attitude. The original program computed the aerodynamic characteristics for missiles operating at angles of attack up to 180 degrees. The effects of control surface deflections for all modes of aerodynamic control are taken into account. The method was based on the then well known linear, nonlinear crossflow and slender body theories with empirical modifications to provide the high angle of attack capability.

The program was modified and, presumably, updated in 1974 by F. A. Kuster, Jr., of the Naval Air Development Center. In 1980, the program was modified for use on the Naval Postgraduate School IBM 360 computer system by D. Redmon [Ref. 1]. The current version of the program was modified for use on the new Naval Postgraduate School IBM 370 computer system. It has been expanded to provide graphical presentation of the output data.

It must be emphasized at this point that the current program edition is not in a completely finished state. Somewhere in the history of the program after its initial establishment on the IBM 360, errors were introduced during the modifications. At present, these errors do not prevent the use of the program and the output data is considered to be correct for trend-observance purposes.

Specifically, the program does not produce any drag-rise phenomena for either the wings or the tails when Cd is observed as a function of Mach number. Additionally, the decline of the drag coefficient above Mach 1.0 is not smooth or as prolonged as is found experimentally. It is very probable that these two failings of the program are linked to a common error inserted accidentally in the process of tailoring the program for use on the IBM 370. In order to temporarily smooth over the graphical discontinuities, exponential decay functions were inserted. They are clearly marked in the the program listing for removal when the program is corrected.

The input to the program is composed of a detailed listing of the dimensions of the missile to be analyzed. The current version of the program will consider a missile which has four symmetrical wings and four symmetrical tails. The missile may be either canard or tail equipped and either wing or canard or tail controlled. The program assumes the control surface is the tail, however, the input data is "mislabelled" to produce the proper configuration. For instance, if the missile is a wing control missile, the wing data is input as the tail and the tail data as the wing. For a canard controlled missile, the canard data is input as the tail. Figures (V-1) and (V-3) show two typical missile configurations and where the input parameters for the program are obtained.

B. USER INSTRUCTIONS

If it is desired to abort the operation of this program prematurely, two methods are available. When the program is waiting for data entry, press ENTER. When the program is not waiting for data entry but is processing, type "HX" and press ENTER. Both actions will return the terminal to CMS mode.

When the screen becomes full, or "MORE...." appears in the status area, clear the screen by pressing ALT and CLEAR simultaneously. At several points in the program, the user will be directed to "CLEAR SCREEN AND INPUT 0". This is to provide proper positioning of the output on the screen for ease of reading. If any other symbol than "4 A" should appear in the lower left of the screen, press RESET and continue.

1. Turn the terminal on with the red I O switch.

2. When the large "NPS" appears after about 30 seconds, press RESET, then press ENTER.

3. When "CP READ" appears in the status area on the lower right of the screen, type "L nnnnP", where nnnn is your 4-digit user number, then press ENTER.

4. You will now be asked for your password. Type it in (the characters will not appear on the screen), then press ENTER.

5. Your personal file must contain a PROFILE EXEC routine with the appropriate Fortran GLOBAL statement. If it does not, type "GLOBAL TXTLIB FORTMOD2 MOD2EEH", then press ENTER.

6. Type "LINK TO xxxxP 191 AS 192 RR", where xxxx is the 4-digit user number for the project file, then press ENTER.

7. You will now be asked for the project file password. Type it in (the characters will not appear on the screen), then press ENTER.

8. Type "ACCESS 192 B" and press ENTER.

9. Press ALT and CLEAR simultaneously to clear screen.

10. Type "LAERO1" and press ENTER.

11. Input the following data as it is requested on the screen by typing the data and then pressing ENTER. Ensure

that the data is input as either decimal or integer as specified. The terms in parenthesis below are the program variable names.

Input the following as integer values unless otherwise noted. The integers must be two digit integers (three=03).

Control constant (ICSC)	01-Tail control 02-Wing control 03-Canard control
Nose shape (INOSE)	01-Ellipsoid 02-Ogive 03-Cone
Number of control deflections (IDT)	Less than 11

You will now be asked for the control deflections in degrees, decimal values.

Number of Mach numbers (IM)	Less than 25

You will now be asked for the Mach numbers, decimal values. Each Mach number will produce a separate table and plot of output data.

Number of angles of attack (IAL)	Less than 25

You will now be asked for the angles of attack in degrees, decimal values.

Number of configurations (NBODY)	No restrictions

Each configuration will restart the program. Only the last configuration will produce the written output.

Wing planform (ISWPW)	01-Not delta 02-Delta
Wing position (IAFBW)	00-Body after wing 01-No body after wing
Wing sweep constant (ISWEPW)	00-Delta planform or Unswept leading edge 01-Swept leading edge
Number of wings (NWING)	04
Tail planform (ISWPT)	01-Not delta 02-Delta
Tail position (IAFBT)	00-No body after tail 01-Body after tail
Tail sweep constant (ISWEPT)	00-Delta planform or Unswept leading edge 01-Swept leading edge
Number of tails (NTAIL)	04

Input the following values as decimal numbers:

Wing tip-to-chord ratio (XLAMW)	
Wing leading edge sweep (CLAMW)	Degrees

Wing span including body (BW)	Feet
Wing root chord (CROOTW)	Feet
Exposed wing area, 2 panels (SW)	Square feet
Wing mean geometric chord (XMACW)	Feet
Distance from nose to wing leading edge (XWING)	Feet
Wing thickness-to-chord ratio (TOVCW)	
Tail tip-to-chord ratio (XLAMT)	
Tail leading edge sweep (CLAMT)	Degrees
Tail span including body (BT)	Feet
Tail root chord (CROOTT)	Feet
Exposed tail area, 2 panels (ST)	Square feet
Tail mean geometric chord (XMACT)	Feet
Distance from nose to tail leading edge (XTAIL)	Feet
Tail thickness-to-chord ratio (TOVCT)	
Altitude (HT)	Feet
Body diameter (D)	Feet
Missile length (XL)	Feet
Nose length (XLNOSE)	Feet
Distance from nose to CG (XCG)	Feet
Reference area (AREA)	Square feet
Reference length (XREF)	Feet
Engine code (ENGINE)	0.0-Turbofan 1.0-Rocket
Inlet code (ENLET)	0.0-Flush 1.0-Extended
Boat tail angle (BETA)	Degrees
Base diameter (DBASE)	Feet
Nozzle exit diameter (DJET)	Feet
Boat tail length (XLABOD)	Feet
Protuberance drag (CDPROT)	(Coefficient value)
If comparing with experimental values, Reynolds number (REFT)	(Dimensionless)

12. If you desire to rerun the problem, or want to run a new problem, answer the questions appropriately when asked by the terminal after the execution of the current problem.

13. To receive the printout and plot of your encounters, answer "no" to rerunning or restarting the problem when asked by the terminal and follow the directions presented on the screen.

14. Upon completion of the program, type "LOGOFF" and press $\boxed{\text{ENTER}}$.

15. Turn the terminal off with the red $\boxed{\begin{smallmatrix}\text{I}\\ \text{O}\end{smallmatrix}}$ switch.

C. EXAMPLE PROBLEMS

Table (V-5) identifies the output variables as they appear in the output tables.

1. Example V-A. Tail control missile

Figure (V-1) illustrates the missile used in this example. The dimensions for this missile and other input parameters are contained in Table (V-1). The output is shown in Table (V-2) and Figure (V-2).

2. Example V-II. Canard control missile

Figure (V-3) illustrates the conard configuration missile used in this example. The input data is contained in Table (V-3). The output is displayed in Table (V-4) and Figure (V-4).

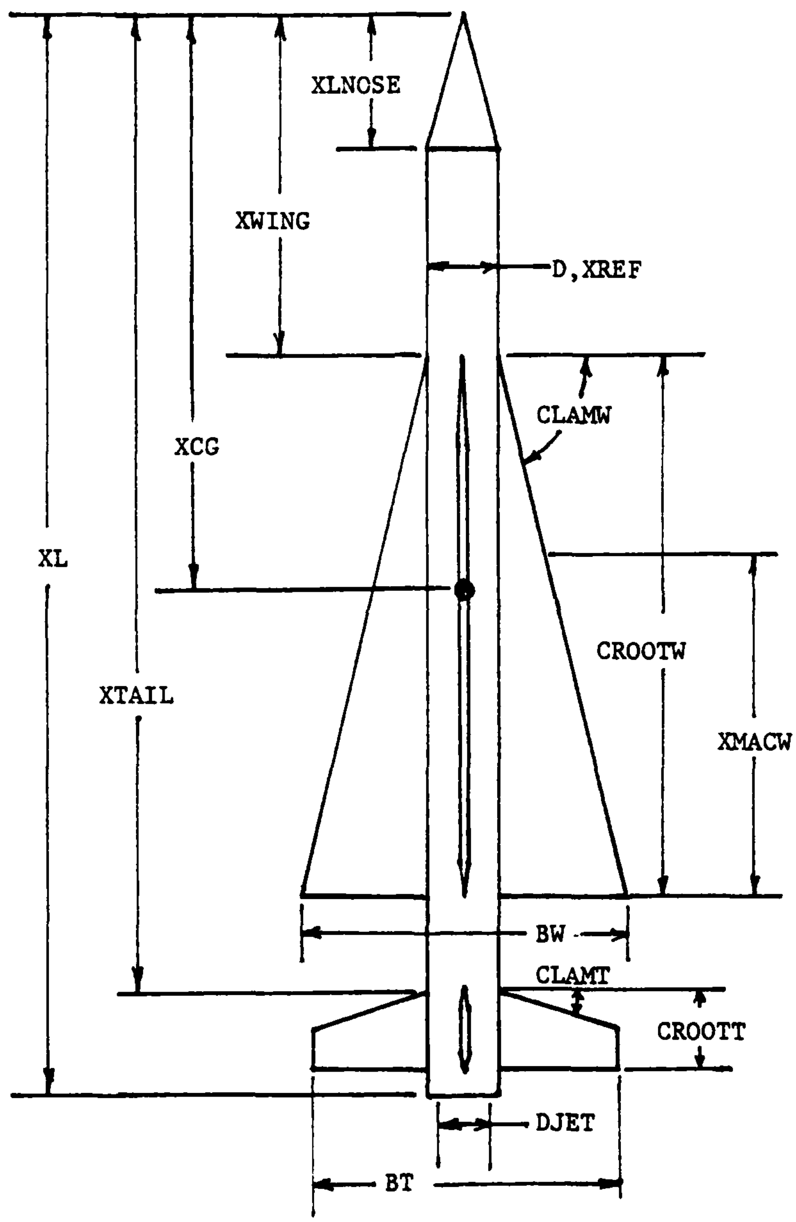


Figure (V-1). Tail control missile as used in Example V-A

THE FOLLOWING TABLE CONTAINS THE INPUT DATA FOR

EXAMPLE V-A. TAIL CONTROL MISSILE

1)	(ICSC) CONTROL CONSTANT; 1=TAIL, 2=WING, 3=CANARD:	1
2)	(INOSE) NOSE SHAPE; 1=ELLIPSE, 2=OGIVE, 3=CONE;	3
3)	(IDT) NUMBER OF CONTROL DEFLECTIONS:	5
4)	(IM) NUMBER OF MACH NUMBERS:	1
5)	(IAL) NUMBER OF ANGLES OF ATTACK:	11
6)	(NBODY) NUMBER OF CONFIGURATIONS:	1
7)	(ISWPW) 1=NON-DELTA WING, 2=DELTA WING:	2
8)	(IAFBW) 0=NO BODY AFTER WING, 1=BODY AFTER WING:	1
9)	(ISWEPW) WING SWEEP CONSTANT (IF DELTA=0) 0=UNSWEPT LEADING EDGE, 1=SWEPT LEADING EDGE:	0
10)	(NWING) NUMBER OF WINGS:	4
11)	(ISWPT) 1=NON-DELTA TAIL, 2=DELTA TAIL:	1
12)	(IAFBT) 0=NO BODY AFTER TAIL,1=BODY AFTER TAIL:	1
13)	(ISWEPT) TAIL SWEEP CONSTANT (IF DELTA=0) 0=UNSWEPT LEADING EDGE, 1=SWEPT LEADING EDGE:	1
14)	(NTAIL) NUMBER OF TAILS:	4
15)	(XLAMW) TIP-TO-CHORD RATIO OF WING:	0.0
16)	(CLAMW) WING LEADING EDGE SWEEP(DEGREES):	77.000
17)	(BW) WING SPAN, INCLUDING BODY:	1.800
18)	(CROOTW) WING ROOT CHORD (AT BODY JUNCTION):	2.960
19)	(SW) EXPOSED WING AREA (TWO PANELS):	2.072
20)	(XMACW) WING MEAN GEOMETRIC CHORD:	1.973
21)	(XWING) DISTANCE FROM NOSE TO WING LE:	1.902
22)	(TOVCW) WING THICKNESS TO CHORD RATIO:	0.030
23)	(XLAMT) TIP-TO-CHORD RATIO OF TAIL:	0.609
24)	(CLAMT) TAIL LEADING EDGE SWEEP (DEGREES):	15.000
25)	(BT) TAIL SPAN, INCLUDING BODY:	1.700
26)	(CROOTT) TAIL ROOT CHORD:	0.400
27)	(ST) EXPOSED TAIL AREA (TWO PANELS):	0.418
28)	(XMACT) TAIL MEAN GEOMETRIC CHORD:	0.328
29)	(XTAIL) DISTANCE FROM NOSE TO TAIL LE:	5.420
30)	(TOVCT) TAIL THICKNESS TO CHORD RATIO:	0.076
31)	(HT) ALTITUDE:	30000.000
32)	(D) BODY DIAMETER:	0.400
33)	(XL) MISSILE LENGTH:	6.000
34)	(XLNOSE) NOSE LENGTH:	0.750
35)	(XCG) DISTANCE TO CG LOCATION FROM NOSE:	3.200
36)	(AREA) REFERENCE AREA:	0.127
37)	(XREF) REFERENCE LENGTH:	0.400
38)	(ENGINE) ENGINE; 0.0=TURBOFAN, 1.0=ROCKET:	1.0
39)	(ENLET) INLET; 0.0=FLUSH, 1.0=EXTENDED:	0.0
40)	(BETA) BOAT-TAIL ANGLE (DEGREES):	0.0
41)	(DBASE) BASE DIAMETER:	0.400
42)	(DJET) NOZZLE EXIT DIAMETER:	0.250
43)	(XLABCD) BOAT-TAIL LENGTH:	0.0
44)	(CDPRCT) PROTUBERANCE DRAG:	0.0

MACH	2.000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
DELTA	0.0	4.00	8.00	12.00	16.00	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ALPHA	0.0	1.00	2.00	3.00	4.00	5.00	6.00	7.00	8.00	9.00	10.00
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Table (V-1). Input data for Example V-A

MACH 2.0000
DELTA= 0.0

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	0.00	0.52	0.00	0.00	-0.00	0.00	0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.92	3.90	5.71	0.00	3.72	-0.00
1.	0.73	0.93	0.61	0.10	-0.03	0.03	0.03	0.01	0.62	-0.03	0.00	0.00	0.75	0.92	3.90	5.71	0.51	3.59	-0.91
2.	1.52	0.97	1.24	0.19	-0.04	0.05	0.08	0.04	1.25	-0.04	0.00	0.00	1.55	0.91	3.90	5.71	0.87	3.72	-2.01
3.	2.36	1.03	1.88	0.29	-0.02	0.08	0.14	0.10	1.90	-0.02	0.00	0.00	2.40	0.90	3.90	5.71	1.15	3.75	-3.30
4.	3.25	1.12	2.53	0.38	0.03	0.10	0.21	0.20	2.55	0.04	0.00	0.00	3.30	0.89	3.90	5.71	1.37	3.78	-4.79
5.	4.18	1.26	3.19	0.48	0.10	0.13	0.29	0.33	3.22	0.11	0.00	0.00	4.25	0.89	3.90	5.71	1.55	3.81	-6.45
6.	5.16	1.43	3.86	0.57	0.19	0.15	0.39	0.51	3.90	0.20	0.00	0.00	5.25	0.88	3.90	5.71	1.70	3.83	-8.24
7.	6.18	1.66	4.54	0.66	0.30	0.18	0.50	0.73	4.59	0.32	0.00	0.00	6.28	0.89	3.90	5.71	1.83	3.85	-10.14
8.	7.23	1.93	5.23	0.75	0.42	0.20	0.62	1.01	5.29	0.44	0.00	0.00	7.35	0.91	3.90	5.71	1.93	3.86	-12.14
9.	8.31	2.26	5.92	0.85	0.56	0.22	0.76	1.34	5.99	0.57	0.00	0.00	8.44	0.93	3.90	5.71	2.03	3.87	-14.22
10.	9.41	2.65	6.62	0.94	0.70	0.25	0.91	1.72	6.71	0.72	0.00	0.00	9.57	0.97	3.90	5.71	2.11	3.88	-16.36

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.3326
CDOW=0.3042 CDOT=0.1489 CDMISC=0.0752 CDOWBT=0.9230

MACH 2.0000
DELTA= 4.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	0.79	0.94	0.00	0.00	0.60	0.19	0.00	0.01	0.00	0.60	0.59	0.01	0.79	0.94	3.90	5.71	0.00	5.78	-5.12
1.	1.54	0.97	0.61	0.10	0.58	0.21	0.03	0.04	0.62	0.58	0.59	0.01	1.55	0.95	3.90	5.71	0.51	4.77	-6.09
2.	2.33	1.03	1.24	0.19	0.58	0.24	0.08	0.10	1.25	0.59	0.59	0.01	2.36	0.95	3.90	5.71	0.87	4.43	-7.24
3.	3.18	1.12	1.88	0.29	0.61	0.26	0.14	0.19	1.90	0.62	0.59	0.01	3.22	0.95	3.90	5.71	1.15	4.27	-8.60
4.	4.08	1.24	2.53	0.38	0.67	0.29	0.21	0.31	2.55	0.68	0.59	0.01	4.13	0.95	3.90	5.71	1.37	4.18	-10.13
5.	5.02	1.40	3.19	0.48	0.75	0.31	0.29	0.47	3.22	0.76	0.59	0.01	5.09	0.96	3.90	5.71	1.55	4.13	-11.42
6.	6.00	1.61	3.86	0.57	0.85	0.34	0.39	0.68	3.90	0.86	0.59	0.01	6.09	0.97	3.90	5.71	1.70	4.10	-13.64
7.	7.02	1.86	4.54	0.66	0.96	0.36	0.50	0.93	4.59	0.97	0.58	0.01	7.12	0.99	3.90	5.71	1.83	4.07	-15.56
8.	8.09	2.17	5.23	0.75	1.09	0.38	0.62	1.24	5.29	1.10	0.58	0.01	8.19	1.02	3.90	5.71	1.93	4.06	-17.57
9.	9.16	2.53	5.92	0.85	1.22	0.41	0.76	1.60	5.99	1.24	0.58	0.01	9.29	1.07	3.90	5.71	2.03	4.05	-19.66
10.	10.27	2.95	6.62	0.94	1.37	0.43	0.91	2.02	6.71	1.38	0.58	0.01	10.42	1.12	3.90	5.71	2.11	4.04	-21.80

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.3326
CDOW=0.3042 CDOT=0.1499 CDMISC=0.0752 CDOWBT=0.9230

MACH 2.0000
DELTA= 8.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	1.59	1.00	0.00	0.00	1.22	0.37	0.00	0.05	0.00	1.22	1.18	0.03	1.59	1.00	3.90	5.71	0.00	5.78	-10.27
1.	2.34	1.05	0.61	0.10	1.20	0.40	0.03	0.10	0.62	1.21	1.17	0.03	2.36	1.01	3.90	5.71	0.51	5.11	-11.29
2.	3.15	1.13	1.24	0.19	1.21	0.42	0.08	0.18	1.25	1.22	1.17	0.03	3.17	1.02	3.90	5.71	0.87	4.77	-12.49
3.	4.00	1.24	1.88	0.29	1.25	0.45	0.14	0.29	1.90	1.26	1.17	0.03	4.04	1.03	3.90	5.71	1.15	4.57	-13.89
4.	4.91	1.39	2.53	0.38	1.31	0.47	0.21	0.44	2.55	1.32	1.16	0.03	4.96	1.04	3.90	5.71	1.37	4.45	-15.45
5.	5.85	1.58	3.19	0.48	1.40	0.49	0.29	0.63	3.22	1.41	1.16	0.03	5.92	1.07	3.90	5.71	1.55	4.36	-17.16
6.	6.84	1.82	3.86	0.57	1.50	0.52	0.39	0.87	3.90	1.51	1.15	0.03	6.92	1.09	3.90	5.71	1.70	4.30	-18.99
7.	7.86	2.10	4.54	0.66	1.62	0.54	0.50	1.15	4.59	1.63	1.15	0.03	7.95	1.13	3.90	5.71	1.83	4.25	-20.92
8.	8.91	2.44	5.23	0.75	1.74	0.56	0.62	1.49	5.29	1.76	1.14	0.03	9.02	1.18	3.90	5.71	1.93	4.22	-22.93
9.	9.99	2.84	5.92	0.85	1.88	0.59	0.76	1.89	5.99	1.89	1.14	0.03	10.12	1.24	3.90	5.71	2.03	4.19	-25.00
10.	11.10	3.29	6.62	0.94	2.03	0.61	0.91	2.34	6.71	2.04	1.13	0.03	11.25	1.31	3.90	5.71	2.11	4.16	-27.13

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.3326
CDOW=0.3042 CDOT=0.1489 CDMISC=0.0752 CDOWBT=0.9230

Table (V-2). Output of Example V-A

MACH 2.0000
DELTA= 12.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	2.38	1.08	0.00	0.00	1.83	0.55	0.00	0.10	0.00	1.83	1.74	0.06	2.38	1.08	3.90	5.71	0.00	5.78	-15.36
1.	3.14	1.16	0.61	0.10	1.82	0.58	0.03	0.18	0.62	1.82	1.73	0.06	3.15	1.10	3.90	5.71	0.51	5.28	-16.41
2.	3.95	1.26	1.24	0.19	1.84	0.60	0.08	0.28	1.25	1.84	1.72	0.06	3.98	1.12	3.90	5.71	0.87	4.98	-17.65
3.	4.81	1.40	1.88	0.29	1.88	0.62	0.14	0.42	1.90	1.89	1.71	0.06	4.85	1.15	3.90	5.71	1.15	4.77	-19.07
4.	5.71	1.57	2.53	0.38	1.95	0.64	0.21	0.59	2.55	1.96	1.71	0.06	5.77	1.17	3.90	5.71	1.37	4.63	-20.65
5.	6.66	1.79	3.19	0.48	2.03	0.67	0.29	0.81	3.22	2.05	1.70	0.06	6.73	1.20	3.90	5.71	1.55	4.53	-22.36
6.	7.65	2.06	3.86	0.57	2.14	0.69	0.39	1.08	3.90	2.15	1.69	0.06	7.73	1.25	3.90	5.71	1.70	4.45	-24.18
7.	8.67	2.37	4.54	0.66	2.25	0.71	0.50	1.39	4.59	2.27	1.68	0.06	8.76	1.30	3.90	5.71	1.83	4.39	-26.10
8.	9.72	2.74	5.23	0.75	2.38	0.73	0.62	1.76	5.29	2.39	1.67	0.06	9.83	1.36	3.90	5.71	1.93	4.34	-28.09
9.	10.80	3.16	5.92	0.85	2.52	0.75	0.76	2.18	5.99	2.53	1.66	0.05	10.92	1.43	3.90	5.71	2.03	4.30	-30.14
10.	11.90	3.65	6.62	0.94	2.66	0.77	0.91	2.67	6.71	2.67	1.65	0.05	12.04	1.52	3.90	5.71	2.11	4.27	-32.23

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.3326
CDOW=0.3042 CDOT=0.1489 CDMISC=0.0752 CDOWBT=0.9230

MACH 2.0000
DELTA= 16.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	3.15	1.20	0.00	0.00	2.43	0.72	0.00	0.18	0.00	2.43	2.26	0.10	3.15	1.20	3.90	5.71	0.00	5.78	-20.26
1.	3.91	1.30	0.61	0.10	2.42	0.74	0.03	0.27	0.62	2.43	2.25	0.10	3.92	1.23	3.90	5.71	0.51	5.38	-21.34
2.	4.72	1.42	1.24	0.19	2.44	0.76	0.08	0.40	1.25	2.45	2.24	0.10	4.75	1.26	3.90	5.71	0.87	5.10	-22.59
3.	5.58	1.58	1.88	0.29	2.49	0.78	0.14	0.56	1.90	2.50	2.22	0.10	5.62	1.29	3.90	5.71	1.15	4.91	-24.02
4.	6.46	1.78	2.53	0.38	2.56	0.80	0.21	0.76	2.55	2.57	2.21	0.10	6.54	1.32	3.90	5.71	1.37	4.77	-25.60
5.	7.43	2.02	3.19	0.48	2.64	0.82	0.29	1.00	3.22	2.66	2.20	0.10	7.50	1.37	3.90	5.71	1.55	4.66	-27.29
6.	8.41	2.31	3.86	0.57	2.74	0.84	0.39	1.29	3.90	2.76	2.18	0.10	8.49	1.42	3.90	5.71	1.70	4.57	-29.10
7.	9.43	2.65	4.54	0.66	2.86	0.86	0.50	1.64	4.59	2.88	2.17	0.09	9.52	1.48	3.90	5.71	1.83	4.50	-30.98
8.	10.47	3.05	5.23	0.75	2.98	0.88	0.62	2.03	5.29	3.00	2.15	0.09	10.58	1.56	3.90	5.71	1.93	4.45	-32.94
9.	11.54	3.50	5.92	0.85	3.12	0.90	0.76	2.48	5.99	3.13	2.13	0.09	11.67	1.65	3.90	5.71	2.03	4.40	-34.95
10.	12.64	4.01	6.62	0.94	3.26	0.92	0.91	2.99	6.71	3.27	2.11	0.09	12.79	1.75	3.90	5.71	2.11	4.36	-37.00

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.3326
CDOW=0.3042 CDOT=0.1489 CDMISC=0.0752 CDOWBT=0.9230

Table (V-2). (Continued)

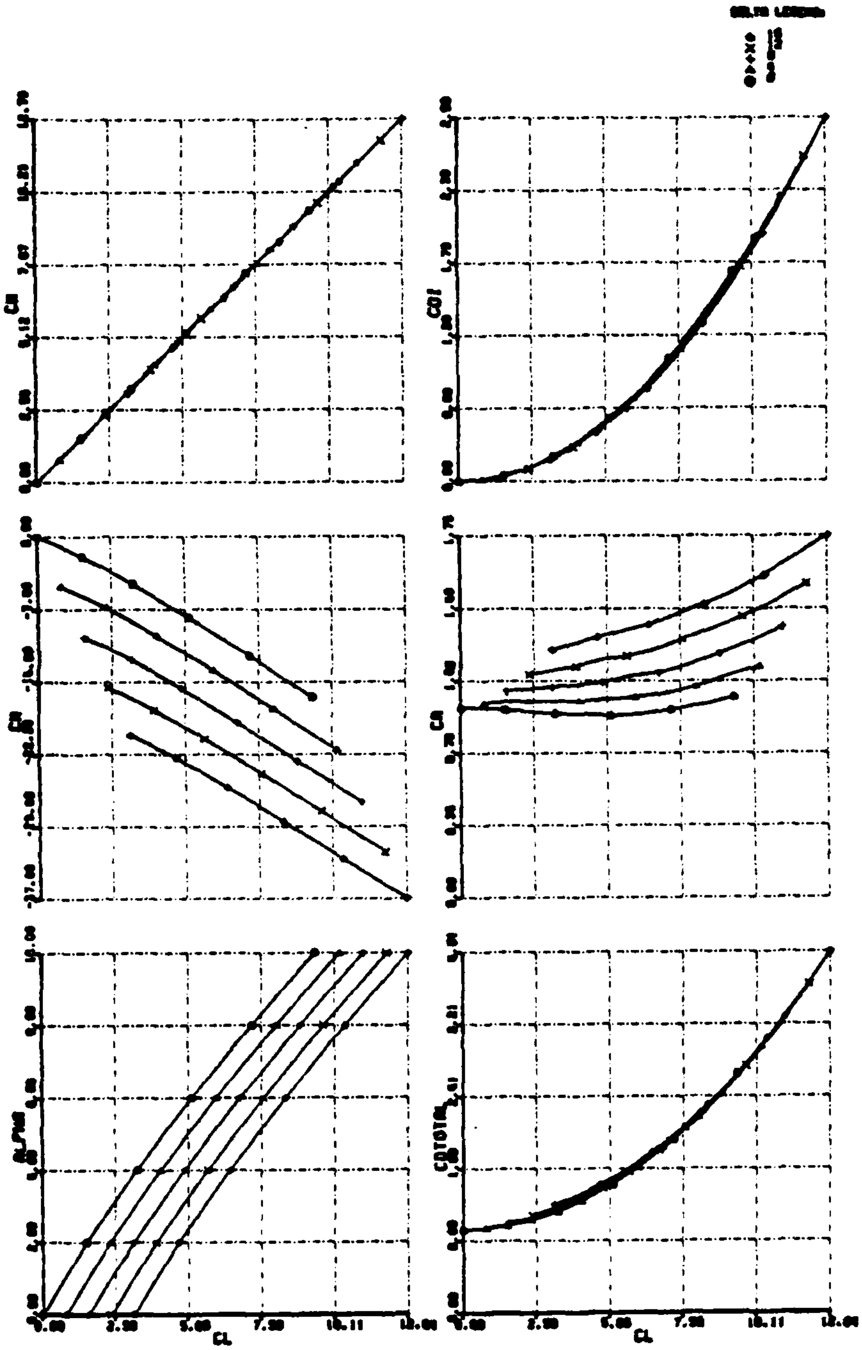


Figure (V-2). Output data plot for Example V-A

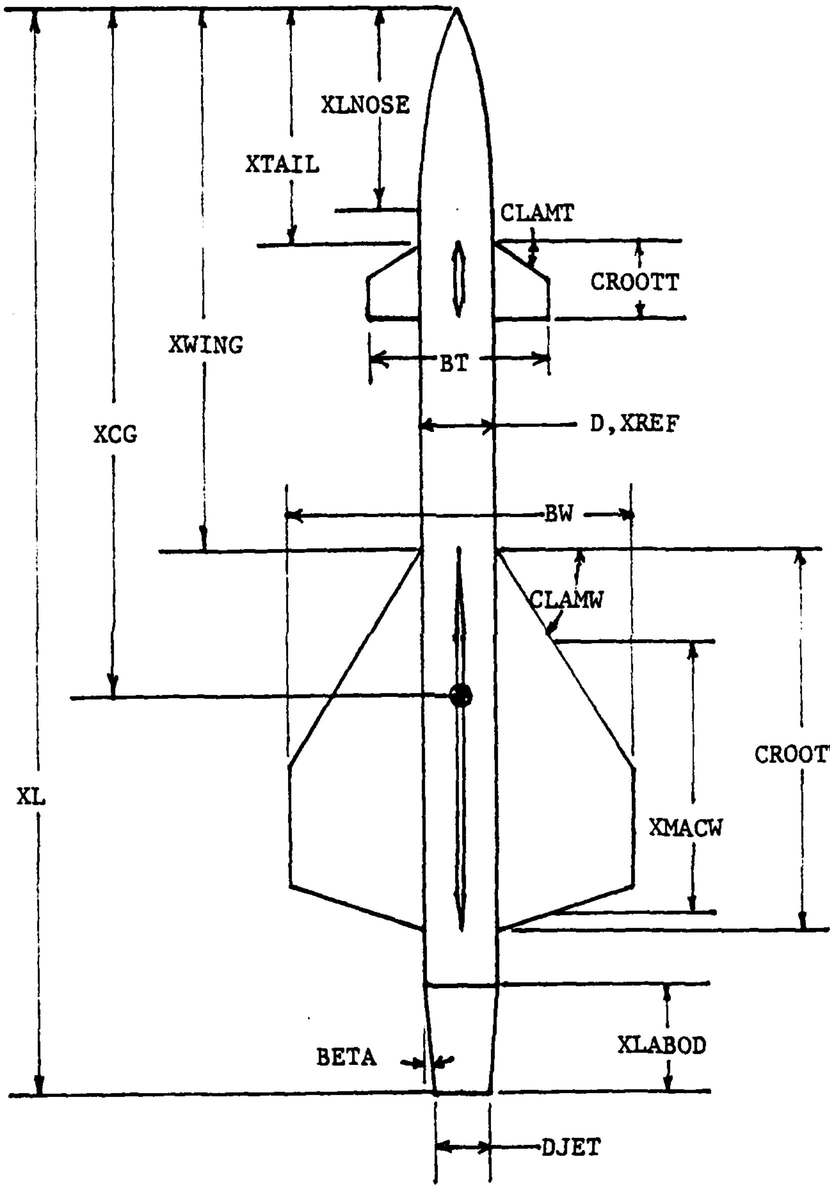


Figure (V-3). Canard control missile for Example V-B

THE FOLLOWING TABLE CONTAINS THE INPUT DATA FOR

EXAMPLE V-B. CANARD CONTROL MISSILE

```
 1) (ICSC) CONTROL CONSTANT; 1=TAIL, 2=WING, 3=CANARD:                3
 2) (INOSE) NOSE SHAPE; 1=ELLIPSE, 2=OGIVE, 3=CONE;                   2
 3) (IDT) NUMBER OF CONTROL DEFLECTIONS:                              5
 4) (IM) NUMBER OF MACH NUMBERS:                                      1
 5) (IAL) NUMBER OF ANGLES OF ATTACK:                                11
 6) (NBODY) NUMBER OF CONFIGURATIONS:                                 1
 7) (ISWPW) 1=NON-DELTA WING, 2=DELTA WING:                           1
 8) (IAFBW) 0=NO BODY AFTER WING, 1=BODY AFTER WING:                  1
 9) (ISWEPW) WING SWEEP CONSTANT (IF DELTA=0)
            0=UNSWEPT LEADING EDGE, 1=SWEPT LEADING EDGE:             1
10) (NWING) NUMBER OF WINGS:                                          4
11) (ISWPT) 1=NON-DELTA TAIL, 2=DELTA TAIL:                           1
12) (IAFBT) 0=NO BODY AFTER TAIL,1=BODY AFTER TAIL:                   1
13) (ISWEPT) TAIL SWEEP CONSTANT (IF DELTA=0)
            0=UNSWEPT LEADING EDGE, 1=SWEPT LEADING EDGE:             1
14) (NTAIL) NUMBER OF TAILS:                                          4
15) (XLAMW) TIP-TO-CHORD RATIO OF WING:                           0.314
16) (CLAMW) WING LEADING EDGE SWEEP(DEGREES):                    57.500
17) (BW) WING SPAN, INCLUDING BODY:                               3.600
18) (CRCCTW) WING ROOT CHORD (AT BODY JUNCTION):                  4.140
19) (SW) EXPOSED WING AREA (TWO PANELS):                          8.070
20) (XMACW) WING MEAN GEOMETRIC CHORD:                            2.522
21) (XWING) DISTANCE FROM NOSE TO WING LE:                        6.060
22) (TOVCW) WING THICKNESS TO CHORD RATIO:                        0.050
23) (XLAMT) TIP-TO-CHORD RATIO OF TAIL:                           0.541
24) (CLAMT) TAIL LEADING EDGE SWEEP (DEGREES):                   32.000
25) (BT) TAIL SPAN, INCLUDING BODY:                               2.000
26) (CRCCTT) TAIL ROOT CHORD:                                     0.614
27) (ST) EXPOSED TAIL AREA (TWO PANELS):                          0.568
28) (XMACT) TAIL MEAN GEOMETRIC CHORD:                            0.473
29) (XTAIL) DISTANCE FROM NOSE TO TAIL LE:                        2.700
30) (TOVCT) TAIL THICKNESS TO CHORD RATIO:                        0.085
31) (HT) ALTITUDE:                                            70000.000
32) (D) BODY DIAMETER:                                            0.800
33) (XL) MISSILE LENGTH:                                         12.000
34) (XLNOSE) NOSE LENGTH:                                         2.200
35) (XCG) DISTANCE TO CG LOCATION FROM NOSE:                      7.600
36) (AREA) REFERENCE AREA:                                        0.503
37) (XREF) REFERENCE LENGTH:                                      0.800
38) (ENGINE) ENGINE; 0.0=TURBOFAN, 1.0=ROCKET:                    1.0
39) (ENLET) INLET; 0.0=FLUSH, 1.0=EXTENDED:                       0.0
40) (BETA) BOAT-TAIL ANGLE (DEGREES):                             5.000
41) (DBASE) BASE DIAMETER:                                        0.500
42) (DJET) NOZZLE EXIT DIAMETER:                                  0.600
43) (XLABCD) BOAT-TAIL LENGTH:                                    1.200
44) (CDPRCT) PROTUBERANCE DRAG:                                   0.0

MACH   3.500 0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
       0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0

DELTA  0.0   4.00  8.00 12.00 16.00  0.0   0.0   0.0   0.0   0.0   0.0
       0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0

ALPHA  0.0   1.00  2.00  3.00  4.00  5.00  6.00  7.00  8.00  9.00 10.00
       0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0   0.0
```

Table (V-3). Input data for Example V-B

MACH 3.5000
DELTA= 0.0

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	0.00	0.49	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.49	9.13	3.20	0.00	7.84	-0.00
1.	0.49	0.50	0.32	0.08	0.05	0.01	0.04	0.00	0.33	0.05	0.00	0.00	0.50	0.49	9.13	3.20	1.02	7.90	-0.19
2.	1.03	0.51	0.68	0.15	0.10	0.01	0.09	0.02	0.69	0.10	0.00	0.00	1.05	0.48	9.13	3.20	1.76	7.91	-0.40
3.	1.61	0.55	1.06	0.23	0.15	0.02	0.15	0.05	1.08	0.15	0.00	0.00	1.63	0.46	9.13	3.20	2.33	7.91	-0.64
4.	2.23	0.59	1.46	0.30	0.20	0.03	0.23	0.10	1.49	0.21	0.00	0.00	2.26	0.44	9.13	3.20	2.79	7.92	-0.90
5.	2.88	0.66	1.89	0.37	0.25	0.04	0.33	0.17	1.92	0.26	0.00	0.00	2.92	0.41	9.13	3.20	3.17	7.92	-1.18
6.	3.57	0.76	2.34	0.45	0.30	0.04	0.43	0.26	2.37	0.31	0.00	0.00	3.62	0.38	9.13	3.20	3.49	7.92	-1.46
7.	4.30	0.88	2.81	0.52	0.36	0.05	0.56	0.38	2.85	0.37	0.00	0.00	4.36	0.35	9.13	3.20	3.76	7.92	-1.75
8.	5.07	1.03	3.30	0.59	0.41	0.06	0.71	0.54	3.34	0.42	0.00	0.00	5.13	0.31	9.13	3.20	4.01	7.92	-2.04
9.	5.88	1.22	3.81	0.67	0.47	0.05	0.87	0.73	3.85	0.48	0.00	0.00	5.94	0.29	9.13	3.20	4.22	7.91	-2.32
10.	6.72	1.45	4.33	0.74	0.52	0.07	1.07	0.96	4.38	0.53	0.00	0.00	6.80	0.26	9.13	3.20	4.42	7.90	-2.59

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.0485
CDOW=0.3331 CDOT=0.1125 CDMISC=0.0527 CDOWBT=0.4942

MACH 3.5000
DELTA= 4.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	0.20	0.50	0.00	0.00	0.13	0.07	0.00	0.00	0.00	0.13	0.12	0.00	0.20	0.50	9.13	3.20	0.00	3.53	1.01
1.	0.70	0.50	0.32	0.08	0.18	0.08	0.04	0.01	0.33	0.18	0.12	0.00	0.70	0.49	9.13	3.20	1.02	6.63	0.85
2.	1.24	0.52	0.68	0.15	0.24	0.08	0.09	0.03	0.69	0.24	0.12	0.00	1.25	0.48	9.13	3.20	1.76	7.18	0.66
3.	1.82	0.56	1.06	0.23	0.29	0.09	0.15	0.06	1.07	0.30	0.12	0.00	1.84	0.46	9.13	3.20	2.33	7.41	0.44
4.	2.44	0.61	1.46	0.30	0.34	0.10	0.23	0.12	1.48	0.35	0.12	0.00	2.47	0.44	9.13	3.20	2.79	7.54	0.20
5.	3.10	0.69	1.89	0.37	0.40	0.10	0.33	0.19	1.92	0.41	0.12	0.00	3.14	0.41	9.13	3.20	3.17	7.61	-0.05
6.	3.79	0.78	2.34	0.45	0.46	0.11	0.43	0.29	2.37	0.47	0.12	0.00	3.84	0.38	9.13	3.20	3.49	7.67	-0.32
7.	4.52	0.91	2.81	0.52	0.51	0.12	0.56	0.42	2.85	0.53	0.12	0.00	4.59	0.35	9.13	3.20	3.76	7.70	-0.59
8.	5.29	1.07	3.30	0.59	0.57	0.13	0.71	0.58	3.34	0.58	0.12	0.00	5.35	0.32	9.13	3.20	4.01	7.73	-0.86
9.	6.10	1.27	3.80	0.67	0.63	0.13	0.87	0.77	3.85	0.64	0.12	0.00	6.17	0.30	9.13	3.20	4.22	7.75	-1.13
10.	6.95	1.51	4.32	0.74	0.69	0.14	1.07	1.02	4.37	0.70	0.12	0.00	7.03	0.28	9.13	3.20	4.42	7.76	-1.39

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.0485
CDOW=0.3331 CDOT=0.1125 CDMISC=0.0527 CDOWBT=0.4942

MACH 3.5000
DELTA= 8.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	0.41	0.50	0.00	0.00	0.27	0.14	0.00	0.00	0.00	0.27	0.24	0.00	0.41	0.50	9.13	3.20	0.00	3.52	2.10
1.	0.91	0.51	0.32	0.08	0.33	0.14	0.04	0.02	0.33	0.33	0.24	0.00	0.92	0.50	9.13	3.20	1.02	5.99	1.96
2.	1.45	0.54	0.67	0.15	0.39	0.15	0.09	0.04	0.69	0.39	0.24	0.00	1.47	0.49	9.13	3.20	1.76	6.03	1.78
3.	2.04	0.58	1.05	0.23	0.44	0.16	0.15	0.08	1.07	0.45	0.24	0.00	2.06	0.47	9.13	3.20	2.33	6.89	1.58
4.	2.66	0.63	1.46	0.30	0.50	0.16	0.23	0.14	1.48	0.51	0.24	0.00	2.69	0.45	9.13	3.20	2.79	7.20	1.36
5.	3.32	0.71	1.89	0.37	0.56	0.17	0.33	0.22	1.91	0.57	0.24	0.00	3.36	0.42	9.13	3.20	3.17	7.33	1.12
6.	4.02	0.82	2.34	0.45	0.62	0.18	0.43	0.32	2.37	0.63	0.24	0.00	4.06	0.39	9.13	3.20	3.49	7.43	0.87
7.	4.75	0.95	2.81	0.52	0.68	0.18	0.56	0.45	2.84	0.69	0.24	0.00	4.81	0.37	9.13	3.20	3.76	7.50	0.61
8.	5.52	1.12	3.29	0.59	0.74	0.19	0.71	0.62	3.33	0.75	0.23	0.00	5.58	0.34	9.13	3.20	4.01	7.55	0.35
9.	6.34	1.32	3.80	0.67	0.80	0.20	0.87	0.83	3.84	0.81	0.23	0.00	6.40	0.32	9.13	3.20	4.22	7.59	0.09
10.	7.19	1.57	4.32	0.74	0.86	0.20	1.07	1.07	4.37	0.87	0.23	0.00	7.26	0.30	9.13	3.20	4.42	7.62	-0.16

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.0485
CDOW=0.3331 CDOT=0.1125 CDMISC=0.0527 CDOWBT=0.4942

Table (V-4). Output of Example V-B

MACH 3.5000
DELTA= 12.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	0.63	0.51	0.00	0.00	0.43	0.20	0.00	0.01	0.00	0.43	0.35	0.01	0.63	0.51	9.13	3.20	0.00	3.50	3.23
1.	1.13	0.52	0.32	0.08	0.49	0.21	0.04	0.02	0.33	0.49	0.35	0.01	1.14	0.50	9.13	3.20	1.02	5.42	3.11
2.	1.68	0.55	0.67	0.15	0.55	0.22	0.09	0.05	0.68	0.55	0.35	0.01	1.69	0.49	9.13	3.20	1.76	6.21	2.95
3.	2.26	0.60	1.05	0.23	0.61	0.22	0.15	0.10	1.07	0.61	0.35	0.01	2.29	0.48	9.13	3.20	2.33	6.63	2.76
4.	2.89	0.66	1.46	0.30	0.67	0.23	0.23	0.16	1.48	0.68	0.35	0.01	2.92	0.46	9.13	3.20	2.79	6.90	2.55
5.	3.55	0.74	1.88	0.37	0.73	0.23	0.33	0.24	1.91	0.74	0.35	0.01	3.59	0.43	9.13	3.20	3.17	7.08	2.32
6.	4.25	0.86	2.33	0.45	0.79	0.24	0.43	0.35	2.37	0.80	0.34	0.01	4.29	0.41	9.13	3.20	3.49	7.21	2.08
7.	4.93	0.99	2.80	0.52	0.85	0.25	0.56	0.49	2.84	0.86	0.34	0.01	5.03	0.38	9.13	3.20	3.76	7.31	1.82
8.	5.75	1.17	3.29	0.59	0.91	0.25	0.71	0.67	3.33	0.93	0.34	0.01	5.81	0.36	9.13	3.20	4.01	7.38	1.57
9.	6.57	1.36	3.80	0.67	0.97	0.26	0.87	0.88	3.84	0.99	0.34	0.01	6.63	0.33	9.13	3.20	4.22	7.44	1.31
10.	7.42	1.63	4.32	0.74	1.03	0.26	1.07	1.13	4.37	1.05	0.34	0.01	7.43	0.32	9.13	3.20	4.42	7.49	1.07

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.0485
CDOW=0.3331 CDOT=0.1125 CDMISC=0.0527 CDOWBT=0.4942

MACH 3.5000
DELTA= 16.00

AL	CLTOT	CDTOT	CLWP	CLBW	CLTP	CLBT	CLB	CDI	CNWP	CNTP	CLTD	CDTD	CN	CA	XCPW	XCPT	XCPB	XCP	CM
0.	0.85	0.52	0.00	0.00	0.59	0.26	0.00	0.01	0.00	0.59	0.46	0.01	0.85	0.52	9.13	3.20	0.00	3.49	4.39
1.	1.35	0.54	0.32	0.08	0.65	0.27	0.04	0.03	0.33	0.65	0.46	0.01	1.36	0.52	9.13	3.20	1.02	5.09	4.27
2.	1.90	0.57	0.67	0.15	0.71	0.28	0.09	0.07	0.68	0.72	0.46	0.01	1.92	0.51	9.13	3.20	1.76	5.88	4.12
3.	2.49	0.62	1.05	0.23	0.77	0.28	0.15	0.11	1.07	0.78	0.45	0.01	2.51	0.49	9.13	3.20	2.33	6.34	3.94
4.	3.11	0.69	1.45	0.30	0.84	0.29	0.23	0.18	1.48	0.84	0.45	0.01	3.14	0.47	9.13	3.20	2.79	6.65	3.74
5.	3.77	0.78	1.88	0.37	0.90	0.29	0.33	0.27	1.91	0.91	0.45	0.01	3.81	0.45	9.13	3.20	3.17	6.66	3.51
6.	4.47	0.89	2.33	0.45	0.96	0.30	0.43	0.39	2.36	0.97	0.45	0.01	4.52	0.42	9.13	3.20	3.49	7.02	3.27
7.	5.21	1.04	2.80	0.52	1.02	0.30	0.56	0.53	2.84	1.04	0.44	0.01	5.26	0.40	9.13	3.20	3.76	7.14	3.02
8.	5.98	1.22	3.29	0.59	1.08	0.31	0.71	0.71	3.33	1.10	0.44	0.01	6.04	0.37	9.13	3.20	4.01	7.23	2.76
9.	6.79	1.44	3.79	0.67	1.14	0.31	0.87	0.93	3.84	1.16	0.44	0.01	6.86	0.36	9.13	3.20	4.22	7.31	2.50
10.	7.64	1.70	4.31	0.74	1.21	0.32	1.07	1.19	4.36	1.22	0.43	0.01	7.71	0.34	9.13	3.20	4.42	7.37	2.25

CDINL=0.0 CDAFT=0.0 CDPROT=0.0 CDOB=0.0485
CDOW=0.3331 CDOT=0.1125 CDMISC=0.0527 CDOWBT=0.4942

Table (V-4). (Continued)

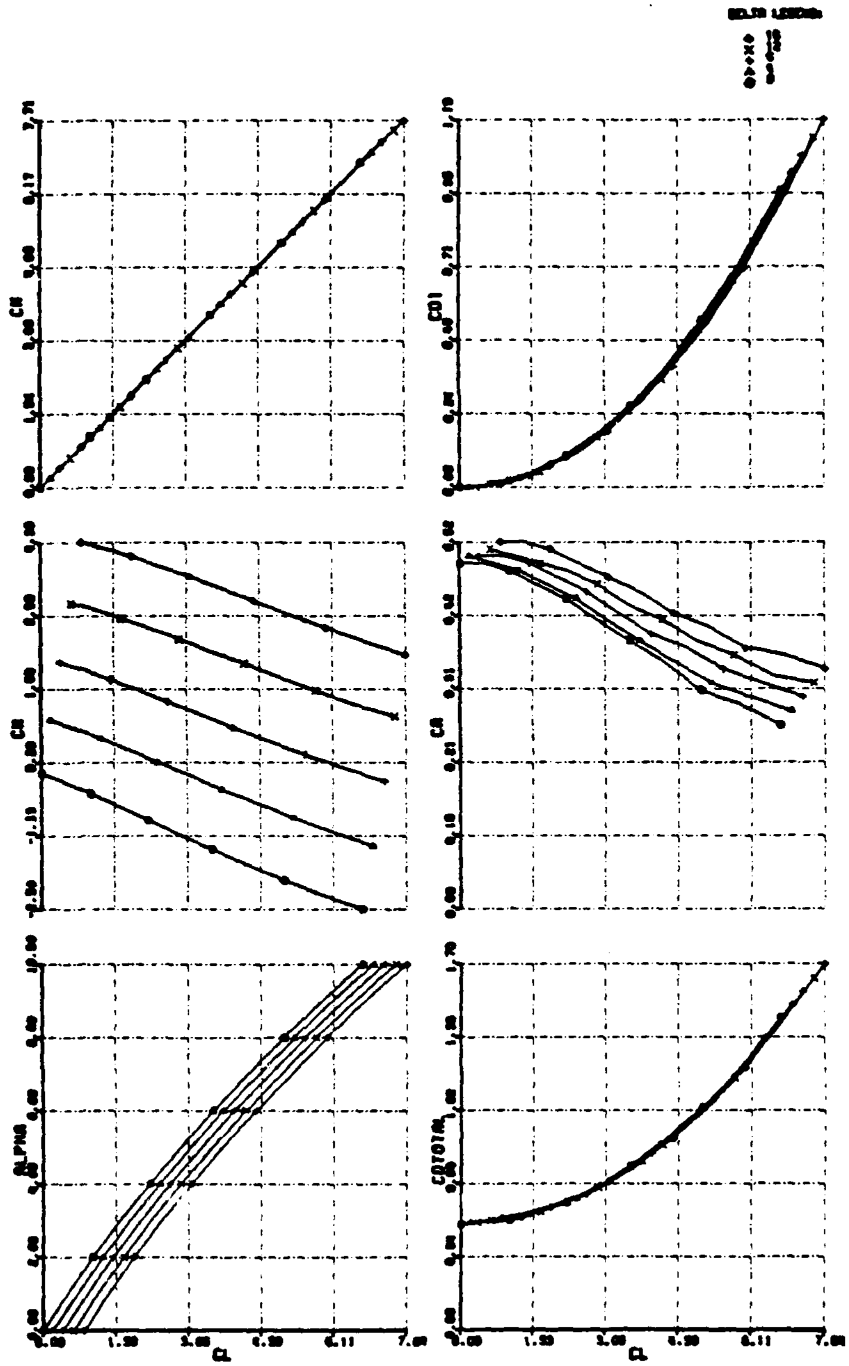


Figure (V-4). Output data plot for Example V-B

Table (V-5). Output variables of LAERO1

AL	Angle of attack
CLTOT	Total coefficient of lift
CDTOT	Total coefficient of drag
CLWP	Wing panel coefficient of lift
CLBW	Additional lift on body due to wing
CLTP	Tail panel coefficient of lift
CLBT	Additional lift on body due to tail
CLB	Body alone lift coefficient
CDI	Induced drag coefficient
CNWP	Wing panel normal force coefficient
CNTP	Tail panel normal force coefficient
CLTD	Lift coefficient due to tail deflection
CDTD	Drag coefficient due to tail deflection
CN	Total normal force coefficient
CA	Total axial force coefficient
XCPW	Wing center of pressure
XCPT	Tail center of pressure
XCP	Total missile center of pressure
CM	Total pitching moment about C.G.
CDOWBT	Zero lift drag coefficient of wing-body-tail combination
CDMISC	Miscellaneous zero lift drag coefficient
CDOT	Zero lift drag coefficient of tail
CDOW	Zero lift drag coefficient of wing
CDOB	Zero lift drag coefficient of body alone
CDPROT	Drag coefficient of body protrusions
CDINL	Drag coefficient of engine inlet
CDAFT	Drag coefficient of boattail region

VI. CONCLUSIONS AND RECOMMENDATIONS

There are many topics which may be the subjects of follow on work contained within this thesis. Although the four programs have been installed on the IBM 370 computer system, these four alone do not fully satisfy the original goal of this work: Provide a computer supplement to the Tactical Missile Conceptual Design handbook. Numerous additional focal algorithms are utilized in the design handbook which deserve the attention of a programmer. Of immediate interest are the areas concerning radar or infrared guidance systems, baseline configuration modelling and weight distribution, and initial control and lifting surface design. Each of these topics can be programmed to provide missile design students interactive learning tools when coupled with the design handbook.

