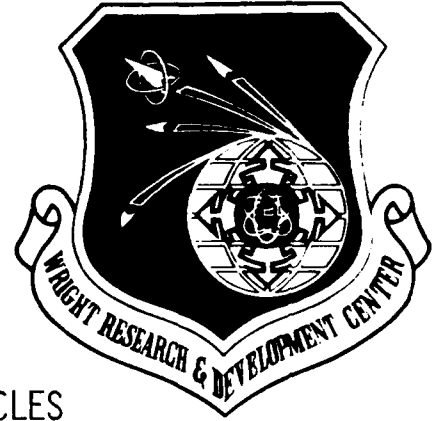


FILE COPY

2

WRDC-TR-89-2023

**AD-A218 157**



ETO - A TRAJECTORY PROGRAM FOR AEROSPACE VEHICLES

J. L. Leingang  
W. A. Donaldson  
K. A. Watson  
L. R. Carreiro, 1LT

Analysis and Applications Branch  
Advanced Propulsion Division

June 1989

DTIC  
ELECTE  
FEB 22 1990  
S D D  
CO

Final Report for Period January 1987 to July 1988

Approved for public release, distribution unlimited.

AERO PROPULSION AND POWER LABORATORY  
WRIGHT RESEARCH DEVELOPMENT CENTER  
AIR FORCE SYSTEMS COMMAND  
WRIGHT-PATTERSON AFB, OHIO 45433-6563

90 02 21 031



## REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSIFIED			1b. RESTRICTIVE MARKINGS None		
2a. SECURITY CLASSIFICATION AUTHORITY			3. DISTRIBUTION/AVAILABILITY OF REPORT Unlimited Distribution Approved for Public Release		
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE					
4. PERFORMING ORGANIZATION REPORT NUMBER(S) WRDC-TR-89-2023			5. MONITORING ORGANIZATION REPORT NUMBER(S)		
6a. NAME OF PERFORMING ORGANIZATION Aero Propulsion and Power Laboratory		6b. OFFICE SYMBOL (If applicable) WRDC/POPA	7a. NAME OF MONITORING ORGANIZATION		
6c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB OH 45433-6563 Phone: (513) 255-4171			7b. ADDRESS (City, State and ZIP Code)		
8a. NAME OF FUNDING/SPONSORING ORGANIZATION		8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER		
8c. ADDRESS (City, State and ZIP Code)			10. SOURCE OF FUNDING NOS.		
			PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.
			61101F & 62204F	3012	08
11. TITLE (Include Security Classification) ETO - A TRAJECTORY PROGRAM FOR AEROSPACE VEHICLES			WORK UNIT NO.	93	
12. PERSONAL AUTHOR(S) L. Leingang, W. A. Donaldson, K. A. Watson, L. R. Carreiro, 1Lt					
13a. TYPE OF REPORT FINAL		13b. TIME COVERED FROM Jan 87 to Jul 88		14. DATE OF REPORT (Yr., Mo., Day) June 1989	15. PAGE COUNT 90
16. SUPPLEMENTARY NOTATION					
17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)		
FIELD	GROUP	SUB. GR.	Trajectory Program, Earth-to-Orbit, Cruise Vehicle Performance, Airbreathing Propulsion, Rocket Propulsion, Horizontal Takeoff, Single-Stage, Two-Stage Vehicles. (EG)		
19. ABSTRACT (Continue on reverse if necessary and identify by block number) This is a formulation report and user guide for a two-degree-of-freedom flight trajectory computer program. The program, called ETO, runs on IBM/compatible XT/AT microcomputers. It is designed for very short run times, i.e., less than 1 to 3 minutes. Horizontal or low angle takeoff single-stage or two-stage vehicles can be studied. Ascent to earth orbit or constant speed cruise can be studied. Any combination of airbreathing or rocket propulsion may be used. Working from tabular data for propulsion performance, aerodynamics, and vehicle weight and fuel, the governing equations of motion are integrated forward in time to track velocity, altitude, range, flight path angle, and weight. A time ordered history of flight parameters is printed out. The program is available in Quick Basic and Fortran versions.					
20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED <input checked="" type="checkbox"/> SAME AS RPT. <input type="checkbox"/> DTIC USERS <input type="checkbox"/>			21. ABSTRACT SECURITY CLASSIFICATION UNCLASSIFIED		
22a. NAME OF RESPONSIBLE INDIVIDUAL John L. Leingang			22b. TELEPHONE NUMBER (Include Area Code) (513) 255-4171	22c. OFFICE SYMBOL WRDC/POPA	

## FOREWORD

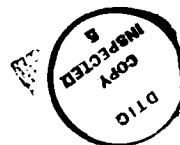
This report describes the development and use of a two-degree-of-freedom flight vehicle trajectory program called ETO (Earth-to-Orbit). The program is intended primarily for study of the payload/range performance of airbreathing horizontal takeoff vehicles. The vehicles may be single stage or two stage and may be intended to reach earth orbit or to cruise at constant speed. Any combination of airbreathing propulsion and rocket propulsion may be used. Low angle ground launch or air launch vehicles may also be studied.

This report discusses the formulation of the computer program and gives detailed instructions on its use. Several example cases are included. The computer program is intended to be used on IBM or compatible -XT or -