The most urgent follow on work to this thesis is the restoration of the program LAERO1 to a reliable, useful program. The program was modified and set up on the IBM 370 computer system during the period immediately following the system's installation at the Naval Postgraduate School. As could be expected, the computer suffered many and varied growing pains in its early life. As a result of this, or of the human manipulation expanding the capability of the program, the effectiveness of LAERO1 was substantially reduced.

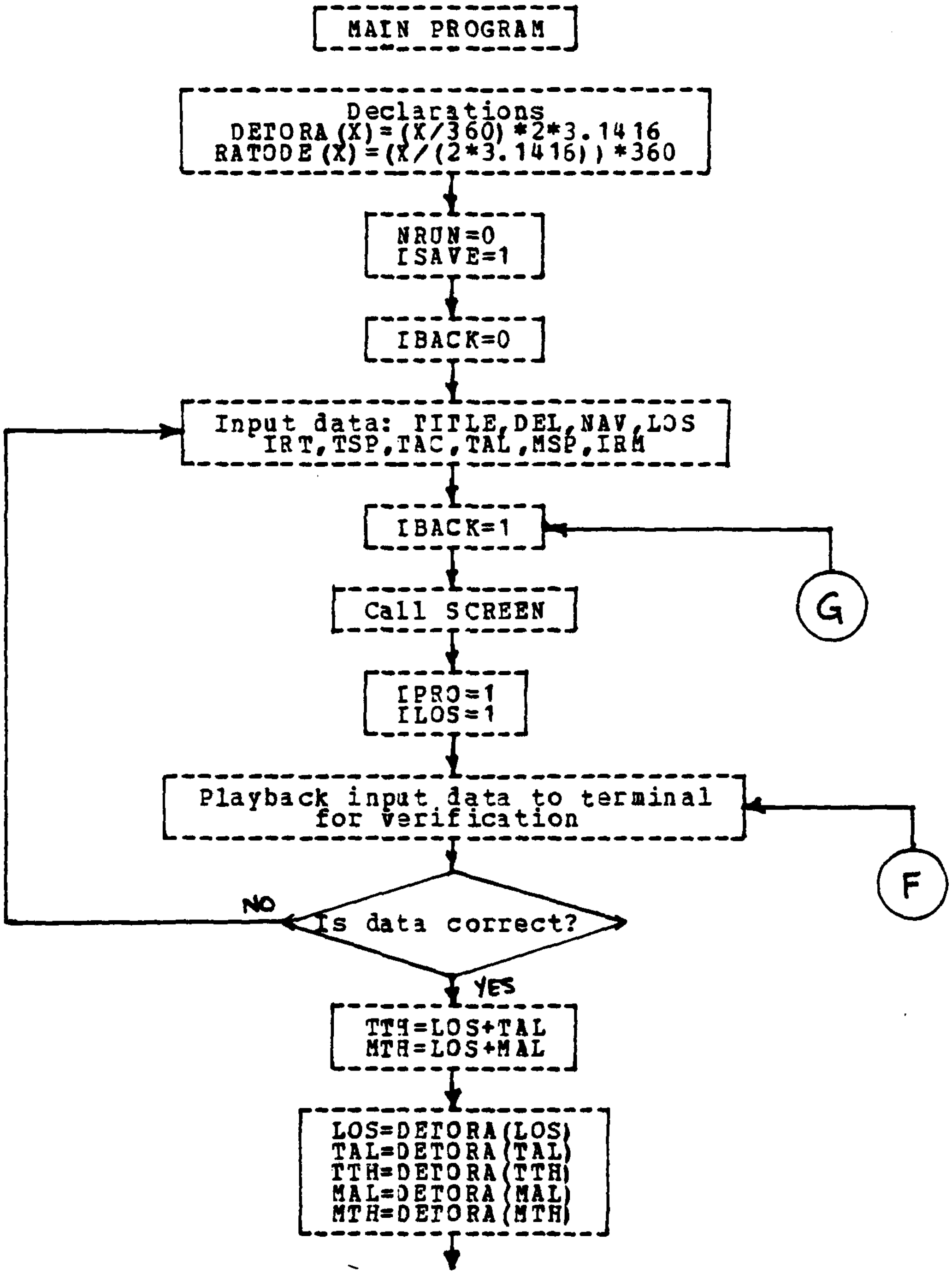
Work involving the other three programs would involve simply expanding their capabilities. The trajectory models program, LPATH, presently considers only two guidance laws: line-of-sight guidance and proportional navigation guidance. Other guidance laws which can easily be included in the program include pursuit, beam rider, and combinations of different laws. It might also prove useful to be able to simulate the entire missile trajectory but still only output the terminal phase of the encounter. Another option would be to provide the target with a controlled trajectory instead of the constant acceleration condition now imposed.

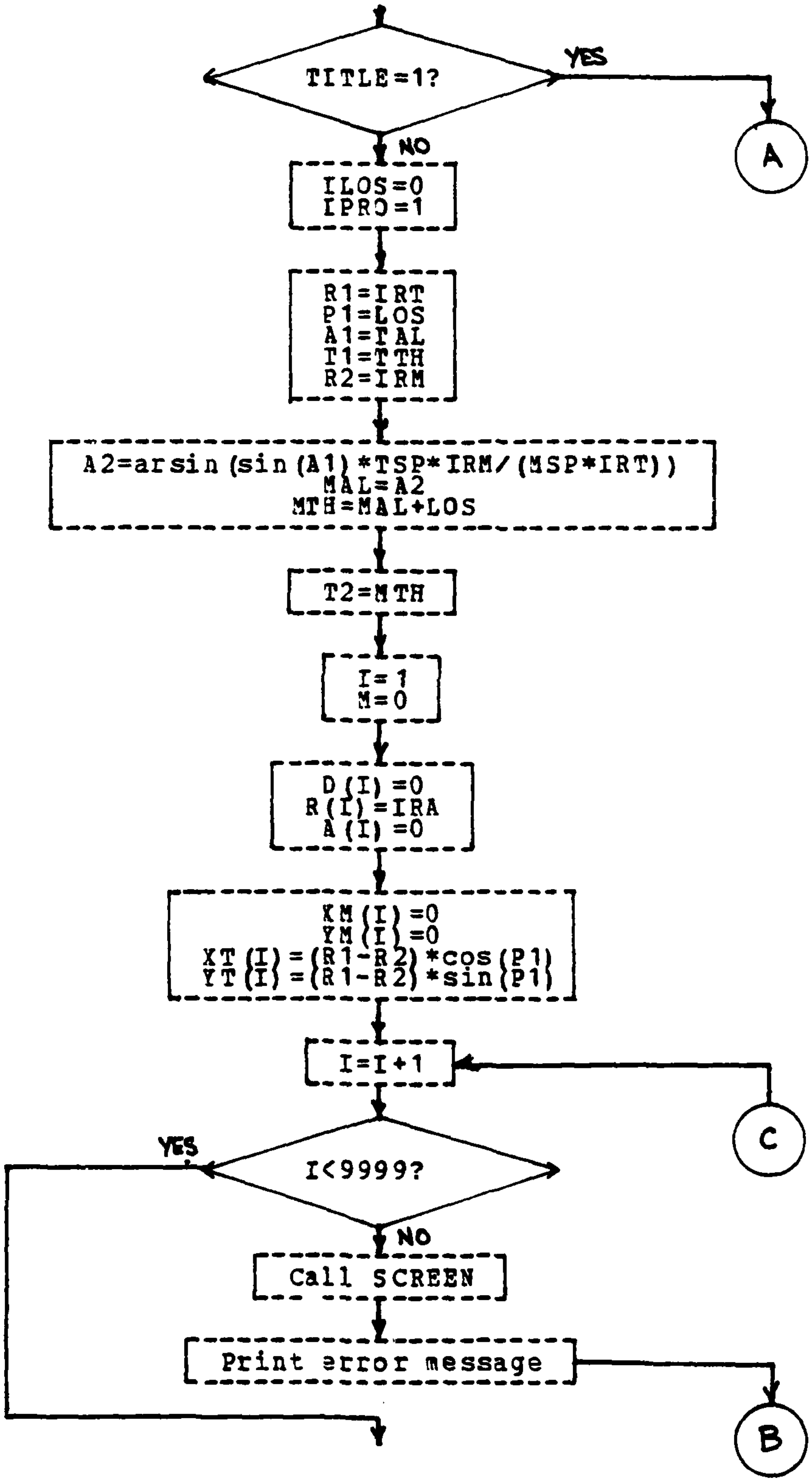
The propulsion sizing program, LPROP, should have the various nozzle options incorporated into the program so that it isn't necessary to manually juggle the program output. Other booster-sustainer grain configurations could be explored, such as the booster grain being cast within the core of the sustainer, or even a motor with only a single grain. Another suggestion for the convenience of the program users is to institute a shopping list of availlable propellants and their characteristics into the program.

A similar list of available explosives and case materials could be put into the warhead sizing program, LBOMB. These shopping lists would provide ready access to reference information and, at the same time, decrease the number of data values to be manually input into the computer. Since the current program is limited to cylindrical warheads, an area of expansion would be the flexibility of warhead styles, such as curved, shaped charge, continuous rod, etc.

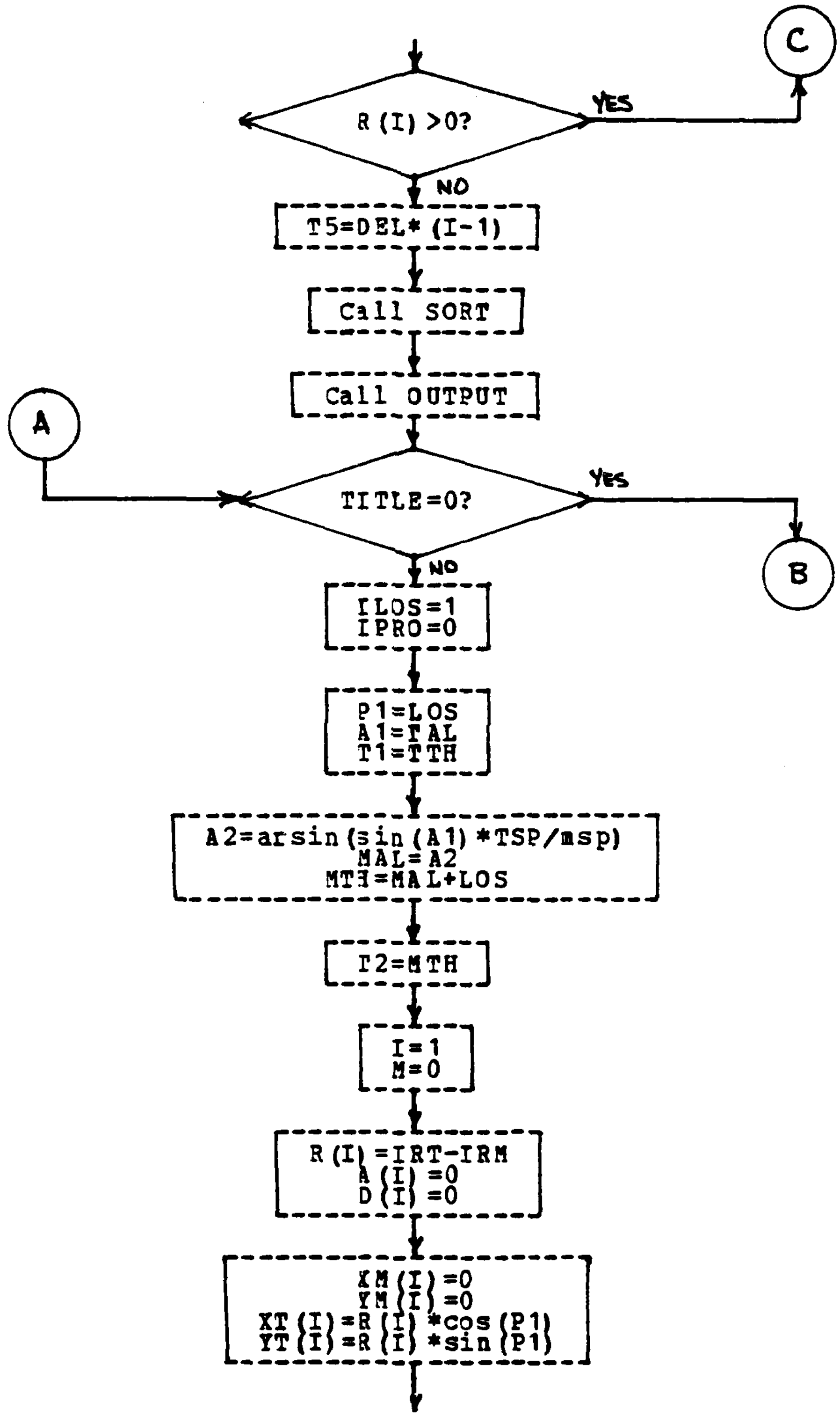
At present, the programs are somewhat hindered by the mechanics involved in producing the printouts and the plots. Due to the results of tailoring a program to be interactive, often it is required to completely exit a program before output can be received. Subsequent design iterations require re-entry into the program, which produces a certain justifyable annoyance to the user. Additionally, the computer center has instituted a new policy of cancelling any jobs with duplicate job names, which can be severely irritating and cumbersome to the persons running the plot routines contained within this thesis. However, the computer center has developed procedures which have the potential to alleviate both of these problems. According to Volume 13, Number 4 of the Computer Center Newsletter, CMS commands can now be invoked from within a FORTRAN program. The print and plot operations presently contained within executive routines can now be placed directly within the source programs. This will remove all current restrictions placed on the numbers of printouts received per session and will label each plot with the user's job name and not the project's job name.

APPENDIX A. TRAJECTORY MODELS PROGRAM FLOWCHART

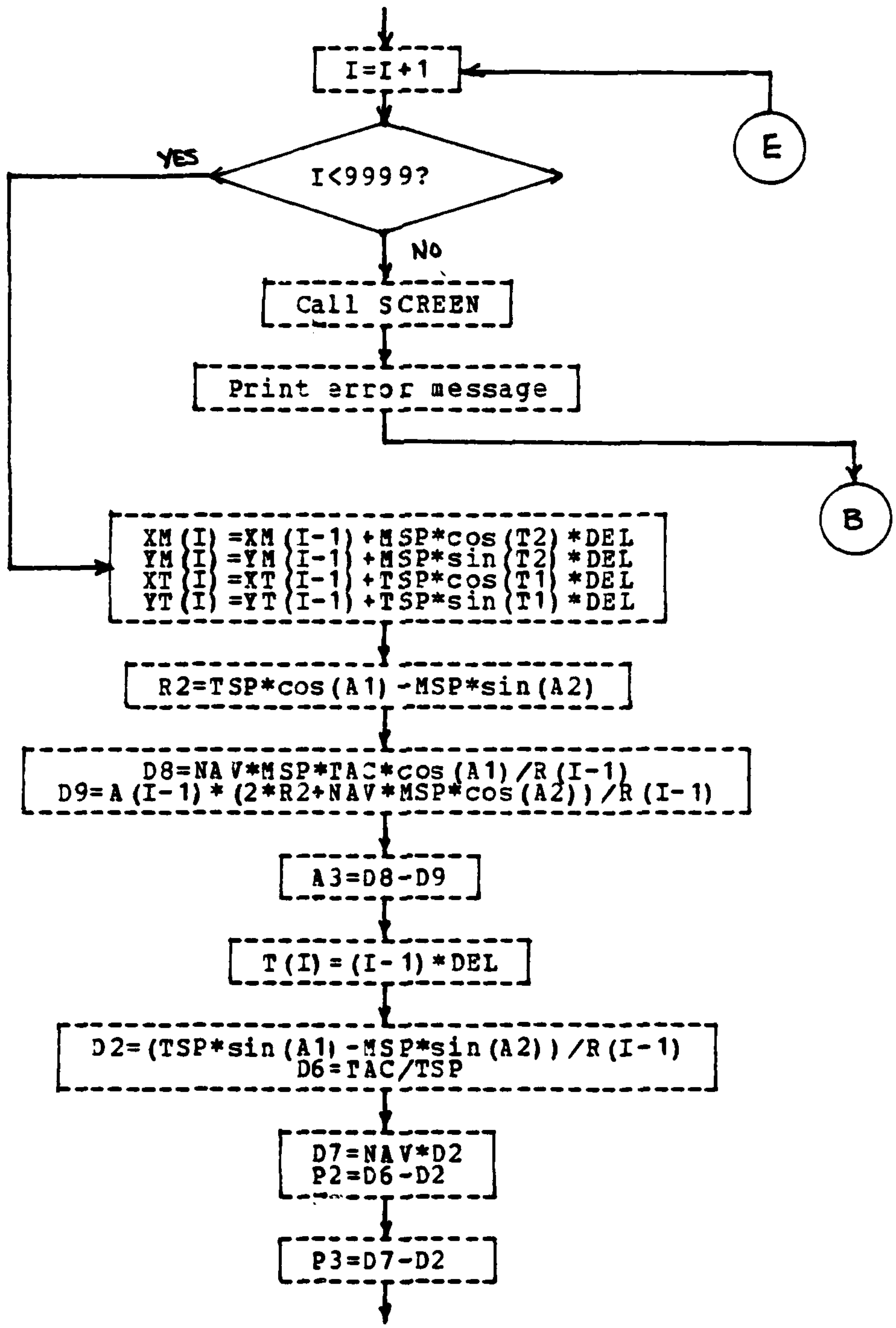



TITLE=1?
YES
A
NO
ILOS=0
IPRO=1
R1=IRT
P1=LOS
A1=TAL
T1=TTH
R2=IRM
A2=arsin(sin(A1)*TSP*IRM/(MSP*IRT))
MAL=A2
MTH=MAL+LOS
T2=MTH
I=1
M=0
D(I)=0
R(I)=IRA
A(I)=0
XM(I)=0
YM(I)=0
XT(I)=(R1-R2)*cos(P1)
YT(I)=(R1-R2)*sin(P1)
I=I+1
C
YES
I<9999?
NO
Call SCREEN
Print error message
B

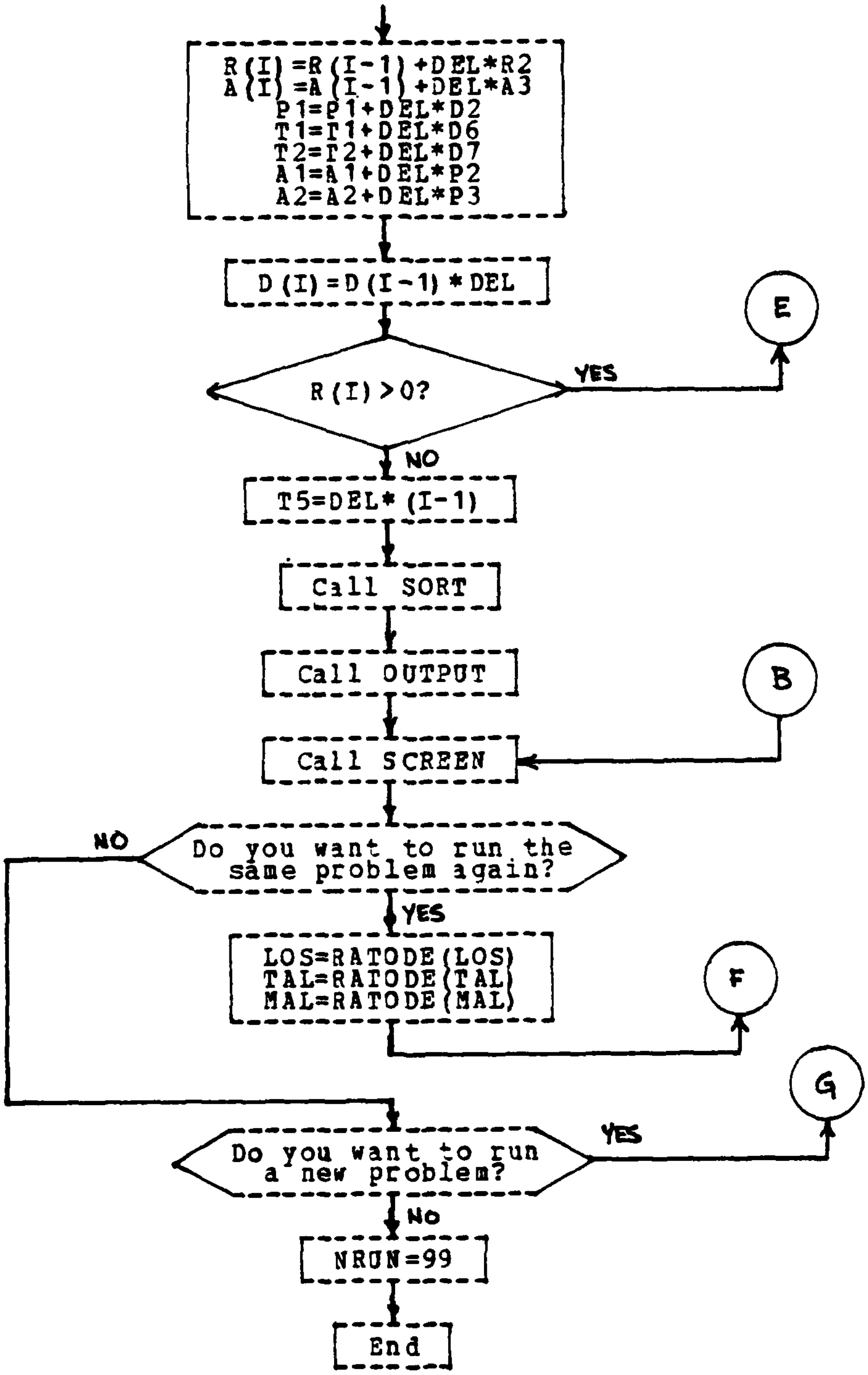
```
T3=TAC/TSP
P3=TSP*sin(A1)/R1

A3=T3-P3
R3=TSP*cos(A1)
R4=MSP*cos(A2)

D6=R4*TSP*sin(A1)
D7=R2*TSP*A3*cos(A1)
D8=-R3*MSP*sin(A2)
D9=R1*MSP*cos(A2)

A4=(D6+D7+D8)/D9

T4=A4+P3

T1=T1+DEL*T3
P1=P1+DEL*P3
A1=A1+DEL*A3
R1=R1+DEL*R3

R2=R2+DEL*R4
A2=A2+DEL*A4
T2=T2+DEL*T4

A(I)=MSP*T4

T2=A2+P1
R(I)=R1-R2

XM(I)=XM(I-1)+DEL*MSP*cos(T2)
YM(I)=YM(I-1)+DEL*MSP*sin(T2)
XT(I)=XT(I-1)+DEL*TSP*cos(T1)
YT(I)=YT(I-1)+DEL*TSP*sin(T1)

D(I)=D(I-1)+DEL
```



R(I)>0?
YES
C
NO
T5=DEL*(I-1)
Call SORT
Call OUTPUT
A
TITLE=0?
YES
B
NO
ILOS=1
IPRO=0
P1=LOS
A1=TAL
T1=TTH
A2=arsin(sin(A1)*TSP/msp)
MAL=A2
MTH=MAL+LOS
T2=MTH
I=1
M=0
R(I)=IRT-IRM
A(I)=0
D(I)=0
XM(I)=0
YM(I)=0
XT(I)=R(I)*cos(P1)
YT(I)=R(I)*sin(P1)



I=I+1
I<9999?
YES
NO
E
Call SCREEN
Print error message
B
XM(I)=XM(I-1)+MSP*cos(T2)*DEL
YM(I)=YM(I-1)+MSP*sin(T2)*DEL
XT(I)=XT(I-1)+TSP*cos(T1)*DEL
YT(I)=YT(I-1)+TSP*sin(T1)*DEL
R2=TSP*cos(A1)-MSP*sin(A2)
D8=NAV*MSP*TAC*cos(A1)/R(I-1)
D9=A(I-1)*(2*R2+NAV*MSP*cos(A2))/R(I-1)
A3=D8-D9
T(I)=(I-1)*DEL
D2=(TSP*sin(A1)-MSP*sin(A2))/R(I-1)
D6=TAC/TSP
D7=NAV*D2
P2=D6-D2
P3=D7-D2


R(I)=R(I-1)+DEL*R2
A(I)=A(I-1)+DEL*A3
P1=P1+DEL*D2
T1=T1+DEL*D6
T2=T2+DEL*D7
A1=A1+DEL*P2
A2=A2+DEL*P3
D(I)=D(I-1)*DEL
R(I)>0?
YES
E
NO
T5=DEL*(I-1)
Call SORT
Call OUTPUT
B
Call SCREEN
Do you want to run the same problem again?
NO
YES
LOS=RATODE(LOS)
TAL=RATODE(TAL)
MAL=RATODE(MAL)
F
G
Do you want to run a new problem?
YES
NO
NRUN=99
End

APPENDIX B. TRAJECTORY MODELS PROGRAM LISTING

Following the next page is the program listing for the Trajectory Models program. It consists of three main divisions; the executive routines, the FORTRAN IV computational program, and the FORTRAN IV plotting program. The executive routines are used to establish required file definitions, initiate operation of the computational program, supervise the transfer of data to the plotting program, and provide operational information to the program user at appropriate times.

The computational program, LPATH FORTRAN, consists of four subprogram divisions. The MAIN program accepts the input data and performs the calculations for the line of sight and the proportional navigation problems. Subroutine SCREEN is used to prompt the user to clear the terminal screen for proper positioning of the output. Subroutine SORT determines the largest missile acceleration value and the value ranges of the X and Y position coordinates for plotting reference. Finally, subroutine OUTPUT formats the calculated data and provides it to the user, the printer file, and the plot program data file.

The plot program, PATHPLOT FORTRAN, is structured for direct submission to the MVS batch reader from the 3278 terminal. No cards need to be punched or read. Attention must be given to the correct JCL accounting data in the first line of the program; those shown in the listing are for illustrative purposes only. This program receives the data from the computational program and produces a single geographic chart of the encounter in the encounter plane. Multiple problems, up to nine, will overlay on the single chart.

FILE: LPATH   EXEC   A   NAVAL POSTGRADUATE SCHOOL

```
FILEDEF 08 DISK LPATH OUTPUT A0
FILEDEF 07 DISK PATHPLOT DATA A0
&BEGTYPE

YOU WILL HAVE THE OPTION TO OBTAIN BOTH A HARDCOPY PRINTOUT
AND A VERSATEC PLOT OF UP TO NINE ENCOUNTER GEOMETRIES.
THE PLOT IS A SINGLE FRAME WITH ALL NINE GEOMETRIES
SUPERIMPOSED ON ONE ANOTHER.  THE HARDCOPY PRINTOUT IS
IDENTICAL IN FORMAT TO THE TERMINAL OUTPUT.

&END
LOAD LPATH
START
&BEGTYPE

TO OBTAIN A HARDCOPY PRINTOUT OF THE RESULTS, TYPE AND
ENTER:

                    lpathpr

THE OUTPUT WILL BE PRINTED ON THE VM PRINTER IN ROOM 140
AND WILL BE IDENTIFIED BY YOUR USER NUMBER.  IT USUALLY
REQUIRES SEVERAL MINUTES TO OBTAIN THE PRINTOUT.

TO OBTAIN THE PLOT OF YOUR ENCOUNTERS, TYPE AND ENTER:

                    lpathpl

THE PLOT WILL BE PRINTED IN THE COMPUTER ROOM AND WILL BE
PLACED ON TOP OF THE ALPHABETIZED OUTPUT FILE IN ROOM 140.
IT WILL BE IDENTIFIED BY THE JCL JOB NAME "PATHPLOT" AND
USUALLY REQUIRES MANY MINUTES (EVEN HOURS!)  TO BE
OBTAINED.  NOTE....IF MANY USERS ARE REQUESTING PLOTS
SIMULTANEOUSLY, THE COMPUTER CENTER PERSONNEL WILL CANCEL
"EXCESS" JOBS USING THE SAME IDENTIFIER.

&END
```

FILE: LPATHPL   EXEC   A   NAVAL POSTGRADUATE SCHOOL

```
COPY LPATH PLOT A PATHPLOT DATA A PLOT FORTRAN A
EXEC SUBMIT PLOT FORTRAN A
ERASE PLOT FORTRAN A
```

FILE: LPATHPR   EXEC   A   NAVAL POSTGRADUATE SCHOOL

```
PRINT LPATH OUTPUT (LINECOUN 70
```

```
C      27 APRIL 1981
C      THIS PROGRAM COMPARES THE TRAJECTORIES OF TWO GUIDANCE SYSTEM
C      INTERCEPT GEOMETRIES:  LINE OF SIGHT AND PROPORTIONAL
C      NAVIGATION.  THE OUTPUT IS A TABLE OF GEOGRAPHIC POSITIONS, THE
C      MAXIMUM LATERAL ACCELERATION THE INTERCEPTING MISSILE MUST
C      ENDURE, AND THE TIME REQUIRED TO MAKE THE INTERCEPT.
C
C  DECLARATIONS
      REAL DEL,IRT,IRM,LOS,TSP,TAC,TAL,TTH,MSP,MAC,MAL,MTH,NAV
      REAL A1,A2,A3,A4,D2,D6,D7,D8,D9,M,M1,P1,P2,P3,R1,R2,R3
      REAL R4,T1,T2,T3,T4,T5,T2LAST
      INTEGER I,W,TITLE,AGAIN,IBACK,IDO,ISAVE,NRUN,ILOS,IPRO
      DIMENSION XM(9999),YM(9999),XT(9999),YT(9999),A(9999),R(9999)
      DIMENSION T(9999),RP(9999),D(9999),APN(9999),DT2(9999)
      COMMON/BLOCK1/A,XM,YM,XT,YT,M,XPOS,XNEG,YPOS,YNEG,I
      COMMON/BLOCK2/DEL,NAV,LOS,IRT,IRM,TSP,TAC,TAL,MSP,MAC,MAL,D,R,T5,
     *ISAVE,NRUN,ILOS,IPRO
      DETORA(X)=(X/360)*2*3.1415926535
      RATODE(X)=(X/(2*3.1415926535))*360
      NRUN=0
      ISAVE=1
    1 IBACK=0
C
C  INPUT PROBLEM PARAMETERS
      WRITE (6,12)
   12 FORMAT (/1X,T13,'INPUT PROBLEM PARAMETERS')
      WRITE (6,14)
   14 FORMAT (1X,T2,'FOR LINE OF SIGHT GUIDANCE INTERCEPT TRAJECTORY, IN
     +PUT "0":'/,T2,'FOR PROPORTIONAL NAVIGATION GUIDANCE INTERCEPT TRAJ
     +ECTORY, INPUT "1":'/,T2,'FOR BOTH INTERCEPT TRAJECTORIES, INPUT "2
     +".  (INTEGER)')
      READ (6,16)TITLE
   16 FORMAT (1I1)
10001 WRITE (6,15)
   15 FORMAT (1X,T2,'INPUT INTEGRATION TIME INCREMENT (SECONDS, DECIMAL)
     *:')
      READ (6,117)DEL
  117 FORMAT (F10.5)
   17 FORMAT (F10.3)
      IF (IBACK.EQ.1) GO TO 204
10002 WRITE (6,18)
   18 FORMAT (1X,T2,'INPUT NAVIGATION CONSTANT FOR PROPORTIONAL NAVIGATI
     *ON (DECIMAL):')
      READ (6,17)NAV
      IF (IBACK.EQ.1) GO TO 204
10003 WRITE (6,19)
   19 FORMAT (1X,T2,'INPUT LINE-OF-SIGHT ANGLE (DEGREES, DECIMAL):')
      READ (6,17)LOS
```

AD-A105 788 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 16/4
COMPUTER PROGRAM APPLICATIONS TO TACTICAL MISSILE CONCEPTUAL DE--ETC(U)
JUN 81 M D SULLIVAN

UNCLASSIFIED NL

2 of 3
AD A
105788

```
      IF (IBACK.EQ.1) GO TO 204                                         LPA00520
10004 WRITE (6,20)                                                      LPA00530
   20 FORMAT (1X,T2,'INPUT TARGET RANGE FROM MISSILE LAUNCH SITE (METERSLPA00540
     *, DECIMAL):')                                                     LPA00550
      READ (6,17)IRT                                                    LPA00560
      IF (IBACK.EQ.1) GO TO 204                                         LPA00570
10005 WRITE (6,21)                                                      LPA00580
   21 FORMAT (1X,T2,'INPUT TARGET SPEED (METERS/SECOND, DECIMAL):')     LPA00590
      READ (6,17)TSP                                                    LPA00600
      IF (IBACK.EQ.1) GO TO 204                                         LPA00610
10006 WRITE (6,22)                                                      LPA00620
   22 FORMAT (1X,T2,'INPUT TARGET LATERAL ACCELERATION (M/SEC/SEC, DECIMLPA00630
     *AL):')                                                            LPA00640
      READ (6,17)TAC                                                    LPA00650
      IF (IBACK.EQ.1) GO TO 204                                         LPA00660
10007 WRITE (6,23)                                                      LPA00670
   23 FORMAT (1X,T2,'INPUT TARGET ALPHA (DEGREES, DECIMAL):')           LPA00680
      READ (6,17)TAL                                                    LPA00690
      IF (IBACK.EQ.1) GO TO 204                                         LPA00700
10008 WRITE (6,25)                                                      LPA00710
   25 FORMAT (1X,T2,'INPUT MISSILE SPEED (METERS/SECOND, DECIMAL):')    LPA00720
      READ (6,17)MSP                                                    LPA00730
      IF (IBACK.EQ.1) GO TO 204                                         LPA00740
10009 WRITE (6,26)                                                      LPA00750
   26 FORMAT (1X,T2,'INPUT MISSILE RANGE FROM LAUNCH SITE (METERS, DECIMLPA00760
     *AL):')                                                            LPA00770
      READ (6,17)IRM                                                    LPA00780
C                                                                       LPA00790
C  COVER PAGE PRINTOUT                                                  LPA00800
      IBACK=1                                                           LPA00810
  204 CALL SCREEN                                                       LPA00820
      IPRO=1                                                            LPA00830
      ILOS=1                                                            LPA00840
      WRITE (6,33)                                                      LPA00850
   33 FORMAT (/1X,T28,'PROBLEM PARAMETERS')                             LPA00860
      WRITE (6,132)DEL,NAV,LOS,IRT,TSP,TAC,TAL,MSP,IRM                  LPA00870
  132 FORMAT (1X,T16,'01) TIME INCREMENT',T40,F10.4,' SECONDS'/,        LPA00880
     +T16,'02) NAVIGATION CONSTANT',T39,F10.3/,                         LPA00890
     +T16,'03) LINE-OF-SIGHT ANGLE',T39,F10.3,' DEGREES'/,              LPA00900
     +T16,'04) TARGET RANGE',T39,F10.3,' METERS'/,                      LPA00910
     +T16,'05) TARGET SPEED',T39,F10.3,' METERS/SEC'/,                  LPA00920
     +T16,'06) TARGET ACCELERATION',T39,F10.3,' M/SEC/SEC'/,            LPA00930
     +T16,'07) TARGET ALPHA',T39,F10.3,' DEGREES'/,                     LPA00940
     +T16,'08) MISSILE SPEED',T39,F10.3,' METERS/SEC'/,                 LPA00950
     +T16,'09) MISSILE RANGE',T39,F10.3,' METERS')                      LPA00960
      WRITE (6,150)                                                     LPA00970
  150 FORMAT (1X/,T15,'IS THIS DATA SUMMARY CORRECT? (00=YES, IF NO,'/, LPA00980
     *T15,'ENTER THE TWO DIGIT NUMBER OF THE INCORRECT ITEM.'/,         LPA00990
```

```
     *T15,'(TWO-DIGIT INTEGER)')
      READ (6,161)AGAIN
  161 FORMAT (1I2)
      IF (AGAIN.EQ.00) GO TO 205
      GO TO ( 10001,10002,10003,10004,10005,10006,10007,10008,10009),
     *AGAIN
C
C
 205  TTH=LOS+TAL
      MTH=LOS+MAL
      LOS=DETORA(LOS)
      TAL=DETORA(TAL)
      TTH=DETORA(TTH)
      MAL=DETORA(MAL)
      MTH=DETORA(MTH)
C
C
      IF (TITLE.EQ.1) GO TO 299
C
C  LINE OF SIGHT GUIDANCE INTERCEPT TRAJECTORY CALCULATIONS
      ILOS=0
      IPRO=1
C
C  PROGRAM
      R1=IRT
      P1=LOS
      A1=TAL
      T1=TTH
      R2=IRM
      A2=ARSIN(SIN(A1)*TSP*IRM/(MSP*IRT))
      MAL=A2
      MTH=LOS+MAL
      T2=MTH
      I=1
      M=0
      D(I)=0
      R(I)=IRA
      A(I)=0
      XM(I)=0
      YM(I)=0
      XT(I)=(R1-R2)*COS(P1)
      YT(I)=(R1-R2)*SIN(P1)
  210 I=I+1
      IF (I.LT.9999) GO TO 211
      CALL SCREEN
      WRITE (6,212)
  212 FORMAT (1X,'YOU HAVE SUCCESSFULLY EXCEEDED THE CAPABILITY OF THIS
     *PROGRAM'/,1X,'BY REQUIRING MORE THAN 9999 ITERATIONS TO COMPLETE
```

```
     *THE SOLUTION.'/,1X,'IF YOU STRONGLY DESIRE A SOLUTION FOR THIS ENCLPA01480
     *OUNTER, INCREASE'/,1X,'THE SIZE OF YOUR TIME INCREMENT AND RERUN. LPA01490
     * IT IS ALSO STRONGLY RECOMMENDED'/,1X,'THAT YOU REVIEW THE GEOMETRLPA01500
     *Y OF THE ENCOUNTER TO ENSURE IT IS POSSIBLE'/,1X,'FOR THE ENCOUNTELPA01510
     *R TO OCCUR.')                                                     LPA01520
      GO TO 999                                                         LPA01530
  211 T3=TAC/TSP                                                        LPA01540
      P3=TSP*SIN(A1)/R1                                                 LPA01550
      A3=T3-P3                                                          LPA01560
      R3=TSP*COS(A1)                                                    LPA01570
      R4=MSP*COS(A2)                                                    LPA01580
      D6=R4*TSP*SIN(A1)                                                 LPA01590
      D7=R2*TSP*A3*COS(A1)                                              LPA01600
      D8=-R3*MSP*SIN(A2)                                                LPA01610
      D9=R1*MSP*COS(A2)                                                 LPA01620
      A4=(D6+D7+D8)/D9                                                  LPA01630
      T4=A4+P3                                                          LPA01640
      T1=T1+DEL*T3                                                      LPA01650
      P1=P1+DEL*P3                                                      LPA01660
      A1=A1+DEL*A3                                                      LPA01670
      R1=R1+DEL*R3                                                      LPA01680
      R2=R2+DEL*R4                                                      LPA01690
      A2=A2+DEL*A4                                                      LPA01700
      T2=T2+DEL*T4                                                      LPA01710
      A(I)=MSP*T4                                                       LPA01720
  220 T2=A2+P1                                                          LPA01730
      R(I)=R1-R2                                                        LPA01740
      XM(I)=XM(I-1)+DEL*MSP*COS(T2)                                     LPA01750
      YM(I)=YM(I-1)+DEL*MSP*SIN(T2)                                     LPA01760
      XT(I)=XT(I-1)+DEL*TSP*COS(T1)                                     LPA01770
      YT(I)=YT(I-1)+DEL*TSP*SIN(T1)                                     LPA01780
      D(I)=D(I-1)+DEL                                                   LPA01790
      IF (R(I).GT.0) GO TO 210                                          LPA01800
      T5=DEL*(I-1)                                                      LPA01810
      CALL SORT                                                         LPA01820
      CALL OUTPUT                                                       LPA01830
C                                                                       LPA01840
C                                                                       LPA01850
C                                                                       LPA01860
  299 IF (TITLE.EQ.0) GO T3 999                                         LPA01870
C                                                                       LPA01880
C  PROPORTIONAL NAVIGATION GUIDANCE INTERCEPT TRAJECTORY CALCULATIONS   LPA01890
      ILOS=1                                                            LPA01900
      IPRO=0                                                            LPA01910
C                                                                       LPA01920
C  PROGRAM                                                              LPA01930
      P1=LOS                                                            LPA01940
      A1=TAL                                                            LPA01950
```

```
      T1=TTH                                                            LPA01960
      A2=ARSIN(SIN(A1)*TSP/MSP)                                         LPA01970
      MAL=A2                                                            LPA01980
      MTH=MAL+LOS                                                       LPA01990
      T2=MTH                                                            LPA02000
      I=1                                                               LPA02010
      M=0                                                               LPA02020
      R(I)=IRT-IRM                                                      LPA02030
      A(I)=MAC                                                          LPA02040
      D(I)=0                                                            LPA02050
      XM(I)=0                                                           LPA02060
      YM(I)=0                                                           LPA02070
      XT(I)=R(I)*COS(P1)                                                LPA02080
      YT(I)=R(I)*SIN(P1)                                                LPA02090
  310 I=I+1                                                             LPA02100
      IF (I.LT.9999) GO TO 311                                          LPA02110
      CALL SCREEN                                                       LPA02120
      WRITE (6,212)                                                     LPA02130
      GO TO 999                                                         LPA02140
  311 XM(I)=XM(I-1)+MSP*COS(T2)*DEL                                     LPA02150
      YM(I)=YM(I-1)+MSP*SIN(T2)*DEL                                     LPA02160
      XT(I)=XT(I-1)+TSP*COS(T1)*DEL                                     LPA02170
      YT(I)=YT(I-1)+TSP*SIN(T1)*DEL                                     LPA02180
      R2=TSP*COS(A1)-MSP*COS(A2)                                        LPA02190
      D8=NAV*MSP*TAC*COS(A1)/R(I-1)                                     LPA02200
      D9=A(I-1)*(2*R2+NAV*MSP*COS(A2))/R(I-1)                           LPA02210
      A3=D8-D9                                                          LPA02220
      T(I)=(I-1)*DEL                                                    LPA02230
      D2=(TSP*SIN(A1)-MSP*SIN(A2))/R(I-1)                               LPA02240
      D6=TAC/TSP                                                        LPA02250
      D7=NAV*D2                                                         LPA02260
      P2=D6-D2                                                          LPA02270
      P3=D7-D2                                                          LPA02280
      R(I)=R(I-1)+DEL*R2                                                LPA02290
      A(I)=A(I-1)+DEL*A3                                                LPA02300
      P1=P1+DEL*D2                                                      LPA02310
      T1=T1+DEL*D6                                                      LPA02320
      T2=T2+DEL*D7                                                      LPA02330
      A1=A1+DEL*P2                                                      LPA02340
      A2=A2+DEL*P3                                                      LPA02350
  320 D(I)=D(I-1)+DEL                                                   LPA02360
      IF (R(I).GT.0) GO TO 310                                          LPA02370
  330 T5=DEL*(I-1)                                                      LPA02380
      CALL SORT                                                         LPA02390
      CALL OUTPUT                                                       LPA02400
C                                                                       LPA02410
C                                                                       LPA02420
  999 CALL SCREEN                                                       LPA02430
```

```
  998 WRITE (6,410)
  410 FORMAT (/1X,'DO YOU WANT TO RUN THIS PROBLEM AGAIN? 0=YES, 1=NO')
      READ (6,16) IDO
      IF (IDO.EQ.1) GO TO 420
      LOS=RATODE(LOS)
      TAL=RATODE(TAL)
      MAL=RATODE(MAL)
      GO TO 204
  420 WRITE (6,430)
  430 FORMAT (1X,'DO YOU WANT TO RUN A NEW PROBLEM? 0=YES, 1=NO')
      READ (6,16) IDO
      IF (IDO.EQ.0) GO TO 1
      NRUN=99
      WRITE (7,144) NRUN,I
      WRITE (7,145)
  145 FORMAT ('/*')
  144 FORMAT (I2,I4)
      STOP
      END
C
C
      SUBROUTINE SCREEN
      WRITE (6,600)
  600 FORMAT (1X,'CLEAR SCREEN AND ENTER "0"')
      READ (6,16) ISCR
   16 FORMAT (1I1)
      RETURN
      END
C
C
      SUBROUTINE SORT
      DIMENSION A(9999),XM(9999),YM(9999),XT(9999),YT(9999)
      REAL A,M,XM,YM,XT,YT,XPOS,XNEG,YPOS,YNEG
      INTEGER I,W,Z
      COMMON/BLOCK1/A,XM,YM,XT,YT,M,XPOS,XNEG,YPOS,YNEG,I
      XPOS=0.0
      XNEG=0.0
      YPOS=0.0
      YNEG=0.0
      Z=I-1
      DO 10 W=1,Z
         IF (ABS(A(W)).GT.ABS(M)) M=A(W)
         IF (XT(W).GT.XPOS) XPOS=XT(W)
         IF (XM(W).GT.XPOS) XPOS=XM(W)
         IF (XT(W).LT.XNEG) XNEG=XT(W)
         IF (XM(W).LT.XNEG) XNEG=XM(W)
         IF (YT(W).GT.YPOS) YPOS=YT(W)
         IF (YM(W).GT.YPOS) YPOS=YM(W)
```

```
      IF (YT(W).LT.YNEG) YNEG=YT(W)
      IF (YM(W).LT.YNEG) YNEG=YM(W)
   10 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE OUTPUT
      REAL D,XM,YM,XT,YT,R,A,M,T5,M1,LOS1,TAL1,MAL1,E,F,G
      REAL DEL,NAV,LOS,IRT,IRM,TSP,TAC,TAL,MSP,MAC,MAL
      INTEGER ISAVE,W,I,NRUN,ILOS,IPRO,II
      DIMENSION D(9999),XM(9999),YM(9999),XT(9999),YT(9999)
      DIMENSION R(9999),A(9999)
      COMMON/BLOCK1/A,XM,YM,XT,YT,M,XPOS,XNEG,YPOS,YNEG,I
      COMMON/BLOCK2/DEL,NAV,LOS,IRT,IRM,TSP,TAC,TAL,MSP,MAC,MAL,D,R,T5,
     *ISAVE,NRUN,ILOS,IPRO
      RATODE(X)=(X/(2*3.1415926535))*360
C
      M1=M/9.80665
      IF (ISAVE.EQ.0) GO TO 250
C
C
      CALL SCREEN
      IF (ILOS.EQ.0) WRITE (6,133)
      IF (IPRO.EQ.0) WRITE (6,134)
      WRITE (6,170)
      DO 240 W=1,I,25
         WRITE (6,107) D(W),XM(W),YM(W),XT(W),YT(W),R(W),A(W)
  240 CONTINUE
      WRITE (6,107) D(I-1),XM(I-1),YM(I-1),XT(I-1),YT(I-1),
     *R(I-1),A(I-1)
  107 FORMAT (1X,F6.3,5(2X,F8.1),2X,F10.3)
      WRITE (6,135) M,M1,T5
  135 FORMAT (T15,'MAXIMUM LATERAL ACCELERATION ON THE MISSILE'/,
     *T15,'IS ',F10.3,' M/SEC/SEC, OR ',F9.2,' G"S.'/,
     *T15,'THE TIME TO INTERCEPT IS ',F6.3,' SECONDS.')
      WRITE (6,143)
  143 FORMAT (/1X,'DO YOU WISH TO INCLUDE THIS OUTPUT IN THE PRINTOUT AN
     *D THE PLOT?'/,1X,'(REMEMBER THAT ALL PLOTS FROM A SINGLE SESSION W
     *ILL OVERLAY EACH OTHER.)'/,1X,'0=YES, 1=NO')
      READ (6,16) ISAVE
   16 FORMAT (I1)
      IF (ISAVE.EQ.1) GO TO 270
C
  250 NRUN=NRUN+1
      IF (NRUN.LT.10) GO TO 255
      WRITE (6,253)
  253 FORMAT (1X,'YOU HAVE REQUESTED MORE THAN 9 ENCOUNTER SITUATIONS BE
```

```
* SAVED AND PLOTTED'/,1X,'WHICH, UNFORTUNATELY, IS NOT POSSIBLE.  ILPA03400
*F YOU WANT ADDITIONAL PLOTS, EXIT THE'/,1X,'PROGRAM, OUTPUT THE LALPA03410
*ST 9 SITUATIONS AND RE-ENTER THE PROGRAM BY TYPING'/,1X,'"LPATH" ALPA03420
*ND ENTERING.')                                                     LPA03430
  GO TO 270                                                         LPA03440
255 WRITE (8,32) NRUN                                               LPA03450
 32 FORMAT (//1X,'$$$$ RUN NUMBER ',I2)                             LPA03460
    WRITE (8,33)                                                    LPA03470
 33 FORMAT (T25,'PROBLEM PARAMETERS')                               LPA03480
    DR=IRT-IRM                                                      LPA03490
    TAL1=RATODE(TAL)                                                LPA03500
    MAL1=RATODE(MAL)                                                LPA03510
    LOS1=RATODE(LOS)                                                LPA03520
    WRITE (8,34) DEL,NAV,LOS1,DR,TSP,TAC,TAL1,MSP,MAL1              LPA03530
 34 FORMAT (1X,T13,'01) TIME INCREMENT',T40,F10.4,' SECONDS'/,      LPA03540
   +T13,'02) NAVIGATION CONSTANT',T39,F10.3/,                       LPA03550
   +T13,'03) LINE-OF-SIGHT ANGLE',T39,F10.3,' DEGREES'/,            LPA03560
   +T13,'04) INITIAL SEPARATION',T39,F10.3,' METERS'/,              LPA03570
   +T13,'05) TARGET SPEED',T39,F10.3,' METERS/SEC'/,                LPA03580
   +T13,'06) TARGET ACCELERATION',T39,F10.3,' M/SEC/SEC'/,          LPA03590
   +T13,'07) TARGET ALPHA',T39,F10.3,' DEGREES'/,                   LPA03600
   +T13,'08) MISSILE SPEED',T39,F10.3,' METERS/SEC'/,               LPA03610
   +T13,'09) MISSILE INITIAL ALPHA',T39,F10.3,' DEGREES')           LPA03620
    IF (ILOS.EQ.0) WRITE (8,133)                                    LPA03630
133 FORMAT (/T12,'LINE OF SIGHT GUIDANCE INTERCEPT TRAJECTORY')     LPA03640
    IF (IPRU.EQ.0) WRITE (8,134)                                    LPA03650
134 FORMAT (/T07,'PROPORTIONAL NAVIGATION GUIDANCE INTERCEPT TRAJECTORLPA03660
   *Y')                                                             LPA03670
    WRITE (8,170)                                                   LPA03680
170 FORMAT (1X,'PROBLEM',T13,'--POSITION COORDINATES (METERS)--',   LPA03690
   *T52,'RANGE',T62,'ACCEL.'/,1X,'TIME(S)',T15,'XM',T25,'YM',T35,   LPA03700
   *'XT',T45,'YT',T51,'(METERS)',T61,'(M/S/S)')                     LPA03710
    II=(I/25)+2                                                     LPA03720
    E=25.                                                           LPA03730
    F=I                                                             LPA03740
    DO 5 J=1,50                                                     LPA03750
       G=F-E                                                        LPA03760
       IF (G.EQ.0.) GO TO 10                                        LPA03770
       E=E+25.                                                      LPA03780
  5 CONTINUE                                                        LPA03790
    GO TO 20                                                        LPA03800
 10 II=II-1                                                         LPA03810
 20 CONTINUE                                                        LPA03820
    WRITE (7,140) NRUN,II                                           LPA03830
140 FORMAT (I2,I4)                                                  LPA03840
    DO 260 W=1,I,25                                                 LPA03850
       WRITE (8,107) D(W),XM(W),YM(W),XT(W),YT(W),R(W),A(W)         LPA03860
       WRITE (7,141) XM(W),YM(W),XT(W),YT(W)                        LPA03870
```

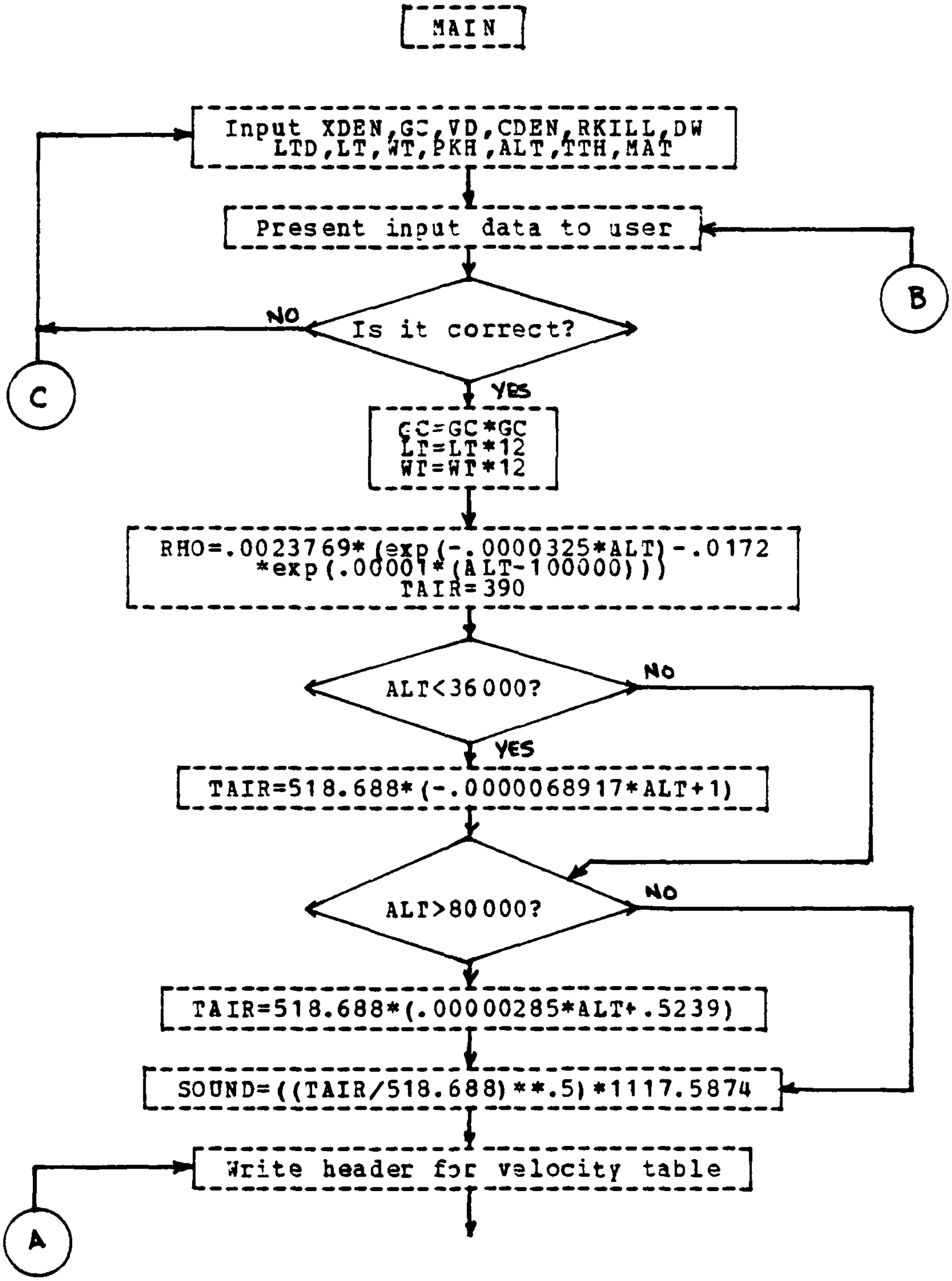
```
260 CONTINUE                                                            LPA03880
    WRITE (8,107) D(I-1),XM(I-1),YM(I-1),XT(I-1),YT(I-1),R(I-1),A(I-1)LPA03890
    WRITE (7,141) XM(I),YM(I),XT(I),YT(I)                              LPA03900
141 FORMAT (4(F10.3))                                                   LPA03910
    WRITE (8,135) M,M1,T5                                               LPA03920
    WRITE (7,141) XPOS,XNEG,YPOS,YNEG                                   LPA03930
    ISAVE=1                                                             LPA03940
270 CONTINUE                                                            LPA03950
    RETURN                                                              LPA03960
    END                                                                 LPA03970
```

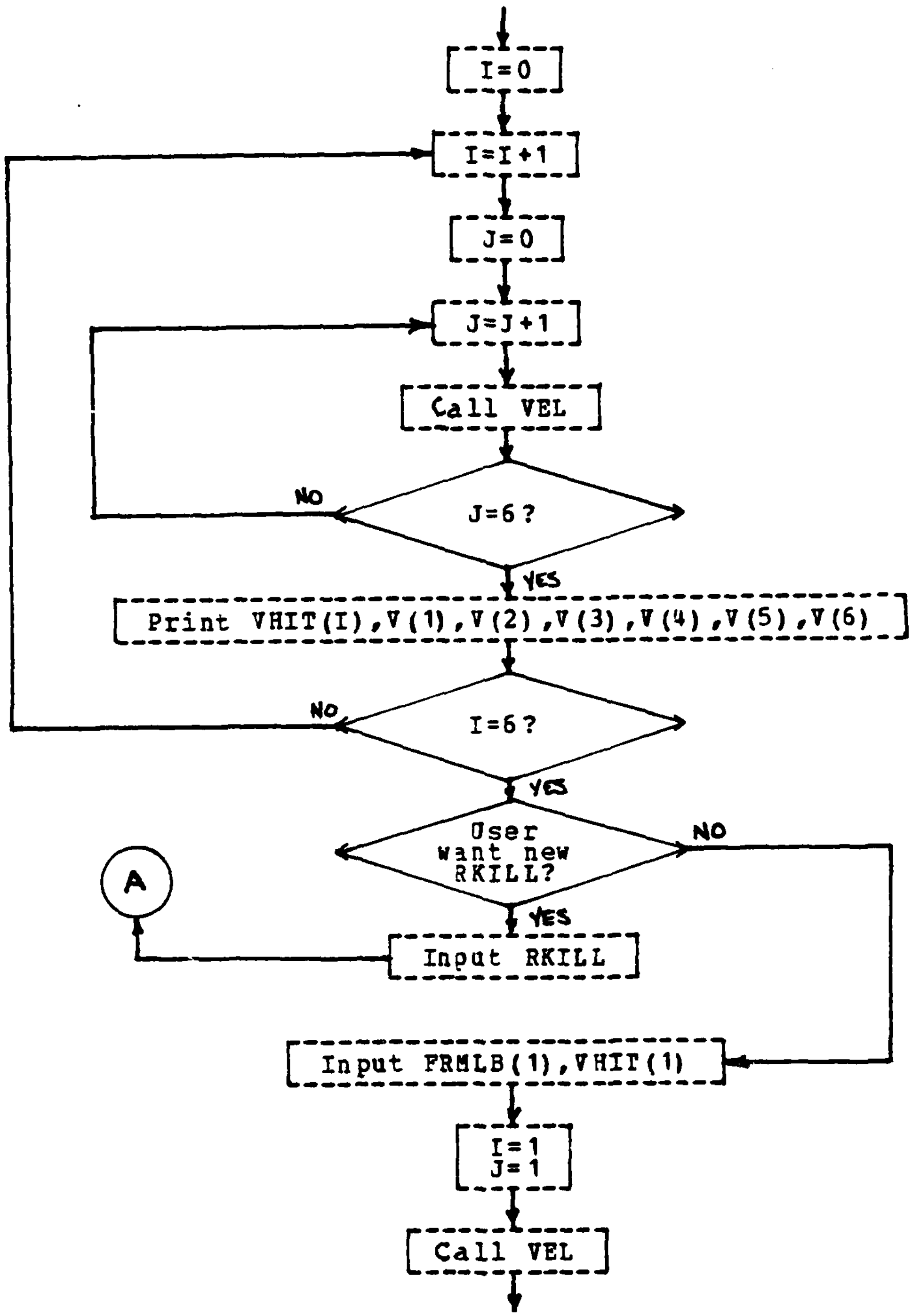
```
//PATHPLOT JOB (1414,0483),'LINDSEY',CLASS=B
// EXEC FRTXCLGP
//FORT.SYSIN DD *
C     LPATH PLOTTING ROUTINE(MAXIMUM OF 9 SIMULTANEOUS PLOTS)
C     READ IN DATA
C
      REAL XM,YM,XT,YT,XPOS,XNEG,YPOS,YNEG,X,Y,XMAX,XMIN,YMAX,YMIN
      REAL XMININ,YMININ,XSPAN,YSPAN,SCALES,XO,YO,SYMB
      INTEGER K,J,N,I,Z
      DIMENSION XM(500,9),YM(500,9),XT(500,9),YT(500,9)
      DIMENSION X(500),Y(500),SYMB(9)
      DIMENSION I(9)
      DATA SYMB/1.,2.,3.,4.,5.,6.,7.,8.,9./
      XMAX=0.0
      XMIN=0.0
      YMAX=0.0
      YMIN=0.0
  100 READ (5,105) NRUN,I(NRUN)
  105 FORMAT (I2,I4)
      IF (NRUN.EQ.99) GO TO 120
      N=NRUN
      DO 110 K=1,500
         READ (5,115) XM(K,N),YM(K,N),XT(K,N),YT(K,N)
         IF (K.EQ.I(N)) GO TO 117
  110 CONTINUE
  115 FORMAT (4(F10.3))
  117 READ (5,115) XPOS,XNEG,YPOS,YNEG
      IF (XPOS.GT.XMAX) XMAX=XPOS
      IF (XNEG.LT.XMIN) XMIN=XNEG
      IF (YPOS.GT.YMAX) YMAX=YPOS
      IF (YNEG.LT.YMIN) YMIN=YNEG
      GO TO 100
  120 CONTINUE
C
C     INITIALIZE PLOTTING
      CALL PLOTS(0,0,0)
C
C     ESTABLISH PLOT "WINDOW"
      XSPAN=XMAX-XMIN
      YSPAN=YMAX-YMIN
      IF ((YSPAN/7.0).GT.(XSPAN/9.0)) GO TO 130
      SCALES=XSPAN/9.0
      YMININ=YMIN/SCALES
      YMAX=(7.0+YMININ)*SCALES
      YSPAN=YMAX-YMIN
      GO TO 135
  130 SCALES=YSPAN/7.0
      XMININ=XMIN/SCALES
```

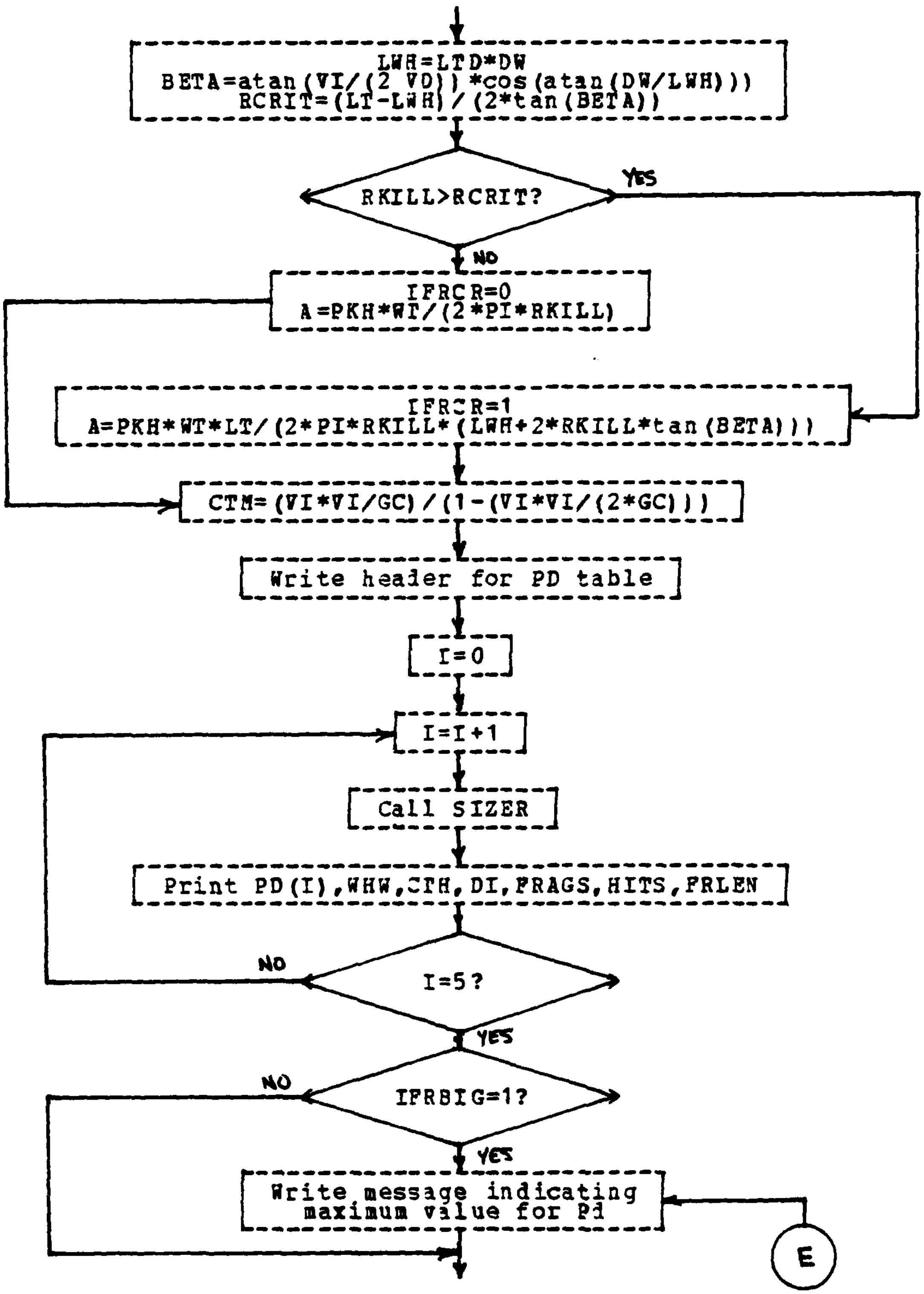
```
      XMAX=(9.0+XMININ)*SCALES                                          LPA00520
      XSPAN=XMAX-XMIN                                                   LPA00530
  135 CONTINUE                                                          LPA00540
C                                                                       LPA00550
C     LOCATE ORIGIN OF AXES                                             LPA00560
      CALL PLOT(2.,2.,-3)                                               LPA00570
C                                                                       LPA00580
C     SCALE X VALUES TO 9.0 INCH AXIS AND Y VALUES TO 7.0 INCH AXIS     LPA00590
C     AND PLOT THEM                                                     LPA00600
      X(1)=XMIN                                                         LPA00610
      X(2)=XMAX                                                         LPA00620
      X(3)=XMAX                                                         LPA00630
      X(4)=0.0                                                          LPA00640
      X(5)=0.0                                                          LPA00650
      Y(1)=YMAX                                                         LPA00660
      Y(2)=YMAX                                                         LPA00670
      Y(3)=YMIN                                                         LPA00680
      Y(4)=0.0                                                          LPA00690
      Y(5)=0.0                                                          LPA00700
      CALL SCALE(X,9.0,3,1)                                             LPA00710
      CALL SCALE(Y,7.0,3,1)                                             LPA00720
      CALL AXIS(0.,0.,'REFERENCE DIRECTION',-19,9.0,0.,XMIN,SCALES)     LPA00730
      CALL AXIS(0.,0.,' ',+1,7.0,90.,YMIN,SCALES)                       LPA00740
C                                                                       LPA00750
      DO 160 K=1,N                                                      LPA00760
      Z=I(K)                                                            LPA00770
C                                                                       LPA00780
C        PLOT TARGET PATH                                               LPA00790
         DO 140 J=1,500                                                 LPA00800
            X(J)=XT(J,K)                                                LPA00810
            Y(J)=YT(J,K)                                                LPA00820
            IF (J.EQ.Z) GO TO 145                                       LPA00830
  140    CONTINUE                                                       LPA00840
  145    X(Z+1)=XMIN                                                    LPA00850
         X(Z+2)=SCALES                                                  LPA00860
         Y(Z+1)=YMIN                                                    LPA00870
         Y(Z+2)=SCALES                                                  LPA00880
         CALL LINE(X,Y,Z,1,+Z,5)                                        LPA00890
C                                                                       LPA00900
C        PLOT MISSILE PATH                                              LPA00910
         DO 150 J=1,500                                                 LPA00920
            X(J)=XM(J,K)                                                LPA00930
            Y(J)=YM(J,K)                                                LPA00940
            IF (J.EQ.Z) GO TO 155                                       LPA00950
  150    CONTINUE                                                       LPA00960
  155    X(Z+1)=XMIN                                                    LPA00970
         X(Z+2)=SCALES                                                  LPA00980
         Y(Z+1)=YMIN                                                    LPA00990
```

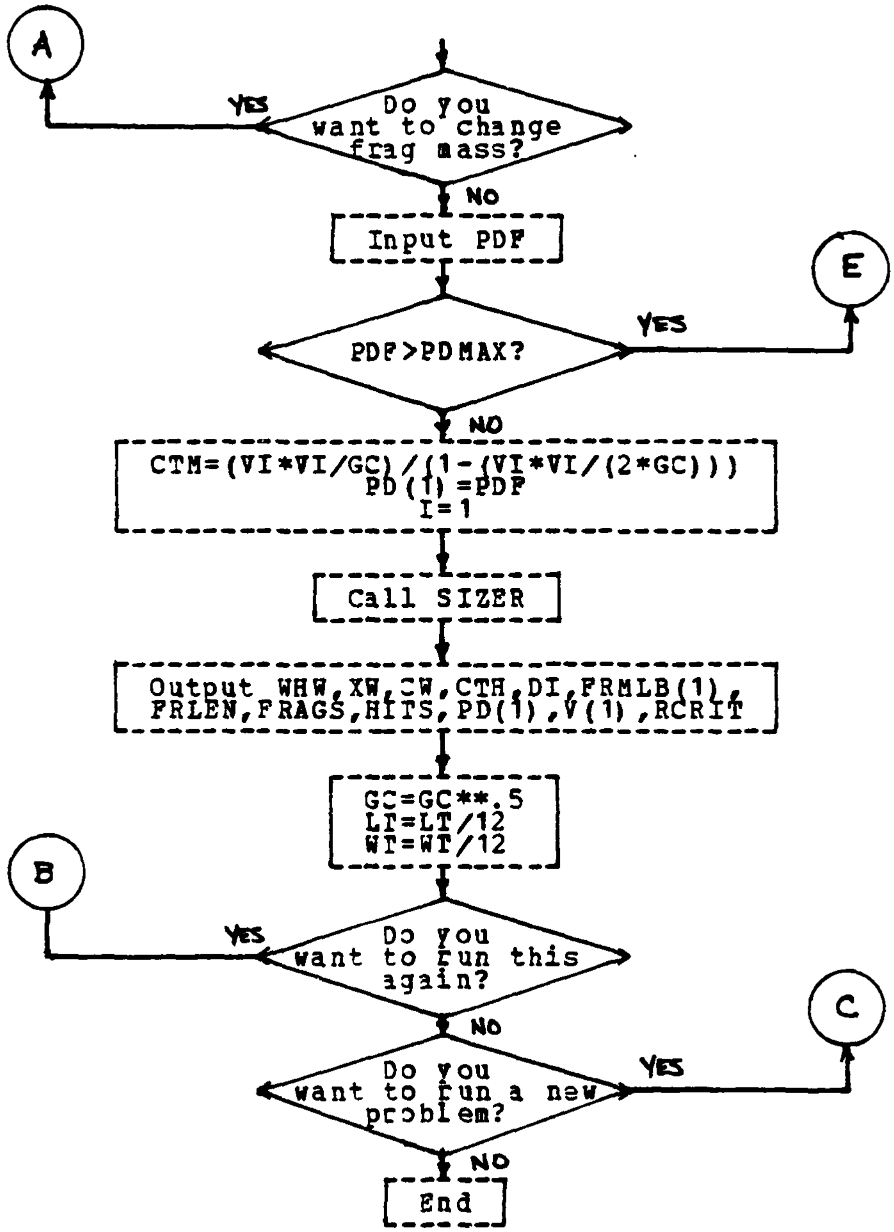
```
      Y(Z+2)=SCALES                                                     LPA01000
      CALL LINE(X,Y,Z,1,+Z,0)                                           LPA01010
      XO=X(Z)/SCALES                                                    LPA01020
      YO=Y(Z)/SCALES                                                    LPA01030
      CALL NUMBER(XO,YO,0.1,SYMB(K),0.0,-1)                             LPA01040
C                                                                       LPA01050
  160 CONTINUE                                                          LPA01060
C                                                                       LPA01070
C     END OF PLOTTING                                                   LPA01080
      CALL PLOT(0.,0.,+999)                                             LPA01090
      STOP                                                              LPA01100
      END                                                               LPA01110
/*                                                                      LPA01120
//GO.PLOTPARM DD *                                                      LPA01130
 &PLOT SCALE=0.5  &END                                                  LPA01140
/*                                                                      LPA01150
//GO.SYSIN DD *                                                         LPA01160
```

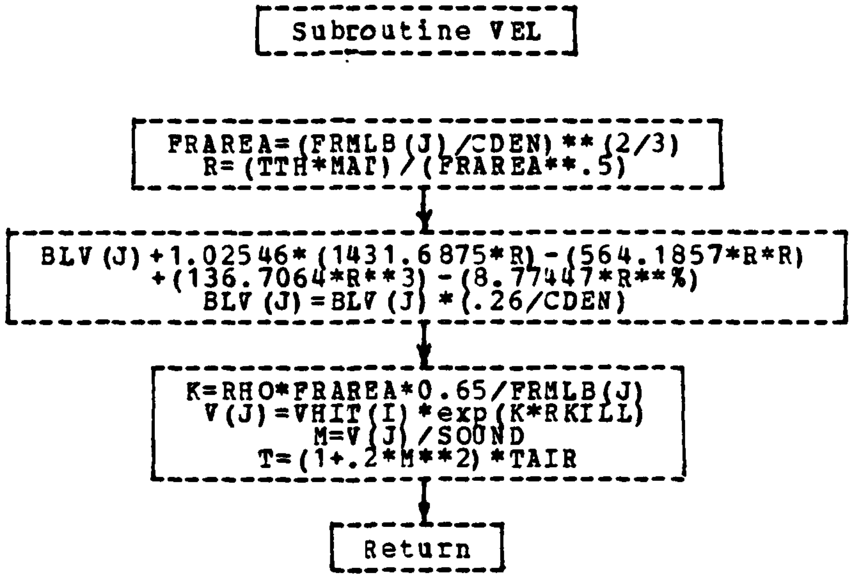
APPENDIX C. WARHEAD DESIGN PROGRAM FLOWCHART

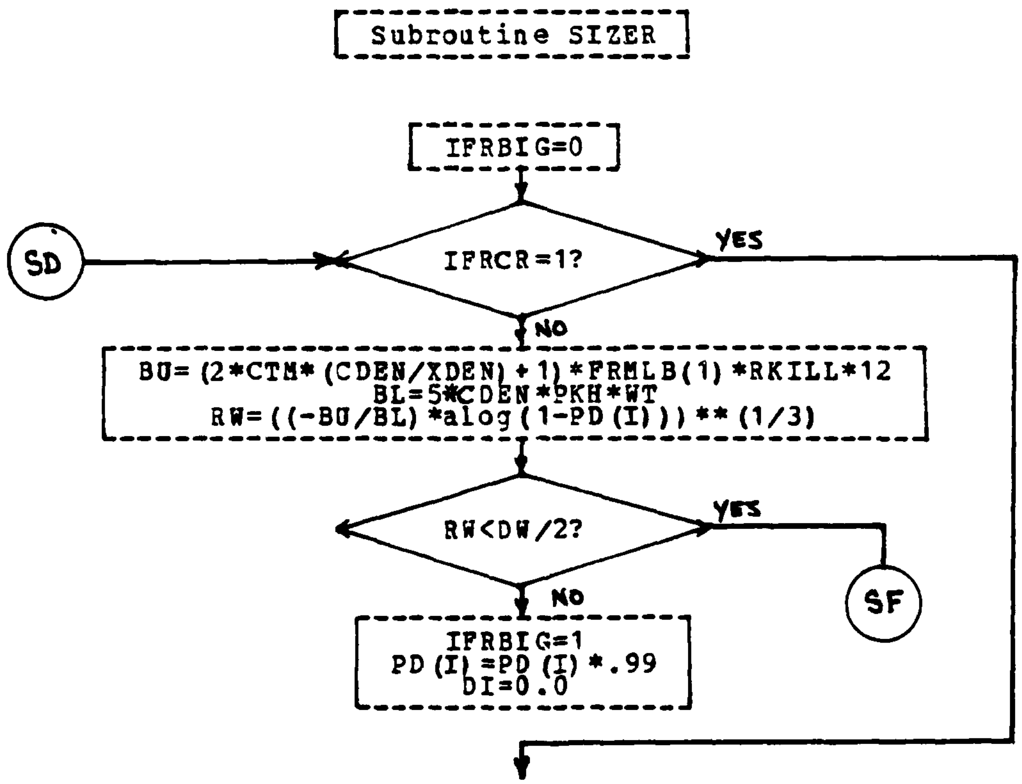


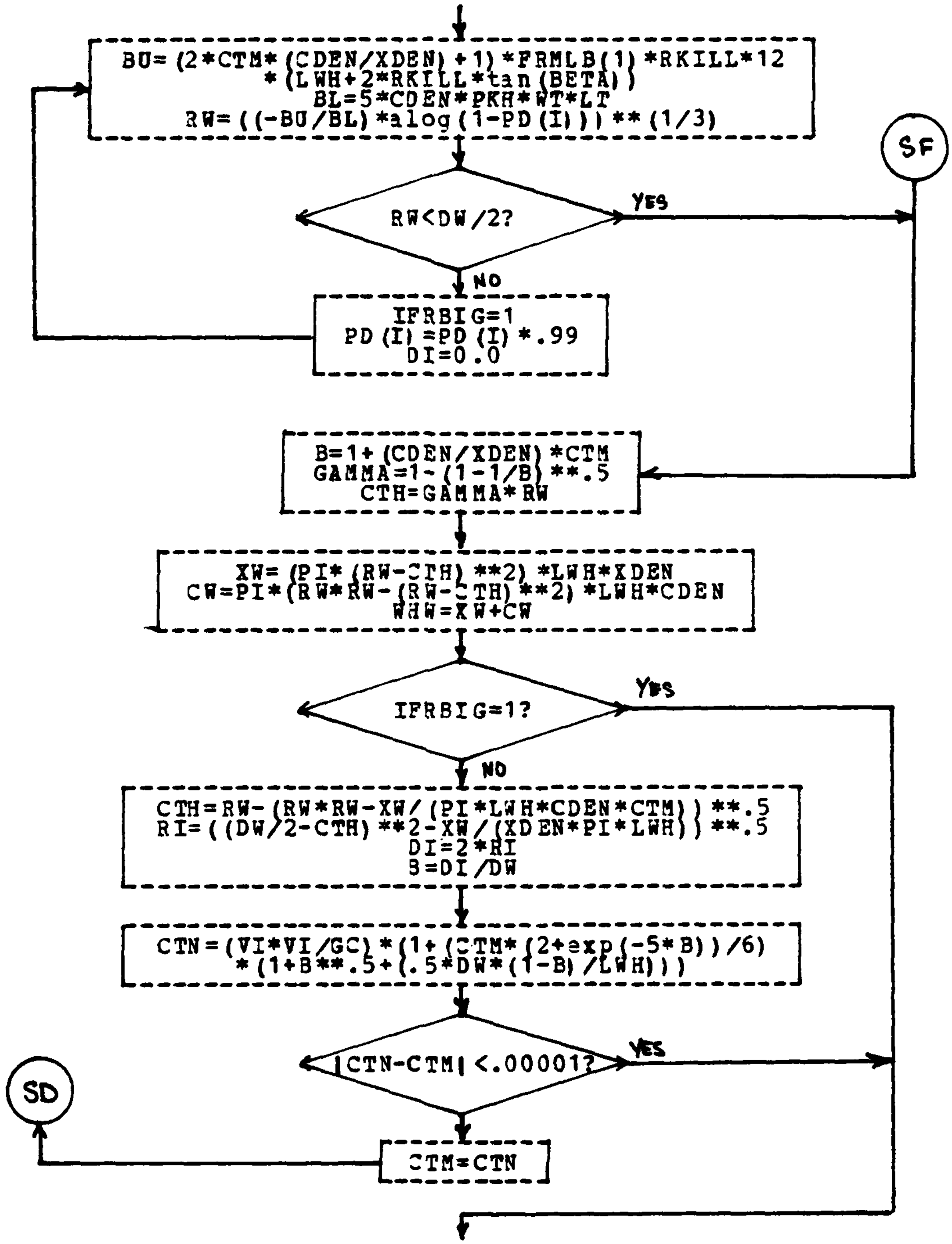

I=0
I=I+1
J=0
J=J+1
Call VEL
J=6?
NO
YES
Print VHIT(I),V(1),V(2),V(3),V(4),V(5),V(6)
I=6?
NO
YES
User want new RKILL?
NO
YES
A
Input RKILL
Input FRMLB(1),VHIT(1)
I=1
J=1
Call VEL


LWH=LTD*DW
BETA=atan(VI/(2 VD))*cos(atan(DW/LWH))
RCRIT=(LT-LWH)/(2*tan(BETA))
RKILL>RCRIT?
YES
NO
IFRCR=0
A=PKH*WT/(2*PI*RKILL)
IFRCR=1
A=PKH*WT*LT/(2*PI*RKILL*(LWH+2*RKILL*tan(BETA)))
CTM=(VI*VI/GC)/(1-(VI*VI/(2*GC)))
Write header for PD table
I=0
I=I+1
Call SIZER
Print PD(I),WHW,CTH,DI,FRAGS,HITS,FRLEN
NO
I=5?
YES
NO
IFRBIG=1?
YES
Write message indicating maximum value for Pd
E


A
YES
Do you want to change frag mass?
NO
Input PDF
E
YES
PDF>PDMAX?
NO
CTM=(VI*VI/GC)/(1-(VI*VI/(2*GC)))
PD(1)=PDF
I=1
Call SIZER
Output WHW,XW,CW,CTH,DI,FRMLB(1),
FRLEN,FRAGS,HITS,PD(1),V(1),RCRIT
GC=GC**.5
LT=LT/12
WT=WT/12
B
YES
Do you want to run this again?
NO
C
YES
Do you want to run a new problem?
NO
End


Subroutine VEL
FRAREA=(FRMLB(J)/CDEN)**(2/3)
R=(TTH*MAT)/(FRAREA**.5)
BLV(J)+1.02546*(1431.6875*R)-(564.1857*R*R)
+(136.7064*R**3)-(8.77447*R**%)
BLV(J)=BLV(J)*(.26/CDEN)
K=RHO*FRAREA*0.65/FRMLB(J)
V(J)=VHIT(I)*exp(K*RKILL)
M=V(J)/SOUND
T=(1+.2*M**2)*TAIR
Return


Subroutine SIZER
IFRBIG=0
SD
IFRCR=1?
YES
NO
BU=(2*CTM*(CDEN/XDEN)+1)*FRMLB(1)*RKILL*12
BL=5*CDEN*PKH*WT
RW=((-BU/BL)*alog(1-PD(I)))**(1/3)
RW<DW/2?
YES
SF
NO
IFRBIG=1
PD(I)=PD(I)*.99
DI=0.0


BU= (2*CTM* (CDEN/XDEN) + 1) *FRMLB(1) *RKILL*12
* (LWH+2*RKILL*tan (BETA))
BL=5*CDEN*PKH*WT*LT
RW= ((-BU/BL) *alog (1-PD (I))) ** (1/3)
SF
RW<DW/2?
YES
NO
IFRBIG=1
PD (I) =PD (I) *.99
DI=0.0
B=1+ (CDEN/XDEN) *CTM
GAMMA=1- (1-1/B) **.5
CTH=GAMMA* RW
XW= (PI* (RW-CTH) **2) *LWH*XDEN
CW=PI* (RW*RW- (RW-CTH) **2) *LWH*CDEN
WHW=XW+CW
IFRBIG=1?
YES
NO
CTH=RW- (RW*RW-XW/ (PI*LWH*CDEN*CTM)) **.5
RI= ((DW/2-CTH) **2-XW/ (XDEN*PI*LWH)) **.5
DI=2*RI
B=DI/DW
CTN= (VI*VI/GC) * (1+ (CTM* (2+exp (-5*B)) /6)
* (1+B**.5+ (.5*DW* (1-B) /LWH)))
|CTN-CTM| <.00001?
YES
SD
CTM=CTN

```
FRAGS=(CDEN*5*PI/FRMLB(1))*(2*RW*RW*CTH-RW*CTH*CTH)
FRLEN=(FRMLB(1)/(CTH*CDEN))**.5
HITS=FRAGS*A
```

```
Return
```

APPENDIX D. WARHEAD DESIGN PROGRAM LISTING

This program has two major sections; the executive routine and the FORTRAN IV computational program. The executive routine establishes the required file definitions and initiates operation of the computational program.

The computational program, LBOMB FORTRAN, consists of four subprogram divisions. The MAIN program accepts the input data, calculates the atmospheric characteristics, and formats and displays the output to the user and sends it to the printer file. Subroutine VEL calculates the initial velocity required to propel a given mass a specified distance through the atmosphere with a particular residual velocity remaining. It also determines the ballistic limit velocities for the situation. Subroutine SIZER sizes the warhead for a given Pd value. It also produces the charge-to-mass ratio, the number of fragments, the fragment size, and the average number of hits received by the target. Subroutine SCREEN prompts the user to clear the terminal screen for proper positioning of the displayed data.

```
FILEDEF 08 DISK LBOMB OUTPUT A0 (PERM
&BEGTYPE

YOU WILL HAVE THE OPTION TO OBTAIN A HARDCOPY PRINTOUT
OF THE LAST SOLUTION THAT YOU SOLVE.  YOU MAY RERUN THE
PROGRAM AS OFTEN AS YOU WISH BUT THE LAST RUN IS THE RUN
THAT IS RECORDED.

&END
LOAD LBOMB
START
&BEGTYPE

TO OBTAIN A HARDCOPY PRINTOUT OF THE RESULTS, TYPE
"PRINT LBOMB OUTPUT" AND ENTER.  THE OUTPUT WILL BE
PRINTED ON THE VM PRINTER IN ROOM 140 AND WILL BE
IDENTIFIED BY YOUR USER NUMBER AND LABEL NAME.  IT
USUALLY REQUIRES SEVERAL MINUTES TO OBTAIN THE
PRINTOUT.

&END
```

```
C LBOMB
C WARHEAD SIZING PROGRAM
C LT. M.D.SULLIVAN,USN                                   3 JUNE 1981
C PROGRAM COORDINATOR:  PROFESSOR GERALD LINDSEY,  DEPT OF AERONAUTICS
C
      REAL FRMLB,VHIT,PD,BLV,CDEN,TTH,MAT,RHO,RKILL,SOUND,TAIR,V,T,XDEN
      REAL CTM,PKH,WT,DW,CTH,CTN,WHW,FRAGS,FRLEN,HITS,A,GC,VD,LTD,LT
      REAL ALT,LWH,BETA,VI,RCRIT,DI,PDF
      INTEGER I,J,K,IWANT,IFRCR,FRMGR,IFRBIG
      DIMENSION FRMLB(6),VHIT(6),PD(5),BLV(6),V(6)
      DIMENSION FRMGR(6)
      DATA FRMLB/.0071428,.0142857,.0214285,.0285714,.0357142,.0428571/
      DATA VHIT/1000.,2000.,3000.,4000.,5000.,6000./
      DATA PD/.999,.99,.98,.95,.90/
      DATA FRMGR/50,100,150,200,250,300/
      PI=3.1415926
C
  100 FORMAT (F15.5)
  110 FORMAT (I2)
  120 FORMAT (I1)
 1000 CALL SCREEN
      IFBACK=0
      WRITE (6,1010)
 1010 FORMAT (1X,'INPUT THE FOLLOWING DATA AS DECIMAL NUMBERS ONLY:'/)
 6001 WRITE (6,1020)
 1020 FORMAT (1X,' 1) INPUT EXPLOSIVE DENSITY (LB/CU.IN):')
      READ (5,100) XDEN
      IF (IFBACK.EQ.1) GO TO 1500
 6002 WRITE (6,1030)
 1030 FORMAT (1X,' 2) INPUT EXPLOSIVE GURNEY CONSTANT (FT/SEC):')
      READ (5,100) GC
      IF (IFBACK.EQ.1) GO TO 1500
 6003 WRITE (6,1040)
 1040 FORMAT (1X,' 3) INPUT EXPLOSIVE DETONATION VELOCITY (FT/SEC):')
      READ (5,100) VD
      IF (IFBACK.EQ.1) GO TO 1500
 6004 WRITE (6,1050)
 1050 FORMAT (1X,' 4) INPUT CASE MATERIAL DENSITY (LB/CU.IN):')
      READ (5,100) CDEN
      IF (IFBACK.EQ.1) GO TO 1500
 6005 WRITE (6,1060)
 1060 FORMAT (1X,' 5) INPUT DESIRED KILL RADIUS (FT):')
      READ (5,100) RKILL
      IF (IFBACK.EQ.1) GO TO 1500
 6006 WRITE (6,1070)
```

```
1070 FORMAT (1X,' 6) INPUT WARHEAD DIAMETER (IN):')
     READ (5,100) DW
     IF (IFBACK.EQ.1) GO TO 1500
6007 WRITE (6,1080)
1080 FORMAT (1X,' 7) INPUT WARHEAD LENGTH-TO-DIAMETER RATIO:')
     READ (5,100) LTD
     IF (IFBACK.EQ.1) GO TO 1500
6008 WRITE (6,1090)
1090 FORMAT (1X,' 8) INPUT TARGET LENGTH (FT):')
     READ (5,100) LT
     IF (IFBACK.EQ.1) GO TO 1500
6009 WRITE (6,1100)
1100 FORMAT (1X,' 9) INPUT TARGET WIDTH (FT):')
     READ (5,100) WT
     IF (IFBACK.EQ.1) GO TO 1500
6010 WRITE (6,1110)
1110 FORMAT (1X,'10) INPUT TARGET VULNERABILITY, P(K/H):')
     READ (5,100) PKH
     IF (IFBACK.EQ.1) GO TO 1500
6011 WRITE (6,1120)
1120 FORMAT (1X,'11) INPUT TARGET ALTITUDE (FT):')
     READ (5,100) ALT
     IF (IFBACK.EQ.1) GO TO 1500
6012 WRITE (6,1130)
1130 FORMAT (1X,'12) INPUT TARGET SKIN THICKNESS (IN):')
     READ (5,100) TTH
     IF (IFBACK.EQ.1) GO TO 1500
6013 WRITE (6,1140)
1140 FORMAT (1X,'13) INPUT SKIN MATERIAL CODE:  1.0=ALUMINUM'/,
    *32X,'2.0=FIBERGLASS/KEVLAR'/,
    *32X,'3.0=STEEL')
     READ (5,100) MAT
     IF (IFBACK.EQ.1) GO TO 1500
C
1500 CALL SCREEN
     REWIND 08
     IFBACK=1
     WRITE (6,1510) XDEN,GC,VD,CDEN,RKILL,DW,LTD
     WRITE (8,1510) XDEN,GC,VD,CDEN,RKILL,DW,LTD
1510 FORMAT (1X,'  THE FOLLOWING IS A SUMMARY OF THE INPUT DATA:'/,
    *1X,'01) EXPLOSIVE DENSITY',T40,F10.5,' LB/CU.IN'/,
    *1X,'02) EXPLOSIVE GURNEY CONSTANT',T40,F10.2,' FT/SEC'/,
    *1X,'03) EXPLOSIVE DETONATION VELOCITY',T40,F10.2,' FT/SEC'/,
    *1X,'04) CASE MATERIAL DENSITY',T40,F10.4,' LB/CU.IN'/,
    *1X,'05) DESIRED KILL RADIUS',T40,F10.1,' FEET'/,
    *1X,'06) WARHEAD DIAMETER',T40,F10.2,' INCHES'/,
    *1X,'07) WARHEAD LENGTH-TO-DIAMETER RATIO',T40,F10.2)
     WRITE (6,1520) LT,WT,PKH,ALT,TTH
```

```
      WRITE (8,1520) LT,WT,PKH,ALT,TTH
 1520 FORMAT (1X,'08) TARGET LENGTH',T40,F10.2,' FEET'/,
     *1X,'09) TARGET WIDTH',T40,F10.2,' FEET'/,
     *1X,'10) TARGET VULNERABILITY, P(K/H)',T40,F10.3/,
     *1X,'11) TARGET ALTITUDE',T40,F10.0,' FEET'/,
     *1X,'12) TARGET SKIN THICKNESS',T40,F10.3,' INCHES')
      IF (MAT.EQ.1.) WRITE (6,1530)
      IF (MAT.EQ.1.) WRITE (8,1530)
 1530 FORMAT (1X,'13) TARGET SKIN MATERIAL',T40,'  ALUMINUM')
      IF (MAT.EQ.2.) WRITE (6,1540)
      IF (MAT.EQ.2.) WRITE (8,1540)
 1540 FORMAT (1X,'13) TARGET SKIN MATERIAL',T40,'FIBERGLASS')
      IF (MAT.EQ.3.) WRITE (6,1550)
      IF (MAT.EQ.3.) WRITE (8,1550)
 1550 FORMAT (1X,'13) TARGET SKIN MATERIAL',T40,'     STEEL')
      WRITE (6,1560)
 1560 FORMAT (/1X,'IS THIS INPUT CORRECT?  IF YES, ENTER "00",'/,
     *1X,'IF NO, ENTER THE TWO-DIGIT NUMBER OF THE WRONG ENTRY.')
      READ (5,110) IGOTO
      IF (IGOTO.EQ.00) GO TO 1600
      GO TO (6001,6002,6003,6004,6005,6006,6007,6008,6009,6010,6011,6012
     *,6013), IGOTO
C
 1600 GC=GC*GC
      LT=LT*12.
      WT=WT*12.
C
      RHO=.0023769*(EXP(-.0000325*ALT)-.0172*EXP(.00001*(ALT-100000.)))
      TAIR=390.
      IF (ALT.LT.36000.) TAIR=518.688*(-.0000068917*ALT+1.)
      IF (ALT.GT.80000.) TAIR=518.688*(.00000285*ALT+.5239)
      SOUND=((TAIR/518.688)**.5)*1117.5874
C
 1990 CALL SCREEN
      VHIT(1)=1000.
      FRMLB(1)=.0071428
      WRITE (6,2000) RKILL
      WRITE (8,2000) RKILL
 2000 FORMAT (/1X,'INITIAL VELOCITY TABLE FOR',F6.1,' FT KILL RADIUS'/,
     *2X,'IMPACT',25X,'FRAGMENT MASSES'/,
     *1X,'VELOCITY',T14,'50 GR.',T23,'100 GR.',T33,'150 GR.',T43,'200 GR
     *.',T53,'250 GR.',T63,'300 GR.')
      DO 2030 I=1,6
         DO 2010 J=1,6
            CALL VEL (FRMLB,VHIT,BLV,V,CDEN,TTH,MAT,RHO,RKILL,SOUND,
     *      TAIR,T,I,J)
 2010    CONTINUE
         WRITE (6,2020) VHIT(I),(V(K),K=1,6)
```

```
      WRITE (8,2020) VHIT(I),(V(K),K=1,6)
2020     FORMAT (3X,F5.0,6(4X,F6.0))
2030 CONTINUE
     WRITE (6,2035) (BLV(I),I=1,6)
     WRITE (8,2035) (BLV(I),I=1,6)
2035 FORMAT (1X,'BALLISTIC'/,1X,'LIMIT',7X,F5.0,5(5X,F5.0))
     WRITE (6,2045)
2045 FORMAT (/1X,'DO YOU WANT A NEW KILL RADIUS? (1=YES,0=NO)')
     READ (5,120) IWANT
     IF (IWANT.EQ.0) GO TO 2049
     WRITE (6,2046)
2046 FORMAT (1X,'INPUT NEW KILL RADIUS (FEET, DECIMAL):')
     READ (5,100) RKILL
     GO TO 1990
2049 WRITE (6,2050)
2050 FORMAT (1X,'INPUT DESIRED FRAGMENT MASS (GRAINS, DECIMAL):')
     READ (5,100) FRMLB(1)
     WRITE (8,2051) FRMLB(1)
2051 FORMAT (/1X,'FRAGMENT MASS........',F6.1,' GRAINS')
     FRMLB(1)=FRMLB(1)*.000142856973
     WRITE (6,2060)
2060 FORMAT (1X,'INPUT DESIRED IMPACT VELOCITY (FT/SEC, DECIMAL):')
     READ (5,100) VHIT(1)
     WRITE (8,2061) VHIT(1)
2061 FORMAT (1X,'IMPACT VELOCITY......',F6.0,' FT/SEC'/)
     CALL SCREEN
     I=1
     J=1
     CALL VEL (FRMLB,VHIT,BLV,V,CDEN,TTH,MAT,RHO,RKILL,SOUND,TAIR,T,I,J
    *)
     VI=V(J)
C
2100 CONTINUE
     LWH=LTD*DW
     BETA=ATAN(VI/(2.*VD)*COS(ATAN(DW/LWH)))
     RCRIT=(LT-LWH)/(2.*TAN(BETA))
     IF (RKILL.GT.RCRIT) GO TO 2105
     IFRCR=0
     A=PKH*WT/(2.*PI*RKILL)
     GO TO 2110
2105 IFRCR=1
     A=PKH*WT*LT/(2.*PI*RKILL*(LWH+2.*RKILL*TAN(BETA)))
C
2110 CONTINUE
     CTM=(VI*VI/GC)/(1.-(VI*VI/(2.*GC)))
2115 CONTINUE
     WRITE (6,2120)
     WRITE (8,2120)
```

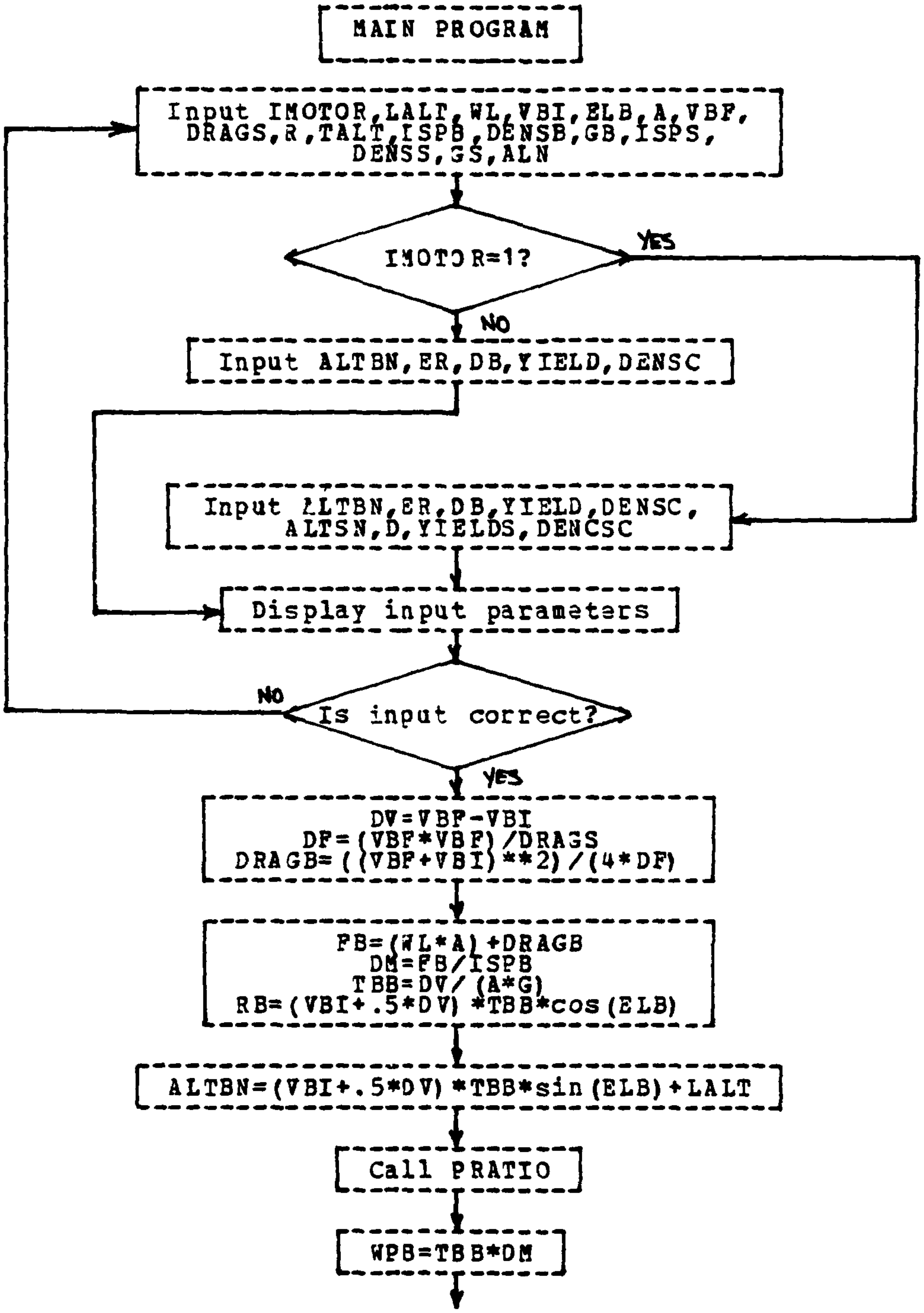
```
2120 FORMAT (T13,'WARHEAD',T25,'CASE',T36,'CORE',T43,'---------FRAGMENT
    *S---------'/,
    *6X,'PD',T14,'WEIGHT',T23,'THICKNESS',T34,'DIAMETER',T44,
    *'NUMBER',T53,'ON TARGET',T64,'LENGTH')
     DO 2200 I=1,5
        CALL SIZER (FRMLB,PD,CDEN,XDEN,CTM,RKILL,PKH,WT,DW,CTH,GC,WHW,
    *   LT,VI,DI,LWH,BETA,FRAGS,FRLEN,HITS,A,XW,CW,IFRBIG,IFRCR,I)
        WRITE (6,2160) PD(I),WHW,CTH,DI,FRAGS,HITS,FRLEN
        WRITE (8,2160) PD(I),WHW,CTH,DI,FRAGS,HITS,FRLEN
2200 CONTINUE
2160 FORMAT (1X,F8.3,6(2X,F8.2))
2205 IF (PD(1).LT..999) WRITE (6,2210)
     IF (PD(1).LT..999) WRITE (8,2210)
2210 FORMAT (1X,'THE FIRST PD TERM IS THE HIGHEST ATTAINABLE WITH THE G
    *IVEN'/,1X,'KILL RADIUS AND FRAGMENT SIZE.')
2215 WRITE (6,2220)
2220 FORMAT (1X,'DO YOU WANT TO CHANGE YOUR FRAGMENT SIZE? (1=YES,0=NO)
    *')
     READ (5,120) IWANT
     IF (IWANT.EQ.1) GO TO 1990
2225 WRITE (6,2230)
2230 FORMAT (1X,'INPUT DESIRED PD (DECIMAL):')
     READ (5,100) PDF
     IF (PDF.LT.PD(1)) GO TO 2228
     WRITE (6,2229)
2229 FORMAT (1X,'TOO LARGE   PD CANNOT BE GREATER THAN THE FIRST ONE ON
    * THE LIST ABOVE '/,1X,'TRY AGAIN')
     GO TO 2225
2228 CALL SCREEN
     WRITE (8,2231) PDF
2231 FORMAT (/1X,'KILL PROBABILITY.....',F6.3/)
     CTM=(VI*VI/GC)/(1.-(VI*VI/(2.*GC)))
     PD(1)=PDF
     I=1
     CALL SIZER (FRMLB,PD,CDEN,XDEN,CTM,RKILL,PKH,WT,DW,CTH,GC,WHW,
    *LT,VI,DI,LWH,BETA,FRAGS,FRLEN,HITS,A,XW,CW,IFRBIG,IFRCR,I)
     FRMLB(1)=FRMLB(1)/.000142856973
     WRITE (6,2300)
     WRITE (8,2300)
2300 FORMAT (1X,'WARHEAD DESCRIPTION-----------------------------------')
     WRITE (6,2310) WHW,XW,CW,CTH,DI,FRMLB(1)
     WRITE (8,2310) WHW,XW,CW,CTH,DI,FRMLB(1)
2310 FORMAT (/4X,'WARHEAD WEIGHT',T35,F8.2,' POUNDS'/,
    *4X,'EXPLOSIVE WEIGHT',T35,F8.2,' POUNDS'/,
    *4X,'CASE WEIGHT',T35,F8.2,' POUNDS'/,
    *4X,'CASE THICKNESS',T35,F8.4,' INCHES'/,
    *4X,'CORE DIAMETER',T35,F8.2,' INCHES'/,
    *4X,'FRAGMENT WEIGHT',T35,F8.2,' GRAINS')
```

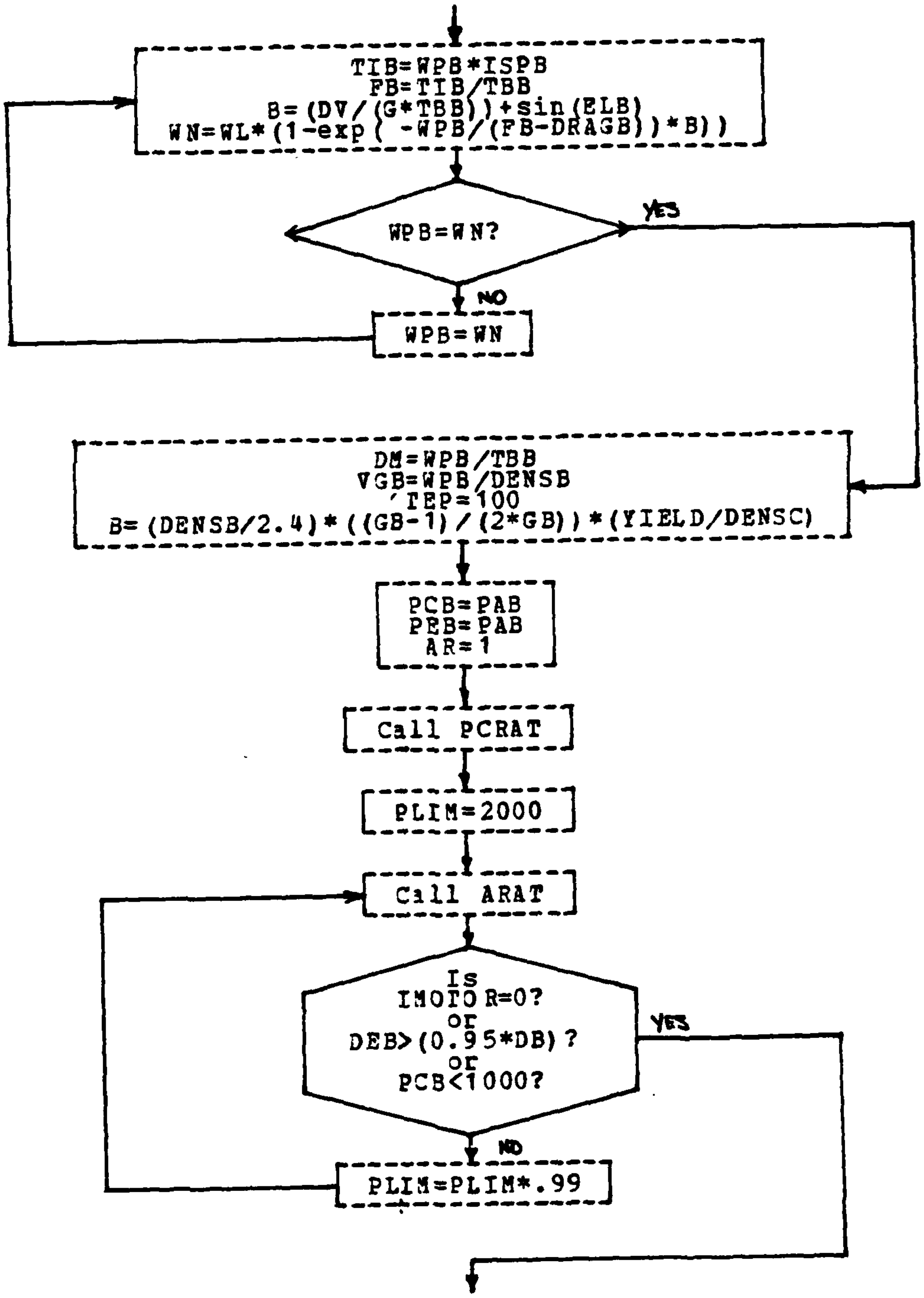
```
      WRITE (6,2320) FRLEN,FRLEN,CTH,FRAGS,HITS,PD(1),V(1),RCRIT
      WRITE (8,2320) FRLEN,FRLEN,CTH,FRAGS,HITS,PD(1),V(1),RCRIT
 2320 FORMAT (4X,'FRAGMENT DIMENSIONS',T26,F5.3,' X ',F5.3,' X ',F5.3,
     *' INCHES'/,4X,'NUMBER OF FRAGMENTS',T35,F8.0/,
     *4X,'NUMBER OF HITS ON TARGET',T35,F8.0/,
     *4X,'PROBABILITY OF KILL (PD)',T35,F8.3/,
     *4X,'INITIAL FRAGMENT VELOCITY',T35,F8.1,' FT/SEC'/,
     *4X,'CRITICAL MISS DISTANCE',T35,F8.1,' FEET'/)
C
      GC=GC**.5
      LT=LT/12.
      WT=WT/12.
      FRMLB(1)=.0071428
      VHIT(1)=1000.
      PD(1)=.999
C
      WRITE (6,2400)
 2400 FORMAT (1X,'DO YOU WANT TO RUN THIS PROBLEM AGAIN? (1=YES,0=NO)')
      READ (5,120) IWANT
      IF (IWANT.EQ.1) GO TO 1500
      WRITE (6,2410)
 2410 FORMAT (1X,'DO YOU WANT TO RUN A NEW PROBLEM? (1=YES,0=NO)')
      READ (5,120) IWANT
      IF (IWANT.EQ.1) GO TO 1000
      CALL SCREEN
      RETURN
      END
C
C
C
C
      SUBROUTINE VEL (FRMLB,VHIT,BLV,V,CDEN,TTH,MAT,RHO,RKILL,SOUND,
     *TAIR,T,I,J)
      REAL FRAREA,FRMLB,CDEN,R,TTH,MAT,BLV,K,RHO,V,VHIT,RKILL,M,SOUND
      REAL T,TAIR
      DIMENSION FRMLB(6),BLV(6),V(6),VHIT(6)
C
      FRAREA=(FRMLB(J)/CDEN)**(2./3.)
      R=(TTH*MAT)/(FRAREA**.5)
      BLV(J)=1.02546+(1431.6875*R)-(564.1857*R*R)+(136.7064*R**3.)-(8.77
     *447*R**4.)
      BLV(J)=BLV(J)*(.26/CDEN)
      K=RHO*FRAREA*0.65/FRMLB(J)
      V(J)=VHIT(I)*EXP(K*RKILL)
      M=V(J)/SOUND
      T=(1.+.2*M**2.)*TAIR
      RETURN
      END
```

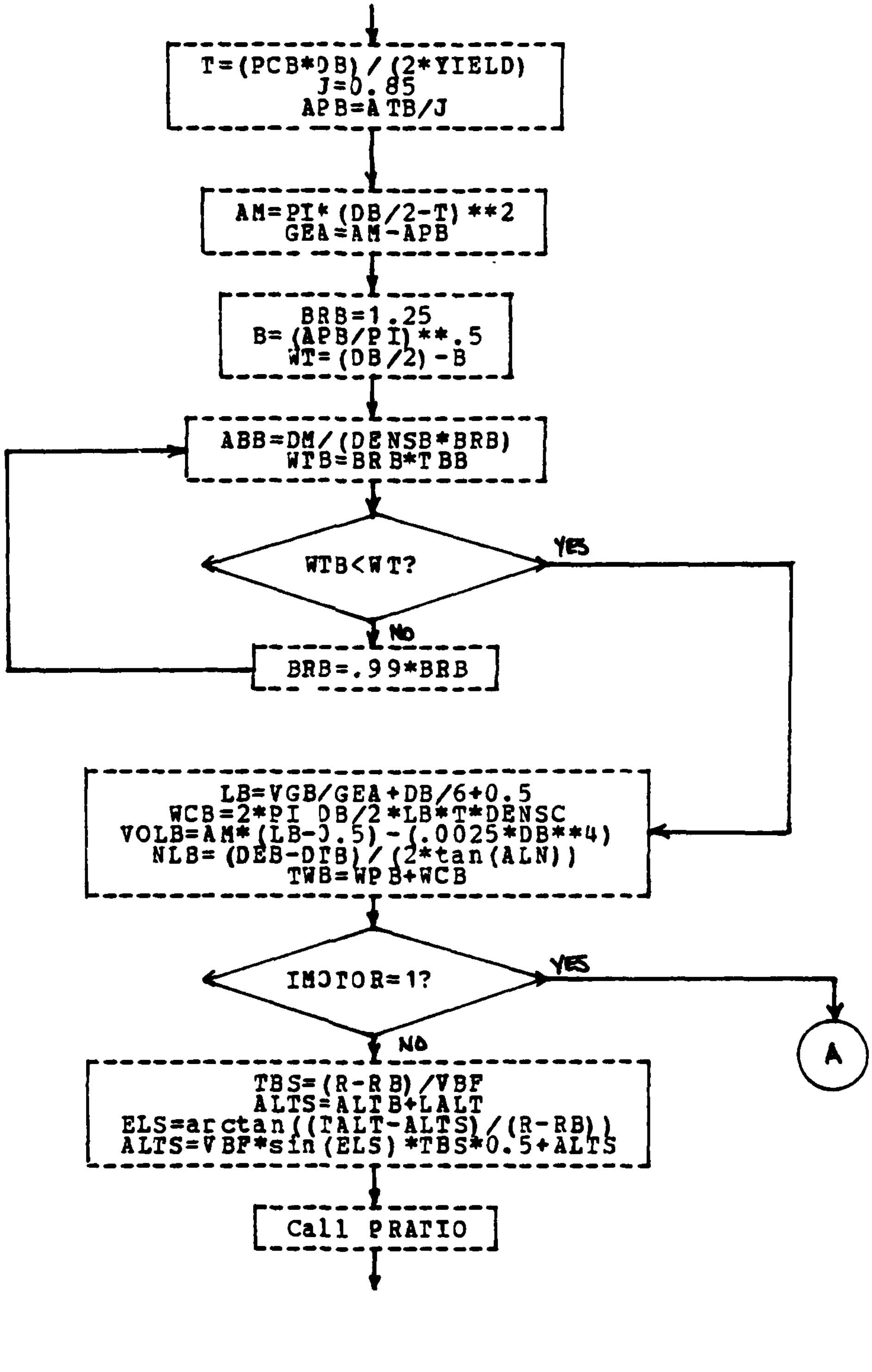
```
C                                                                        LB002920
C                                                                        LB002930
C                                                                        LB002940
C                                                                        LB002950
      SUBROUTINE SIZER (FRMLB,PD,CDEN,XDEN,CTM,RKILL,PKH,WT,DW,CTH,      LB002960
     *GC,WHW,LT,VI,DI,LWH,BETA,FRAGS,FRLEN,HITS,A,XW,CW,IFRBIG,IFRCR,I) LB002970
      REAL PI,BU,BL,CDEN,XDEN,CTM,RKILL,FRMLB,PKH,WT,GAMMA,B,LT          LB002980
      REAL RW,PD,DW,DI,CTH,XW,CW,LWH,WHW,RI,CTN,VI,GC,FRAGS,FRLEN,HITS   LB002990
      INTEGER IFRBIG,IFRCR,I                                             LB003000
      DIMENSION FRMLB(6),PD(5)                                           LB003010
C                                                                        LB003020
      PI=3.1415926                                                       LB003030
      IFRBIG=0                                                           LB003040
   10 IF (IFRCR.EQ.1) GO TO 25                                           LB003050
      BU=(2.*CTM*(CDEN/XDEN)+1.)*FRMLB(1)*RKILL*12.                      LB003060
      BL=5.*CDEN*PKH*WT                                                  LB003070
   20 RW=((-BU/BL)*ALOG(1.-PD(I)))**(1./3.)                              LB003080
      IF (RW.LT.DW/2.) GO TO 30                                          LB003090
      IFRBIG=1                                                           LB003100
      PD(I)=PD(I)*.99                                                    LB003110
      DI=0.0                                                             LB003120
      GO TO 20                                                           LB003130
   25 BU=(2.*CTM*(CDEN/XDEN)+1.)*FRMLB(1)*RKILL*12.*(LWH+2.*RKILL*TAN(BELB003140
     *TA))                                                               LB003150
      BL=5.*CDEN*PKH*WT*LT                                               LB003160
   26 RW=((-BU/BL)*ALOG(1.-PD(I)))**(1./3.)                              LB003170
      IF (RW.LT.DW/2.) GO TO 30                                          LB003180
      IFRBIG=1                                                           LB003190
      PD(I)=PD(I)*.99                                                    LB003200
      DI=0.0                                                             LB003210
      GO TO 26                                                           LB003220
   30 CONTINUE                                                           LB003230
      B=1.+(CDEN/XDEN)*CTM                                               LB003240
      GAMMA=1.-(1.-1./B)**.5                                             LB003250
      CTH=GAMMA*RW                                                       LB003260
      XW=(PI*(RW-CTH)**2.)*LWH*XDEN                                      LB003270
      CW=PI*(RW*RW-(RW-CTH)**2.)*LWH*CDEN                                LB003280
      WHW=XW+CW                                                          LB003290
      IF (IFRBIG.EQ.1) GO TO 50                                          LB003300
      CTH=RW-(RW*RW-XW/(PI*LWH*CDEN*CTM))**.5                            LB003310
      RI=((DW/2.-CTH)**2.-XW/(XDEN*PI*LWH))**.5                          LB003320
      DI=2*RI                                                            LB003330
      B=DI/DW                                                            LB003340
   40 CTN=(VI*VI/GC)*(1.+(CTM*(2.+EXP(-5.*B))/6.)*(1.+B**.5+(.5*DW*(1.-BLB003350
     *)/LWH)))                                                           LB003360
      IF (ABS(CTN-CTM).LT..00001) GO TO 50                               LB003370
      CTM=CTN                                                            LB003380
      GO TO 10                                                           LB003390
```

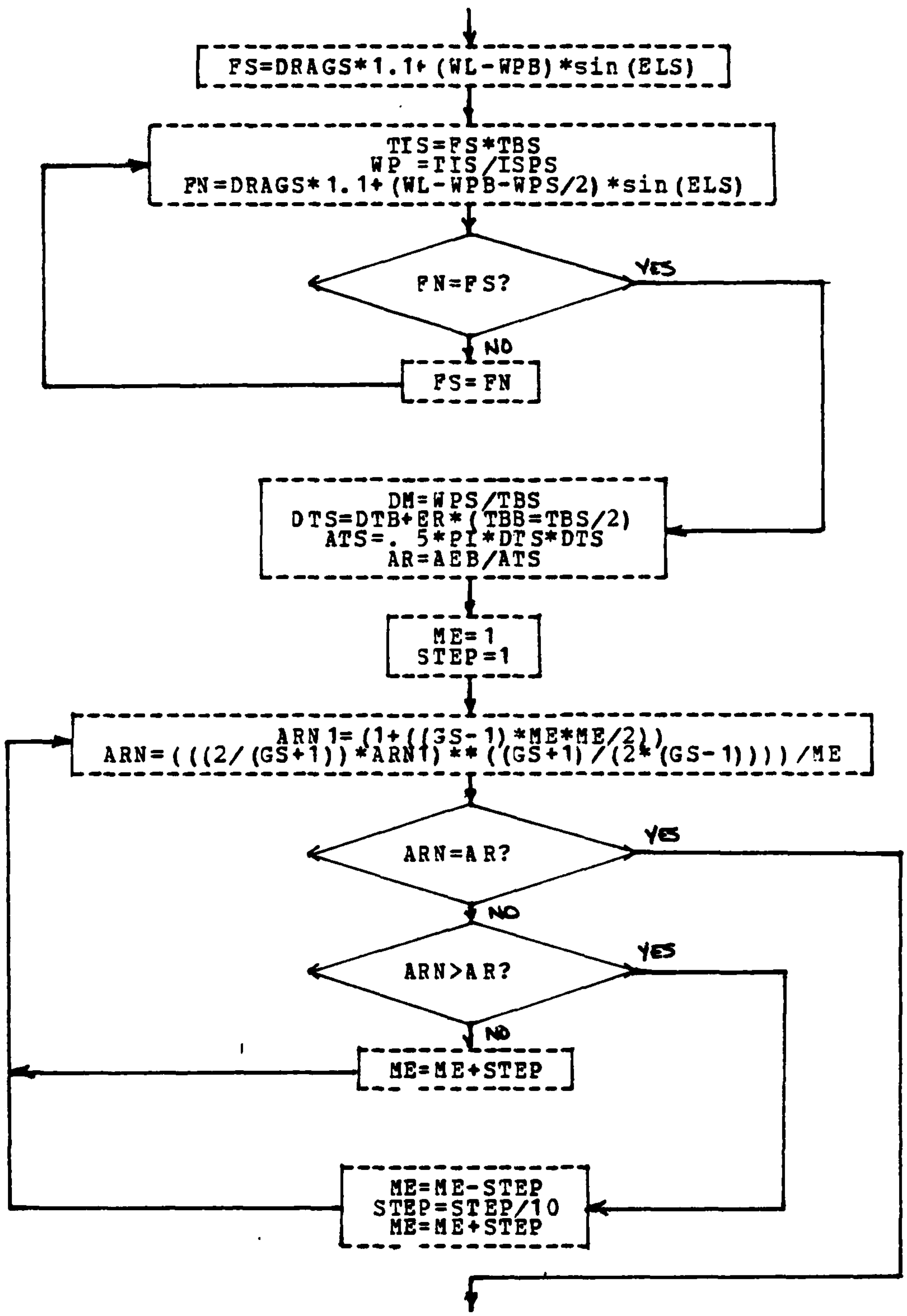
```
   50 FRAGS=(CDEN*5.*PI/FRMLB(1))*(2.*RW*RW*CTH-RW*CTH*CTH)
      FRLEN=(FRMLB(1)/(CTH*CDEN))**.5
      HITS=FRAGS*A
      RETURN
      END
C
C
C
C
      SUBROUTINE SCREEN
      WRITE (6,600)
  600 FORMAT (1X,'CLEAR SCREEN AND ENTER "0"')
      READ (6,16) ISCR
   16 FORMAT (1I1)
      RETURN
      END
```

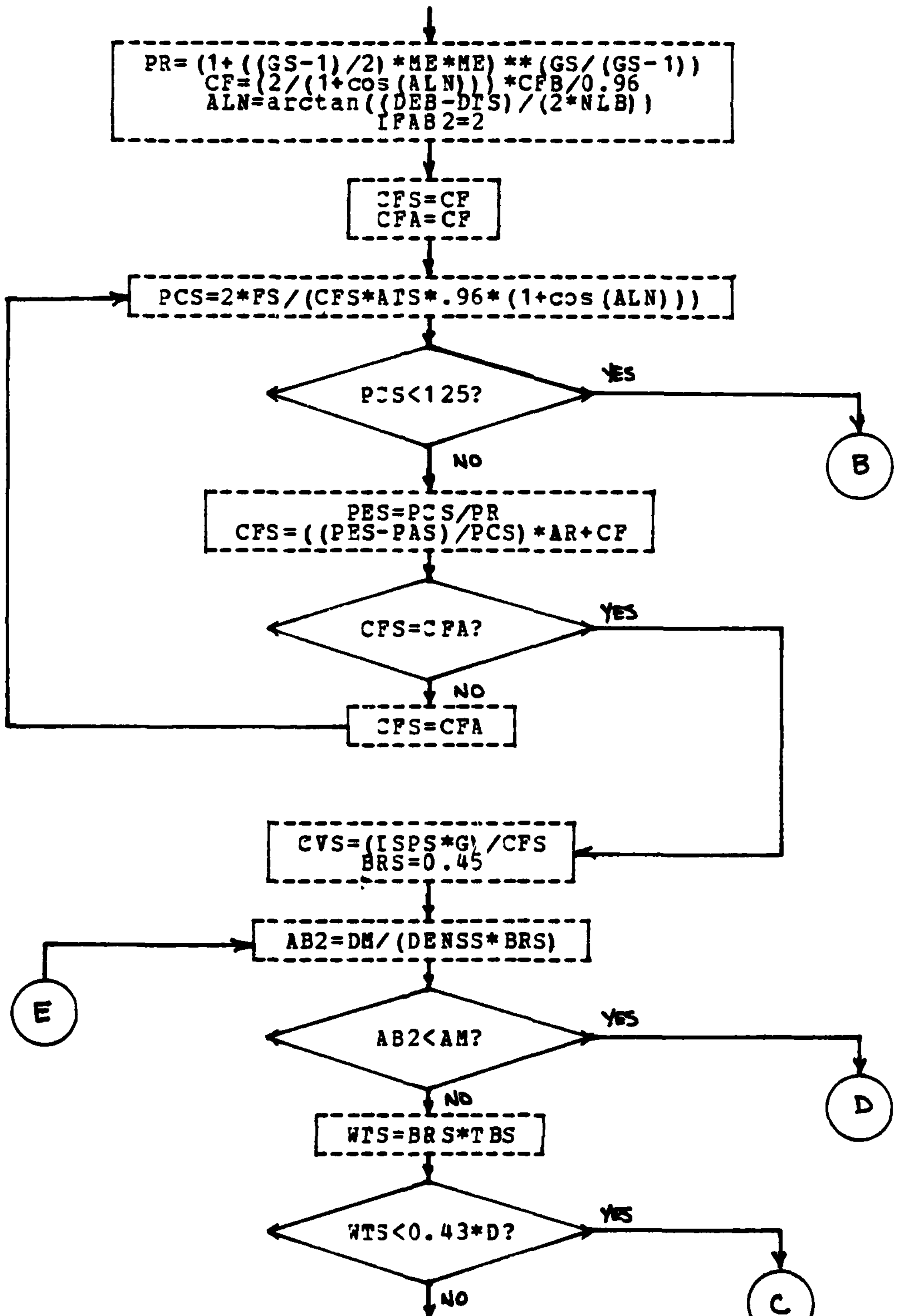
APPENDIX E. PROPULSION SIZING PROGRAM FLOWCHART



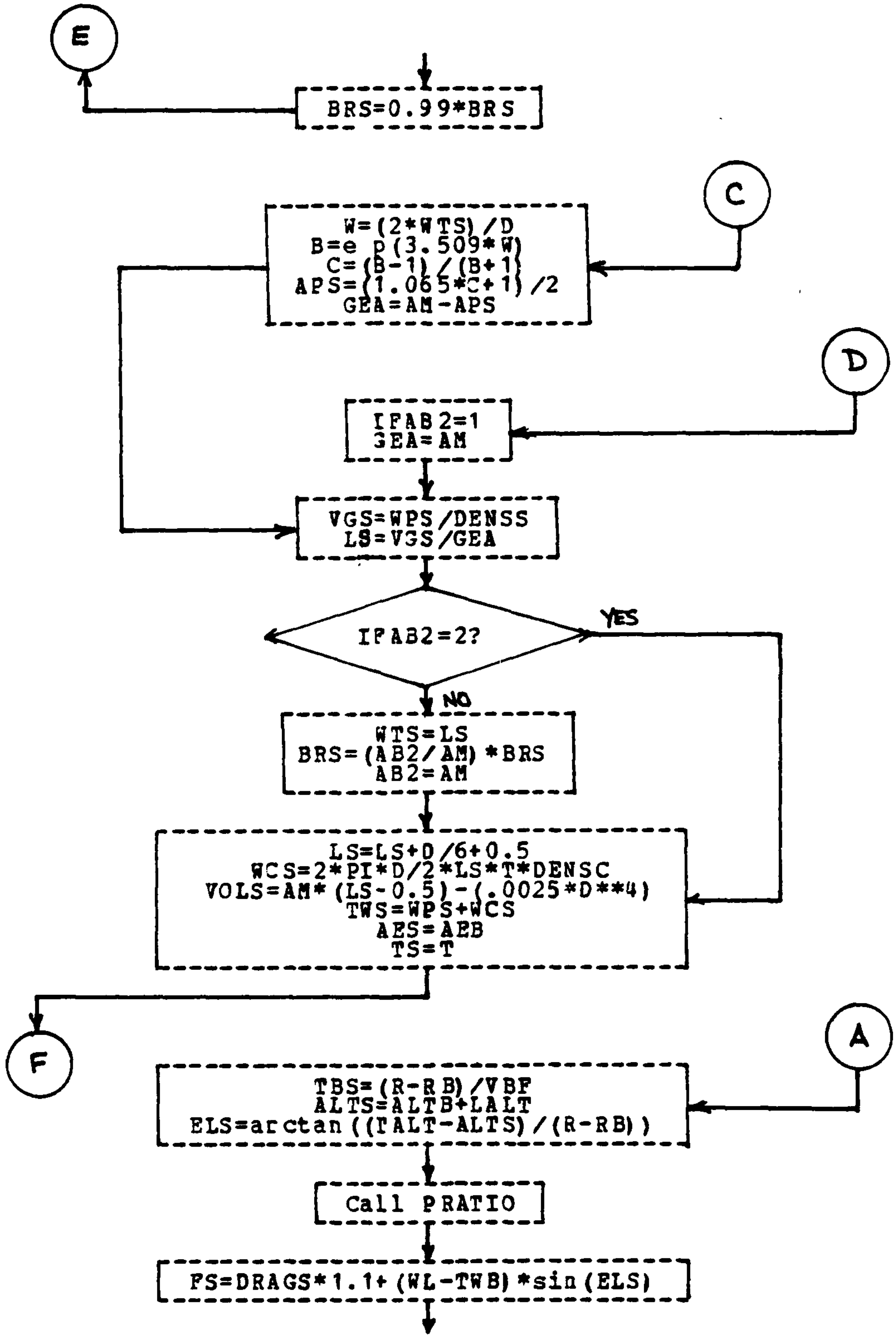

TIB=WPB*ISPB
FB=TIB/TBB
B=(DV/(G*TBB))+sin(ELB)
WN=WL*(1-exp(-WPB/(FB-DRAGB))*B))
WPB=WN?
YES
NO
WPB=WN
DM=WPB/TBB
VGB=WPB/DENSB
TEP=100
B=(DENSB/2.4)*((GB-1)/(2*GB))*(YIELD/DENSC)
PCB=PAB
PEB=PAB
AR=1
Call PCRAT
PLIM=2000
Call ARAT
Is
IMOTOR=0?
or
DEB>(0.95*DB)?
or
PCB<1000?
YES
NO
PLIM=PLIM*.99

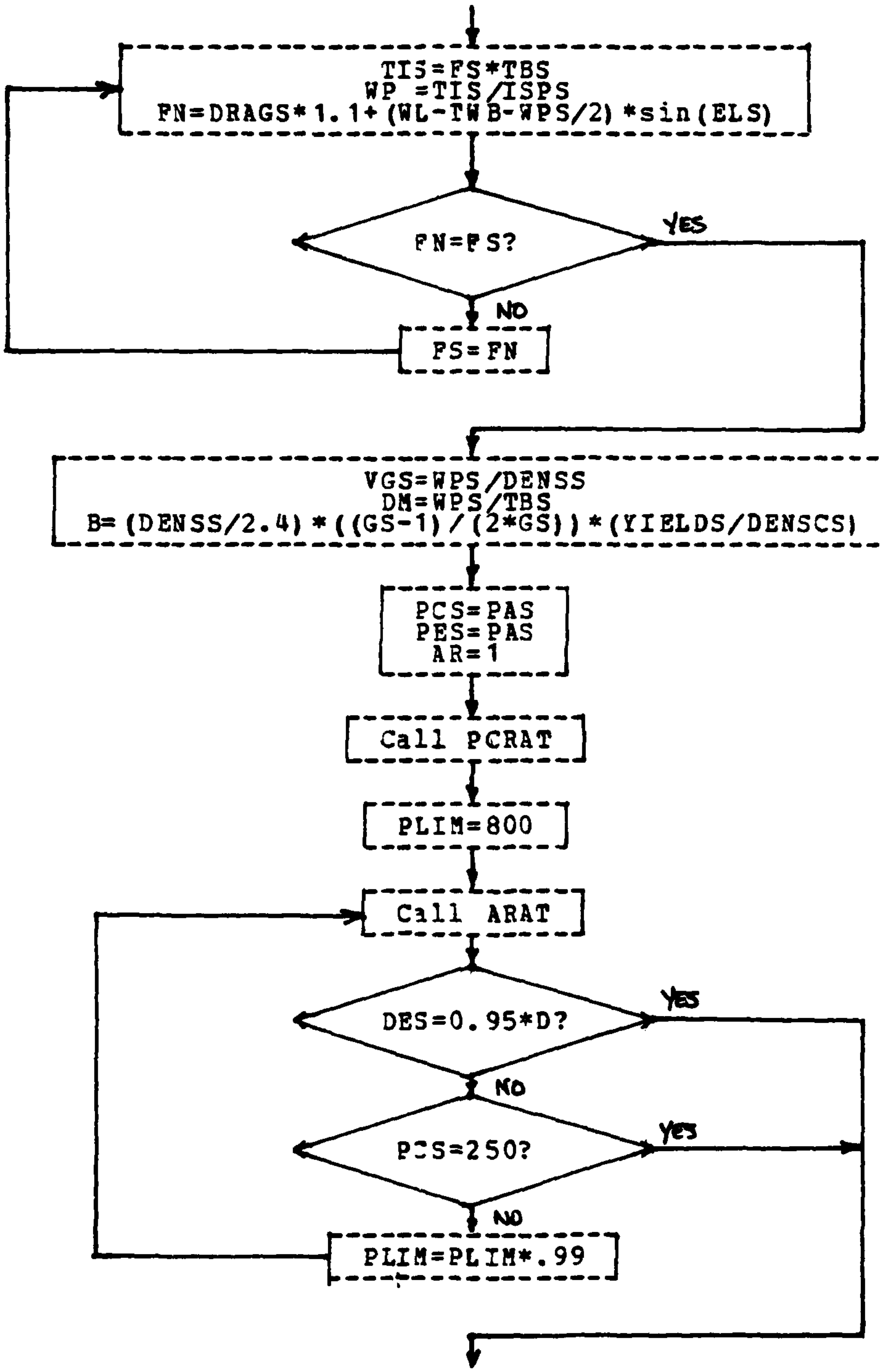

T=(PCB*DB)/(2*YIELD)
J=0.85
APB=ATB/J
AM=PI*(DB/2-T)**2
GEA=AM-APB
BRB=1.25
B=(APB/PI)**.5
WT=(DB/2)-B
ABB=DM/(DENSB*BRB)
WTB=BRB*TBB
WTB<WT?
YES
NO
BRB=.99*BRB
LB=VGB/GEA+DB/6+0.5
WCB=2*PI DB/2*LB*T*DENSC
VOLB=AM*(LB-0.5)-(.0025*DB**4)
NLB=(DEB-DIB)/(2*tan(ALN))
TWB=WPB+WCB
IMOTOR=1?
YES
A
NO
TBS=(R-RB)/VBF
ALTS=ALTB+LALT
ELS=arctan((TALT-ALTS)/(R-RB))
ALTS=VBF*sin(ELS)*TBS*0.5+ALTS
Call PRATIO

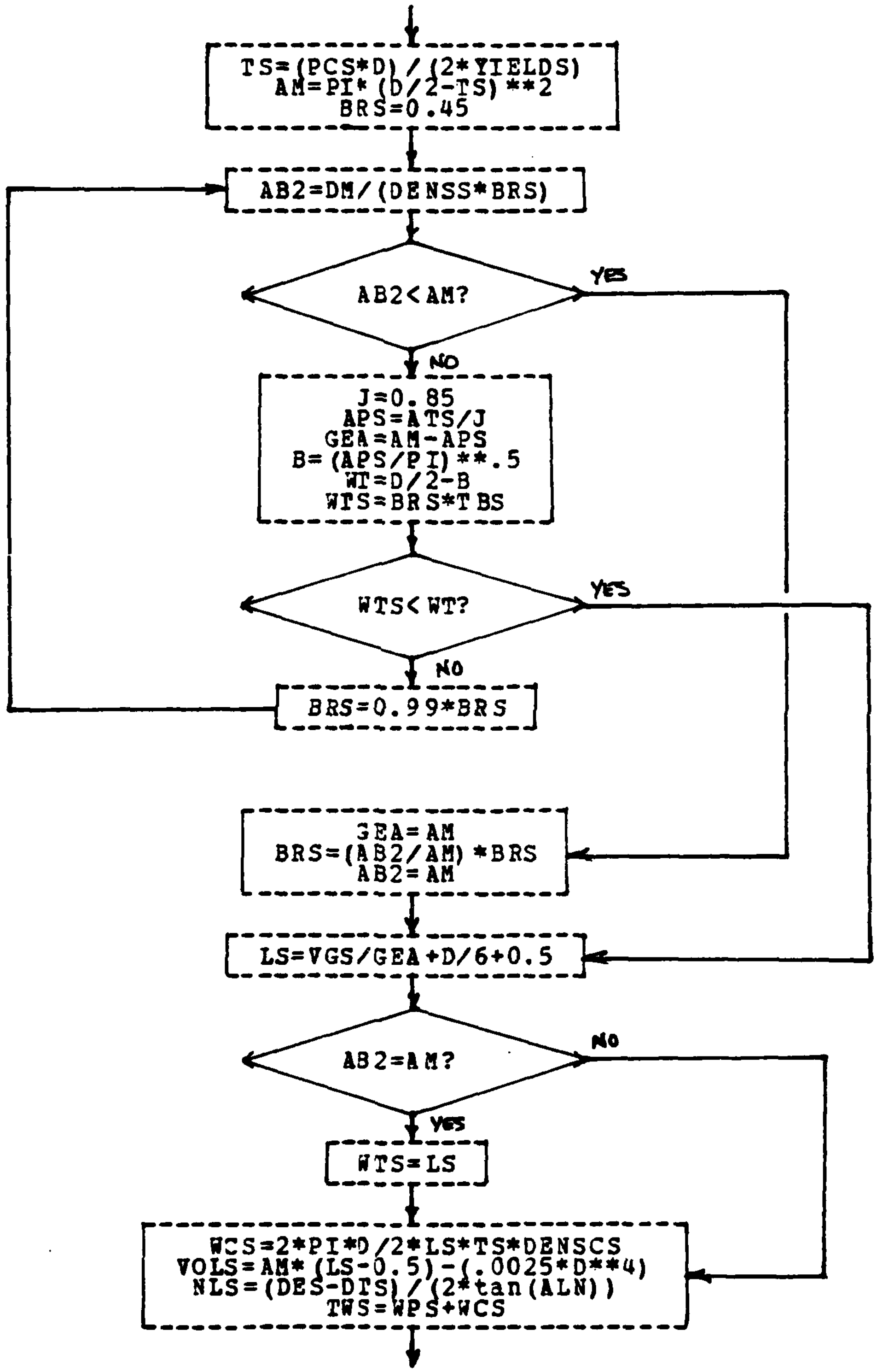

FS=DRAGS*1.1+(WL-WPB)*sin(ELS)
TIS=FS*TBS
WP =TIS/ISPS
FN=DRAGS*1.1+(WL-WPB-WPS/2)*sin(ELS)
FN=FS?
YES
NO
FS=FN
DM=WPS/TBS
DTS=DTB+ER*(TBB=TBS/2)
ATS=.5*PI*DTS*DTS
AR=AEB/ATS
ME=1
STEP=1
ARN1=(1+((GS-1)*ME*ME/2))
ARN=(((2/(GS+1))*ARN1)**((GS+1)/(2*(GS-1))))/ME
ARN=AR?
YES
NO
ARN>AR?
YES
NO
ME=ME+STEP
ME=ME-STEP
STEP=STEP/10
ME=ME+STEP

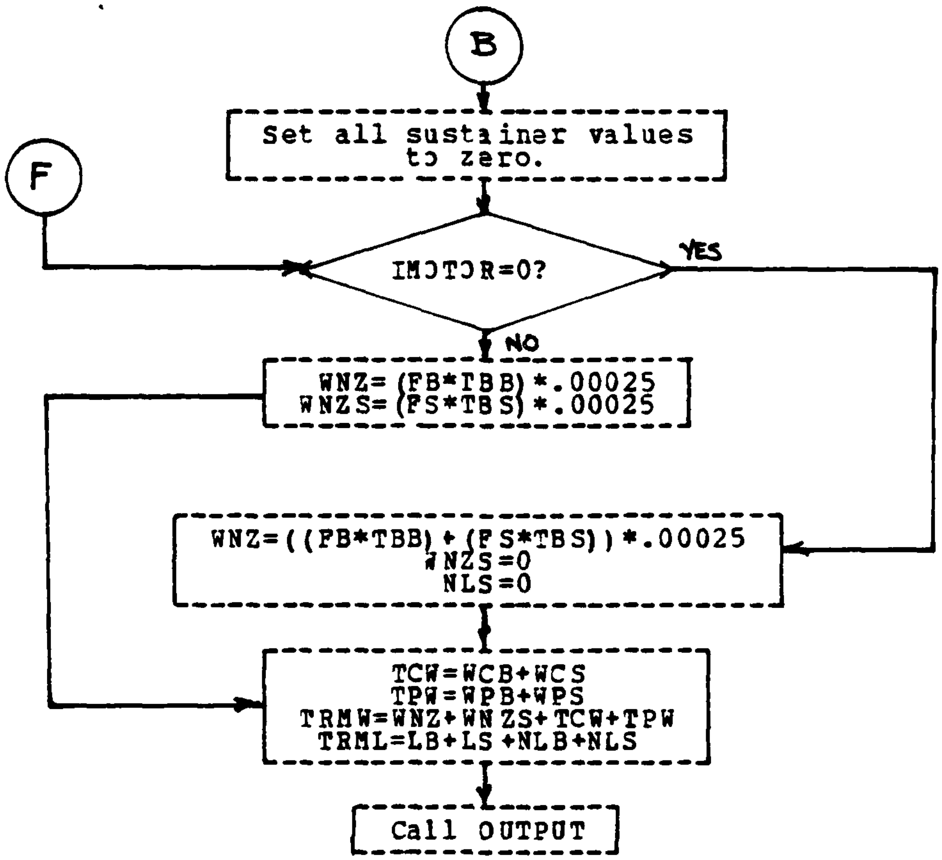


PR= (1+ ((GS-1)/2) *ME*ME) ** (GS/(GS-1))
CF=(2/(1+cos(ALN)))*CFB/0.96
ALN=arctan((DEB-DTS)/(2*NLB))
IFAB2=2
CFS=CF
CFA=CF
PCS=2*FS/(CFS*ATS*.96*(1+cos(ALN)))
PCS<125?
YES
NO
B
PES=PCS/PR
CFS=((PES-PAS)/PCS)*AR+CF
CFS=CFA?
YES
NO
CFS=CFA
CVS=(ISPS*G)/CFS
BRS=0.45
AB2=DM/(DENSS*BRS)
E
AB2<AM?
YES
NO
D
WTS=BRS*TBS
WTS<0.43*D?
YES
NO
C

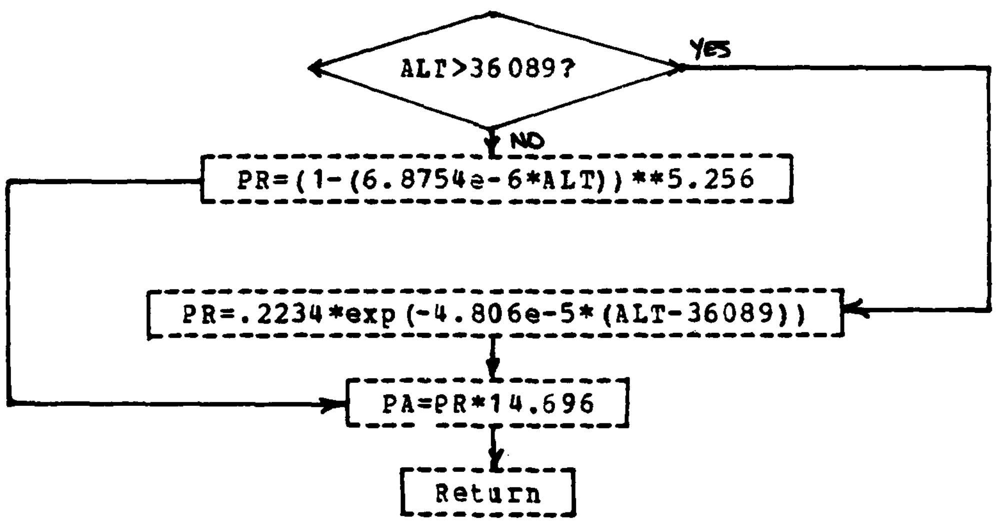

E
BRS=0.99*BRS
C
W=(2*WTS)/D
B=e p(3.509*W)
C=(B-1)/(B+1)
APS=(1.065*C+1)/2
GEA=AM-APS
D
IFAB2=1
GEA=AM
VGS=WPS/DENSS
LS=VGS/GEA
IFAB2=2?
YES
NO
WTS=LS
BRS=(AB2/AM)*BRS
AB2=AM
LS=LS+D/6+0.5
WCS=2*PI*D/2*LS*T*DENSC
VOLS=AM*(LS-0.5)-(.0025*D**4)
TWS=WPS+WCS
AES=AEB
TS=T
F
A
TBS=(R-RB)/VBF
ALTS=ALTB+LALT
ELS=arctan((TALT-ALTS)/(R-RB))
Call PRATIO
FS=DRAGS*1.1+(WL-TWB)*sin(ELS)

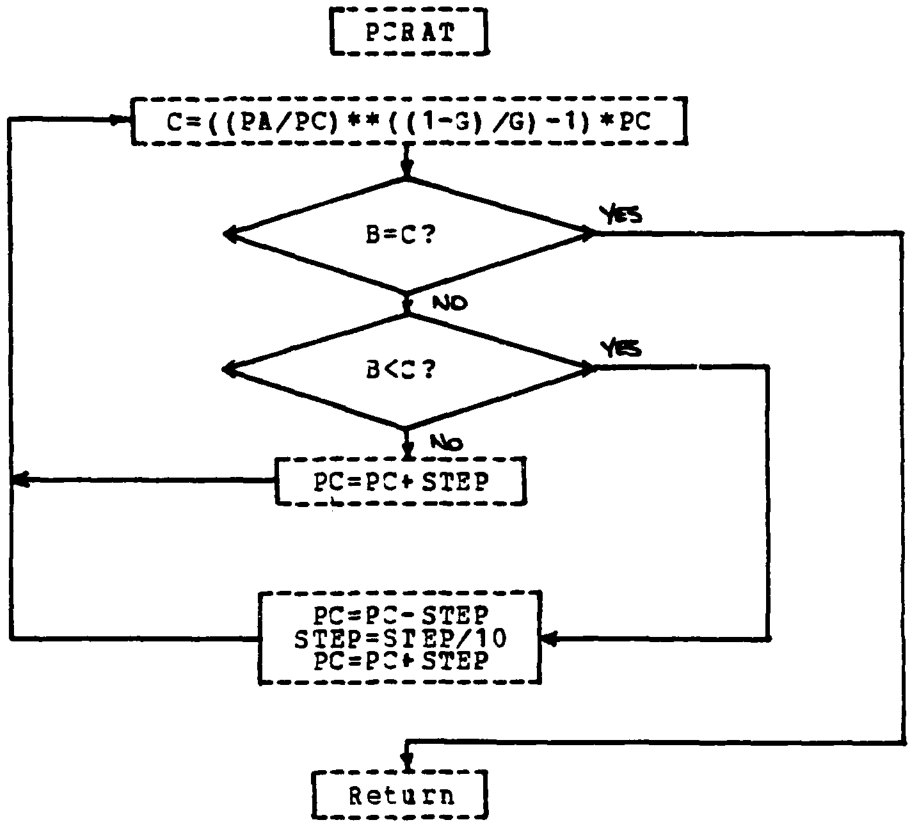

TIS=FS*TBS
WP =TIS/ISPS
FN=DRAGS*1.1+(WL-TWB-WPS/2)*sin(ELS)
FN=FS?
YES
NO
FS=FN
VGS=WPS/DENSS
DM=WPS/TBS
B=(DENSS/2.4)*((GS-1)/(2*GS))*(YIELDS/DENSCS)
PCS=PAS
PES=PAS
AR=1
Call PCRAT
PLIM=800
Call ARAT
DES=0.95*D?
YES
NO
PCS=250?
YES
NO
PLIM=PLIM*.99

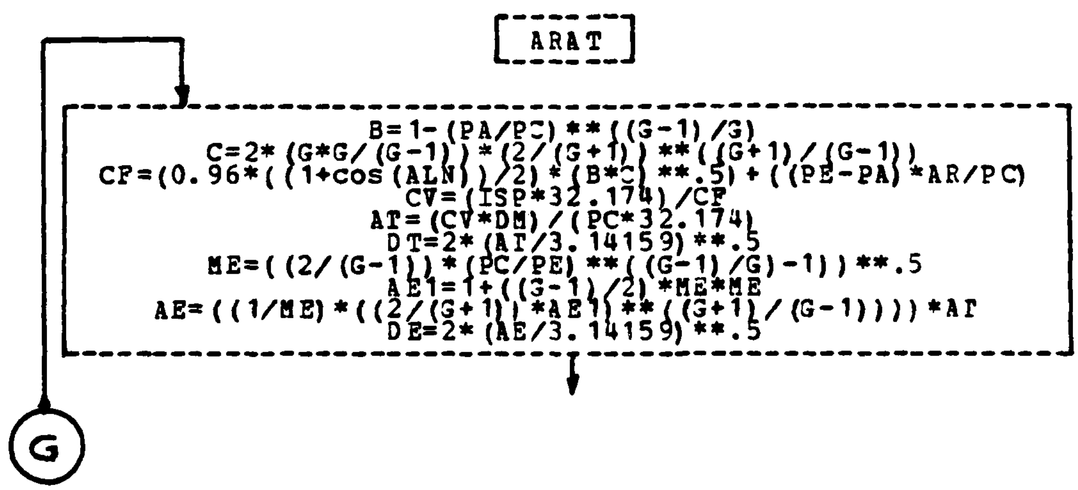

TS=(PCS*D)/(2*YIELDS)
AM=PI*(D/2-TS)**2
BRS=0.45
AB2=DM/(DENSS*BRS)
AB2<AM?
YES
NO
J=0.85
APS=ATS/J
GEA=AM-APS
B=(APS/PI)**.5
WT=D/2-B
WTS=BRS*TBS
WTS<WT?
YES
NO
BRS=0.99*BRS
GEA=AM
BRS=(AB2/AM)*BRS
AB2=AM
LS=VGS/GEA+D/6+0.5
AB2=AM?
NO
YES
WTS=LS
WCS=2*PI*D/2*LS*TS*DENSCS
VOLS=AM*(LS-0.5)-(.0025*D**4)
NLS=(DES-DTS)/(2*tan(ALN))
TWS=WPS+WCS

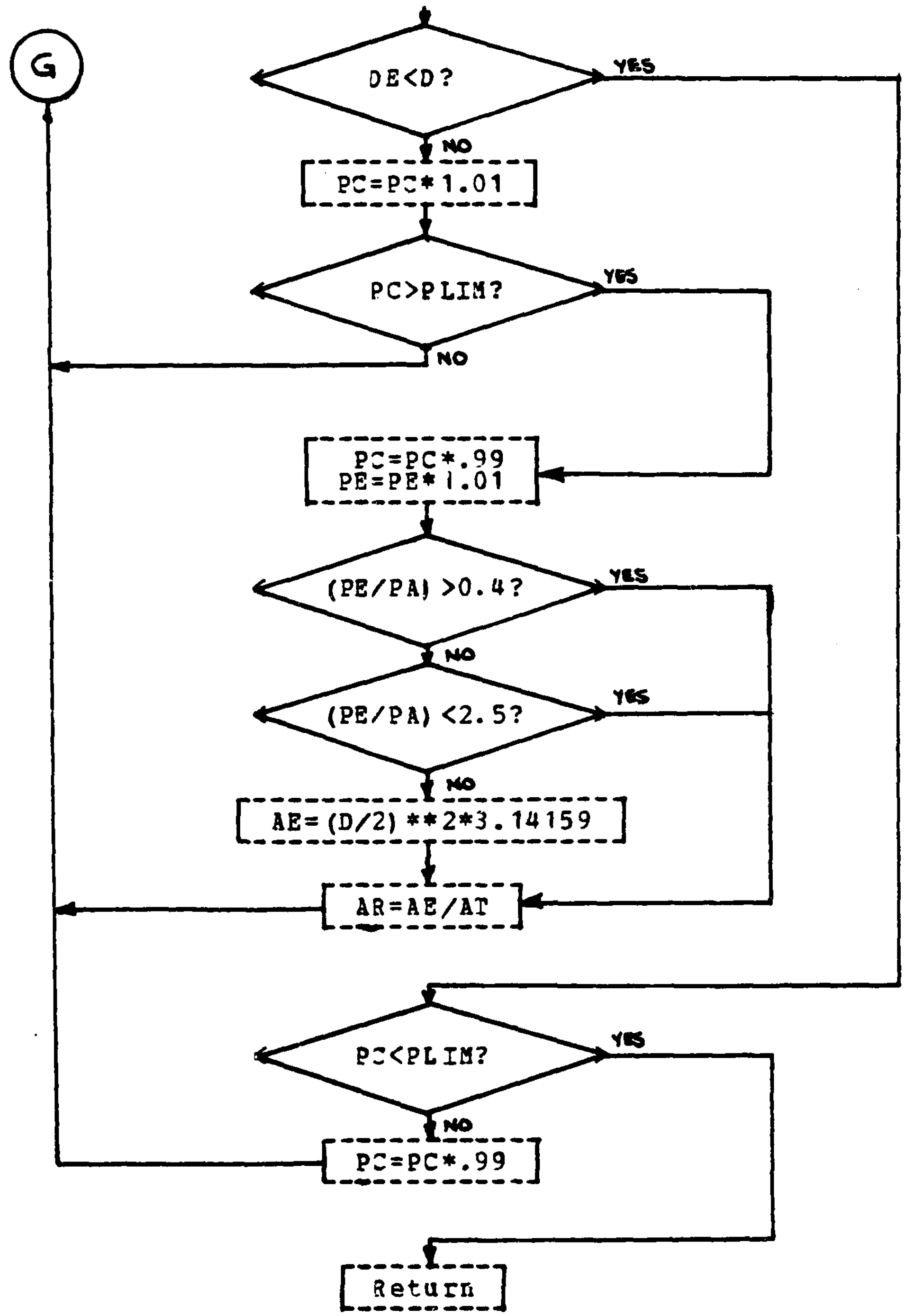

B
Set all sustainer values to zero.
F
IMOTOR=0?
YES
NO
WNZ= (FB*TBB)*.00025
WNZS= (FS*TBS)*.00025
WNZ=((FB*TBB)+(FS*TBS))*.00025
WNZS=0
NLS=0
TCW=WCB+WCS
TPW=WPB+WPS
TRMW=WNZ+WNZS+TCW+TPW
TRML=LB+LS+NLB+NLS
Call OUTPUT

PRATIO


ALT>36089?
YES
NO
PR=(1-(6.8754e-6*ALT))**5.256
PR=.2234*exp(-4.806e-5*(ALT-36089))
PA=PR*14.696
Return


PCRAT
C=((PA/PC)**((1-G)/G)-1)*PC
B=C?
YES
NO
B<C?
YES
NO
PC=PC+STEP
PC=PC-STEP
STEP=STEP/10
PC=PC+STEP
Return


ARAT
B=1-(PA/PC)**((G-1)/G)
C=2*(G*G/(G-1))*(2/(G+1))**((G+1)/(G-1))
CF=(0.96*((1+cos(ALN))/2)*(B*C)**.5)+((PE-PA)*AR/PC)
CV=(ISP*32.174)/CF
AT=(CV*DM)/(PC*32.174)
DT=2*(AT/3.14159)**.5
ME=((2/(G-1))*(PC/PE)**((G-1)/G)-1))**.5
AE1=1+((G-1)/2)*ME*ME
AE=((1/ME)*((2/(G+1))*AE1)**((G+1)/(G-1))))*AT
DE=2*(AE/3.14159)**.5
G


G
DE<D?
YES
NO
PC=PC*1.01
PC>PLIM?
YES
NO
PC=PC*.99
PE=PE*1.01
(PE/PA)>0.4?
YES
NO
(PE/PA)<2.5?
YES
NO
AE=(D/2)**2*3.14159
AR=AE/AT
PC<PLIM?
YES
NO
PC=PC*.99
Return

APPENDIX F. PROPULSION SIZING PROGRAM LISTING

Following this page is the program listing for the Propulsion Sizing Program. It has two segments; the executive routine and the FORTRAN IV computational program. The executive routine establishes the required file definitions and initiates operation of the computational program.

The computational program, LPROP FORTRAN, consists of six subprogram divisions. The MAIN program accepts the input data from the user and performs the guiding calculations for the booster and sustainer motors. Subroutine PRATIO determines the ambient pressures at the design altitudes. Subroutine PCRAT defines the optimum chamber pressure to ambient pressure ratio with respect to the case material properties. Subroutine ARAT solves for the area ratio of the nozzle and tries to size the nozzle to the missile diameter by varying the chamber pressure, characteristic velocity, and thrust coefficient. Subroutine SCREEN is used to prompt the user to clear the terminal screen for proper positioning of the output. And subroutine OUTPUT formats the calculated solutions and provides them to the user and to the printer file, if so directed by the user.

```
FILE: LPROP    EXEC     A    NAVAL POSTGRADUATE SCHOOL

FILEDEF 08 DISK LPROP OUTPUT A0  (RECFM VA BLOCK 133 PERM
&BEGTYPE

YOU WILL HAVE THE OPTION TO OBTAIN A HARDCOPY PRINTOUT OF
AS MANY ALTERNATIVES AS YOU WISH.  THE PROGRAM WILL ASK
YOU IF YOU DESIRE TO SAVE A PARTICULAR RUN, SIMPLY ANSWER
ACCORDINGLY.

&END
LOAD LPROP
START
&BEGTYPE

TO OBTAIN A HARDCOPY PRINTOUT OF THE RESULTS, TYPE "PRINT
LPROP OUTPUT" AND ENTER.  THE OUTPUT WILL BE PRINTED ON
THE VM PRINTER IN ROOM 140 AND WILL BE IDENTIFIED BY YOUR
USER NUMBER AND LABEL NAME.  IT USUALLY REQUIRES SEVERAL
MINUTES TO OBTAIN THE PRINTOUT.

&END
```

```
C       PROPULSION SIZING PROGRAM                                          LPR00040
C       LT. MARTIN D. SULLIVAN, USN        21 MAY 1981                     LPR00050
C       PROGRAM COORDINATOR:  PROFESSOR GERALD LINDSEY, AERO. DEPARTMENT   LPR00060
C                                                                          LPR00070
        REAL PI,G,DV,VBF,VBI,DF,DRAGS,DRAGB,FB,WL,A,DM,ISPB,TBB,WPB        LPR00080
        REAL ABB,DENSB,BRB,WTB,VGB,B,GB,YIELD,DENSC,PCB,PAB,C,AEB1         LPR00090
        REAL ATB,DTB,ME,AEB,DEB,D,T,J,APB,AM,GEA,LB,WCB,VOLB,NLB           LPR00100
        REAL FS,ELS,TIS,ISPS,FN,DTS,ER,ATS,AR,ARN,GS,PR,CF,CFS,ARN1        LPR00110
        REAL ISPS,CVS,AB2,BRS,WTS,W,APS,DENSS,VGS,LS,WCS,VOLS,TWS          LPR00120
        REAL TIB,WN,ELB,STEP,CFB,ALN,CVB,RB,R,TWB,TBS,CFA,PCS,PES,PAS      LPR00130
        REAL LALT,TALT,PEB,ALTB,ALTS,TCW,WNZ,TPW,TRMW,TRML,AES             LPR00140
        REAL YIELDS,DENSCS,TS,WNZS,NLS,PLIM,DB,ALTBN,ALTSN                 LPR00150
        INTEGER ICOR,IGO,IFBRB,IFDLMB,IFAB2,IFBRS,IFPCS,IFDLMS,IMOTOR      LPR00160
        INTEGER IFPCB,IFTRMW,IFBSTO,IFTRML                                 LPR00170
        COMMON WPB,WCB,TWB,CFB,CVB,FB,TBB,RB,PCB,ABB,WTB,APB,LB,VOLB       LPR00180
        COMMON ATB,BRB,NLB,AEB,WNZ,TPW,TCW,TRMW,TRML,T                     LPR00190
        COMMON WPS,WCS,TWS,CFS,CVS,FS,TBS,PCS,AB2,WTS,APS,LS,VOLS,ATS,BRS  LPR00200
        COMMON VBF,VBI,YIELD,DENSC,ALN,ER,D,R,WL,A,ELB,ISPB                LPR00210
        COMMON DENSB,GB,LALT,DRAGS,TALT,ISPS,GS,DENSS,DB                   LPR00220
        COMMON YIELDS,DENSCS,TS,WNZS,NLS,AES,ALTBN,ALTSN                   LPR00230
        COMMON IFBRB,IFDLMB,IFAB2,IFBRS,IFPCB,IFTRMW,IFBSTO,IFTRML,IFPCS   LPR00240
        COMMON IFDLMS,IMOTOR                                               LPR00250
        DETORA(X)=(X/360)*2*3.1415926535                                   LPR00260
        RATODE(X)=(X/(2*3.1415926535))*360                                 LPR00270
  997   ICOR=0                                                             LPR00280
 1100   FORMAT (F15.4)                                                     LPR00290
 1110   FORMAT (I2)                                                        LPR00300
 1120   FORMAT (I1)                                                        LPR00310
        WRITE (6,1300)                                                     LPR00320
 1300   FORMAT (1X,'INPUT "0" FOR INTEGRAL MOTORS (COMMON NOZZLE) OR "1" FLPR00330
       +OR STAGED'/,1X,'MOTORS (SEPARATE NOZZLES).')                       LPR00340
        READ (5,1120) IMOTOR                                               LPR00350
    1   WRITE (6,1310)                                                     LPR00360
 1310   FORMAT (1X,'ENTER THE FOLLOWING DATA AS DECIMAL NUMBERS ONLY ')    LPR00370
 6001   WRITE (6,1001)                                                     LPR00380
 1001   FORMAT (1X,'INPUT LAUNCH ALTITUDE (FEET)')                         LPR00390
        READ (5,1100) LALT                                                 LPR00400
        IF (ICOR.EQ.1) GO TO 1260                                          LPR00410
 6002   WRITE (6,1002)                                                     LPR00420
 1002   FORMAT (1X,'INPUT LAUNCH WEIGHT (POUNDS)')                         LPR00430
        READ (5,1100) WL                                                   LPR00440
        IF (ICOR.EQ.1) GO TO 1260                                          LPR00450
 6003   WRITE (6,1003)                                                     LPR00460
 1003   FORMAT (1X,'INPUT LAUNCH VELOCITY (FT/SEC)')                       LPR00470
        READ (5,1100) VBI                                                  LPR00480
        IF (ICOR.EQ.1) GO TO 1260                                          LPR00490
 6004   WRITE (6,1004)                                                     LPR00500
 1004   FORMAT (1X,'INPUT LAUNCH ANGLE (DEGREES)')                         LPR00510
```

```
      READ (5,1100) ELB                                                 LPR00520
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00530
 6005 WRITE (6,1005)                                                    LPR00540
 1005 FORMAT (1X,'INPUT AVERAGE ACCELERATION DURING BOOST (G"S)')       LPR00550
      READ (5,1100) A                                                   LPR00560
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00570
 6006 WRITE (6,1006)                                                    LPR00580
 1006 FORMAT (1X,'INPUT CRUISE VELOCITY (FT/SEC)')                      LPR00590
      READ (5,1100) VBF                                                 LPR00600
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00610
 6007 WRITE (6,1007)                                                    LPR00620
 1007 FORMAT (1X,'INPUT DRAG ON MISSILE AT CRUISE VELOCITY (POUNDS)')   LPR00630
      READ (5,1100) DRAGS                                               LPR00640
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00650
 6008 WRITE (6,1008)                                                    LPR00660
 1008 FORMAT (1X,'INPUT MAXIMUM RANGE (NAUTICAL MILES)')                LPR00670
      READ (5,1100) R                                                   LPR00680
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00690
 6009 WRITE (6,1009)                                                    LPR00700
 1009 FORMAT (1X,'INPUT FINAL (TARGET) ALTITUDE (FEET)')                LPR00710
      READ (5,1100) TALT                                                LPR00720
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00730
 6010 WRITE (6,1010)                                                    LPR00740
 1010 FORMAT (1X,'INPUT BOOSTER PROPELLANT SPECIFIC IMPULSE (SECONDS)') LPR00750
      READ (5,1100) ISPB                                                LPR00760
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00770
 6011 WRITE (6,1011)                                                    LPR00780
 1011 FORMAT (1X,'INPUT BOOSTER PROPELLANT DENSITY (LBS/CU.IN)')        LPR00790
      READ (5,1100) DENSB                                               LPR00800
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00810
 6012 WRITE (6,1012)                                                    LPR00820
 1012 FORMAT (1X,'INPUT BOOSTER EXHAUST SPECIFIC HEAT RATIO')           LPR00830
      READ (5,1100) GB                                                  LPR00840
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00850
 6013 WRITE (6,1013)                                                    LPR00860
 1013 FORMAT (1X,'INPUT SUSTAINER PROPELLANT SPECIFIC IMPULSE (SEC)')   LPR00870
      READ (5,1100) ISPS                                                LPR00880
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00890
 6014 WRITE (6,1014)                                                    LPR00900
 1014 FORMAT (1X,'INPUT SUSTAINER PROPELLANT DENSITY (LBS/CU.IN)')      LPR00910
      READ (5,1100) DENSS                                               LPR00920
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00930
 6015 WRITE (6,1015)                                                    LPR00940
 1015 FORMAT (1X,'INPUT SUSTAINER EXHAUST SPECIFIC HEAT RATIO')         LPR00950
      READ (5,1100) GS                                                  LPR00960
      IF (ICOR.EQ.1) GO TO 1260                                         LPR00970
 6016 WRITE (6,1016)                                                    LPR00980
 1016 FORMAT (1X,'INPUT NOZZLE HALF ANGLE (DEGREES)')                   LPR00990
```

```
      READ (5,1100) ALN                                                 LPRO1000
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1010
      IF (IMOTOR.EQ.1) GO TO 90                                         LPRO1020
 6017 WRITE (6,1017)                                                    LPRO1030
 1017 FORMAT (1X,'INPUT NOZZLE DESIGN ALTITUDE (FEET)')                 LPRO1040
      READ (5,1100) ALTBN                                               LPRO1050
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1060
 6018 WRITE (6,1018)                                                    LPRO1070
 1018 FORMAT (1X,'INPUT NOZZLE EROSION RATE (IN/SEC)')                  LPRO1080
      READ (5,1100) ER                                                  LPRO1090
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1100
 6019 WRITE (6,1019)                                                    LPRO1110
 1019 FORMAT (1X,'INPUT MISSILE DIAMETER (INCHES)')                     LPRO1120
      READ (5,1100) DB                                                  LPRO1130
      D=DB                                                              LPRO1140
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1150
 6020 WRITE (6,1020)                                                    LPRO1160
 1020 FORMAT (1X,'INPUT YIELD STRENGTH OF CASE MATERIAL (PSI)')         LPRO1170
      READ (5,1100) YIELD                                               LPRO1180
      YIELDS=YIELD                                                      LPRO1190
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1200
 6021 WRITE (6,1021)                                                    LPRO1210
 1021 FORMAT (1X,'INPUT DENSITY OF THE CASE MATERIAL (LB/CU.IN)')       LPRO1220
      READ (5,1100) DENSC                                               LPRO1230
      DENSCS=DENSC                                                      LPRO1240
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1250
      GO TO 1201                                                        LPRO1260
   90 CONTINUE                                                          LPRO1270
 6022 WRITE (6,1022)                                                    LPRO1280
 1022 FORMAT (1X,'INPUT BOOSTER NOZZLE DESIGN ALTITUDE (FEET)')         LPRO1290
      READ (5,1100) ALTBN                                               LPRO1300
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1310
 6023 WRITE (6,1023)                                                    LPRO1320
 1023 FORMAT (1X,'INPUT BOOSTER DIAMETER (INCHES)')                     LPRO1330
      READ (5,1100) DB                                                  LPRO1340
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1350
 6024 WRITE (6,1024)                                                    LPRO1360
 1024 FORMAT (1X,'INPUT YIELD STRENGTH OF BOOSTER CASE MATERIAL (PSI)') LPRO1370
      READ (5,1100) YIELD                                               LPRO1380
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1390
 6025 WRITE (6,1025)                                                    LPRO1400
 1025 FORMAT (1X,'INPUT DENSITY OF BOOSTER CASE MATERIAL (LBS/CU.IN)')  LPRO1410
      READ (5,1100) DENSC                                               LPRO1420
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1430
 6026 WRITE (6,1026)                                                    LPRO1440
 1026 FORMAT (1X,'INPUT SUSTAINER NOZZLE DESIGN ALTITUDE (FEET)')       LPRO1450
      READ (5,1100) ALTSN                                               LPRO1460
      IF (ICOR.EQ.1) GO TO 1260                                         LPRO1470
```

```
6027 WRITE (6,1027)
1027 FORMAT (1X,'INPUT SUSTAINER DIAMETER (INCHES)')
     READ (5,1100) D
     IF (ICOR.EQ.1) GO TO 1260
6028 WRITE (6,1028)
1028 FORMAT(1X,'INPUT YIELD STRENGTH OF SUSTAINER CASE MATERIAL (PSI)')
     READ (5,1100) YIELDS
     IF (ICOR.EQ.1) GO TO 1260
6029 WRITE (6,1029)
1029 FORMAT (1X,'INPUT DENSITY OF SUSTAINER CASE MATERIAL (LBS/CU.IN)')
     READ (5,1100) DENSCS
     IF (ICOR.EQ.1) GO TO 1260
1201 CONTINUE
     CALL SCREEN
     WRITE (6,1202)
1202 FORMAT (/1X,'REVIEW THE FOLLOWING LIST OF INPUT PARAMETERS AND REC
    +ORD THE NUMBERS'/,1X,'OF THOSE TO BE CHANGED.'/)
     ICOR=1
     WRITE (6,1210) LALT,WL,VBI,ELB,A,VBF,DRAGS,R,TALT,ISPB,DENSB,GB
1210 FORMAT (1X,'SUMMARY OF INPUT PARAMETERS=========================',
    +/5X,'01) LAUNCH ALTITUDE',T45,F12.1,' FEET',
    +/5X,'02) LAUNCH WEIGHT',T45,F12.2,' POUNDS',
    +/5X,'03) LAUNCH VELOCITY',T45,F12.1,' FT/SEC',
    +/5X,'04) LAUNCH ANGLE',T45,F12.1,' DEGREES',
    +/5X,'05) AVERAGE ACCELERATION',T45,F12.2,' G"S',
    +/5X,'06) CRUISE VELOCITY',T45,F12.1,' FT/SEC',
    +/5X,'07) DRAG AT CRUISE VELOCITY',T45,F12.1,' POUNDS',
    +/5X,'08) MAXIMUM RANGE',T45,F12.0,' MILES',
    +/5X,'09) FINAL (TARGET) ALTITUDE',T45,F12.1,' FEET',
    +/5X,'10) BOOSTER PROPELLANT SPECIFIC IMPULSE',T45,F12.1,' SEC',
    +/5X,'11) BOOSTER PROPELLANT DENSITY',T45,F12.4,' LBS/CU.IN',
    +/5X,'12) BOOSTER EXHAUST SPECIFIC HEAT RATIO',T50,F7.5)
     WRITE (6,1220) ISPS,DENSS,GS,ALN
1220 FORMAT (5X,'13) SUSTAINER PROPELLANT SPECIFIC IMPULSE',T50,F7.1,
    +' SEC',
    +/5X,'14) SUSTAINER PROPELLANT DENSITY',T45,F12.4,' LBS/CU.IN',
    +/5X,'15) SUSTAINER EXHAUST SPECIFIC HEAT RATIO',T50,F7.5,
    +/5X,'16) NOZZLE HALF ANGLE',T45,F12.2,' DEGREES')
     IF (IMOTOR.EQ.1) GO TO 1240
     WRITE (6,1230) ALTBN,ER,DB,YIELD,DENSC
1230 FORMAT (5X,'17) NOZZLE DESIGN ALTITUDE',T45,F12.1,' FEET',
    +/5X,'18) NOZZLE EROSION RATE',T45,F12.5,' IN/SEC',
    +/5X,'19) MISSILE DIAMETER',T45,F12.1,' INCHES',
    +/5X,'20) YIELD STRENGTH OF CASE MATERIAL',T45,F12.1,' PSI',
    +/5X,'21) DENSITY OF CASE MATERIAL',T45,F12.4,' LBS/CU.IN')
     GO TO 1250
1240 WRITE (6,1245) ALTBN,DB,YIELD,DENSC,ALTSN,D,YIELDS,DENSCS
1245 FORMAT (5X,'17) BOOSTER DESIGN ALTITUDE',T45,F12.1,' FEET',
```

```
     +/5X,'18) BOOSTER DIAMETER',T45,F12.2,' INCHES',
     +/5X,'19) YIELD STRENGTH OF BOOSTER CASE',T45,F12.1,' PSI',
     +/5X,'20) DENSITY OF BOOSTER CASE MATERIAL',T45,F12.4,' LBS/CU.IN',
     +/5X,'21) SUSTAINER DESIGN ALTITUDE',T45,F12.1,' FEET',
     +/5X,'22) SUSTAINER DIAMETER',T45,F12.2,' INCHES',
     +/5X,'23) YIELD STRENGTH OF SUSTAINER CASE',T45,F12.1,' PSI',
     +/5X,'24) DENSITY OF SUSTAINER CASE MATERIAL',T45,F12.4,' LBS/CU.IN
     +')
 1250 CONTINUE
      WRITE (6,1254)
 1254 FORMAT (/1X,'HOW MANY INPUT PARAMETERS DO YOU WISH TO CHANGE? (TWO
     + DIGIT INTEGER, PLEASE)')
      READ (5,1110) N
      IF (N.EQ.0) GO TO 1301
      DO 1260 I=1,N
 1255    WRITE (6,1256)
 1256    FORMAT (1X,'INPUT TWO DIGIT ITEM NUMBER OF PARAMETER TO BE CHAN
     +GED.')
         READ (5,1110) IGO
         IF ((IGO.GE.1).AND.(IGO.LE.24)) GO TO 1258
         WRITE (6,1257) IGO
 1257    FORMAT(/1X,'WRONG   ',I2,' IS NOT A VALID CHOICE.  TRY AGAIN.')
         GO TO 1255
 1258    IF ((IMOTOR.EQ.1).AND.(IGO.GT.16)) IGO=IGO+5
 1261    GO TO (6001,6002,6003,6004,6005,6006,6007,6008,6009,6010,6011,
     +6012,6013,6014,6015,6016,6017,6018,6019,6020,6021,6022,6023,6024,
     +6025,6026,6027,6028,6029),IGO
 1260 CONTINUE
      GO TO 1201
 1301 CONTINUE
      CALL SCREEN
      PI=3.1415927
      G=32.174
      ALN=DETORA(ALN)
      ELB=DETORA(ELB)
      R=R*6080.
C
C=====BOOSTER MOTOR CALCULATIONS=======================================
C
      DV=VBF-VBI
      DF=(VBF*VBF)/DRAGS
      DRAGB=(VBF+VBI)**2./(4.*DF)
      FB=(WL*A)+DRAGB
      DM=FB/ISPB
      TBB=DV/(A*G)
      RB=(VBI+0.5*DV)*TBB*COS(ELB)
      ALTB=(VBI+0.5*DV)*TBB*SIN(ELB)+LALT
      CALL PRATIO (ALTBN,PAB)
```

```
      WPB=TBB*DM                                                        LPR02440
C                                                                       LPR02450
  100 TIB=WPB*ISPB                                                      LPR02460
      FB=TIB/TBB                                                        LPR02470
      B=DV/(G*TBB)+SIN(ELB)                                             LPR02480
      WN=WL*(1.-EXP((-WPB/(FB-DRAGB))*B))                               LPR02490
      IF (ABS(WPB-WN).LT.0.01) GO TO 110                                LPR02500
      WPB=WN                                                            LPR02510
      GO TO 100                                                         LPR02520
C                                                                       LPR02530
  110 CONTINUE                                                          LPR02540
      WPB=WN                                                            LPR02550
      DM=WPB/TBB                                                        LPR02560
C                                                                       LPR02570
  113 CONTINUE                                                          LPR02580
      VGB=WPB/DENSB                                                     LPR02590
      STEP=100.                                                         LPR02600
      B=(DENSB/2.4)*((GB-1.)/(2.*GB))*(YIELD/DENSC)                     LPR02610
      PCB=PAB                                                           LPR02620
      PEB=PAB                                                           LPR02630
      AR=1.                                                             LPR02640
      IFDLMB=0                                                          LPR02650
      CALL PCRAT (PAB,PCB,GB,B,STEP)                                    LPR02660
      PLIM=2000.                                                        LPR02670
  140 CALL ARAT (PAB,PCB,GB,PEB,ISPB,DB,CFB,CVB,ATB,DTB,AEB,DEB,IFDLMB, LPR02680
     +IFPCB,AR,ALN,DM,PLIM,IMOTOR)                                      LPR02690
      IF ((IMOTOR.EQ.0).OR.(DEB.GE.(0.95*DB)).OR.(PCB.LT.1000))GO TO 150LPR02700
      PLIM=PLIM*.99                                                     LPR02710
      GO TO 140                                                         LPR02720
C                                                                       LPR02730
  150 CONTINUE                                                          LPR02740
      T=(PCB*DB)/(2.*YIELD)                                             LPR02750
      J=0.85                                                            LPR02760
  160 APB=ATB/J                                                         LPR02770
      AM=PI*(DB/2.-T)**2.                                               LPR02780
      GEA=AM-APB                                                        LPR02790
      IFBRB=0                                                           LPR02800
      BRB=1.25                                                          LPR02810
      B=(APB/PI)**0.5                                                   LPR02820
      C=DB/2.-B                                                         LPR02830
C                                                                       LPR02840
  165 ABB=DM/(DENSB*BRB)                                                LPR02850
      WTB=BRB*TBB                                                       LPR02860
      IF (WTB.LT.C) GO TO 170                                           LPR02870
      BRB=0.99*BRB                                                      LPR02880
      IFBRB=1                                                           LPR02890
      GO TO 165                                                         LPR02900
C                                                                       LPR02910
```

```
  170 LB=VGB/GEA
      LB=LB+DB/6.+0.5
      WCB=2.*PI*DB/2.*LB*T*DENSC
      VOLB=AM*(LB-0.5)-(.0025*DB**4.)
      NLB=(DEB-DTB)/(2*TAN(ALN))
      TWB=WPB+WCB
             WRITE (6,9000)
 9000        FORMAT (1X,'MADE IT THIS FAR.')
      IFBSTO=1
      IF (RB.GT.R) GO TO 999
      IFBSTO=0
      IF (IMOTOR.EQ.1) GO TO 300
C
C=====SUSTAINER MOTOR CALCULATIONS (COMMON NOZZLE)=====================
C
  200 TBS=(R-RB)/VBF
      ALTS=ALTB+LALT
      ELS=ATAN((TALT-ALTS)/(R-RB))
      ALTS=VBF*SIN(ELS)*TBS*0.5+ALTS
      CALL PRATIO (ALTSN,PAS)
      FS=DRAGS*1.1+(WL-WPB)*SIN(ELS)
C
  201 TIS=FS*TBS
      WPS=TIS/ISPS
      FN=DRAGS*1.1+(WL-WPB-WPS/2)*SIN(ELS)
      IF (ABS(FN-FS).LT.0.01) GO TO 210
      FS=FN
      GO TO 201
C
  210 CONTINUE
      FS=FN
      DM=WPS/TBS
      DTS=DTB+ER*(TBB+TBS*0.5)
      ATS=.25*PI*DTS*DTS
      AR=AEB/ATS
      ME=1.
      STEP=1.
C
  220 ARN1=(1.+((GS-1.)*ME*ME/2.))
      ARN=(((2./(GS+1.))*ARN1)**((GS+1.)/(2.*(GS-1.))))/ME
      IF (ABS(ARN-AR).LT.0.001) GO TO 240
      IF (ARN.GT.AR) GO TO 230
      ME=ME+STEP
      GO TO 220
  230 CONTINUE
      ME=ME-STEP
      STEP=STEP/10.
      ME=ME+STEP
```

```
      GO TO 220                                                         LPR03400
C                                                                       LPR03410
  240 CONTINUE                                                          LPR03420
      PR=(1.+((GS-1.)/2.)*ME*ME)**(GS/(GS-1.))                          LPR03430
      CF=(2/(1.+COS(ALN)))*CFB/0.96                                     LPR03440
      ALN=ATAN((DEB-DTS)/(2.*NLB))                                      LPR03450
      CFS=CF                                                            LPR03460
      CFA=CF                                                            LPR03470
      IFPCS=0                                                           LPR03480
      IFAB2=2                                                           LPR03490
      IFBRS=0                                                           LPR03500
C                                                                       LPR03510
  250 PCS=2.*FS/(CFS*ATS*0.96*(1.+COS(ALN)))                            LPR03520
      IF (PCS.GT.125.) GO TO 251                                        LPR03530
      IFAB2=0                                                           LPR03540
      IFBSTO=1                                                          LPR03550
      IFPCS=2                                                           LPR03560
      GO TO 999                                                         LPR03570
  251 PES=PCS/PR                                                        LPR03580
      CFS=((PES-PAS)/PCS)*AR+CF                                         LPR03590
      IF (ABS(CFS-CFA).LT.0.001) GO TO 260                              LPR03600
      CFA=CFS                                                           LPR03610
      GO TO 250                                                         LPR03620
C                                                                       LPR03630
  260 CONTINUE                                                          LPR03640
      CVS=(ISPS*G)/CFS                                                  LPR03650
      BRS=0.45                                                          LPR03660
C                                                                       LPR03670
  270 AB2=DM/(DENSS*BRS)                                                LPR03680
      IF (AB2.LT.AM) GO TO 280                                          LPR03690
      WTS=BRS*TBS                                                       LPR03700
      IF (WTS.LT.(0.43*D)) GO TO 275                                    LPR03710
      BRS=0.99*BRS                                                      LPR03720
      IFBRS=1                                                           LPR03730
      GO TO 270                                                         LPR03740
C                                                                       LPR03750
  275 W=(2.*WTS)/D                                                      LPR03760
      B=EXP(3.509*W)                                                    LPR03770
      C=(B-1.)/(B+1.)                                                   LPR03780
      APS=(1.065*C+1.)/2.                                               LPR03790
      GEA=AM-APS                                                        LPR03800
      GO TO 285                                                         LPR03810
  280 IFAB2=1                                                           LPR03820
      GEA=AM                                                            LPR03830
  285 VGS=WPS/DENSS                                                     LPR03840
      LS=VGS/GEA                                                        LPR03850
      IF (IFAB2.EQ.2) GO TO 290                                         LPR03860
      WTS=LS                                                            LPR03870
```

```
      BRS=(AB2/AM)*BRS
      AB2=AM
      IFBRS=1
C
  290 CONTINUE
      LS=LS+D/6.+.5
      WCS=2*PI*D/2.*LS*T*DENSC
      VOLS=AM*(LS-.5)-(.0025*D**4.)
      TWS=WPS+WCS
      AES=1000000000000000000.
      ATS=1000000000000000000.
      TS=T
      GO TO 998
C
C=====SUSTAINER MOTOR CALCULATIONS (STAGED MOTORS)====================
C
  300 TBS=(R-RB)/VBF
      ALTS=ALTB+LALT
      ELS=ATAN((TALT-ALTS)/(R-RB))
      CALL PRATIO (ALTSN,PAS)
      FS=DRAGS*1.1+(WL-TWB)*SIN(ELS)
C
  305 TIS=FS*TBS
      WPS=TIS/ISPS
      FN=DRAGS*1.1+(WL-TWB-WPS/2.)*SIN(ELS)
      IF (ABS(FN-FS).LT.0.01) GO TO 310
      FS=FN
      GO TO 305
C
  310 CONTINUE
      VGS=WPS/DENSS
      DM=WPS/TBS
      STEP=100.
      B=(DENSS/2.4)*((GS-1.)/(2.*GS))*(YIELDS/DENSCS)
      PCS=PAS
      PES=PAS
      AR=1.
      IFDLMS=0
      CALL PCRAT (PAS,PCS,GS,B,STEP)
      PLIM=800.
  315 CALL ARAT (PAS,PCS,GS,PES,ISPS,D,CFS,CVS,ATS,DTS,AES,DES,IFDLMS,
     +IFPCS,AR,ALN,DM,PLIM,IMOTOR)
      IF ((DES.GE.(0.95*D)).OR.(PCS.LT.250)) GO TO 320
      PLIM=PLIM*.99
      GO TO 315
C
  320 CONTINUE
      TS=(PCS*D)/(2.*YIELDS)
```

```
      AM=PI*(D/2.-TS)**2.
      BRS=0.45
      IFAB2=2
      IFBRS=0
C
  330 AB2=DM/(DENSS*BRS)
      IF (AB2.LT.AM) GO TO 340
      J=0.85
      APS=ATS/J
      GEA=AM-APS
      B=(APS/PI)**0.5
      C=D/2-B
      WTS=BRS*TBS
      IF (WTS.LT.C) GO TO 360
      BRS=0.99*BRS
      IFBRS=1
      GO TO 330
C
  340 CONTINUE
      IFAB2=1
      GEA=AM
      BRS=(AB2/AM)*BRS
      AB2=AM
C
  360 CONTINUE
      LS=VGS/GEA
      LS=LS+D/3.+0.5
      LB=LB+DB/6.
      IF (AB2.EQ.AM) WTS=LS
      WCS=2.*PI*D/2.*LS*TS*DENSCS
      VOLS=AM*(LS-0.5)-(.0025*D**4.)
      NLS=(DES-DTS)/(2.*TAN(ALN))
      TWS=WPS+WCS
      GO TO 998
C
C=====================================================================
C
  999 WPS=0.
      WCS=0
      TWS=0.
      CFS=0.
      CVS=0.
      FS=0.
      TBS=0.
      PCS=0.
      AB2=0.
      WTS=0.
      APS=0.
```

```
      BRS=0.                                                            LPR04840
      LS=0.                                                             LPR04850
      VOLS=0.                                                           LPR04860
      ATS=0.                                                            LPR04870
      AES=0.                                                            LPR04880
      NLS=0.                                                            LPR04890
      TS=0.                                                             LPR04900
  998 IF (IMOTOR.EQ.0) GO TO 910                                        LPR04910
      WNZ=(FB*TBB)*.00025                                               LPR04920
      WNZS=(FS*TBS)*.00025                                              LPR04930
      GO TO 915                                                         LPR04940
  910 WNZ=((FB*TBB)+(FS*TBS))*.00025                                    LPR04950
      WNZS=0.                                                           LPR04960
      NLS=0.                                                            LPR04970
  915 TCW=WCB+WCS                                                       LPR04980
      TPW=WPB+WPS                                                       LPR04990
      TRMW=WNZ+WNZS+TCW+TPW                                             LPR05000
      IFTRMW=0                                                          LPR05010
      IF (TRMW.GT.(0.75*WL)) IFTRMW=1                                   LPR05020
      IF (TRMW.LT.(0.25*WL)) IFTRMW=1                                   LPR05030
      TRML=LB+LS+NLB+NLS                                                LPR05040
      IFTRML=0                                                          LPR05050
      IF (TRML.GT.(15.*D)) IFTRML=1                                     LPR05060
      IF (IMOTOR.EQ.0) WNZS=1000000000000000000000.                     LPR05070
      IF (IMOTOR.EQ.0) NLS=100000000000000000000.                       LPR05080
      ALN=ATAN((DEB-DTB)/(2.*NLB))                                      LPR05090
      ALN=RATODE(ALN)                                                   LPR05100
      ELB=RATODE(ELB)                                                   LPR05110
      R=R/6080.                                                         LPR05120
      CALL OUTPUT                                                       LPR05130
C                                                                       LPR05140
      WRITE (6,800)                                                     LPR05150
  800 FORMAT (1X,'DO YOU WANT TO REPEAT THIS PROBLEM? (0=NO,1=YES)')    LPR05160
      READ (5,1120) IRPT                                                LPR05170
      IF (IRPT.EQ.1) GO TO 1201                                         LPR05180
      WRITE (6,810)                                                     LPR05190
  810 FORMAT (1X,'DO YOU WANT TO RUN A NEW PROBLEM? (0=NO,1=YES)')      LPR05200
      READ (5,1120) IRPT                                                LPR05210
      IF (IRPT.EQ.1) GO TO 997                                          LPR05220
      CALL SCREEN                                                       LPR05230
      STOP                                                              LPR05240
      END                                                               LPR05250
C                                                                       LPR05260
C                                                                       LPR05270
      SUBROUTINE PRATIO (ALT,PA)                                        LPR05280
      REAL ALT,PA,PR                                                    LPR05290
      IF (ALT.GT.36089.) GO TO 10                                       LPR05300
      PR=(1.-(6.8754E-6*ALT))**5.256                                    LPR05310
```

```
      GO TO 20
   10 PR=.2234*EXP(-4.806E-5*(ALT-36089.))
   20 PA=PR*14.696
      RETURN
      END
C
C
      SUBROUTINE PCRAT (PA,PC,G,B,STEP)
      REAL PA,PC,G,B,STEP
   10 C=((PA/PC)**((1.-G)/G)-1.)*PC
      IF (ABS(B-C).LT.0.01) GO TO 30
      IF (B.LT.C) GO TO 20
      PC=PC+STEP
      GO TO 10
   20 CONTINUE
      PC=PC-STEP
      STEP=STEP/10.
      PC=PC+STEP
      GO TO 10
   30 CONTINUE
      RETURN
      END
C
C
      SUBROUTINE ARAT (PA,PC,G,PE,ISP,D,CF,CV,AT,DT,AE,DE,IFDLM,IFPC,AR,
     +ALN,DM,PLIM,IMOTOR)
      REAL PA,PC,G,PE,ISP,D,CF,CV,AT,DT,AE,DE,B,C,ME,ALN,DM,AR,PLIM
      INTEGER IFDLM,IFPC,IMOTOR
      IFPC=0
   10 B=1.-(PA/PC)**((G-1.)/G)
      C=2.*(G*G/(G-1.))*(2./(G+1.))**((G+1.)/(G-1.))
      CF=(0.96*((1.+COS(ALN))/2.)*(B*C)**0.5)+((PE-PA)*AR/PC)
      CV=(ISP*32.174)/CF
      AT=(CV*DM)/(PC*32.174)
      DT=2*(AT/3.1415927)**0.5
      ME=((2./(G-1.))*((PC/PE)**((G-1.)/G)-1.))**0.5
      AE1=1.+((G-1.)/2.)*ME*ME
      AE=((1./ME)*((2./(G+1.))*AE1)**((G+1.)/(2.*(G-1.))))*AT
      DE=2.*(AE/3.1415927)**0.5
      IF ((DE.LE.D).AND.(IMOTOR.EQ.1)) GO TO 30
      IF (((PE/PA).LT.0.5).OR.((PE/PA).GT.2.2)) GO TO 30
      IFDLM=1
      PC=PC/.999
      IF (PC.GT.PLIM) GO TO 20
      GO TO 10
   20 PC=PC*.999
      PE=PE/.999
      IF (((PE/PA).GT.0.4).AND.((PE/PA).LT.2.5)) GO TO 25
```

```
      AE=(D/2.)**2.*3.1415927
   25 AR=AE/AT
      IFPC=1
      GO TO 10
   30 CONTINUE
      IF (PC.LT.PLIM) GO TO 40
      PC=PC*.999
      GO TO 10
   40 RETURN
      END
C
C
      SUBROUTINE SCREEN
      WRITE (6,600)
  600 FORMAT (1X,'CLEAR SCREEN AND ENTER "0"')
      READ (5,16) ISCR
   16 FORMAT (1I1)
      RETURN
      END
C
C
      SUBROUTINE OUTPUT
      REAL WPB,WCB,TWB,CFB,CVB,FB,TBB,RB,PCB,ABB,WTB,APB,LB,VOLB,ATB,BRB
      REAL WPS,WCS,TWS,CFS,CVS,FS,TBS,PCS,ABS,WTS,APS,LS,VOLS,ATS,BRS
      REAL NLB,AEB,WNZ,TPW,TCW,TRMW,TRML,T,AES
      REAL VBF,VBI,YIELD,DENSC,ALN,ER,D,R,WL,A,ELB,ISPB
      REAL DENSB,GB,LALT,DRAGS,TALT,ISPS,GS,DENSS,DB
      REAL YIELDS,DENSCS,TS,WNZS,NLS,ALTBN,ALTSN
      INTEGER IFBRB,IFDLMB,IFAB2,IFBRS,IFPCB,IFTRMW,IFBSTO,IFTRML,IFPCS
      INTEGER IMOTOR
      COMMON WPB,WCB,TWB,CFB,CVB,FB,TBB,RB,PCB,ABB,WTB,APB,LB,VOLB
      COMMON ATB,BRB,NLB,AEB,WNZ,TPW,TCW,TRMW,TRML,T
      COMMON WPS,WCS,TWS,CFS,CVS,FS,TBS,PCS,AB2,WTS,APS,LS,VOLS,ATS,BRS
      COMMON VBF,VBI,YIELD,DENSC,ALN,ER,D,R,WL,A,ELB,ISPB
      COMMON DENSB,GB,LALT,DRAGS,TALT,ISPS,GS,DENSS,DB
      COMMON YIELDS,DENSCS,TS,WNZS,NLS,AES,ALTBN,ALTSN
      COMMON IFBRB,IFDLMB,IFAB2,IFBRS,IFPCB,IFTRMW,IFBSTO,IFTRML,IFPCS
      COMMON IFDLMS,IMOTOR
      WRITE (6,1000)
 1000 FORMAT (35X,'BOOSTER',T55,'SUSTAINER')
      WRITE (6,1010) WPB,WPS,WCB,WCS,TWB,TWS,CFB,CFS,CVB,CVS,FB,FS
 1010 FORMAT (1X,'PROPELLANT WEIGHT',T35,F7.2,' LBS',T55,F7.2,' LBS'/,
     +1X,'CASING WEIGHT',T35,F7.2,' LBS',T55,F7.2,' LBS'/,
     +1X,'TOTAL WEIGHT',T35,F7.2,' LBS',T55,F7.2,' LBS'/,
     +1X,'THRUST COEFFICIENT',T35,F7.4,T55,F7.4/,
     +1X,'CHARACTERISTIC VELOCITY',T35,F7.1,' FT/SEC',T55,F7.1,' FT/SEC'
     +/,1X,'THRUST',T35,F7.1,' LBS',T55,F7.1,' LBS')
      WRITE (6,1020) TBB,TBS,RB,PCB,PCS,ABB,AB2,WTB,WTS,APB,APS
```

```
1020 FORMAT (1X,'BURN TIME',T35,F7.3,' SEC',T55,F7.3,' SEC'/,          LPRO6280
    +1X,'HORIZONTAL BURNOUT RANGE',T35,F7.0,' FEET'/,                   LPRO6290
    +1X,'CHAMBER PRESSURE',T35,F7.2,' PSI',T55,F7.2,' PSI'/,            LPRO6300
    +1X,'GRAIN BURN AREA',T34,F8.3,' SQ.IN',T54,F8.3,' SQ.IN'/,        LPRO6310
    +1X,'GRAIN WEB THICKNESS',T35,F7.3,' IN',T55,F7.3,' IN'/,          LPRO6320
    +1X,'GRAIN PORT AREA',T35,F7.3,' SQ.IN',T55,F7.3,' SQ.IN')         LPRO6330
     WRITE (6,1030) BRB,BRS,LB,LS,VOLB,VOLS,ATB,ATS,AEB,AES,NLB,NLS    LPRO6340
1030 FORMAT (1X,'REQUIRED BURN RATE',T35,F7.3,' IN/SEC',T55,F7.3,' IN/SLPRO6350
    +EC'/,1X,'MOTOR CASE LENGTH',T35,F7.2,' IN',T55,F7.2,' IN'/,       LPRO6360
    +1X,'MOTOR CASE VOLUME',T33,F9.2,' CU.IN',T53,F9.2,' CU.IN'/,      LPRO6370
    +1X,'NOZZLE THROAT AREA',T35,F7.3,' SQ.IN',T55,F7.3,' SQ.IN'/,     LPRO6380
    +1X,'NOZZLE EXIT AREA',T35,F7.3,' SQ.IN',T55,F7.3,' SQ.IN'/,       LPRO6390
    +1X,'NOZZLE LENGTH',T35,F7.3,' IN',T55,F7.3,' IN')                 LPRO6400
     WRITE (6,1040) WNZ,WNZS,T,TS,TCW,TPW,TRMW,TRML                    LPRO6410
1040 FORMAT (1X,'NOZZLE WEIGHT',T35,F7.2,' LBS',T55,F7.2,' LBS'/,      LPRO6420
    +1X,'CASE THICKNESS',T35,F7.5,' IN',T55,F7.5,' IN'/,               LPRO6430
    +1X,'TOTAL CASE WEIGHT',T45,F7.2,' LBS'/,                          LPRO6440
    +1X,'TOTAL PROPELLANT WEIGHT',T45,F7.2,' LBS'/,                    LPRO6450
    +1X,'TOTAL ROCKET MOTOR WEIGHT',T45,F7.2,' LBS'/,                  LPRO6460
    +1X,'TOTAL ROCKET MOTOR LENGTH',T45,F7.2,' IN')                    LPRO6470
     IF (IFBSTO.EQ.0) GO TO 1209                                        LPRO6480
     IF (IFPCS.EQ.0) WRITE (6,1205)                                     LPRO6490
1205 FORMAT (/1X,'SUSTAINER NOT CALCULATED SINCE THE BOOSTER BURNOUT RALPRO6500
    +NGE EXCEEDS'/,1X,'THE DESIGN RANGE.')                              LPRO6510
     IF (IFPCS.EQ.2) WRITE (6,1206)                                     LPRO6520
1206 FORMAT (/1X,'SUSTAINER NOT CALCULATED; THE BOOSTER NOZZLE DESIGN PLPRO6530
    +REVENTS'/,1X,'SUSTAINER OPERATION.  RECOMMEND STAGING OR INDEPENDELPRO6540
    +NT NOZZLES.')                                                      LPRO6550
1209 IF (IFBRB.EQ.1) WRITE (6,1210)                                     LPRO6560
1210 FORMAT (/1X,'BOOSTER BURN RATE WAS ADJUSTED TO CORRESPOND WITH THELPRO6570
    + WEB THICKNESS.')                                                  LPRO6580
     IF (IFDLMB.EQ.1) WRITE (6,1220)                                    LPRO6590
1220 FORMAT (/1X,'THE BOOSTER NOZZLE DESIGN WAS RESTRICTED DUE TO THE MLPRO6600
    +ISSILE DIAMETER.')                                                 LPRO6610
     IF (IFPCB.EQ.1) WRITE (6,1230)                                     LPRO6620
1230 FORMAT (/1X,'BOOSTER NOZZLE DESIGN IS NOT OPTIMUM DUE TO EXCESSIVELPRO6630
    + BOOSTER CHAMBER'/,1X,'PRESSURES.')                                LPRO6640
     IF (IFDLMS.EQ.1) WRITE (6,1233)                                    LPRO6650
1233 FORMAT (/1X,'THE SUSTAINER NOZZLE DESIGN WAS RESTRICTED DUE TO THELPRO6660
    + MISSILE DIAMETER.')                                               LPRO6670
     IF (IFPCS.EQ.1) WRITE (6,1235)                                     LPRO6680
1235 FORMAT (/1X,'SUSTAINER NOZZLE DESIGN IS NOT OPTIMUM DUE TO EXCESSILPRO6690
    +VE SUSTAINER'/,1X,'CHAMBER PRESSURES.')                            LPRO6700
     IF (IFAB2.EQ.2) WRITE (6,1241)                                     LPRO6710
1241 FORMAT (/1X,'THE SUSTAINER MOTOR HAS A CORE-BURNING GRAIN.')      LPRO6720
     IF (IFBRS.EQ.1) WRITE (6,1245)                                     LPRO6730
1245 FORMAT (/1X,'SUSTAINER BURN RATE WAS ADJUSTED TO CORRESPOND WITH TLPRO6740
    +HE WEB THICKNESS.')                                                LPRO6750
```

```
      IF (IFAB2.EQ.1) WRITE (6,1250)                                    LPR06760
 1250 FORMAT (/1X,'THE SUSTAINER MOTOR HAS AN END-BURNING GRAIN.')      LPR06770
      IF (IFTRMW.EQ.1) WRITE (6,1260)                                   LPR06780
 1260 FORMAT (/1X,'REESTIMATION OF LAUNCH WEIGHT IS REQUIRED FOR THESE MLPR06790
     +ISSILE PERFORMANCE'/,1X,'PARAMETERS.')                            LPR06800
      IF (IFTRML.EQ.1) WRITE (6,1270)                                   LPR06810
 1270 FORMAT (/1X,'ENLARGEMENT OF DIAMETER IS RECOMMENDED DUE TO A VERY LPR06820
     +HIGH LENGTH-TO-DIAMETER'/,1X,'RATIO FOR THE MOTOR.')              LPR06830
      WRITE (6,1300)                                                    LPR06840
 1300 FORMAT (/1X,'DO YOU WANT TO HAVE THIS AS HARDCOPY OUTPUT? (0=NO,1=LPR06850
     +YES)')                                                            LPR06860
      READ (5,1310) ICOPY                                               LPR06870
 1310 FORMAT (I1)                                                       LPR06880
      IF (ICOPY.EQ.0) GO TO 1500                                        LPR06890
                                                                        LPR06900
      IF (IMOTOR.EQ.0) WRITE (8,208)                                    LPR06910
  208 FORMAT (//1X,'INTEGRAL MOTORS (COMMON NOZZLE)')                   LPR06920
      IF (IMOTOR.EQ.1) WRITE (8,209)                                    LPR06930
  209 FORMAT (//1X,'STAGED MOTORS (INDEPENDENT NOZZLES)')               LPR06940
      WRITE (8,210) LALT,WL,VBI,ELB,A,VBF,DRAGS,R,TALT,ISPB,DENSB,GB    LPR06950
  210 FORMAT (1X,'SUMMARY OF INPUT PARAMETERS=========================',LPR06960
     +/5X,' 1) LAUNCH ALTITUDE',T45,F12.1,' FEET',                      LPR06970
     +/5X,' 2) LAUNCH WEIGHT',T45,F12.2,' POUNDS',                      LPR06980
     +/5X,' 3) LAUNCH VELOCITY',T45,F12.1,' FT/SEC',                    LPR06990
     +/5X,' 4) LAUNCH ANGLE',T45,F12.1,' DEGREES',                      LPR07000
     +/5X,' 5) AVERAGE ACCELERATION',T45,F12.2,' G"S',                  LPR07010
     +/5X,' 6) CRUISE VELOCITY',T45,F12.1,' FT/SEC',                    LPR07020
     +/5X,' 7) DRAG AT CRUISE VELOCITY',T45,F12.1,' POUNDS',            LPR07030
     +/5X,' 8) MAXIMUM RANGE',T45,F12.0,' MILES',                       LPR07040
     +/5X,' 9) FINAL (TARGET) ALTITUDE',T45,F12.1,' FEET',              LPR07050
     +/5X,'10) BOOSTER PROPELLANT SPECIFIC IMPULSE',T45,F12.1,' SEC',   LPR07060
     +/5X,'11) BOOSTER PROPELLANT DENSITY',T45,F12.4,' LBS/CU.IN',      LPR07070
     +/5X,'12) BOOSTER EXHAUST SPECIFIC HEAT RATIO',T50,F7.5)           LPR07080
      WRITE (8,220) ISPS,DENSS,GS,ALN                                   LPR07090
  220 FORMAT (5X,'13) SUSTAINER PROPELLANT SPECIFIC IMPULSE',T50,F7.1,  LPR07100
     +' SEC',                                                           LPR07110
     +/5X,'14) SUSTAINER PROPELLANT DENSITY',T45,F12.4,' LBS/CU.IN',    LPR07120
     +/5X,'15) SUSTAINER EXHAUST SPECIFIC HEAT RATIO',T50,F7.5,         LPR07130
     +/5X,'16) NOZZLE HALF ANGLE',T45,F12.2,' DEGREES')                 LPR07140
      IF (IMOTOR.EQ.1) GO TO 240                                        LPR07150
      WRITE (8,230) ALTBN,ER,DB,YIELD,DENSC                             LPR07160
  230 FORMAT (5X,'17) NOZZLE DESIGN ALTITUDE',T45,F12.1,' FEET',        LPR07170
     +/5X,'18) NOZZLE EROSION RATE',T45,F12.5,' IN/SEC',                LPR07180
     +/5X,'19) MISSILE DIAMETER',T45,F12.1,' INCHES',                   LPR07190
     +/5X,'20) YIELD STRENGTH OF CASE MATERIAL',T45,F12.1,' PSI',       LPR07200
     +/5X,'21) DENSITY OF CASE MATERIAL',T45,F12.4,' LBS/CU.IN')        LPR07210
      GO TO 249                                                         LPR07220
  240 WRITE (8,245) ALTBN,DB,YIELD,DENSC,ALTSN,D,YIELDS,DENSCS          LPR07230
```

```
245 FORMAT (5X,'17) BOOSTER DESIGN ALTITUDE',T45,F12.1,' FEET',              LPRO7240
   +/5X,'18) BOOSTER DIAMETER',T45,F12.2,' INCHES',                           LPRO7250
   +/5X,'19) YIELD STRENGTH OF BOOSTER CASE',T45,F12.1,' PSI',                LPRO7260
   +/5X,'20) DENSITY OF BOOSTER CASE MATERIAL',T45,F12.4,' LBS/CU.IN',       LPRO7270
   +/5X,'21) SUSTAINER DESIGN ALTITUDE',T45,F12.1,' FEET',                    LPRO7280
   +/5X,'22) SUSTAINER DIAMETER',T45,F12.2,' INCHES',                         LPRO7290
   +/5X,'23) YIELD STRENGTH OF SUSTAINER CASE',T45,F12.1,' PSI',              LPRO7300
   +/5X,'24) DENSITY OF SUSTAINER CASE MATERIAL',T45,F12.4,' LBS/CU.INLPRO7310
   +'/)                                                                       LPRO7320
249 CONTINUE                                                                  LPRO7330
    WRITE (8,1000)                                                            LPRO7340
    WRITE (8,1010) WPB,WPS,WCB,WCS,TWB,TWS,CFB,CFS,CVB,CVS,FB,FS              LPRO7350
    WRITE (8,1020) TBB,TBS,RB,PCB,PCS,ABB,AB2,WTB,WTS,APB,APS                 LPRO7360
    WRITE (8,1030) BRB,BRS,LB,LS,VOLB,VOLS,ATB,ATS,AEB,AES,NLB,NLS            LPRO7370
    WRITE (8,1040) WNZ,WNZS,T,TS,TCW,TPW,TRMW,TRML                            LPRO7380
    IF (IFBSTO.EQ.0) GO TO 250                                                LPRO7390
    IF (IFPCS.EQ.0) WRITE (8,1205)                                            LPRO7400
    IF (IFPCS.EQ.2) WRITE (8,1206)                                            LPRO7410
250 IF (IFBRB.EQ.1) WRITE (8,1210)                                            LPRO7420
    IF (IFDLMB.EQ.1) WRITE (8,1220)                                           LPRO7430
    IF (IFPCB.EQ.1) WRITE (8,1230)                                            LPRO7440
    IF (IFDLMS.EQ.1) WRITE (8,1233)                                           LPRO7450
    IF (IFPCS.EQ.1) WRITE (8,1235)                                            LPRO7460
    IF (IFAB2.EQ.2) WRITE (8,1241)                                            LPRO7470
    IF (IFBRS.EQ.1) WRITE (8,1245)                                            LPRO7480
    IF (IFAB2.EQ.1) WRITE (8,1250)                                            LPRO7490
    IF (IFTRMW.EQ.1) WRITE (8,1260)                                           LPRO7500
    IF (IFTRML.EQ.1) WRITE (8,1270)                                           LPRO7510
1500 CONTINUE                                                                 LPRO7520
    RETURN                                                                    LPRO7530
    END                                                                       LPRO7540
```

APPENDIX G. AERODYNAMIC COEFFICIENTS PROGRAM LISTING

This program is divided into three major subdivisions; the executive routines, the FORTRAN II/IV computational program, and the FORTRAN IV plotting program. The executive routines are used to establish the required file definitions, initiate operation of the computational program, supervise the transfer of data to the plotting program, and provide operational information to the user at appropriate times.

The computational program, LAERO1 FORTRAN, consists of five subprogram divisions. The MAIN program accepts the input data, conducts calc lations not done by other subroutines, formats the calculated data and provides the output data to the user, the printer file, and the plot program data file. Subroutine GEOSUB calculates the missile wetted area and the Reynolds number per foot based on the flight altitude. Subroutine CLASUB calculates the aerodynamic surface lift-curve slopes. Subroutine CATSUB calculates center of pressure locations, cross-flow drag coefficients, and interference factors. Subroutine SCREEN prompts the user to clear the terminal screen for proper positioning of the output.

The plot program, AEROPLOT FORTRAN, is structured for direct submission to the MVS batch reader from the 3278 terminal. No cards need to be punched or read. However, attention must be given to the correct JCL accounting data in the first line of the program; those shown in the listing are for illustrative purposes only. This program receives the data from the computational program and produces a group of six charts for each Mach number entered. The plots represent the relationships of Cl to alpha, Cm, Cn, Cd, Ca, and Cdi. The program can produce up to 24 sets of plots for a single run.

```
FILE: LAERO1   EXEC        A    NAVAL POSTGRADUATE SCHOOL

FILEDEF 08 DISK LAERO1 OUTPUT A0 (RECFM VA BLOCK 133 PERM
FILEDEF 07 DISK LAERO1 PLOT A0 (RECFM VA BLOCK 80 PERM
&BEGTYPE
YOU WILL HAVE THE OPPORTUNITY TO OBTAIN BOTH A HARDCOPY
PRINTOUT AND A SET OF PLOTS FOR ONE SET OF INPUT PARAMETERS
EACH TIME YOU ENTER THIS PROGRAM.  THE PROGRAM MAY BE
RE-RUN CONTINUOUSLY AND YOU WILL HAVE THE OPTION TO CHANGE
INPUT PARAMETERS FOR EACH SUCCESSIVE RUN, BUT YOU CAN
OBTAIN THE PRINTOUT AND PLOTS PERTAINING TO THE LAST RUN
ONLY.  IF ADDITIONAL OUTPUT IS REQUIRED, RE-ENTER THE
PROGRAM.
&END
LOAD LAERO1
START
&BEGTYPE

TO OBTAIN THE HARDCOPY PRINTOUT OF THE DATA TABLES, TYPE
AND ENTER:

                        LAERO1PR

TO OBTAIN THE VERSATEC PLOT OF THE TABULAR DATA, TYPE AND
ENTER:

                        LAERO1PL

&END
```

```
FILE:  LAERO1PR  EXEC     A    NAVAL POSTGRADUATE SCHOOL

PRINT LAERO1 OUTPUT  (LINECOUN 70
&BEGTYPE
THE OUTPUT WILL BE PRINTED ON THE VM PRINTER IN ROOM 140
AND WILL BE IDENTIFIED WITH YOUR USER NUMBER AND LAST NAME.
IT USUALLY REQUIRES SEVERAL MINUTES TO OBTAIN THE PRINTOUT.
&END
```

```
FILE:  LAERO1PL EXEC     A    NAVAL POSTGRADUATE SCHOOL

COPY LAERO1 PLOT A LAERO  PLOTDATA A PLOT LAERO1 A
EXEC SUBMIT PLOT LAERO1 A
ERASE PLOT LAERO1 A
&BEGTYPE
THE PLOT WILL BE DRAWN IN THE COMPUTER ROOM AND PLACED OVER
THE ALPHABETIZED OUTPUT FILE IN ROOM 140.  IT WILL BE
IDENTIFIED BY "AEROFILE" AND USUALLY REQUIRES MANY MINUTES
(HOURS, DAYS) TO OBTAIN.
&END
```

```fortran
C     AERO FORTRAN
C     26 MAY 1981
C     PLOTTING OUTPUT FACILITY INSTALLED
C
C     AERO1 CONSISTS OF A MAIN PROGRAM AND THREE SUBROUTINES.  THE INPUTS
C     ARE THE GEOMETRIC CHARACTERISTICS OF THE MISSILE, THE FLIGHT COND-
C     ITIONS, ENGINE AND INLET TYPE AND THE PROTUBERANCE DRAG.  THE OUTPUT
C     CONSISTS OF THE AERODYNAMIC COEFFICIENTS FOR LIFT, DRAG AND MOMENT.
C     THE COMPONENT CONTRIBUTION TO THESE COEFFICIENTS, AND THE COMPONENT
C     AND OVERALL CENTERS OF PRESSURE ARE ALSO DETERMINED.
C
C     AERO1 IS A MODIFICATION BY LT. D. R. REDMON OF THE PROGRAM AEROCF
C     WHICH WAS DEVELOPED AT THE NAVAL AIR DEVELOPEMENT CENTER BY
C     MR. F. A. KUSTER, JR.  THE PROGRAM IS ESSENTIALLY AS WRITTEN EXCEPT
C     FOR CALCULATION OF PLANFORM AREAS AND CENTROIDS.  IT WAS FURTHER
C     ADAPTED BY LT. M. D. SULLIVAN FOR USE AS AN INTERACTIVE PROGRAM ON
C     THE NAVAL POSTGRADUATE SCHOOL IBM 3033 VM/370 SYSTEM.
C
C     THE MAIN PROGRAM ASSUMES THE CONTROL SURFACES IS THE TAIL.  THIS IS
C     REGARDLESS OF THE METHOD OF CONTROL (WING, TAIL, CANARD).  BECAUSE OF
C     THIS, CARE MUST BE TAKEN TO INPUT THE "RIGHT" DATA FOR THE TAIL.  FOR
C     INSTANCE, IF THE MISSILE IS A WING-CONTROL MISSILE, THE WING DATA
C     MUST BE INPUT FOR THE TAIL AND THE TAIL DATA INPUT FOR THE WING.
C
      DIMENSION XVXM(24),XDT(24),XAL(48)
      DIMENSION CLM(24,24,24),CDM(24,24,24),CMM(24,24,24),CAM(24,24,24)
      DIMENSION CNM(24,24,24),CDIM(24,24,24)
      COMMON REFTI,REFT,SSUBS,XLOB,ZF,VXM,IZZY,LLKK,LLLL,NSURF
      COMMON ICSC, INOSE,ISWPW,IAFBW,ISWPW2,IAFBW2,ISWPT,IAFBT,IDUM1,
     *CLAMT,CLAMW,CLAMW2,D,XLNOSE,BT,BW,BW2,CROOTW,CTOOTT,CTOOW2,
     *ST,SW,SW2,TOVCW,TOVCW2,TOVCT,XLAMT,XLAMW,XLAMW2,
     *XMACW,XMACW2,XMACT,XWING,XWING2,XTAIL,HT,XL,AREA,XREF,
     *ISWEPW,ISWEPT,ISWEP2,NWING,NTAIL,NWING2,ENGINE,ENLET,BETA,DBASE,
     *DJET,XLABOD
      COMMON ART,ARW,ARW2,BAR,BCOLAM,BETA1,B1,CLALT,CLALW,CLALW2,CLAL1,
     *COLAM,CROOT,IAFB,ISWP,TOVC,ODC,XBCRBW,XBCRWB,XKBT,XKBW,XKTB,XKTBT,
     *XKWB,XKWBI,ISWP1,RATIO,XLAM1,XMAC,D1
      CDW2=0.0
      CDOW2=0.0
      IL = 0
      IABC=0
      ISWPW2=0
      IAFBW2=0
      XLAMW2=.0
      CLAMW2=.0
      BW2=.0
      CROOW2=.0
      SW2=.0
```

```
      XMACW2=.0
      XWING2=.0
      ISWEP2=0
      NWING2=0
C+++++++++++++++++++++++++++++++++++++++++++++
C  THE FOLLOWING INSTRUCTIONS ARE FOR THE PLOT CAPABILITY
      CLMAX=0.
      CLMIN=0.
      DO 4000 I=1,24
        XAL(I)=000.
        XDT(I)=000.
        XVXM(I)=000.
        DO 4000 J=1,24
          DO 4000 K=1,24
            CLM(I,J,K)=000.
            CDM(I,J,K)=000.
            CMM(I,J,K)=000.
            CAM(I,J,K)=000.
            CNM(I,J,K)=000.
            CDIM(I,J,K)=000.
 4000       CONTINUE
C+++++++++++++++++++++++++++++++++++++++++++++
 1000 FORMAT(2I5,7F10.5)
 1010 FORMAT(6I5)
 1020 FORMAT(15A4)
 1030 FORMAT(3F10.3,6I2)
 1040 WRITE (6,1050)
 1050 FORMAT(1X,'INPUT TITLE')
      READ 1020, TITL1,TITL2,TITL3,TITL4,TITL5,TITL6,TITL7,TITL8,TITL9,
     *TITLO,TITLO1,TITLO2,TITLO3,TITLO4,TITLO5
 1060 FORMAT (I2)
 1070 FORMAT (F10.3)
 1080 WRITE (6,1090)
 1090 FORMAT(/,1X,'NOTE                '/,
     *1X,'UNLESS OTHERWISE INDICATED, INPUT THE FOLLOWING DATA AS'/,
     *1X,'TWO-DIGIT INTEGERS; I.E., THREE=03.'/)
60001 WRITE (6,10010)
10010 FORMAT(1X,'INPUT CONTROL CONSTANT, 01=TAIL CONTROL, 02=WING CONTRO
     *L, 03=CANARD CONTROL')
      READ 1060,ICSC
      IF(IABC.EQ.1)GO TO 1100
60002 WRITE (6,10020)
10020 FORMAT(1X,'INPUT NOSE SHAPE, 01=ELLIPSOID, 02=OGIVE, 03=CONE')
      READ 1060,INOSE
      IF(IABC.EQ.1)GO TO 1100
60003 WRITE (6,10030)
10030 FORMAT(1X,'INPUT NUMBER OF CONTROL DEFLECTIONS (LESS THAN 25)')
      READ 1060,IDT
```

```
      DO 10000 I=1,IDT                                                  LAE01000
      WRITE (6,10450)                                                   LAE01010
10450 FORMAT(1X,'****INPUT CONTROL DEFLECTION (DECIMAL NUMBER)')        LAE01020
10000 READ 1070, XDT(I)                                                 LAE01030
      IF(IABC.EQ.1)GO TO 1100                                           LAE01040
60004 WRITE (6,10040)                                                   LAE01050
10040 FORMAT(1X,'INPUT NUMBER OF MACH NUMBERS (LESS THAN 25)')          LAE01060
      READ 1060,IM                                                      LAE01070
      DO 20000 I=1,IM                                                   LAE01080
      WRITE (6,10460)                                                   LAE01090
10460 FORMAT(1X,'****INPUT MACH NUMBER (DECIMAL NUMBER)')               LAE01100
20000 READ 1070, XVXM(I)                                                LAE01110
      IF(IABC.EQ.1)GO TO 1100                                           LAE01120
60005 WRITE (6,10050)                                                   LAE01130
10050 FORMAT(1X,'INPUT NUMBER OF ANGLES OF ATTACK (LESS THAN 25)')      LAE01140
      READ 1060,IAL                                                     LAE01150
      ICL=IAL                                                           LAE01160
      DO 30000 I=1,IAL                                                  LAE01170
      WRITE (6,10470)                                                   LAE01180
10470 FORMAT(1X,'****INPUT ANGLE OF ATTACK (DECIMAL NUMBER)')           LAE01190
30000 READ 1070, XAL(I)                                                 LAE01200
      IF(IABC.EQ.1)GO TO 1100                                           LAE01210
60006 WRITE (6,10060)                                                   LAE01220
10060 FORMAT(1X,'INPUT NUMBER OF CONFIGURATIONS')                       LAE01230
      READ 1060,NBODY                                                   LAE01240
      IF(IABC.EQ.1)GO TO 1100                                           LAE01250
60007 WRITE (6,10070)                                                   LAE01260
10070 FORMAT(1X,'INPUT, 01=NON-DELTA WING, 02=DELTA WING')              LAE01270
      READ 1060, ISWPW                                                  LAE01280
      IF(IABC.EQ.1)GO TO 1100                                           LAE01290
60008 WRITE (6,10080)                                                   LAE01300
10080 FORMAT(1X,'00=NO BODY AFTER WING, 01=BODY AFTER WING')            LAE01310
      READ 1060, IAFBW                                                  LAE01320
      IF(IABC.EQ.1)GO TO 1100                                           LAE01330
60009 WRITE (6,10090)                                                   LAE01340
10090 FORMAT(1X,'INPUT WING SWEEP CONSTANT')                            LAE01350
      WRITE (6,55001)                                                   LAE01360
55001 FORMAT(1X,'IF DELTA PLANFORM ENTER 00; OTHERWISE,'/,              LAE01370
     *1X,'00=UNSWEPT LEADING EDGE, 01=SWEPT LEADING EDGE')              LAE01380
      READ 1060,ISWEPW                                                  LAE01390
      IF(IABC.EQ.1)GO TO 1100                                           LAE01400
60010 WRITE (6,10100)                                                   LAE01410
10100 FORMAT(1X,'INPUT NUMBER OF WINGS')                                LAE01420
      READ 1060,NWING                                                   LAE01430
      IF(IABC.EQ.1)GO TO 1100                                           LAE01440
60011 WRITE (6,10110)                                                   LAE01450
10110 FORMAT(1X,'INPUT, 01=NON-DELTA TAIL, 02=DELTA TAIL')              LAE01460
      READ 1060, ISWPT                                                  LAE01470
```

```
      IF(IABC.EQ.1)GO TO 1100
60012 WRITE (6,10120)
10120 FORMAT(1X,'00=NO BODY AFTER TAIL, 01=BODY AFTER TAIL')
      READ 1060,IAFBT
      IF(IABC.EQ.1)GO TO 1100
60013 WRITE (6,10130)
10130 FORMAT(1X,'INPUT TAIL SWEEP CONSTANT')
      WRITE (6,55001)
      READ 1060, ISWEPT
      IF(IABC.EQ.1)GO TO 1100
60014 WRITE (6,10140)
10140 FORMAT(1X,'INPUT NUMBER OF TAILS')
      READ 1060, NTAIL
      IF(IABC.EQ.1)GO TO 1100
90002 WRITE (6,99998)
99998 FORMAT(/,1X,'NOTE                '/
     *1X,'INPUT THE FOLLOWING DATA AS DECIMAL NUMBERS '/)
60015 WRITE (6,10150)
10150 FORMAT(1X,'INPUT TIP-TO-CHORD RATIO OF WING')
      READ 1070,XLAMW
      IF(IABC.EQ.1)GO TO 1100
60016 WRITE (6,10160)
10160 FORMAT(1X,'INPUT LEADING EDGE SWEEP OF WING,(DEGS)')
      READ 1070, CLAMW
      IF(IABC.EQ.1)GO TO 1100
60017 WRITE (6,10170)
10170 FORMAT(1X,'INPUT WING SPAN, INCLUDE BODY')
      READ 1070, BW
      IF(IABC.EQ.1)GO TO 1100
60018 WRITE (6,10180)
10180 FORMAT(1X,'INPUT WING ROOT CHORD (AT BODY JUNCTION)')
      READ 1070, CROOTW
      IF(IABC.EQ.1)GO TO 1100
60019 WRITE (6,10190)
10190 FORMAT(1X,'INPUT EXPOSED WING AREA (2 PANELS)')
      READ 1070,SW
      IF(IABC.EQ.1)GO TO 1100
60020 WRITE (6,10200)
10200 FORMAT(1X,'INPUT WING MEAN GEOMETRIC CHORD')
      READ 1070, XMACW
      IF(IABC.EQ.1)GO TO 1100
60021 WRITE (6,10210)
10210 FORMAT(1X,'INPUT DISTANCE FROM NOSE TO WING LEADING EDGE')
      READ 1070, XWING
      IF(IABC.EQ.1)GO TO 1100
60022 WRITE (6,10220)
10220 FORMAT(1X,'INPUT WING THICKNESS-TO-CHORD RATIO')
      READ 1070, TOVCW
```

```
      IF(IABC.EQ.1)GO TO 1100
60023 WRITE (6,10230)
10230 FORMAT(1X,'INPUT TIP-TO-ROOT CHORD RATIO OF TAIL')
      READ 1070, XLAMT
      IF(IABC.EQ.1)GO TO 1100
60024 WRITE (6,10240)
10240 FORMAT( 1X,'INPUT TAIL LEADING EDGE SWEEP (DEGS)')
      READ 1070, CLAMT
      IF(IABC.EQ.1)GO TO 1100
60025 WRITE (6,10250)
10250 FORMAT(1X,'INPUT TAIL SPAN, INCLUDING BODY')
      READ 1070,BT
      IF(IABC.EQ.1)GO TO 1100
60026 WRITE (6,10260)
10260 FORMAT( 1X,'INPUT TAIL ROOT CHORD')
      READ 1070, CROOTT
      IF(IABC.EQ.1)GO TO 1100
60027 WRITE (6,10270)
10270 FORMAT(1X,'INPUT EXPOSED TAIL AREA (2 PANELS)')
      READ 1070,ST
      IF(IABC.EQ.1)GO TO 1100
60028 WRITE (6,10280)
10280 FORMAT( 1X,'INPUT TAIL MEAN GEOMETRIC CHORD')
      READ 1070, XMACT
      IF(IABC.EQ.1)GO TO 1100
60029 WRITE ( 6,10290)
10290 FORMAT(1X,'INPUT DISTANCE FROM NOSE TO TAIL LEADING EDGE')
      READ 1070,XTAIL
      IF(IABC.EQ.1)GO TO 1100
60030 WRITE (6,10300)
10300 FORMAT( 1X,'INPUT TAIL THICKNESS-TC-CHORD RATIO')
      READ 1070, TOVCT
      IF(IABC.EQ.1)GO TO 1100
60031 WRITE (6,10310)
10310 FORMAT(1X,'INPUT ALTITUDE')
      READ 1070, HT
      IF(IABC.EQ.1)GO TO 1100
60032 WRITE (6,10320)
10320 FORMAT( 1X,'INPUT BODY DIAMETER')
      READ 1070, D
      IF(IABC.EQ.1)GO TO 1100
60033 WRITE (6,10330)
10330 FORMAT(1X,'INPUT MISSILE LENGTH')
      READ 1070,XL
      IF(IABC.EQ.1)GO TO 1100
60034 WRITE (6,10340)
10340 FORMAT( 1X,'INPUT NOSE LENGTH')
      READ 1070,XLNOSE
```

```
      IF(IABC.EQ.1)GO TO 1100                                          LAE02440
60035 WRITE (6,10350)                                                  LAE02450
10350 FORMAT(1X,'INPUT CG LOCATION (FROM NOSE)')                       LAE02460
      READ 1070, XCG                                                   LAE02470
      IF(IABC.EQ.1)GO TO 1100                                          LAE02480
60036 WRITE (6,10360)                                                  LAE02490
10360 FORMAT( 1X,'INPUT REFERENCE AREA')                               LAE02500
      READ 1070, AREA                                                  LAE02510
      IF(IABC.EQ.1)GO TO 1100                                          LAE02520
60037 WRITE (6,10370)                                                  LAE02530
10370 FORMAT(1X,'INPUT REFERENCE LENGTH')                              LAE02540
      READ 1070, XREF                                                  LAE02550
      IF(IABC.EQ.1)GO TO 1100                                          LAE02560
60038 WRITE (6,10380)                                                  LAE02570
10380 FORMAT( 1X,'ENGINE CODE, 0.0=TURBOFAN, 1.0=ROCKET')              LAE02580
      READ 1070, ENGINE                                                LAE02590
      IF(IABC.EQ.1)GO TO 1100                                          LAE02600
60039 WRITE (6,10390)                                                  LAE02610
10390 FORMAT(1X,'INLET CODE, 0.0=FLUSH, 1.0=EXTENDED')                 LAE02620
      READ 1070, ENLET                                                 LAE02630
      IF(IABC.EQ.1)GO TO 1100                                          LAE02640
60040 WRITE (6,10400)                                                  LAE02650
10400 FORMAT( 1X,'INPUT BOATTAIL ANGLE (DEGS)')                        LAE02660
      READ 1070, BETA                                                  LAE02670
      IF(IABC.EQ.1)GO TO 1100                                          LAE02680
60041 WRITE (6,10410)                                                  LAE02690
10410 FORMAT(1X,'INPUT BASE DIAMETER')                                 LAE02700
      READ 1070, DBASE                                                 LAE02710
      IF(IABC.EQ.1)GO TO 1100                                          LAE02720
60042 WRITE (6,10420)                                                  LAE02730
10420 FORMAT( 1X,'INPUT NOZZLE EXIT DIAMETER')                         LAE02740
      READ 1070,DJET                                                   LAE02750
      IF(IABC.EQ.1)GO TO 1100                                          LAE02760
60043 WRITE (6,10430)                                                  LAE02770
10430 FORMAT(1X,'INPUT BOATTAIL LENGTH')                               LAE02780
      READ 1070, XLABOD                                                LAE02790
      IF(IABC.EQ.1)GO TO 1100                                          LAE02800
60044 WRITE (6,10440)                                                  LAE02810
10440 FORMAT( 1X,'INPUT PROTUBERANCE DRAG')                            LAE02820
      READ 1070,CDPROT                                                 LAE02830
      IF(IABC.EQ.1)GO TO 1100                                          LAE02840
C                                                                      LAE02850
 1100 REWIND 08                                                        LAE02860
      REWIND 07                                                        LAE02870
      CALL SCREEN                                                      LAE02880
 1110 WRITE (6,1130)                                                   LAE02890
      WRITE (6,1140) TITL1,TITL2,TITL3,TITL4,TITL5,TITL6,TITL7,TITL8,  LAE02900
     *TITL9,TITL0,TITL01,TITL02,TITL03,TITL04,TITL05                   LAE02910
```

```
      WRITE (6,1120)                                                    LAE02920
 1120 FORMAT(1X,'BE SURE TO NOTE NUMBER OF ANY INCORRECT ENTRIES'//)    LAE02930
      WRITE (6,1150) ICSC,INOSE,IDT,IM,IAL,NBODY,ISWPW,IAFBW,ISWEPW,    LAE02940
     *NWING,ISWPT,IAFBT                                                 LAE02950
      WRITE (6,1160) ISWEPT,NTAIL,XLAMW,CLAMW,BW,CROOTW,SW,XMACW,       LAE02960
     *XWING,TOVCW,XLAMT,CLAMT,BT,CROOTT,ST                              LAE02970
      WRITE (6,1170) XMACT,XTAIL,TOVCT,HT,D,XL,XLNOSE,XCG,AREA,XREF,    LAE02980
     *ENGINE,ENLET,BETA,DBASE,DJET,XLABOD,CDPROT                        LAE02990
      WRITE (6,1180)(XVXM(I), I=1,24)                                   LAE03000
      WRITE (6,1190)(XDT(I), I=1,24)                                    LAE03010
      WRITE (6,1200)(XAL(I), I=1,24)                                    LAE03020
 1130 FORMAT(1X,'THE FOLLOWING TABLE CONTAINS THE INPUT DATA FOR'/)     LAE03030
 1140 FORMAT(1X,15A4//)                                                 LAE03040
 1150 FORMAT(1X,' 1) (ICSC) CONTROL CONSTANT; 1=TAIL, 2=WING, 3=CANARD:'LAE03050
     *,T63,I2/,                                                         LAE03060
     *1X,' 2) (INOSE) NOSE SHAPE; 1=ELLIPSE, 2=OGIVE, 3=CONE;',T63,I2/, LAE03070
     *1X,' 3) (IDT) NUMBER OF CONTROL DEFLECTIONS:',T63,I2/,            LAE03080
     *1X,' 4) (IM) NUMBER OF MACH NUMBERS:',T63,I2/,                    LAE03090
     *1X,' 5) (IAL) NUMBER OF ANGLES OF ATTACK:',T63,I2/,               LAE03100
     *1X,' 6) (NBODY) NUMBER OF CONFIGURATIONS:',T63,I2/,               LAE03110
     *1X,' 7) (ISWPW) 1=NON-DELTA WING, 2=DELTA WING:',T63,I2/,         LAE03120
     *1X,' 8) (IAFBW) 0=NO BODY AFTER WING, 1=BODY AFTER WING:',T63,I2/,LAE03130
     *1X,' 9) (ISWEPW) WING SWEEP CONSTANT (IF DELTA=0)'/,              LAE03140
     *15X,'0=UNSWEPT LEADING EDGE, 1=SWEPT LEADING EDGE:',T63,I2/,      LAE03150
     *1X,'10) (NWING) NUMBER OF WINGS:',T63,I2/,                        LAE03160
     *1X,'11) (ISWPT) 1=NON-DELTA TAIL, 2=DELTA TAIL:',T63,I2/,         LAE03170
     *1X,'12) (IAFBT) 0=NO BODY AFTER TAIL,1=BODY AFTER TAIL:',T63,I2)  LAE03180
 1160 FORMAT(1X,'13) (ISWEPT) TAIL SWEEP CONSTANT (IF DELTA=0)'/,       LAE03190
     *15X,'0=UNSWEPT LEADING EDGE, 1=SWEPT LEADING EDGE:',T63,I2/,      LAE03200
     *1X,'14) (NTAIL) NUMBER OF TAILS:',T63,I2/,                        LAE03210
     *1X,'15) (XLAMW) TIP-TO-CHORD RATIO OF WING:',T56,F12.3/,          LAE03220
     *1X,'16) (CLAMW) WING LEADING EDGE SWEEP(DEGREES):',T56,F12.3/,    LAE03230
     *1X,'17) (BW) WING SPAN, INCLUDING BODY:',T56,F12.3/,              LAE03240
     *1X,'18) (CROOTW) WING ROOT CHORD (AT BODY JUNCTION):',T56,F12.3/, LAE03250
     *1X,'19) (SW) EXPOSED WING AREA (TWO PANELS):',T56,F12.3/,         LAE03260
     *1X,'20) (XMACW) WING MEAN GEOMETRIC CHORD:',T56,F12.3/,           LAE03270
     *1X,'21) (XWING) DISTANCE FROM NOSE TO WING LE:',T56,F12.3/,       LAE03280
     *1X,'22) (TOVCW) WING THICKNESS TO CHORD RATIO:',T56,F12.3/,       LAE03290
     *1X,'23) (XLAMT) TIP-TO-CHORD RATIO OF TAIL:',T56,F12.3/,          LAE03300
     *1X,'24) (CLAMT) TAIL LEADING EDGE SWEEP (DEGREES):',T56,F12.3/,   LAE03310
     *1X,'25) (BT) TAIL SPAN, INCLUDING BODY:',T56,F12.3/,              LAE03320
     *1X,'26) (CROOTT) TAIL ROOT CHORD:',T56,F12.3/,                    LAE03330
     *1X,'27) (ST) EXPOSED TAIL AREA (TWO PANELS):',T56,F12.3)          LAE03340
 1170 FORMAT(1X,'28) (XMACT) TAIL MEAN GEOMETRIC CHORD:',T56,F12.3/,    LAE03350
     *1X,'29) (XTAIL) DISTANCE FROM NOSE TO TAIL LE:',T56,F12.3/,       LAE03360
     *1X,'30) (TOVCT) TAIL THICKNESS TO CHORD RATIO:',T56,F12.3/,       LAE03370
     *1X,'31) (HT) ALTITUDE:',T56,F12.3/,                               LAE03380
     *1X,'32) (D) BODY DIAMETER:',T56,F12.3/,                           LAE03390
```

```
     *1X,'33) (XL) MISSILE LENGTH:',T56,F12.3/,                         LAE03400
     *1X,'34) (XLNOSE) NOSE LENGTH:',T56,F12.3/,                        LAE03410
     *1X,'35) (XCG) DISTANCE TO CG LOCATION FROM NOSE:',T56,F12.3/,     LAE03420
     *1X,'36) (AREA) REFERENCE AREA:',T56,F12.3/,                       LAE03430
     *1X,'37) (XREF) REFERENCE LENGTH:',T56,F12.3/,                     LAE03440
     *1X,'38) (ENGINE) ENGINE: 0.0=TURBOFAN, 1.0=ROCKET:',T63,F3.1/,    LAE03450
     *1X,'39) (ENLET) INLET: 0.0=FLUSH, 1.0=EXTENDED:',T63,F3.1/,       LAE03460
     *1X,'40) (BETA) BOAT-TAIL ANGLE (DEGREES):',T56,F12.3/,            LAE03470
     *1X,'41) (DBASE) BASE DIAMETER:',T56,F12.3/,                       LAE03480
     *1X,'42) (DJET) NOZZLE EXIT DIAMETER:',T56,F12.3/,                 LAE03490
     *1X,'43) (XLABOD) BOAT-TAIL LENGTH:',T56,F12.3/,                   LAE03500
     *1X,'44) (CDPROT) PROTUBERANCE DRAG:',T56,F12.3//)                 LAE03510
 1180 FORMAT(1X,'MACH ',12F6.3/,6X,12F6.3/)                             LAE03520
 1190 FORMAT(1X,'DELTA',12F6.2/,6X,12F6.2/)                             LAE03530
 1200 FORMAT(1X,'ALPHA',12F6.2/,6X,12F6.2/)                             LAE03540
C                                                                       LAE03550
      IABC=1                                                            LAE03560
      WRITE (6,1210)                                                    LAE03570
 1210 FORMAT(1X,'IS INPUT DATA CORRECT? 00=YES, IF NO,'/                LAE03580
     *1X,'ENTER THE TWO-DIGIT NUMBER OF THE VARIABLE TO BE CHANGED.')   LAE03590
      READ 1060, IVAR                                                   LAE03600
      IF(IVAR.EQ.0)GO TO 1220                                           LAE03610
      GO TO (60001,60002,60003,60004,60005,60006,60007,60008,60009,     LAE03620
     *60010,60011,60012,60013,60014,60015,60016,60017,60018,60019,      LAE03630
     *60020,60021,60022,60023,60024,60025,60026,60027,60028,60029,      LAE03640
     *60030,60031,60032,60033,60034,60035,60036,60037,60038,60039,      LAE03650
     *60040,60041,60042,60043,60044),IVAR                               LAE03660
 1220 CONTINUE                                                          LAE03670
      WRITE(8,1130)                                                     LAE03680
      WRITE(8,1140) TITL1,TITL2,TITL3,TITL4,TITL5,TITL6,TITL7,TITL8,TITLLAE03690
     *9,TITL0,TITL01,TITL02,TITL03,TITL04,TITL05                        LAE03700
      WRITE(8,1150) ICSC,INOSE,IDT,IM,IAL,NBODY,ISWPW,IAFBW,ISWEPW,     LAE03710
     *NWING,ISWPT,IAFBT                                                 LAE03720
      WRITE(8,1160) ISWEPT,NTAIL,XLAMW,CLAMW,BW,CROOTW,SW,XMACW,        LAE03730
     *XWING,TOVCW,XLAMT,CLAMT,BT,CROOTT,ST                              LAE03740
      WRITE(8,1170) XMACT,XTAIL,TOVCT,HT,D,XL,XLNOSE,XCG,AREA,XREF,     LAE03750
     *ENGINE,ENLET,BETA,DBASE,DJET,XLABOD,CDPROT                        LAE03760
      WRITE(8,1180)(XVXM(I), I=1,24)                                    LAE03770
      WRITE(8,1190)(XDT(I), I=1,24)                                     LAE03780
      WRITE(8,1200)(XAL(I), I=1,24)                                     LAE03790
 1230 PIE=3.14159                                                       LAE03800
        IL = 1 + IL                                                     LAE03810
      LLKK=0                                                            LAE03820
      LLLL=0                                                            LAE03830
       XCG2 = XCG                                                       LAE03840
       IZZY = 0                                                         LAE03850
      IF(INOSE.EQ.3)GO TO 1250                                          LAE03860
      IF(INOSE.EQ.2)GO TO 1240                                          LAE03870
```

```
      XC=XL/2.                                                          LAE03880
      AP=XL*D                                                           LAE03890
      GO TO 1260                                                        LAE03900
 1240 XLNOD=XLNOSE/D                                                    LAE03910
      RADIUS=XLNOD**2+.25                                               LAE03920
      APOD2=XLNOD*SQRT(RADIUS**2-XLNOD**2)+RADIUS**2                    LAE03930
     **ARSIN(XLNOD/RADIUS)-2.*XLNOD*(RADIUS-.5)                         LAE03940
      APN=APOD2*D*D                                                     LAE03950
      AP=(XL-XLNOSE)*D+APN                                              LAE03960
      XCOD1=RADIUS**3-(RADIUS**2-XLNOD**2)**1.5                         LAE03970
      XCOD=XLNOD-(.6667*XCOD1-XLNOD**2*(RADIUS-.5))/APOD2               LAE03980
      XCMOVE=XCOD*D-XLNOSE/2.                                           LAE03990
      XC=XL/2.+XCMOVE                                                   LAE04000
      GO TO 1260                                                        LAE04010
 1250 APN=.5*D*XLNOSE                                                   LAE04020
      AP=APN+(XL-XLNOSE)*D                                              LAE04030
      XCN=.6667*XLNOSE                                                  LAE04040
      XCMOVE=XCN-XLNOSE/2.                                              LAE04050
      XC=XL/2.+XCMOVE                                                   LAE04060
 1260 CONTINUE                                                          LAE04070
      CALL GEOSUB                                                       LAE04080
 1270 CONTINUE                                                          LAE04090
      CALL SCREEN                                                       LAE04100
      VXM=XVXM(1)                                                       LAE04110
C     RE=REFT*VXM                                                       LAE04120
      RE=REFTI                                                          LAE04130
      SQ=2.                                                             LAE04140
      DO 3590 IJ=1,IM                                                   LAE04150
      REFT=REFTI/VXM                                                    LAE04160
      DELTA1=XDT(1)                                                     LAE04170
      DO 3580 II=1,IDT                                                  LAE04180
      ALPHA=XAL(1)                                                      LAE04190
      WRITE (6,1280) VXM                                                LAE04200
C                                                                       LAE04210
 1280 FORMAT(//,1X,'MACH ',F7.4)                                        LAE04220
      WRITE (6,1290) DELTA1                                             LAE04230
 1290 FORMAT(1X,'DELTA=',F6.2)                                          LAE04240
      WRITE (6,1300)                                                    LAE04250
 1300 FORMAT(1X,'AL',3X,'CLTOT',2X,'CDTCT',3X,'CLWP',3X,'CLBW',3X,'CLTP'LAE04260
     *,3X,'CLBT',4X,'CLB',4X,'CDI',3X,'CNWP',3X,'CNTP')                 LAE04270
      WRITE (6,1310)                                                    LAE04280
 1310 FORMAT(7X,'CLTD',3X,'CDTD',5X,'CN',5X,'CA',3X,'XCPW',3X,'XCPT',3X,LAE04290
     *'XCPB',4X,'XCP',5X,'CM')                                          LAE04300
      WRITE(8,1280) VXM                                                 LAE04310
      WRITE(8,1290) DELTA1                                              LAE04320
      WRITE(8,1320)                                                     LAE04330
 1320 FORMAT(1X,'AL',2X,'CLTOT',1X,'CDTCT',2X,'CLWP',2X,'CLBW',2X,'CLTP'LAE04340
     *,2X,'CLBT',3X,'CLB',3X,'CDI',2X,'CNWP',2X,'CNTP',2X,'CLTD',2X,    LAE04350
```

```
     *'CDTD',4X,'CN',4X,'CA',2X,'XCPW',2X,'XCPT',2X,'XCPB',3X,'XCP',4X, LAE04360
     *'CM',/)                                                          LAE04370
C                                                                      LAE04380
 1330 DELTA=DELTA1/57.29578+.000000001                                 LAE04390
      DO 3540 J=1,IAL                                                  LAE04400
      AL=ALPHA/57.29578+.00000001                                      LAE04410
      SINAAL=SIN(AL)                                                   LAE04420
      COSAAL=COS(AL)                                                   LAE04430
    1 VXMR1=VXM                                                        LAE04440
      IZZY=IZZY+1                                                      LAE04450
      IF (IZZY -4) 1340,1340,1350                                      LAE04460
 1340 VXM=.6                                                           LAE04470
 1350 CALL CLASUB                                                      LAE04480
      IF(LLLL-1) 1360,1640,1710                                        LAE04490
 1360 IF(IZZY-4) 1380,1380,1370                                        LAE04500
 1370 CALL CATSUB                                                      LAE04510
 1380 XLAM14=ATAN((.5*(B1-D)*1./COLAM+.25*XLAM1*CROOT-.25*CROOT)/      LAE04520
     *(.5*(B1-D)))                                                     LAE04530
      RE=REFT*VXM*XMAC                                                 LAE04540
      IF (RE-1.E06) 1390,1400,1400                                     LAE04550
 1390 AA=.0835                                                         LAE04560
      XNN=-.211                                                        LAE04570
       GO TO 1450                                                      LAE04580
 1400 IF (RE-1.E07) 1410,1420,1420                                     LAE04590
 1410 AA=.052                                                          LAE04600
      XNN=-.177                                                        LAE04610
       GO TO 1450                                                      LAE04620
 1420 IF (RE-1.E08) 1430,1440,1440                                     LAE04630
 1430 AA=.0333                                                         LAE04640
      XNN=-.1488                                                       LAE04650
      GO TO 1450                                                       LAE04660
 1440 AA=.0221                                                         LAE04670
      XNN=-.127                                                        LAE04680
 1450 CF=AA*RE**XNN                                                    LAE04690
      SURF=FLOAT(NSURF)                                                LAE04700
      CDO=SURF*CF*(1.+2.*TOVC+100.*TOVC**4.)                           LAE04710
      IF (IZZY-4) 1460,1460,1470                                       LAE04720
 1460 IF (IZZY-3) 1580,1630,1670                                       LAE04730
 1470 IF (AL)  1480,1490,1490                                          LAE04740
 1480 ODC=-ODC                                                         LAE04750
 1490 LLKK=LLKK+1                                                      LAE04760
      IF (LLKK-2) 1500,1520,1540                                       LAE04770
 1500 IF (SW) 1510,1510,1540                                           LAE04780
 1510 LLKK=LLKK+1                                                      LAE04790
 1520 IF (SW2) 1530,1530,1540                                          LAE04800
 1530 LLKK=LLKK+1                                                      LAE04810
 1540 IF (LLKK-2) 1550,1600,1660                                       LAE04820
 1550 XKWBW=XKWB                                                       LAE04830
```

```
      XKBWW=XKBW                                                        LAE04840
      IF(ISWEPW.EQ.0) GO TO 1560                                        LAE04850
      SHIFT=TAN(CLAMW)*(BW-D)/4.0                                       LAE04860
      GO TO 1570                                                        LAE04870
 1560 SHIFT=0.0                                                         LAE04880
 1570 XCPWB=XWING+XBCRWB*CROOT+SHIFT                                    LAE04890
      XCPBW=XWING+XBCRBW*CROOT                                          LAE04900
      ODCW=ODC                                                          LAE04910
      CLW=SIN(AL)*(XKWBW+XKBWW)*CLALW*SW*COS(AL)/AREA                   LAE04920
      CLWB=SIN(AL)*XKWBW*CLALW*SW*COS(AL)/AREA                          LAE04930
      CLBW=CLW-CLWB                                                     LAE04940
      CLVISW=    (SIN(AL)*SIN(AL)*SW*COS(AL)/AREA)*ODCW                 LAE04950
      CLW=CLW+CLVISW                                                    LAE04960
      CLWP=CLWB+CLVISW                                                  LAE04970
 1580 CDOW=CDO*(SW)/AREA                                                LAE04980
      XLAMW4=XLAM14                                                     LAE04990
      TOVCW=TOVC                                                        LAE05000
      SWTOT=SW                                                          LAE05010
      IZZY=IZZY+1                                                       LAE05020
      IF(SW2) 1640,1640,1590                                            LAE05030
 1590 COLAM=COS(CLAMW2)/SIN(CLAMW2)                                     LAE05040
      CDOW2=0.                                                          LAE05050
      BCOLAM=BETA1*COLAM                                                LAE05060
      CROOT=CROOW2                                                      LAE05070
      B1=BW2                                                            LAE05080
      IAFB=IAFBW2                                                       LAE05090
      CLAL1=CLALW2                                                      LAE05100
      XLAM1=XLAMW2                                                      LAE05110
      TOVC=TOVCW2                                                       LAE05120
      XMAC=XMACW2                                                       LAE05130
      ISWP1=ISWPW2                                                      LAE05140
      BAR=BETA1*ARW2                                                    LAE05150
      RATIO=CROOT/(BETA1*D)                                             LAE05160
      IF(IZZY-4) 1380,1380,1370                                         LAE05170
 1600 XKWBW2=XKWB                                                       LAE05180
      XKBWW2=XKBW                                                       LAE05190
      IF(ISWEP2.EQ.0) GO TO 1610                                        LAE05200
      SHIFT=TAN(CLAMW2)*(BW2-D)/4.0                                     LAE05210
      GO TO 1620                                                        LAE05220
 1610 SHIFT=0.                                                          LAE05230
 1620 XCPWB2=XWING2+XBCRWB*CROOT+SHIFT                                  LAE05240
      XCPBW2=XWING2+XBCRBW*CROOT                                        LAE05250
      ODCW2=ODC                                                         LAE05260
      CLW2=SIN(AL)*(XKWBW2+XKBWW2)*CLALW2*SW2*COS(AL)/AREA              LAE05270
      CLWB2=SIN(AL)*XKWBW2*CLALW2*SW2*COS(AL)/AREA                      LAE05280
      CLBW2=CLW2-CLWB2                                                  LAE05290
      CLVIW2=(SIN(AL)*SIN(AL)*SW2*COS(AL)/AREA)*ODCW2                   LAE05300
      CLW2=CLW2+CLVIW2                                                  LAE05310
```

```
1630 CDOW2=CDO*(SW2)/AREA                                             LAE05320
     XLAM24=XLAM14                                                    LAE05330
     SW2TOT=SW2                                                       LAE05340
1640 IZZY = IZZY +1                                                   LAE05350
     LLKK=LLKK+2                                                      LAE05360
     IF (ST) 1710,1710,1650                                           LAE05370
1650 COLAM=COS(CLAMT)/SIN(CLAMT)                                      LAE05380
     ART=(BT-D)**2/ST                                                 LAE05390
     BCOLAM=BETA1*COLAM                                               LAE05400
     CROOT=CROOTT                                                     LAE05410
     B1=BT                                                            LAE05420
     BAR=BETA1*ART                                                    LAE05430
     CLAL1=CLALT                                                      LAE05440
     IAFB=IAFBT                                                       LAE05450
     XMAC=XMACT                                                       LAE05460
     TOVC=TOVCT                                                       LAE05470
     ISWP1=ISWPT                                                      LAE05480
     XLAM1=XLAMT                                                      LAE05490
     RATIO=CROOT/(BETA1*D)                                            LAE05500
     IF (IZZY-4) 1380,1380,1370                                       LAE05510
1660 XKWBT=XKWB                                                       LAE05520
     XKBWT=XKBW                                                       LAE05530
     XCPBT=XTAIL+XBCRBW*CROOT                                         LAE05540
     ODCT=ODC                                                         LAE05550
     CLT=((XKWBT+XKBWT)*SIN(AL))*CLALT*ST*COS(AL)/AREA                LAE05560
     CLTB=SIN(AL)*XKWBT*CLALT*ST*COS(AL)/AREA                         LAE05570
     CLBT=CLT-CLTB                                                    LAE05580
     CLTD=XKTB*CLALT*SIN(DELTA)*ST*COS(AL+DELTA)/AREA                 LAE05590
     CLTDB=(XKTB+XKBT)*CLALT*SIN(DELTA)*ST*COS(AL+DELTA)/AREA         LAE05600
     CLBDT=CLTDB-CLTD                                                 LAE05610
     CLBT=CLBT+CLBDT                                                  LAE05620
     CLVIST=((SIN(AL+DELTA)*SIN(AL+DELTA))*ST*COS(AL+DELTA)/AREA)*ODCT LAE05630
     CLT=CLT+CLVIST                                                   LAE05640
     CLTP=CLTB+CLVIST+CLTD                                            LAE05650
1670 CDOT=CDO*(ST)/AREA                                               LAE05660
     CT=CDOT                                                          LAE05670
     S1TOT=ST                                                         LAE05680
     XLAMT4=XLAM14                                                    LAE05690
     IF (IZZY-4) 1900,1900,1680                                       LAE05700
1680 IF(ISWEPT.EQ.0) GO TO 1690                                       LAE05710
     SHIFT=TAN(CLAMT)*(BT-D)/4.0                                      LAE05720
     GO TO 1700                                                       LAE05730
1690 SHIFT=0.0                                                        LAE05740
1700 XCPTB=XTAIL+((XKWBT*SIN(AL)*XBCRWB+XKTB*SIN(DELTA)*XBCRWB)/      LAE05750
    *(XKWBT*SIN(AL)+XKTB*SIN(DELTA)))*CROOTT+SHIFT                    LAE05760
1710  IF (IZZY - 4) 1900, 1900, 1720                                  LAE05770
1720  XLOB = XL/D                                                     LAE05780
     ZXM=VXM*ABS(SIN(AL))                                             LAE05790
```

```
      IF (ZXM-.8) 1730,1740,1740                                        LAE05800
 1730 CDC=2.4-SQRT(1.5129-1.5129*ZXM*ZXM)                               LAE05810
      GO TO 1790                                                        LAE05820
 1740 IF (ZXM-1.15) 1750,1760,1760                                      LAE05830
 1750 CDC=1.6+SQRT(.344-(ZXM-.975)**2)                                  LAE05840
      GO TO 1790                                                        LAE05850
 1760 IF (ZXM-3.) 1770,1780,1780                                        LAE05860
 1770 CDC=1.9-SQRT(.361-.09*(ZXM-3.)**2)                                LAE05870
      GO TO 1790                                                        LAE05880
 1780 CDC=1.3                                                           LAE05890
                                                                        LAE05900
 1790 ETA=(0.0000475*(XLOB**3))-(0.00173*(XLOB**2))+(0.0298*XLOB)+0.5146LAE05910
      IF (VXM-.5) 1830,1830,1800                                        LAE05920
 1800 IF (VXM-1.4) 1810,1820,1820                                       LAE05930
 1810 ETA=ETA+(1.-ETA)*(VXM-.5)*1.111                                   LAE05940
      GO TO 1830                                                        LAE05950
 1820 ETA=1.                                                            LAE05960
 1830 IF (XLOB-10.) 1840,1850,1860                                      LAE05970
 1840 XK2K1=-0.0054*(XLOB**2)+0.104*XLOB+0.437                          LAE05980
      GO TO 1870                                                        LAE05990
 1850 XK2K1=0.939                                                       LAE06000
      GO TO 1870                                                        LAE06010
 1860 XK2K1=0.939+(0.001525*(XLOB-10.0))                                LAE06020
 1870 ALP=AL                                                            LAE06030
      IF(AL) 1880,1890,1890                                             LAE06040
 1880 CDC=-CDC                                                          LAE06050
 1890 CNB=(XK2K1*SIN(2.*ALP)*COS(ALP/2.))*3.14159*D*D/(4.*AREA)         LAE06060
     *+ETA*CDC*((AP   )/AREA)*((SIN(ALP))**2)                           LAE06070
      XQ=XCG/D                                                          LAE06080
      CMB1=(XK2K1*XQ*SIN(2.*ALP)*COS(ALP/2.))*3.14159*D*D*D/            LAE06090
     *(4.*AREA*XREF)                                                    LAE06100
      CMB2=(ETA*CDC*((AP   )/AREA)*((XCG-(XC   ))/D)*((SIN(ALP)**2)))   LAE06110
     **D/XREF                                                           LAE06120
      CMB=CMB1+CMB2                                                     LAE06130
 1900 RE=REFT*VXM*XL                                                    LAE06140
      IF (RE-1.E06) 1910,1920,1920                                      LAE06150
 1910 AA=.0835                                                          LAE06160
      XNN=-.211                                                         LAE06170
      GO TO 1970                                                        LAE06180
 1920 IF (RE-1.E07) 1930,1940,1940                                      LAE06190
 1930 AA=.052                                                           LAE06200
      XNN=-.177                                                         LAE06210
      GO TO 1970                                                        LAE06220
 1940 IF (RE-1.E08) 1950,1960,1960                                      LAE06230
 1950 AA=.033                                                           LAE06240
      XNN=-.1488                                                        LAE06250
      GO TO 1970                                                        LAE06260
 1960 AA=.0221                                                          LAE06270
```

```
      XNN=-.127                                                         LAE06280
 1970 CFBOD=AA*RE**XNN                                                  LAE06290
      CDOB=(1.02*CFBOD*(1.0+.0025*XLOB+60.0/(XLOB**3.0))*SSUBS/AREA)    LAE06300
     *+CDPROT                                                           LAE06310
      IF(ENGINE.EQ.1.0) GO TO 1990                                      LAE06320
      IF(ENLET.EQ.1.0) GO TO 1980                                       LAE06330
      CDINL=.038*CDOB                                                   LAE06340
      GO TO 2000                                                        LAE06350
 1980 CDINL=0.05*CDOB                                                   LAE06360
      GO TO 2000                                                        LAE06370
 1990 CDINL=.0                                                          LAE06380
C   BASE DRAG (NO BOATTAIL)                                             LAE06390
 2000 CBPRI=-(((((2.8*VXM-.4)/2.4)/VXM**2.28)*.8123)-1.)*1.4286/VXM**2. LAE06400
      SB=(DBASE-DJET)*(DBASE+DJET)*3.14159/4.                           LAE06410
      CB=CBPRI*SB/AREA                                                  LAE06420
      IF (CB.LT.0.0) CB=0.0                                             LAE06430
      IF (BETA.NE.0.0) CB=0.0                                           LAE06440
      CDOB=CDOB+CB                                                      LAE06450
C   AFTERBODY DRAG CALCULATIONS (BOATTAIL)                              LAE06460
      DELCPO=0.                                                         LAE06470
      DELCPD=0.                                                         LAE06480
      DRATIO=(DBASE**2-DJET**2)/(D**2)                                  LAE06490
      DRATO1=(DJET**2)/(DBASE*D)                                        LAE06500
      DBDM=DBASE/D                                                      LAE06510
      DELBAS=-.1532+(.0247*BETA1)-(.002632*(BETA1**2))                  LAE06520
      CPBSO1=-.11905+(.00017*DBDM)-(.0283*(DBDM**2))                    LAE06530
      CPBSO2=-.0273+(.00425*DBDM)-(.1143*(DBDM**2))                     LAE06540
      CPBSO3=-.0612+(.3485*DBDM)-(.4254*(DBDM**2))                      LAE06550
      CPBSO4=.0789+(.5252*DBDM)-(.7434*(DBDM**2))                       LAE06560
      CPBSO5=-.085+(1.324*DBDM)-(1.407*(DBDM**2))                       LAE06570
      IF(BETA-3.0)2010,2010,2020                                        LAE06580
 2010 CPBSO=((CPBSO1+0.15)*BETA/3.0)-.15                                LAE06590
      GO TO 2090                                                        LAE06600
 2020 IF(BETA-5.6)2030,2030,2040                                        LAE06610
 2030 CPBSO=((CPBSO2-CPBSO2)*(BETA-3.0)/2.6)+CPBSO1                     LAE06620
      GO TO 2090                                                        LAE06630
 2040 IF(BETA-8.0)2050,2050,2060                                        LAE06640
 2050 CPBSO=((CPBSO3-CPBSO2)*(BETA-5.6)/2.4)+CPBSO2                     LAE06650
      GO TO 2090                                                        LAE06660
 2060 IF(BETA-16.0)2070,2070,2080                                       LAE06670
 2070 CPBSO=((CPBSO4-CPBSO3)*(BETA-8.0)/8.0)+CPBSO3                     LAE06680
      GO TO 2090                                                        LAE06690
 2080 CPBSO=((CPBSO5-CPBSO4)*(BETA-16.0)/4.0)+CPBSO4                    LAE06700
 2090 CDBASE=(-DRATIO)*(CPBSO+DELCPO+(DELCPO*DRATO1))                   LAE06710
      IF(ENGINE.EQ.1.0)GO TO 2470                                       LAE06720
      IF(DBDM-.76)2100,2100,2110                                        LAE06730
 2100 CBOAT5=.1013*DBDM                                                 LAE06740
      GO TO 2130                                                        LAE06750
```

```
2110 IF(DBDM.LT.0.82) GO TO 2120                                    LAE06760
     CBOAT5=-.4*(DBDM-.82)+.077                                     LAE06770
     GO TO 2130                                                     LAE06780
2120 CBOAT5=.077                                                    LAE06790
2130 IF(DBDM-.245)2140,2140,2150                                    LAE06800
2140 CBOAT4=.0653*DBDM                                              LAE06810
     GO TO 2180                                                     LAE06820
2150 IF(DBDM.GT.0.76) GO TO 2160                                    LAE06830
     CBOAT4=.1048*(DBDM-.245)+.016                                  LAE06840
     GO TO 2180                                                     LAE06850
2160 IF(DBDM.GT.0.82) GO TO 2170                                    LAE06860
     CBOAT4=.071                                                    LAE06870
     GO TO 2180                                                     LAE06880
2170 CBOAT4=-.369*(DBDM-.82)+.07                                    LAE06890
2180 IF(DBDM-.36)2190,2190,2200                                     LAE06900
2190 CBOAT3=.0292*DBDM                                              LAE06910
     GO TO 2230                                                     LAE06920
2200 IF(DBDM.GT.0.76) GO TO 2210                                    LAE06930
     CDOAT3=.10125*(DBDM-.36)+.0105                                 LAE06940
     GO TO 2230                                                     LAE06950
2210 IF(DBDM.GT.0.81) GO TO 2220                                    LAE06960
     CBOAT3=.057                                                    LAE06970
     GO TO 2230                                                     LAE06980
2220 CBOAT3=-.2579*(DBDM-.81)+.051                                  LAE06990
2230 IF(DBDM-.32)2240,2240,2250                                     LAE07000
2240 CBOAT2=.0125*DBDM                                              LAE07010
     GO TO 2290                                                     LAE07020
2250 IF(DBDM.GT.0.5) GO TO 2260                                     LAE07030
     CBOAT2=.0333*(DBDM-.32)+.004                                   LAE07040
     GO TO 2290                                                     LAE07050
2260 IF(DBDM.GT.0.72) GO TO 2270                                    LAE07060
     CBOAT2=.1068*(DBDM-.5)+.01                                     LAE07070
     GO TO 2290                                                     LAE07080
2270 IF(DBDM.GT.0.76) GO TO 2280                                    LAE07090
     CBOAT2=.0335                                                   LAE07100
     GO TO 2290                                                     LAE07110
2280 CBOAT2=-.1354*(DBDM-.76)+.0335                                 LAE07120
2290 IF(DBDM-.6)2300,2300,2310                                      LAE07130
2300 CBOAT1=.0055-(.03375*DBDM)+(.06875*(DBDM**2))                  LAE07140
     GC TO 2340                                                     LAE07150
2310 IF(DBDM.GT.0.75) GO TO 2320                                    LAE07160
     CBOAT1=.1*(DBDM-.6)+.01                                        LAE07170
     GO TO 2340                                                     LAE07180
2320 IF(DBDM.GT.0.78) GO TO 2330                                    LAE07190
     CBOAT1=.025                                                    LAE07200
     GO TO 2340                                                     LAE07210
2330 CBOAT1=-.091*(DBDM-.78)+.025                                   LAE07220
2340 IF(BETA-3.0)2350,2350,2360                                     LAE07230
```

```
2350 CBOAT1=(CBOAT1*BETA)/3.0                                                   LAE07240
     GO TO 2460                                                                 LAE07250
2360 IF(BETA-5.6)2370,2370,2380                                                 LAE07260
2370 CBOAT1=((CBOAT2-CBOAT1)*(BETA-3.0)/2.6)+CBOAT1                             LAE07270
     GO TO 2460                                                                 LAE07280
2380 IF(BETA-8.0)2390,2390,2400                                                 LAE07290
2390 CBOAT1=((CBOAT3-CBOAT2)*(BETA-5.6)/2.4)+CBOAT2                             LAE07300
     GO TO 2460                                                                 LAE07310
2400 IF(BETA-16.0)2410,2410,2420                                                LAE07320
2410 CBOAT1=((CBOAT4-CBOAT3)*(BETA-8.0)/8.0)+CBOAT3                             LAE07330
     GO TO 2460                                                                 LAE07340
2420 IF(BETA-24.0)2430,2430,2440                                                LAE07350
2430 CBOAT1=((CBOAT5-CBOAT4)*(BETA-16.0)/8.0)+CBOAT4                            LAE07360
     GO TO 2460                                                                 LAE07370
2440 PRINT 2450                                                                 LAE07380
2450 FORMAT(1X,'BOATTAIL ANGLE GREATER THAN 24 DEGREES')                        LAE07390
     CBOAT1=CBOAT5                                                              LAE07400
2460 CDBOAT=CBOAT1+DELBAS*(DELCPO+(DELCPD)*(DRATO1))                            LAE07410
     CDAFT=CDBOAT+CDBASE                                                        LAE07420
2470 FR=XLNOSE/D                                                                LAE07430
     CDWN1=0.000407*(FR**8.0)-.0102*(FR**7.0)+.108*(FR**6.0)-.616*(             LAE07440
    *FR**5.)+2.074*(FR**4.0)-4.183*(FR**3.0)+4.891*(FR**2.0)-3.017*             LAE07450
    *FR+.7795                                                                   LAE07460
     IF(FR.GT.1.0) CDWN1=.7432093E-04*FR**5.0-.15368398E-02*FR**4.0             LAE07470
    *+.011741209*FR**3.0-0.039555503*FR**2.0+0.050635882*FR+.005431397 LAE07480
     CDPTR=(CDWN1/.8)*VXM                                                       LAE07490
     CDOB1=CDOB                                                                 LAE07500
     CDOB=CDOB+CDINL+CDAFT+CDPTR                                                LAE07510
     CDOWBT=CDOW+CDOW2+CDOT+CDOB                                                LAE07520
     CDMISC=CDOWBT*.1                                                           LAE07530
     CDOWBT=CDOWBT+CDMISC                                                       LAE07540
     IF (IZZY-4) 2480,2480,2550                                                 LAE07550
2480 VXM=1.1999999                                                              LAE07560
     IF (SW) 2490,2490,2500                                                     LAE07570
2490 CDOW=0.                                                                    LAE07580
2500 IF (SW2) 2510,2510,2520                                                    LAE07590
2510 CDOW2=0.                                                                   LAE07600
2520 IF (ST) 2530,2530,2540                                                     LAE07610
2530 CDOT=0.                                                                    LAE07620
2540 HONST=CDOWBT                                                               LAE07630
     GO TO 2560                                                                 LAE07640
2550 IF (VXM-1.2) 2560,2560,3020                                                LAE07650
2560  IF (SW) 2570,2570,2580                                                    LAE07660
2570  DCDOSW = 0.0                                                              LAE07670
      SWTOT=0.0                                                                 LAE07680
      CDOW = 0.0                                                                LAE07690
      GO TO 2740                                                                LAE07700
2580  XXM = VXM*SQRT(COS(XLAMW4))                                               LAE07710
```

```
      SCMITC=SQRT(ABS((XXM*XXM)-1.0))/(TOVCW**0.33333)                  LAE07720
      ATC=ARW*(TOVCW**.333333)                                          LAE07730
      IZT=1                                                             LAE07740
 2590 IF (ATC-1.5) 2630,2600,2600                                       LAE07750
 2600 IF (VXM-1.) 2610,2610,2620                                        LAE07760
 2610 FUNCT=3.3081-1.88779*SQMITC+11.0916*SQMITC*SQMITC-18.6087*        LAE07770
     *SQMITC**3+7.4633*SQMITC**4                                        LAE07780
      FUNCT=FUNCT+(ATC-2.0)*0.1                                         LAE07790
      GO TO 2720                                                        LAE07800
 2620 FUNCT=3.0+(ATC-1.5)*0.3                                           LAE07810
      GO TO 2720                                                        LAE07820
 2630 IF (ATC-.5) 2690,2690,2640                                        LAE07830
 2640 IF (VXM-1.) 2650,2650,2680                                        LAE07840
 2650 FUNCT=2.47917-1.42798*SQMITC-.324405*SQMITC*SQMITC                LAE07850
      IF(SQMITC-0.4)2660,2670,2670                                      LAE07860
 2660 FUNCT=FUNCT+(1.8*(ATC-1.0))                                       LAE07870
 2670 FUNCT=FUNCT+(0.72*(ATC-1.0))/SQMITC                               LAE07880
      GO TO 2720                                                        LAE07890
 2680 FUNCT=(0.25*SQMITC)+(ATC*2.3)                                     LAE07900
      GO TO 2720                                                        LAE07910
 2690 IF (VXM-1.) 2700,2700,2710                                        LAE07920
 2700 FUNCT=0.9-(0.7*SQMITC)                                            LAE07930
      GO TO 2720                                                        LAE07940
 2710 FUNCT=.333*SQMITC+ATC*1.8                                         LAE07950
 2720 IF (IZT-2) 2730,2770,2810                                         LAE07960
 2730 DCDOSW=FUNCT*(TOVCW**1.66667)*((COS(XLAMW4))**2.5)                LAE07970
      IF(SQMITC.GT.1.3)DCDOSW=0.0                                       LAE07980
      IF(DCDOSW.LT.0.0)DCDOSW=0.0                                       LAE07990
      DCDOW=DCDOSW                                                      LAE08000
 2740  IF (SW2) 2750,2750,2760                                          LAE08010
 2750  DCDOS2 = 0.0                                                     LAE08020
       SW2TOT = 0.0                                                     LAE08030
       CDOW2= 0.0                                                       LAE08040
       GO TO 2780                                                       LAE08050
 2760  IZT = 2                                                          LAE08060
      XXM=VXM*SQRT(COS(XLAM24))                                         LAE08070
      SQMITC=SQRT(ABS((XXM*XXM)-1.))/(TOVCW2**0.33333)                  LAE08080
      ATC=ARW2*(TOVCW2**.33333)                                         LAE08090
      GO TO 2590                                                        LAE08100
 2770 DCDOS2=FUNCT*(TOVCW2**1.66667)*((COS(XLAM24))**2.5)               LAE08110
      IF(SQMITC.GT.1.3)DCDOS2=0.0                                       LAE08120
      IF(DCDOS2.LT.0.0)DCDOS2=0.0                                       LAE08130
      DCDOW2=DCDOS2                                                     LAE08140
 2780  IF (ST ) 2790,2790,2800                                          LAE08150
 2790  DCDOST = 0.0                                                     LAE08160
      STTOT=0.0                                                         LAE08170
       CDOT = 0.0                                                       LAE08180
       GO TO 2820                                                       LAE08190
```

```
2800  IZT = 3                                                          LAE08200
      XXM=VXM*SQRT(COS(XLAMT4))                                        LAE08210
      SQMITC=SQRT(ABS((XXM*XXM)-1.0))/(TOVCT**0.33333)                 LAE08220
      ATC=ART*(TOVCT**.33333)                                          LAE08230
      GO TO 2590                                                       LAE08240
2810  DCDOST=FUNCT*(TOVCT**1.66667)*((COS(XLAMT4))**2.5)               LAE08250
      IF(SQMITC.GT.1.3) DCDOST=0.0                                     LAE08260
      IF(DCDOST.LT.0.0) DCDOST=0.0                                     LAE08270
      DCDOT=DCDOST                                                     LAE08280
2820  IF(VXM-1.0)2840,2840,2830                                        LAE08290
2830  IF(VXM-2.0)2850,2860,2860                                        LAE08300
C   TRANSONIC BODY DRAG                                                LAE08310
2840  COVC=1.0-(0.08*VXM)                                              LAE08320
      GO TO 2870                                                       LAE08330
2850  COVC=0.92-(0.12*(VXM-1.0))                                       LAE08340
      GO TO 2870                                                       LAE08350
2860   COVC = 1.31213-0.36633*VXM+.06038*VXM**2-.00601*VXM**3+.000275* LAE08360
     *VXM**4                                                           LAE08370
2870  CDFPTR=1.02*CFBOD*COVC*SSUBS/AREA                                LAE08380
        CDPPTR=CDOB1-1.02*CFBOD*SSUBS/AREA                             LAE08390
      IF(VXM-1.0) 2890,2890,2880                                       LAE08400
2880  CDPPTR=(CDPPTR/0.2)*(1.2-VXM)                                    LAE08410
2890  IF(FR.GT.2.0) GO TO 2900                                         LAE08420
      CDWN2=0.000172*(FR**8)-0.00453*(FR**7)+0.050*(FR**6)-0.304*(FR**5)LAE08430
     *+1.096*(FR**4)-2.406*(FR**3)+3.160*(FR**2)-2.391*(FR)+1.00       LAE08440
      CDWN3=.000125*(FR**8)-.00370*(FR**7)+.0447*(FR**6)-.288*(FR**5)  LAE08450
     *+1.076*(FR**4)-2.385*(FR**3)+3.141*(FR**2)-2.529*FR+1.300        LAE08460
      GO TO 2910                                                       LAE08470
2900  CDWN2=-.33793095E-03*FR**5.0+.76402056E-02*FR**4.0-.67397615E-01* LAE08480
     *FR**3.0+.29429971*FR**2.0-.65782772*FR+.65397474                 LAE08490
      CDWN3=-.36714382E-03*FR**5.0+.88164977E-02*FR**4.0-.83661489E-01* LAE08500
     *FR**3.0+.39748005*FR**2.0-.9686696*FR+1.0327385                  LAE08510
2910  IF(VXM-.8) 3000,2920,2930                                        LAE08520
2920  CDPTR=CDWN1                                                      LAE08530
      GO TO 3000                                                       LAE08540
2930  IF(VXM-1.0) 2950,2940,2960                                       LAE08550
2940  CDPTR=CDWN2                                                      LAE08560
      GO TO 3000                                                       LAE08570
2950  CDPTR=((CDWN2-CDWN1)/0.2)*(VXM-0.8)+CDWN1                        LAE08580
      GO TO 3000                                                       LAE08590
2960  IF(VXM-1.2) 2980,2970,2990                                       LAE08600
2970  CDPTR=CDWN3                                                      LAE08610
      GO TO 3000                                                       LAE08620
2980  CDPTR=((CDWN3-CDWN2)/0.2)*(VXM-1.0)+CDWN2                        LAE08630
      GO TO 3000                                                       LAE08640
2990  CDPTR=0.0                                                        LAE08650
3000  CDOB=CB+CDFPTR+CDPPTR+CDINL+CDAFT+CDPTR                          LAE08660
C THE FOLLOWING INDENTED PROCEDURE WAS INSERTED TO FORCE THE CDOB       LAE08670
```

```
C CURVE TO SMOOTHLY DECAY.  THE ORIGINAL PROGRAM DOES NOT CAUSE
C THE DRAG CURVE TO DECAY PROPERLY.
                 IF (VXM.LE.1.0) F=0.0
                 IF (VXM.GT.1.0) F=48.3841618-106.7812601*VXM+24.23729924*
     *VXM**2.+67.95235861*VXM**3.-1.179384979*VXM**4.-55.05489748*VXM**5
     *.+22.44144432*VXM**6.
                 CDOB=CDOB+F
      CDALZ=CDOB
      CDOT=1.1*(DCDOST*(STTOT/AREA)+CDOT)
                 IF (VXM.GT.1.1) CDOT =CDOT*2.*EXP(VXM-1.1)
      CDOW=1.1*(DCDOSW*(SWTOT/AREA)+CDOW)
      CDOW2=1.1*(CDOW2+DCDOS2*(SWTOT/AREA))
      CDOWBT=CDOB+CDOT+CDOW+CDOW2+CDMISC
                 IF (VXM.GT.1.1) CDOWBT=CDOWBT*1.1*EXP(1.1-VXM)
      TONST=CDOWBT
      HONST=HONST+ZF*(TONST-HONST)
      IF (IZZY-4) 3010,3010,3030
 3010 VXM=VXMR1
      IZZY = IZZY + 1
      GO TO 1
 3020 CDOWBT=((HONST-TONST)/(SQRT(3.)-SQRT(1.2)))*SQRT(VXM)+TONST
     *+(HONST-TONST)/(1.-SQRT(3.)/SQRT(1.2))
                 IF (VXM.GT.1.2) CDOWBT=CDOWBT*.5*EXP(.5*(1.2-VXM))+CDOWBT
     */2.
      CDOT=1.1*(DCDOT *(STTOT/AREA)+CDOT)
                 IF ((SQM(TC.GT.1.3).OR.(VXM.GT.1.0)) CDOT=CDOT*.4*EXP(2.0
     **(1.0-VXM))+CDOT*1.5
      CDOW=1.1*(DCDOW *(SWTOT/AREA)+CDOW)
      CDOW2=1.1*(CDOW2+DCDOW2*(SWTOT/AREA))
      CDOB=CDOWBT-CDOT-CDOW-CDOW2-CB
      CDALZ=CDOB
 3030 CDB=XK2K1*SIN(2.*AL)*SIN(AL/2.)*3.14159*D*D/(4.*AREA)+ETA*
     *CDC*AP*((SIN(AL))**3)/AREA
      CLB=XK2K1*(SIN(2.*AL))*COS(AL/2.)*COS(AL)*3.14159*D*D/(4.*AREA)+ET
     *A*CDC*AP*((SIN(AL))**2)*COS(AL)/AREA-CDALZ*COS(AL)*COS(AL)*SIN(A
     *L)
      CA5=CDOWBT
      IF(AL) 3050,3040,3050
 3040 XCPB=0.0
      GO TO 3060
 3050 XCPB=((XCG/XREF)-(CMB/CNB))*XREF
 3060 IF (SW) 3080,3080,3070
 3070 IF (ST) 3210,3210,3090
 3080 IF (ST) 3340,3340,3220
 3090 R=D/2.
      IF (ICSC -1) 3110,3110,3100
 3100 XB1=BW/2.
      XB2=BT/2.
```

```
      TT=D/BT                                                           LAE09160
      HW1=-0.5*CROOTT*ABS(SIN(DELTA))+(XCPWB-XTAIL-CROOTT)*ABS(SIN(AL)) LAE09170
      XLAM1=XLAMW                                                       LAE09180
      GO TO 3120                                                        LAE09190
 3110 XB1=BT/2.                                                         LAE09200
      XB2=BW/2.                                                         LAE09210
      TT=D/BW                                                           LAE09220
      HW1=-0.5*CROOTT*ABS(SIN(DELTA))+(XCPWB-XTAIL-CROOTT)*ABS(SIN(AL)) LAE09230
      XLAM1=XLAMT                                                       LAE09240
 3120 FTRT=((XB2-R)/(2.*(1.-TT)))*((3.14159/4.)-((3.14159*TT**2)/4.)-TT+LAE09250
     *(((1.+TT**2)**2)/(2.*(1.-TT**2)))*ARSIN((1.-TT**2)/(1.+TT**2)))   LAE09260
      FW=FTRT+R                                                         LAE09270
      FI2=(FW*R**2)/((FW**2)+(HW1**2))                                  LAE09280
      HI2=(HW1*R**2)/((FW**2)+(HW1**2))                                 LAE09290
      ZC=FW                                                             LAE09300
      ZD=HW1                                                            LAE09310
      ZLT=0.0                                                           LAE09320
      DO 3180 I=1,4                                                     LAE09330
      ZL1=((XB1-(XLAM1*R))-(ZC*(1.-XLAM1)))/(2.*(XB1-R))                LAE09340
      ZL2=ALOG(((ZD**2)+((ZC-XB1)**2))/((ZD**2)+((ZC-R)**2)))           LAE09350
      ZL3=((1.-XLAM1)/(XB1-R))*((XB1-R)+(ZD*(ATAN((ZC-XB1)/ZD)-ATAN((ZC-LAE09360
     *R)/ZD))))                                                         LAE09370
      ZL=(ZL1*ZL2)-ZL3                                                  LAE09380
      IF(I-2) 3130,3140,3140                                            LAE09390
 3130 ZC=-ZC                                                            LAE09400
      GO TO 3170                                                        LAE09410
 3140 IF(I-3) 3150,3160,3170                                            LAE09420
 3150 ZL=-ZL                                                            LAE09430
      ZC=FI2                                                            LAE09440
      ZD=HI2                                                            LAE09450
      GO TO 3170                                                        LAE09460
 3160 ZL=-ZL                                                            LAE09470
      ZC=-FI2                                                           LAE09480
 3170 ZLT=ZLT+ZL                                                        LAE09490
 3180 CONTINUE                                                          LAE09500
      IF(ICSC-1) 3190,3190,3200                                         LAE09510
 3190 ART=(BT-D)**2/ST                                                  LAE09520
      XBT=BT/2.                                                         LAE09530
      CLI=(CLALW*CLALT*XKWBI*SIN(AL)*SW*2.*ZLT*(XBT-R))/(2.*3.14159*ART*LAE09540
     *FTRT*AREA*(1.+XLAMT))                                             LAE09550
      CLI=CLI*COS(AL)                                                   LAE09560
      CLTP=CLTP+CLI                                                     LAE09570
      CLT=CLT+CLI                                                       LAE09580
      CLIT=CLI                                                          LAE09590
      CLIW=0.                                                           LAE09600
      XCPTV=XCPTB                                                       LAE09610
      GO TO 3230                                                        LAE09620
 3200 ARW=(BW-D)**2/SW                                                  LAE09630
```

```
      XBW=BW/2.                                                         LAE09640
      CLI=(CLALW*CLALT*(XKWBI*SIN(AL)+XKTBI*SIN(DELTA))*ST*2.*ZLT*(XBW- LAE09650
     *R))/(2.*3.14159*ARW*FTRT*AREA*(1.+XLAMW))                         LAE09660
      CLI=CLI*COS(AL)                                                   LAE09670
      CLW=CLW+CLI                                                       LAE09680
      CLWP=CLWP+CLI                                                     LAE09690
      CLIW=CLI                                                          LAE09700
      CLIT=0.                                                           LAE09710
      XCPTV=XCPWB                                                       LAE09720
      GO TO 3230                                                        LAE09730
 3210 CLT=0.                                                            LAE09740
      CLALT=0.                                                          LAE09750
      CLTD=0.                                                           LAE09760
      CDTD=0.                                                           LAE09770
      CLBT=0.                                                           LAE09780
      CLBDT=0.                                                          LAE09790
      CLTB=0.                                                           LAE09800
      CLTDB=0.                                                          LAE09810
      CLVIST=0.                                                         LAE09820
      CLIT=0.                                                           LAE09830
      CLIW=0.                                                           LAE09840
      CLI=0.                                                            LAE09850
      CLTP=0.                                                           LAE09860
      GO TO 3230                                                        LAE09870
 3220 CLW=0.                                                            LAE09880
      CLALW=0.                                                          LAE09890
      CLBW=0.                                                           LAE09900
      CLWB=0.                                                           LAE09910
      CLVISW=0.                                                         LAE09920
      CLIW=0.                                                           LAE09930
      CLIT=0.                                                           LAE09940
      CLI=0.                                                            LAE09950
 3230 IF (SW2) 3240,3240,3250                                           LAE09960
 3240 CLW2=0.                                                           LAE09970
      CLALW2=0.                                                         LAE09980
      CLBW2=0.                                                          LAE09990
      CLWB2=0.                                                          LAE10000
C INDUCED DRAG                                                          LAE10010
 3250 ALPHA=AL                                                          LAE10020
      CLTOT=CLWP+CLTP+CLB+CLBW+CLBT                                     LAE10030
      IF(ARW.GT.3.0) GO TO 3260                                         LAE10040
      IF(VXM.GE..85) GO TO 3260                                         LAE10050
      CDI=ABS(CLTOT*TAN(ALPHA))                                         LAE10060
      GO TO 3270                                                        LAE10070
 3260 CDI=(CLTOT)**2/(PIE*BW**2*.7/AREA)                                LAE10080
 3270 IF(BT)3280,3290,3280                                              LAE10090
 3280 AR1=BT**2/AREA                                                    LAE10100
      CDTD=CLTD**2/(PIE*AR1*.7)                                         LAE10110
```

```
      CCT=CDOT+(CLTP**2/(PIE*AR1*.7))                                   LAE10120
 3290 IF(BW2)3300,3310,3300                                             LAE10130
 3300 AR2=BW2**2/AREA                                                   LAE10140
      CDW2=CDOW2+(CLW2**2/(PIE*AR2*.7))                                 LAE10150
 3310 IF(BW)3320,3330,3320                                              LAE10160
 3320 AR3=BW**2/AREA                                                    LAE10170
      CDW=CDOW+(CLWP**2/(PIE*AR3*.7))                                   LAE10180
C   AXIAL AND NORMAL COMPONENTS                                         LAE10190
 3330 CATD=CDTD*COSAAL-CLTD*SINAAL                                      LAE10200
      CNTD=CLTD*COSAAL+CDTD*SINAAL                                      LAE10210
      CNT=CLT*COSAAL+CDT*SINAAL                                         LAE10220
      CNTP=(CLTP+CLTD)*COSAAL+(CDT+CDTD)*(SINAAL)                       LAE10230
      CNW=CLW*COSAAL+CDW*SINAAL                                         LAE10240
      CNWP=CLWP*COSAAL+CDW*SINAAL                                       LAE10250
      CNW2=CLW2*COSAAL+CDW2*SINAAL                                      LAE10260
      CAB=(CDOB+CDB)*COSAAL -CLB*SINAAL                                 LAE10270
      CAW=CDW*COSAAL-CLW*SINAAL                                         LAE10280
      CAW2=CDW2*COSAAL-CLW2*SINAAL                                      LAE10290
      CAT=CDT*COSAAL-CLT*SINAAL                                         LAE10300
      CA5=CAT+CAW+CAW2+CATD+CAB                                         LAE10310
      GO TO 3370                                                        LAE10320
 3340 CN=CNB                                                            LAE10330
      XCP2=XCPB                                                         LAE10340
      IF(XAL(J).GT.90.)GO TO 3350                                       LAE10350
      CA=CA5*COS(AL)*COS(AL)                                            LAE10360
      GO TO 3360                                                        LAE10370
 3350 CA51=-(.5667+.893*VXM-.1727*VXM*VXM)                              LAE10380
      CA=CA51*COS(AL)*COS(AL)                                           LAE10390
 3360 CONTINUE                                                          LAE10400
      CLTOT=CLB                                                         LAE10410
      CDTOT=CDB                                                         LAE10420
      CLALW=0.                                                          LAE10430
      CLALT=0.                                                          LAE10440
      CDT=0.                                                            LAE10450
      CDD=0.0                                                           LAE10460
      CDW=0.                                                            LAE10470
      CATD=0.                                                           LAE10480
      CNTD=0.                                                           LAE10490
      CNT=0.                                                            LAE10500
      CNTP=0.                                                           LAE10510
      CAT=0.                                                            LAE10520
      CAW=0.                                                            LAE10530
      CNW=0.                                                            LAE10540
      CNWP=0.                                                           LAE10550
      CLW=0.                                                            LAE10560
      CLWP=0.                                                           LAE10570
      CLT=0.                                                            LAE10580
      CLTP=0.                                                           LAE10590
```

```
      CLTT=0.
      CLI=0.
      CLWT=0.
      CLALW2=0.
      CPW2=0.
      CAW2=0.
      CNTT=0.
      CNWT=0.
      CNWT2=0.
      CNW2=0.
      CLW2=0.
      XCPW2=0.
      XCPT=0.
      XCPW=0.
      GO TO 3490
C TOTAL NORMAL,AXIAL, AND DRAG FORCE COEFFICIENTS
 3370 CNBC=(CLBW+CLBT+CLBW2)*COSAAL
      CNTP=CLTP*COSAAL+CDT*SINAAL
      CNWP=CLWP*COSAAL+CDW*SINAAL
      CNW2=CLW2*COSAAL+CDW2*SINAAL
      CN=CNWP+CNTP+CNB+CNW2+CNBC
      CDI=CDI+CDB
      CDTOT=CDOWBT+CDI+CDTD
      CA=CDTOT*COSAAL-CLTOT*SINAAL
C  TOTAL CENTER OF PRESSURE CALCULATIONS
      XCPBC=( XCPBW*CLBW*COSAAL+XCPBT*CLBT*COSAAL+XCPBW2*CLBW2*COSAAL)
     */CNBC
      IF (ST) 3390,3390,3380
 3380 XCPT=XCPTB
 3390 IF (SW) 3420,3420,3400
 3400 XCPW=XCPWB
      IF (SW2) 3420,3420,3410
 3410 XCPW2=XCPWB2
 3420 IF (SW2) 3430,3430,3440
 3430 XCPW2=0.
      CNW2=0.0
      IF(SW) 3450,3450,3440
 3440 XCPW=((XCPW)*CNWP+(XCPW2)*CNW2)/(CNWP+CNW2)
      CNWP=CNWP+CNW2
      IF (SW) 3450,3450,3460
 3450 XCPW=0.
 3460 IF (ST) 3470,3470,3480
 3470 XCPT=0.
 3480 XCP2=(CNB*XCPB+XCPT*CNTP+XCPW*CNWP+XCPBC*CNBC)/(CNB+CNTP+CNWP+CNBC
     *)
C  PITCHING MOMENT
 3490 CM=CN*(XCG2-XCP2)/XREF
      AL1=AL*57.29578
```

```
C
C OUTPUT
      WRITE(8,3500) AL1,CLTOT,CDTOT,CLWP,CLBW,CLTP,CLBT,CLB,CDI,CNWP,
     *CNTP,CLTD,CDTD,CN,CA,XCPW,XCPT,XCPB,XCP2,CM
 3500 FORMAT(1X,F3.0,18(1X,F5.2),1X,F6.2)
 3510 WRITE(6,3520)AL1,CLTOT,CDTOT,CLWP,CLBW,CLTP,CLBT,CLB,CDI,CNWP,CNTP
 3520 FORMAT(1X,F3.0,10(1X,F6.2))
      WRITE (6,3530)CLTD,CDTD,CN,CA,XCPW,XCPT,XCPB,XCP2,CM
 3530 FORMAT(4X,9(1X,F6.2))
C++++++++++++++++++++++++++++++++++++++++++++++++
C  THE FOLLOWING INSTRUCTIONS ARE FOR THE PLOT OUTPUT FACILITY
      IF (CLTOT.LT.CLMIN) CLMIN=CLTOT
      IF (CLTOT.GT.CLMAX) CLMAX=CLTOT
      CLM(IJ,II,J)=CLTOT
      CDM(IJ,II,J)=CDTOT
      CMM(IJ,II,J)=CM
      CAM(IJ,II,J)=CA
      CNM(IJ,II,J)=CN
      CDIM(IJ,II,J)=CDI
C++++++++++++++++++++++++++++++++++++++++++++++++
      ALPHA = XAL(J+1)
 3540 CONTINUE
      WRITE(8,3560)CDINL,CDAFT,CDPROT,CDOB
      WRITE(8,3570)CDOW,CDOT,CDMISC,CDOWBT
 3550 WRITE (6,3560)CDINL,CDAFT,CDPROT,CDOB
 3560 FORMAT(1X,'CDINL=',F6.4,' CDAFT=',F6.4,' CDPROT=',F6.4,' CDOB='
     *,F6.4)
      WRITE (6,3570)CDOW,CDOT,CDMISC,CDOWBT
 3570 FORMAT(1X,'CDOW=',F6.4,' CDOT=',F6.4,' CDMISC=',F6.4,' CDOWBT=',F6
     *.4)
C
      DELTA1=XDT(II+1)
 3580 CONTINUE
      VXM=XVXM(IJ+1)
 3590 CONTINUE
      IF (NBODY-IL) 3600,3600,1040
 3600 CONTINUE
      CLAMW=CLAMW*57.29578
      CLAMT=CLAMT*57.29578
      WRITE (6,3610)
 3610 FORMAT(/,1X,'DO YOU WANT ANOTHER RUN, 00=YES, 01=NO')
      READ(5,1060)IZXI
      IF(IZXI.EQ.0)GO TO 1100
      CALL SCREEN
C++++++++++++++++++++++++++++++++++++++++++++++++
C  THE FOLLOWING INSTRUCTIONS ARE FOR THE PLOT OUTPUT FACILITY
      DC 4010 I=1,IM
         DO 4010 J=1,IDT
```

```
      DO 4010 K=1,IAL
         ALK=XAL(K)
         IF (ALK.LT.ALMIN)  ALMIN=ALK
         IF (ALK.GT.ALMAX)  ALMAX=ALK
         CDK=CDM(I,J,K)
         IF (CDK.LT.CDMIN)  CDMIN=CDK
         IF (CDK.GT.CDMAX)  CDMAX=CDK
         CMK=CMM(I,J,K)
         IF (CMK.LT.CMMIN)  CMMIN=CMK
         IF (CMK.GT.CMMAX)  CMMAX=CMK
         CAK=CAM(I,J,K)
         IF (CAK.LT.CAMIN)  CAMIN=CAK
         IF (CAK.GT.CAMAX)  CAMAX=CAK
         CNK=CNM(I,J,K)
         IF (CNK.LT.CNMIN)  CNMIN=CNK
         IF (CNK.GT.CNMAX)  CNMAX=CNK
         CDIK=CDIM(I,J,K)
         IF (CDIK.LT.CDIMIN) CDIMIN=CDIK
         IF (CDIK.GT.CDIMAX) CDIMAX=CDIK
4010     CONTINUE
      WRITE(7,4020) IM,IDT,ICL,CLMAX,CLMIN
4020  FORMAT(3I2,5X,2F6.2)
      DO 4060 I=1,IM
        WRITE(7,4030) XVXM(I)
4030    FORMAT(F6.2)
        DO 4070 J=1,IDT
          WRITE(7,4030) XDT(J)
          WRITE(7,4040)(CLM(I,J,K), K=1,24)
4040      FORMAT(12F6.2/,12F6.2)
          WRITE(7,4050) ALMAX,ALMIN
4050      FORMAT(2F6.2)
          WRITE(7,4040) (XAL(K), K=1,24)
          WRITE(7,4050) CDMAX,CDMIN
          WRITE(7,4040) (CDM(I,J,K), K=1,24)
          WRITE(7,4050) CMMAX,CMMIN
          WRITE(7,4040) (CMM(I,J,K), K=1,24)
          WRITE(7,4050) CAMAX,CAMIN
          WRITE(7,4040) (CAM(I,J,K), K=1,24)
          WRITE(7,4050) CNMAX,CNMIN
          WRITE(7,4040) (CNM(I,J,K), K=1,24)
          WRITE(7,4050) CDIMAX,CDIMIN
          WRITE(7,4040) (CDIM(I,J,K), K=1,24)
4070      CONTINUE
4060    CONTINUE
      WRITE (7,4080)
4080  FORMAT (1X,'/*')
      STOP
      END
```

```
C
C
C
      SUBROUTINE GEOSUB
C   THIS SUBROUTINE CLACULATES THE MISSILE WETTED AREA AND THE
C   REYNOLDS NUMBER PER FOOT BASED ON THE FLIGHT ALTITUDE.
      DIMENSION XVXM(16),XDT(16),XAL(48)
      COMMON REFTI,REFT,SSUBS,XLOB,ZF,VXM,IZZY,LLKK,LLLL,NSURF
      COMMON ICSC,INOSE,ISWPW,IAFBW,ISWPW2,IAFBW2,ISWPT,IAFBT,IDUM1,
     *CLAMT,CLAMW,CLAMW2,D,XLNOSE,BT,BW,BW2,CROOTW,CROOTT,CROOW2,
     *ST,SW,SW2,TOVCW,TOVCW2,TOVCT,XLAMT,XLAMW,XLAMW2,
     *XMACW,XMACW2,XMACT,XWING,XWING2,XTAIL,HT,XL,AREA,XREF,
     *ISWEPW,ISWEPT,ISWEP2,NWING,NTAIL,NWING2,ENGINE,ENLET,BETA,DBASE,
     *DJET,XLABOD
      COMMON ART,ARW,ARW2,BAR,BCOLAM,BETA1,B1,CLALT,CLALW,CLALW2,CLAL1,
     *COLAM,CROOT,IAFB,ISWP,TOVC,ODC,XBCRBW,XBCRWB,XKBT,XKBW,XKTB,XKTBT,
     *XKWB,XKWBI,ISWP1,RATIO,XLAM1,XMAC,D1
      DRAT=XLNOSE/D
      DRAA=XLABOD/D
      D1=D
      XLOB=XL/D
      IF(CLAMW-5.) 10,10,20
   10 CLAMW=5.
   20 IF(CLAMW2-5.) 30,30,40
   30 CLAMW2=5.
   40 IF(CLAMT-5.) 50,50,60
   50 CLAMT=5.
   60 CONTINUE
      CLAMW=CLAMW/57.29578
      CLAMT=CLAMT/57.29578
      CLAMW2=CLAMW2/57.29578
      IF (INOSE-1) 80,70,80
   70 SS1=(3.14*DRAT)+0.4
      GO TO 120
   80 IF(INOSE-2) 100,90,100
   90 SS1=(2.68*DRAT)+0.1
      GO TO 120
  100 IF(INOSE-3) 120,110,120
  110 SS1=(2.0*DRAT)+.2
  120 SSUBS=3.14159*D*(XL-XLNOSE-XLABOD)+(SS1+2.68*DRAA*0.9)*(3.14159*D*
     *D/4.0)
      WRITE (6,130)SSUBS
  130 FORMAT(1X,'NOSE WETTED AREA = ',F10.4,/)
      IF(DRAT-.6) 140,140,150
  140 ZF=1.
      GO TO 160
  150 ZF=1.522*EXP(-.7*DRAT)
  160 Z=HT*0.001
```

```
      IF(HT-35332.0) 170,170,180
  170 T=519.-HT/1280.
      PS=(1.91-0.01315*Z)**5.256
      GO TO 190
  180 T=393.
      B=1.69-0.0478*Z
      PS=6.49*EXP(B)
  190 C=49.1*SQRT(T)
      PS=PS*70.9
      RHO=PS/(1715.*T)
      XMU=2.270*(T**1.5)/((T+198.6)*(10.**8))
      REFT=(C*RHO)/XMU
      WRITE (6,200)
  200 FORMAT(1X,'IS THIS PROGRAM COMPARING EXPERIMENTAL DATA')
      WRITE (6,210)
  210 FORMAT(1X,'ENTER 00=NO, 01=YES')
      READ 220,IEXP
  220 FORMAT(I2)
      REFTI=1000000.
      IF(IEXP.EQ.0)GO TO 250
      WRITE (6,230)
  230 FORMAT(1X,'ENTER REYNOLDS NUMBER/XREF OF EXPERIMENTAL DATA')
      READ 240, REFTI
C     READ 240, REFT
  240 FORMAT(F14.2)
  250 CONTINUE
      RETURN
      END
C
C
C
      SUBROUTINE CLASUB
C  THIS SUBROUTINE CALCULATES THE AERODYNAMIC SURFACE LIFT-CURVE SLOPES.
      DIMENSION XVXM(16),XDT(16),XAL(48)
      COMMON REFTI,REFT,SSUBS,XLOB,ZF,VXM,IZZY,LLKK,LLLL,NSURF
      COMMON ICSC,INOSE,ISWPW,IAFBW,ISWPW2,IAFBW2,ISWPT,IAFBT,IDUM1,
     *CLAMT,CLAMW,CLAMW2,D,XLNOSE,BT,BW,BW2,CROOTW,CROOTT,CROOW2,
     *ST,SW,SW2,TOVCW,TOVCW2,TOVCT,XLAMT,XLAMW,XLAMW2,
     *XMACW,XMACW2,XMACT,XWING,XWING2,XTAIL,HT,XL,AREA,XREF,
     *ISWEPW,ISWEPT,ISWEP2,NWING,NTAIL,NWING2,ENGINE,ENLET,BETA,DBASE,
     *DJET,XLABOD
      COMMON ART,ARW,ARW2,BAR,BCOLAM,BETA1,B1,CLALT,CLALW,CLALW2,CLAL1,
     *COLAM,CROOT,IAFB,ISWP,TOVC,DDC,XBCRBW,XBCRWB,XKBT,XKBW,XKTB,XKTBT,
     *XKWB,XKWBI,ISWP1,RATIO,XLAM1,XMAC,D1
      IF (VXM-1.)10,10,20
   10 BETA1=SQRT (1.-VXM**2)
      GO TO 30
   20 BETA1=SQRT (VXM**2-1.)
```

```
 30 IF(IZZY-4)510,510,40
 40 IF(VXM-1.)60,50,60
 50 BETA1=0.0000001
 60 KFIN=0
    KFIN=KFIN+1
    IF (SW) 420,420,70
 70 ARW = (BW - D)**2/SW
    BAR=BETA1*ARW
    ISWP=ISWPW
    XLAM=XLAMW
    AR=ARW
 80 IF(BAR.GT.5.0.AND.VXM.GE.1.0)  BAR=5.0
    IF(BAR.GT.5.0.AND.VXM.LT.1.0)  BAR=5.0
    IF(ISWP-1) 90,90,200
 90 IF(XLAM-.25) 170,100,100
100 IF(VXM-1.0) 110,110,120
110 CLAR=-.1833*BAR+1.6
    GO TO 370
120 IF(XLAM-.50)130,140,150
130 CLAR=-.001032518*BAR**8+.0200677*BAR**7-.1557225*BAR**6
   *+.607131*BAR**5-1.21537*BAR**4+1.13786*BAR**3-.5718*BAR**2
   *+.458487*BAR+1.57588
    GO TO 370
140 CLAR=.00121006*BAR**8-.0254597*BAR**7+.22119793*BAR**6
   *- 1.020448*BAR**5+2.651997*BAR**4-3.6809957*BAR**3
   * +1.97903*BAR**2+.27640134*BAR+1.57129
    GO TO 370
150 CLAR=-.00007776*BAR**8-.001814086*BAR**7+.051777483*BAR**6
   *-.43555555*BAR**5+1.7240831*BAR**4-3.3472653*BAR**3
   *+2.6137236*BAR**2-.23389542*BAR+1.57998
    GO TO 370
170 IF(VXM-1.) 180,180,190
180 CLAR=-.1667*BAR+1.575
    GO TO 370
190 IF(BAR-2.)200,200,210
200 CLAR=.00133364*BAR**5-.0318237*BAR**4+.26049*BAR**3-.8643*BAR**2
   *+.669396*BAR+2.2008
    GO TO 370
210 CLAR=-.040277*BAR**4+.222522*BAR**3-.48931*BAR**2+.47287*BAR+1.57
    GO TO 370
220 IF (XLAM-.1) 230,300,300
230 IF(VXM-1.) 240,240,250
240 CLAR=-.2077*BAR+1.575
    GO TO 370
250 IF(BAR-.25) 260,260,270
260 CLAR=.2077*BAR+1.575
    GO TO 370
270 IF(BAR-4.) 280,280,290
```

```
280 CLAR=-.1668*BAR+1.667                                           LAE13480
    GO TO 370                                                       LAE13490
290 CLAR=1.587*(1.26)**(2.-BAR)                                     LAE13500
    GO TO 370                                                       LAE13510
300 IF (XLAM-.3) 310,340,340                                        LAE13520
310 IF(VXM-1.) 320,320,330                                          LAE13530
320 CLAR=-.2065*BAR+1.6                                             LAE13540
    GO TO 370                                                       LAE13550
330 CLAR=.002119027*BAR**5-.03283282*BAR**4+.20366375*BAR**3        LAE13560
   *-.61820362*BAR**2+.62703849*BAR+1.550048                        LAE13570
    GO TO 370                                                       LAE13580
340 IF(VXM-1.) 350,350,360                                          LAE13590
350 CLAR=-.225*BAR+1.675                                            LAE13600
    GC TO 370                                                       LAE13610
360 IF(XLAM.LE..40)CLAR=.00070428*BAR**8-.0158837*BAR**7            LAE13620
   *+.1489347*BAR**6-.74733673*BAR**5+2.1861823*BAR**4              LAE13630
   *-3.3445976*BAR**3+2.2919403*BAR**2-.21045286*BAR+1.6203524      LAE13640
    IF(XLAM.GT..40)CLAR=-.00012278513*BAR**8+.0000879756*BAR**7     LAE13650
   *+.025071933*BAR**6-.25587433*BAR**5+1.0834904*BAR**4            LAE13660
   *-2.1562044*BAR**3+1.6307710*BAR**2-.084287165*BAR+1.7039        LAE13670
370 CLAL=CLAR*AR                                                    LAE13680
    IF (KFIN-2) 380,390,400                                         LAE13690
380 CLALW=CLAL                                                      LAE13700
    GO TO 410                                                       LAE13710
390 CLALW2=CLAL                                                     LAE13720
    GO TO 410                                                       LAE13730
400 CLALT=CLAL                                                      LAE13740
410 IF (KFIN-2) 420,440,500                                         LAE13750
420 KFIN=KFIN+1                                                     LAE13760
    IF (SW2) 440,440,430                                            LAE13770
430 AR=(BW2-D)**2/SW2                                               LAE13780
    BAR=BETA1*AR                                                    LAE13790
    ISWP=ISWPW2                                                     LAE13800
    ARW2=AR                                                         LAE13810
    XLAM=XLAMW2                                                     LAE13820
    GO TO 80                                                        LAE13830
440 KFIN=KFIN+1                                                     LAE13840
    IF (ST) 460,460,450                                             LAE13850
450 AR=(BT-D)**2/ST                                                 LAE13860
    BAR=BETA1*AR                                                    LAE13870
    ART=AR                                                          LAE13880
    ISWP=ISWPT                                                      LAE13890
    NSURF=NTAIL                                                     LAE13900
    XLAM=XLAMT                                                      LAE13910
    GO TO 80                                                        LAE13920
460 IF (SW) 470,470,500                                             LAE13930
470 IF (SW2) 480,480,500                                            LAE13940
480 IF (ST) 490,490,500                                             LAE13950
```

```
  490 LLLL=2                                                            LAE13960
      RETURN                                                            LAE13970
  500 LLKK=0                                                            LAE13980
  510 IZZY = IZZY + 1                                                   LAE13990
      IF (SW) 530,530,520                                               LAE14000
  520 COLAM= COS(CLAMW)/(SIN(CLAMW+.1))                                 LAE14010
      ARW=(BW-D)**2/SW                                                  LAE14020
      BCOLAM=BETA1*COLAM                                                LAE14030
      CROOT=CROOTW                                                      LAE14040
      B1=BW                                                             LAE14050
      IAFB=IAFBW                                                        LAE14060
      XMAC=XMACW                                                        LAE14070
      TOVC=TOVCW                                                        LAE14080
      CLAL1=CLALW                                                       LAE14090
      XLAM1=XLAMW                                                       LAE14100
      ISWP1=ISWPW                                                       LAE14110
      NSURF=NWING                                                       LAE14120
      BAR=BETA1*ARW                                                     LAE14130
      RATIO=CROOT/(BETA1*D)                                             LAE14140
      LLLL=0                                                            LAE14150
      RETURN                                                            LAE14160
  530 IZZY = IZZY + 1                                                   LAE14170
      IF (SW2) 540,540,550                                              LAE14180
  540 LLLL=1                                                            LAE14190
      RETURN                                                            LAE14200
  550 COLAM=COS(CLAMW2)/(SIN(CLAMW2+.1))                                LAE14210
      BCOLAM=BETA1*COLAM                                                LAE14220
      CROOT=CROOW2                                                      LAE14230
      B1=BW2                                                            LAE14240
      IAFB=IAFBW2                                                       LAE14250
      CLAL1=CLALW2                                                      LAE14260
      XLAM1=XLAMW2                                                      LAE14270
      NSURF=NWING2                                                      LAE14280
      XMAC=XMACW2                                                       LAE14290
      TOVC=TOVCW2                                                       LAE14300
      ISWP1=ISWPW2                                                      LAE14310
      ARW2=(BW2-D)**2/SW2                                               LAE14320
      BAR=BETA1*ARW2                                                    LAE14330
      RATIO=CROOT/(BETA1*D)                                             LAE14340
      LLLL=0                                                            LAE14350
      RETURN                                                            LAE14360
      END                                                               LAE14370
C                                                                       LAE14380
C                                                                       LAE14390
C                                                                       LAE14400
      SUBROUTINE CATSUB                                                 LAE14410
C  THIS SUBROUTINE CALCULATES CENTER OF PRESSURE LOCATIONS, CROSS-FLOW  LAE14420
C  DRAG COEFFICIENTS, AND INTERFERENCE FACTORS.                         LAE14430
```

```
      DIMENSION XVXM(16),XDT(16),XAL(48)
      COMMON REFTI,REFT,SSUBS,XLOB,ZF,VXM,IZZY,LLKK,LLLL,NSURF
      COMMON ICSC,INOSE,ISWPW,IAFBW,ISWPW2,IAFBW2,ISWPT,IAFBT,IDUM1,
     1CLAMT,CLAMW,CLAMW2,D,XLNOSE,BT,BW,BW2,CROOTW,CROOTT,CROOW2,
     2ST,SW,SW2,TOVCW,TOVCW2,TOVCT,XLAMT,XLAMW,XLAMW2,
     3XMACW,XMACW2,XMACT,XWING,XWING2,XTAIL,HT,XL,AREA,XREF,
     4ISWEPW,ISWEPT,ISWEP2,NWING,NTAIL,NWING2,ENGINE,ENLET,BETA,DBASE,
     5DJET,XLABOD
      COMMON ART,ARW,ARW2,BAR,BCOLAM,BETA1,B1,CLALT,CLALW,CLALW2,CLAL1,
     1COLAM,CROOT,IAFB,ISWP,TOVC,ODC,XBCRBW,XBCRWB,XKBT,XKBW,XKTB,XKTBT,
     2XKWB,XKWBI,ISWP1,RATIO,XLAM1,XMAC,D1
      XKWB=(2./3.14159)*((1.+D**4/B1**4)*(.5*ATAN(.5*(B1/D-D/B1))+3.1415
     19/4.)-(D**2/B1**2)*((B1/D-D/B1)+2.*ATAN(D/B1)))/(1.-D/B1)**2
   10 XKTB1=((3.14159)**2*D**2*(B1/D+1.)**2)/(4.*B1**2)+(3.14159*D**2*(B
     11**2/D**2+1.)**2/(B1**2*(B1/D-1.)**2))*ARSIN((B1**2/D**2-1.)/(B1**
     22/D**2+1.))
      XKTB2=2.*3.14159*D*(B1/D+1.)/(B1*(B1/D-1.))-((B1**2/D**2+1.)**2/((
     1B1**2/D**2)*(B1/D-1.)**2))*(ARSIN((B1**2/D**2-1.)/(B1**2/D**2+1.))
     2)**2
      XKTB3=(4.*D*(B1/D+1.)/(B1*(B1/D-1.)))*ARSIN((B1**2/D**2-1.)/(B1**2
     1/D**2+1.))-(8./(B1/D-1.)**2)*ALOG10((B1**2/D**2+1.)/(2.*B1/D))
      XKTB=(1./(3.14159)**2)*(XKTB1-XKTB2-XKTB3)
      IF (ICSC-1) 20,20,40
   20 IF (LLKK) 30,30,50
   30 XKWBI=D/B1+1.
       XKTBI=XKTB
      GO TO 50
   40 IF (LLKK-1) 50,50,30
   50 BAREF=BAR*(1.+XLAM1)*((1./(BETA1*COLAM))+1.)
      IF (BAREF-4.) 60,60,70
   60 XKBW=(1.+D/B1)**2-XKWB
      GO TO 180
   70 IF (IAFB) 80,110,80
   80 IF (BCOLAM-1.) 100,90,90
   90 XKBW1=(BCOLAM/(1.+BCOLAM))*((((BCOLAM+1.)*(1./RATIO)+BCOLAM)/BCOLA
     1M)**2)*ARCOS((1.+(1.+BCOLAM)*(1./RATIO))/(BCOLAM+(BCOLAM+1.)*(1./
     2RATIO)))
      XKBW2=(SQRT(BCOLAM**2-1.)/(BCOLAM+1.))*(SQRT(1.+2.*BETA1*D/CROOT)-
     11.)-(SQRT(BCOLAM**2-1.)/BCOLAM)*(BETA1*D/CROOT)**2*ALOG((1.+CROOT/
     2(BETA1*D))+SQRT((1.+CROOT/(BETA1*D))**2-1.))-(BCOLAM/(1.+BCOLAM))*
     3ARCOS(1./BCOLAM)
      XKBW=(8.*BCOLAM/(3.14159*SQRT(BCOLAM**2-1.)*(1.+XLAM1)*(BETA1*D/
     1CROOT)*(B1/D-1.)*(BETA1*CLAL1)))*(XKBW1+XKBW2)
      GO TO 180
  100 XKBW1=((BCOLAM+(1.+BCOLAM)*BETA1*D/CROOT)/BCOLAM)**1.5+((BCOLAM+(1
     1.+BCOLAM)*BETA1*D/CROOT)/BCOLAM)**.5-2.
      XKBW2=(((1.+BCOLAM)*BETA1*D/CROOT)/BCOLAM)**2*0.5*(ALOG(1.+SQRT(BC
     1OLAM/(BCOLAM+(1.+BCOLAM)*BETA1*D/CROOT)))-ALOG(1.-SQRT(BCOLAM/(BCO
```

```
    2LAM+(1.+BCOLAM)*BETA1*D/CROOT))))
     XKBW=(16.*(BCOLAM/(1.+BCOLAM))**2/(3.14159*(1.+XLAM1)*(BETA1*D/
    1CROOT)*(B1/D-1.)*(BETA1*CLAL1)))*(XKBW1-XKBW2)
     GO TO 180
110  IF(BCOLAM-1.) 150,120,120
120  IF(RATIO-1.) 130,140,140
130  RATIO=1.
     D=CROOT/BETA1
140  XKB1=(1.+COLAM*BETA1*RATIO)**2*ARCOS((BCOLAM+RATIO)/(1.+COLAM*BETA
    11*RATIO))
     XKB2=BCOLAM**2*RATIO**2*ARCOS(1./BCOLAM)-BCOLAM*RATIO**2*SQRT(BCOL
    1AM**2-1.)*ARSIN(1./RATIO)
     XKB3=SQRT(BCOLAM**2-1.)*ALOG(RATIO+SQRT(RATIO**2-1.))
     XKBW=((8./(3.14159*SQRT(BCOLAM**2-1.)*BETA1*CLAL1*(XLAM1+1.)*(B1/D
    11-1.)))*(1./RATIO))*(XKB1-XKB2-XKB3)
     GO TO 180
150  IF(RATIO-1.) 160,170,170
160  RATIO=1.
     D=CROOT/BETA1
170  XKB1=(1.+COLAM*BETA1*RATIO)*SQRT((RATIO-1.)*(COLAM*BETA1*RATIO+1.)
    1)
     XKB2=RATIO**2*(BCOLAM)**1.5-BCOLAM*RATIO**2*(BCOLAM+1.)*(ATAN(SQRT
    1(1./BCOLAM))-ATAN(SQRT((RATIO-1.)/(COLAM*BETA1*RATIO+1.))))
     XKB3=((BCOLAM+1.)/SQRT(BCOLAM))*0.5*(ALOG(1.+SQRT(BCOLAM*(RATIO-1.
    1)/(COLAM*BETA1*RATIO+1.)))-ALOG(1.-SQRT(BCOLAM*(RATIO-1.)/(COLAM*
    2BETA1*RATIO+1.))))
     XKBW=((16.*SQRT(BCOLAM)*(1./RATIO)/(BETA1*CLAL1*(XLAM1+1.)*(B1/D1
    1-1.)*3.14159*(BCOLAM+1.))))*(XKB1-XKB2-XKB3)
180  XKBT=XKWB-XKTB
     D=D1
     IF(VXM-1.0) 190,190,320
190  IF(ISWP1-1   ) 200,200,230
200  IF(BAR-2.0) 210,220,220
210  XBCRWB=0.35-XLAM1*(0.35+17.885-SQRT(328.8782-(BAR-3.)**2))
     GO TO 260
220  XBCRWB=0.35-0.1*XLAM1
     GO TO 260
230  IF(BAR-2.0) 240,250,250
240  XCR1=SQRT(2029.5+(BAR-3.)**2)
     XBCRWB=((-44.5+XCR1)-XLAM1*(-44.5+XCR1+17.885-SQRT(328.8782-(BAR
    1-3.)**2)))
     GO TO 290
250  XBCRWB=0.55-0.3*XLAM1
     GO TO 290
260  IF(BAR-4.0) 270,280,280
270  XBCRBW=0.25-XLAM1*(32.125-SQRT(1032.02-(BAR-4.)**2))+(1.-XLAM1)*
    1ALOG(1.04+0.1*D/B1)*(-7.5+SQRT(72.25-(BAR-4.0)**2))
     GO TO 520
```

```
280 XBCRBW=0.25+((1.-XLAM1)*ALOG(1.04+0.1*D/B1))                        LAE15400
    GO TO 520                                                            LAE15410
290 IF(BAR-4.0) 300,310,310                                              LAE15420
300 XBCRBW=0.25+(1.-2.*XLAM1)*(32.125-SQRT(1032.02-(BAR-4.)**2))+        LAE15430
   1(1.-XLAM1)*ALOG(1.12+0.3*D/B1)*(-7.5+SQRT(72.25-(BAR-4.)**2))        LAE15440
    GO TO 520                                                            LAE15450
310 XBCRBW=0.25+(1.-XLAM1)*ALOG(1.12+0.3*D/B1)                           LAE15460
    GO TO 520                                                            LAE15470
320 IF(ISWP1-1  ) 330,330,360                                            LAE15480
330 IF(BAR-3.0) 340,350,350                                              LAE15490
340 XBCRWB=-(9.235+25.*(1.-XLAM1))+SQRT((9.71+25.*(1.-XLAM1))**2-        LAE15500
   1(BAR-3.)**2)                                                         LAE15510
    GO TO 390                                                            LAE15520
350 XBCRWB=0.005*BAR+0.46                                                LAE15530
    GO TO 390                                                            LAE15540
360 IF(BAR-3.) 370,380,380                                               LAE15550
370 XBCRWB=0.675-XLAM1*(0.675+9.235-SQRT(94.1-(BAR-3.)**2))              LAE15560
    GO TO 490                                                            LAE15570
380 XBCRWB=0.005*BAR+0.46+0.2*(1.-XLAM1)                                 LAE15580
    GO TO 490                                                            LAE15590
390 BARLAM=BAR*(1.-XLAM1)*(1.+(1./BCOLAM))                               LAE15600
    IF(BARLAM-4.0) 400,400,440                                           LAE15610
400 IF(BAR-2.0) 410,430,430                                              LAE15620
410 XBMID1 = ALOG(1.32 - 0.32*XLAM1)                                     LAE15630
420 XBMID2=4.+(XBMID1**2)-((0.5+0.5139*(D/B1)*(1.17+XLAM1)*(1./(0.331    LAE15640
   1**(D/B1))))**2)                                                      LAE15650
    XBMID3=2.*((XBMID1)-(0.5+0.5139*(D/B1)*(1.17+XLAM1)*(1./(0.331       LAE15660
   1**(D/B1)))))                                                         LAE15670
    XBMID=XBMID2/XBMID3                                                  LAE15680
    XCMID=4.+(XBMID1-XBMID)**2                                           LAE15690
    XBCRBW=SQRT(XCMID-((BAR-2.)**2))+XBMID                               LAE15700
    GO TO 520                                                            LAE15710
430 XBCRBW=0.5+0.25695*(D/B1)*(1./(0.331**(D/B1)))*BAR*(1.17+XLAM1)      LAE15720
    GO TO 520                                                            LAE15730
440 IF(IAFB) 450,450,480                                                 LAE15740
450 IF(RATIO-1.0) 460,460,470                                            LAE15750
460 XBCRBW=0.67                                                          LAE15760
    GO TO 520                                                            LAE15770
470 XBCRBW=-2.32+SQRT(8.9401-((1./RATIO)-1.)**2)                         LAE15780
    GO TO 520                                                            LAE15790
480 XBCRBW=(0.429/RATIO)+0.5                                             LAE15800
    GO TO 520                                                            LAE15810
490 BARLAM=BAR*(1.-XLAM1)*(1.+(1./BCOLAM))                               LAE15820
    IF(BARLAM-4.0) 500,500,440                                           LAE15830
500 IF(BAR-2.0) 510,430,430                                              LAE15840
510 XBMID1=ALOG(1.65-0.65*XLAM1)                                         LAE15850
    GO TO 420                                                            LAE15860
520 D=D1                                                                 LAE15870
```

```
      ARAT=BAR/BETA1                                                    LAE15880
      IF(ARAT-4.0) 540,540,530                                          LAE15890
  530 ARAT=4.0                                                          LAE15900
  540 ODC=2.*(1.+XLAM1)**1.6*EXP(-.4*ARAT)                              LAE15910
      IF(ARAT-1.0) 580,550,550                                          LAE15920
  550 IF(XLAM1.LE.0.25) GO TO 560                                       LAE15930
      IF(XLAM1.LE.0.75) GO TO 570                                       LAE15940
      ODC=ODC-.25*(ARAT-1.)                                             LAE15950
      IF(ARAT.LT.2.0) GO TO 580                                         LAE15960
      ODC=(2.*(1.+XLAM1)**1.6)*.45-(.85*(ARAT-2.))                      LAE15970
      GO TO 580                                                         LAE15980
  560 ODC=(2.*(1.+XLAM1)**1.6)*.67-(.3*(ARAT-1.0))                      LAE15990
      GO TO 580                                                         LAE16000
  570 ODC=(2.*(1.+XLAM1)**1.6)*.67-(.725*(ARAT-1.))                     LAE16010
  580 CONTINUE                                                          LAE16020
      RETURN                                                            LAE16030
      END                                                               LAE16040
C                                                                       LAE16050
C                                                                       LAE16060
      SUBROUTINE SCREEN                                                 LAE16070
      WRITE (6,500)                                                     LAE16080
  500 FORMAT (/1X,'CLEAR SCREEN AND ENTER "0"')                         LAE16090
      READ (6,16) ISCR                                                  LAE16100
   16 FORMAT (1I1)                                                      LAE16110
      RETURN                                                            LAE16120
      END                                                               LAE16130
```

```
//AEROPLOT JOB (1414,0483,,24),'LINDSEY',CLASS=A
// EXEC FRTXCLGP
//FORT.SYSIN DD *
C.***********PROGRAM AEROPLOT****************
C
C   PROGRAM AEROPLOT PLOTS DIMENSIONLESS AERODYNAMIC COEFFIEIENTS
C   CD TOTAL, CM,CA,DN,CD INDUCED AND ANGLE OF ATTACK AS A
C.  FUNCTION OF LIFT COEFFICIENT.  A SEPERATE PLOT IS GENERATED
C   FOR EACH MACH NUMBER (MAX OF 30 MACH NUMBERS) ALLOWING FOR UP
C.  TO 10 DIFFERENT CONTROL SURFACE DEFLECTIONS FOR EACH PLOT.
C.  THE PROGRAM USES VERSAPLOT SOFTWARE FOR USE ON THE IBM 3033
C.  COMPUTER.  ACTUAL PLOTTING IS ACCOMPLISHED VIA SUBROUTINE
C.  "GRAPH" TO PERMIT RELATIVELY PAINLESS TRANSITION TO OTHER
C.  SOFTWARE, THOUGH LOCATION OF ORIGIN FOR THE VARIOUS PLOTS ARE
C.  DETERMINED IN MAIN PROGRAM.
C
C.VARIABLES USED IN COMMON BLOCKS ARE AS FOLLOWS.
C.       1. ASCALE - CONTAINS THE INITIAL PLOTTING VALUE AND PLOT
C.            INCREMENT FOR ALL 6 PLOT PARAMETERS.  PLOT INCREMENTS
C.            ARE DETERMINED SUCH THAT EACH PLOT IS ON A 5" X 5"
C.            AXIS
C
C.       2. ISYMBL - CONTAINS THE INTEGER DESIGNATION OF THE PLOTTING
C.            SYMBOL USED IN GRAPHING
C
C.  OTHER VARIABLES:
C        1. LEGEND - ARRAY OF 10 PLOTTING SYMBOL INTEGERS USED TO
C.            PRODUCE A LEGEND OF SYMBOLS CORRESPONDING TO THE VARIOUS
C.            CONTROL SURFACE DEFLECTIONS USED.
C
C.       2. DELTA - ARRAY OF CONTROL SURFACE DEFLECTION ANGLES
C
C.       3. IFLAG - FLAG WHICH DETERMINES IF COORDINATE AXES ARE
C.            TO BE PLOTTED (1), OR IF ANOTHER DELTA VALUE IS TO
C.            BE PLOTTED ON AN ALREADY EXISTING GRAPH (0).
C
C.  ALL DATA TO BE PLOTTED IS GENERATED FROM AN EXTERNAL SOURCE.
C.  NO COEFFICIENT CALCULATIONS ARE MADE IN THIS PROGRAM.
C.
C.  LT. GREY HOBBY USN                     MARCH 1981
C****************************************
C.
      DIMENSION CL(24),DELTA(14),LEGEND(10)
      REAL MACH
      COMMON/ASCALE/FVALX,FVALY,DVX,DVY,FCM,DCM,FCD,DCD,
     +FCDT,DCDT,FCA,DCA,FCN,DCN
      COMMON/ISYMBL/NSYMB
      DATA LEGEND /1,2,3,4,5,6,7,8,9,10/
```

```
C
C
C
C.   ENTER NUMBER OF MACH,NUMBER OF DELTA/MACH,NUMBER OF CL AND
C.   MAX,MIN CL
       READ(5,900)NMACH,NDEL,NCL,CLMAX,CLMIN
C
C.   SET UP PLOTTING WINDOW TO ALLOW PLOTTING SPACE FOR UP
C.   TO 30 MACH NUMBERS
       XMACH=FLOAT(NMACH)
       XPLOT=21.1 *XMACH
       CALL WINDOW(0.,XPLOT,0.,XPLOT)
C
C==============PLOTTING STARTS HERE==============
C  INITIALIZE PLOT
       CALL PLOTS(0,0,0)
C.   MOVE ORIGIN TO ALLOW SPACE FOR AXIS LABELS AND TITLES
       CALL PLOT(1.,1.,-3)
C
C.   READ MACH AND PLOT ALPHA, CM,CDTOT,CA,CLTOT,CN
       DO 100  I=1,NMACH
C.   SET UP FLAG FOR AXIS PLOT AND PLOT SYMBOL DESIGNATION
       NSYMB=1
       IFLAG=1
       READ(5,902) MACH
C
C.   PLOT DATA FOR EACH MACH NUMBER
       DO 90 J=1,NDEL
C.   READ DELTA
       READ(5,902) DELTA(J)
C.   READ CL DATA
       READ(5,901)CL
C.
C.   PLOT ANGLE OF ATTACK
       CALL ALPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
C.   REORIGIN AND PLOT TOTAL DRAG COEFFICIENT
       CALL PLOT(5.75,0.0,-3)
       CALL CDTPLT(NCL,CL,CLMIN,CLMAX,IFLAG)
       CALL PLOT(-5.75,0.,-3)
C.   REORIGIN AND PLOT PITCHING MOMENT
       CALL PLOT(0.,5.75,-3)
       CALL CMPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
       CALL PLOT(0.,-5.75,-3)
C.   REORIGIN AND PLOT CA
       CALL PLOT(5.75,5.75,-3)
       CALL CAPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
       CALL PLOT(-5.75,-5.75,-3)
C.   REORIGIN AND PLOT NORMAL FORCE
```

```
      CALL PLOT(0.,11.5,-3)
      CALL CNPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
      CALL PLOT(0.,-11.5,-3)
C.  REORIGIN AND PLOT INDUCED DRAG
      CALL PLOT(5.75,11.5,-3)
      CALL CDPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
      CALL PLOT(-5.75,-11.5,-3)
C
C.  INCREMENT SYMBOL NUMBER AND ZERO AXIS FLAG
      NSYMB=NSYMB+1
      IFLAG=0
90    CONTINUE
C.
C.  LABEL PLOT WITH MACH NUMBER
      CALL NEWPEN(3)
      CALL SYMBOL(4.5,-1.0,.25,'MACH = ',0.0,7)
      CALL NUMBER(6.25,-1.0,.25,MACH,0.0,2)
C.  PRINT LEGEND FOR DELTA PLOTTING SYMBOLS
      XSYMB=9.75
      YSYMB=17.0
      XDEL=10.0
      DO 95 J=1,NDEL
C.  PRINT PLOTTING SYMBOL
      CALL SYMBOL(XSYMB,YSYMB,0.1,LEGEND(J),0.,-1)
C.  PRINT DELTA VALUE
      CALL NUMBER(XDEL,YSYMB,0.1,DELTA(J),0.,-1)
C.  INCREMENT PLOTTING SYMBOL AND PRINT COORDINATES
      YSYMB=YSYMB+0.15
   95 CONTINUE
C.  LABEL THE LEGEND
      YSYMB=YSYMB+0.15
      CALL SYMBOL(9.5,YSYMB,0.1,'DELTA LEGEND:',0.,13)
C.  LABEL PLOTTING GRAPH
      CALL SYMBOL(3.,19.5,.25,'AERODYNAMIC COEFFICIENTS',0.,24)
C
C.  MOVE TO NEXT MACH PLOTTING POSITION
      CALL PLOT(12.5,0.,-3)
C
100   CONTINUE
C
C.  TERMINATE PLOT
      CALL PLOT(0.,0.,+999)
C
C.***********FORMATS**********
900   FORMAT(3I2,5X,2F6.2)
901   FORMAT(12F6.2,/,12F6.2)
902   FORMAT(F6.2)
C*******************************
```

AD-A105 788 NAVAL POSTGRADUATE SCHOOL MONTEREY CA F/G 16/4
COMPUTER PROGRAM APPLICATIONS TO TACTICAL MISSILE CONCEPTUAL DE--ETC(U)
JUN 81 M D SULLIVAN
UNCLASSIFIED NL

3 of 3

END
DATE
FILMED
11-81
DTIC

```
C
      STOP
      END
C.
C.
      SUBROUTINE GETDAT(YMAX,YMIN,Y)
      DIMENSION Y(24)
C
C.   SUBROUTINE INPUTS YCOORD. ARRAY FOR PROCESSING BY PLOT ROUTINE
      READ(5,900)YMAX,YMIN
      READ(5,901)Y
C
C===============FORMATS================
900   FORMAT(2F6.2)
901   FORMAT(12F6.2,/,12F6.2)
C======================================
C
      RETURN
      END
C
C
      FUNCTION YSCALE(YMAX,YMIN)
C.   FUNCTION GENERATES PLOTTING INCREMENT FOR A 5" AXIS
      YSCALE=(YMAX-YMIN)/5.0
      RETURN
      END
C
C
      SUBROUTINE GRAPH(NPOINT,XCOORD,YCOORD,FVX,FVY,XINC,YINC)
      DIMENSION XCOORD(NPOINT),YCOORD(NPOINT),X(26),Y(26)
      COMMON/ISYMBL/NSYMB
C
C.   SUBROUTINE GRAPHS N POINTS WITHOUT ALTERING INPUT ARRAYS
C
      I1=NPOINT+1
      I2=NPOINT+2
C.   SET UP THE PLOTTING ARRAYS
      DO 10 I=1,NPOINT
      X(I)=XCOORD(I)
      Y(I)=YCOORD(I)
10    CONTINUE
C.   INSERT SCALING VALUES: FIRST VALUE AND AXIS INCREMENT
      X(I1)=FVX
      X(I2)=XINC
      Y(I1)=FVY
      Y(I2)=YINC
C.   PLOT
      CALL LINE(X,Y,NPOINT,1,+2,NSYMB)
```

```
C
      RETURN
      END
C
C
      SUBROUTINE ALPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
      DIMENSION CL(NCL)
      COMMON/ASCALE/FVALX,FVALY,DVX,DVY,FCM,DCM,FCD,DCD,
     +FCDT,DCDT,FCA,DCA,FCN,DCN
      COMMON/ISYMBL/NSYMB
      REAL ALPHA(24)
      DATA LMASK1/Z18FF/
C
C.   SUBROUTINE PLOTS ANGLE OF ATTACK AS A FUNCTION OF LIFT COEFFICIENT
C.
C.   READ MAX,MIN ALPHA, AND ALPHA DATA
      CALL GETDAT(AMAX,AMIN,ALPHA)
C.   GENERATE SCALE DATA
      FVALY=AMIN
      DVY=YSCALE(AMAX,AMIN)
      FVALX=CLMIN
      DVX=YSCALE(CLMAX,CLMIN)
      CALL NEWPEN(3)
C.   IF 1ST ALPHA PLOT, DRAW COORDINATE AXES
      IF(IFLAG.EQ.0) GO TO 10
C.   DRAW X,Y AXES
      CALL AXIS(0.,0.,'CL',-2,5.0,0.,FVALX,DVX)
      CALL AXIS(0.,0.,'ALPHA',5,5.0,90.,FVALY,DVY)
C.   DRAW GRID
      CALL NEWPEN(1)
      CALL GRID(0.,0.,5,1.0,5,1.0,LMASK1)
C
C.   PLOT ALPHA
10    CALL NEWPEN(2)
      CALL GRAPH(NCL,CL,ALPHA,FVALX,FVALY,DVX,DVY)
C
      RETURN
      END
C
C
      SUBROUTINE CMPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
      DIMENSION CL(NCL),CM(24)
      COMMON/ASCALE/FVALX,FVALY,DVX,DVY,FCM,DCM,FCD,DCD,
     +FCDT,DCDT,FCA,DCA,FCN,DCN
      COMMON/ISYMBL/NSYMBL
      DATA LMASK1/Z18FF/
C.
C.   SUBROUTINE PLOTS PITCHING COEFFICIENT AS A FUCNTION OF LIFT
```

```
C.   COEFFICIENT
C
C.   INPUT MAX,MIN CM AND CM DATA
        CALL GETDAT(CMMAX,CMMIN,CM)
C.   GENERATE SCALING DATA FOR PLOT: INIT.CM VAL(FCM) AND PLOT
C.   INCREMENT(DCM)
        FCM=CMMIN
        DCM=YSCALE(CMMAX,CMMIN)
C.   IF FIRST PLOT OF CM, DRAW COORDINATE AXES, OTHERWISE PROCEED
C.   WITH PLOTTING.
        IF(IFLAG.EQ.0) GO TO 10
        CALL NEWPEN(3)
        CALL AXIS(0.,0.,'CM',2,5.0,90.,FCM,DCM)
C.   DRAW THE GRID
        CALL NEWPEN(1)
        CALL GRID(0.,0.,5,1.0,5,1.0,LMASK1)
C
C.   PLOT CM
10      CALL NEWPEN(2)
        CALL GRAPH(NCL,CL,CM,FVALX,FCM,DVX,DCM)
C
C. DONE
        RETURN
        END
C
C
        SUBROUTINE CDPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
        DIMENSION CL(NCL),CD(24)
        COMMON/ASCALE/FVALX,FVALY,DVX,DVY,FCM,DCM,FCD,DCD,
       +FCDT,DCDT,FCA,DCA,FCN,DCN
        COMMON/ISYMBL/NSYMB
        DATA LMASK/Z18FF/
C
C.   SUBROUTINE PLOTS DRAG AS A FUCNTION OF LIFT
C
C.   INPUT MAX,MIN CD, AND CD DATA
        CALL GETDAT(CDMAX,CDMIN,CD)
C.   GENERATE SCALE DATA: INITIAL CD (FCD), PLOT INCREMENT (DCD)
        FCD=CDMIN
        DCD=YSCALE(CDMAX,CDMIN)
C.   IF 1ST PLOT OF CD, DRAW Y AXIS, OTHERWISE PROCEED
C.   WITH PLOT
        IF(IFLAG.EQ.0) GO TO 10
        CALL NEWPEN(3)
        CALL AXIS(0.,0.,'CDI',3,5.0,90.0,FCD,DCD)
C.   DRAW GRID
        CALL NEWPEN(1)
        CALL GRID(0.,0.,5,1.0,5,1.0,LMASK)
```

```
C.
C.   PLOT CD
10     CALL NEWPEN(2)
       CALL GRAPH(NCL,CL,CD,FVALX,FCD,DVX,DCD)
C
C.   DONE
       RETURN
       END
C
C
       SUBROUTINE CDTPLT(NCL,CL,CLMIN,CLMAX,IFLAG)
       DIMENSION CL(NCL),CDT(24)
       COMMON/ASCALE/FVALX,FVALY,DVX,DVY,FCM,DCM,FCD,DCD,
     +FCDT,DCDT,FCA,DCA,FCN,DCN
       COMMON/ISYMBL/NSYMB
       DATA LMASK/Z18FF/
C.
C.   PLOT TOTAL DRAG AS A FUNCTION OF ANGLE OF ATTACK
C
C.   INPUT CD DATA
       CALL GETDAT(CDTMAX,CDTMIN,CDT)
C.   GENERATE SCALING DATA
       FCDT=CDTMIN
       DCDT=YSCALE(CDTMAX,CDTMIN)
C.   IF 1ST PLOT DRAW Y AXIS
       IF(IFLAG.EQ.0) GO TO 10
       CALL NEWPEN(3)
       CALL AXIS(0.,0.,'CDTOTAL',7,5.0,90.,FCDT,DCDT)
       CALL AXIS(0.,0.,'CL',-2,5.0,0.0,FVALX,DVX)
C.   DRAW GRID
       CALL NEWPEN(1)
       CALL GRID(0.,0.,5,1.0,5,1.0,LMASK)
C
C.   PLOT CD TOTAL
    10 CALL NEWPEN(2)
       CALL GRAPH(NCL,CL,CDT,FVALX,FCDT,DVX,DCDT)
C
C.   DONE
       RETURN
       END
C
C
       SUBROUTINE CAPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
       DIMENSION CL(NCL),CA(24)
       COMMON/ASCALE/FVALX,FVALY,DVX,DVY,FCM,DCM,FCD,DCD,
     +FCDT,DCDT,FCA,DCA,FCN,DCN
       COMMON/ISYMBL/NSYMB
       DATA LMASK/Z18FF/
```

```
C
C.   PLOT CA AS A FUNCTION OF ANGLE OF ATTACK
C
C.   INPUT CA DATA
        CALL GETDAT(CAMAX,CAMIN,CA)
C.   GENERATE SCALE DATA
        FCA=CAMIN
        DCA=YSCALE(CAMAX,CAMIN)
C.   IF 1ST PLOT DRAW Y AXIS
        IF(IFLAG.EQ.0) GO TO 10
        CALL NEWPEN(3)
        CALL AXIS(0.,0.,'CA',2,5.,90.,FCA,DCA)
C.   DRAW GRID
        CALL NEWPEN(1)
        CALL GRID(0.,0.,5,1.0,5,1.0,LMASK)
C
C.   PLOT CA
   10 CALL NEWPEN(2)
        CALL GRAPH(NCL,CL,CA,FVALX,FCA,DVX,DCA)
C
C.   DONE
        RETURN
        END
C
C
        SUBROUTINE CNPLOT(NCL,CL,CLMIN,CLMAX,IFLAG)
        DIMENSION CL(NCL),CN(24)
        COMMON/ASCALE/FVALX,FVALY,DVX,DVY,FCM,DCM,FCD,DCD,
       +FCDT,DCDT,FCA,DCA,FCN,DCN
        COMMON/ISYMBL/NSYMB
        DATA LMASK/Z18FF/
C
C.   PLOT NORMAL FORCE AS A FUNCTION OF ANGLE OF ATTACK
C.
C.   INPUT CN DATA
        CALL GETDAT(CNMAX,CNMIN,CN)
C.   GENERATE SCALE DATA
        FCN=CNMIN
        DCN=YSCALE(CNMAX,CNMIN)
C.   IF 1ST PLOT, DRAW Y AXIS
        IF(IFLAG.EQ.0) GO TO 10
        CALL NEWPEN(3)
        CALL AXIS(0.,0.,'CN',2,5.,90.,FCN,DCN)
C.   DRAW GRID
        CALL NEWPEN(1)
        CALL GRID(0.,0.,5,1.,5,1.,LMASK)
C
C.   PLOT CN
```

```
   10 CALL NEWPEN(2)
      CALL GRAPH(NCL,CL,CN,FVALX,FCN,DVX,DCN)
C
C.  DONE
      RETURN
      END
/*
//GO.PLOTPARM DD *
 &PLOT SCALE=0.85 &END
/*
//GO.SYSIN DD *
```

LIST OF REFERENCES

1. Redmon, D.R., Tactical Missile Conceptual Design, Engineer's Thesis, Naval Postgraduate School, Monterey, Ca., 1980.

2. Platzek, H., Preliminary Solid Rocket Motor Design Techniques, NWC-TM-2953, Naval Weapons Center, China Lake, Ca., 1975.

3. Hill, P.G., and Peterson, C.R., Mechanics and Thermodynamics of Propulsion, Chapter 11, Addison-Wesley, 1965.

4. Barrere, M., and others, Rocket Propulsion, Chapter 2, Elsevier PublishingCo., 1960.

5. Saffell, B.F., and others, A Method for Predicting the Static Aerodynamic Characteristics of Typical Missile Configurations for Angles of Attack up to 180 Degrees, Naval Ship Research and Development Center Report 3645, Washington, D.C., 1971.

6. Fuhs, A.E., Fragmenting Warhead Design, Course Notes, Department of Aeronautics, Naval Postgraduate School, Monterey, Ca. 1976

INITIAL DISTIBUTION LIST

		No. Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0142 Naval Postgraduate School Monterey, California 93940	2
3.	Department Chairman, Code 67 Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93940	1
4.	Professor G. H. Lindsey, Code 67Li Department of Aeronautical Engineering Naval Postgraduate School Monterey, California 93940	2
5.	Lieutenant M. D. Sullivan, USN c/o D. M. Sullivan Star Route 1, Box 30-1 Dyke, Virginia 22935	1
6.	Lieutenant D. R. Redmon, USN 925 Skylark Ontario, California 91761	1

DATE
ILME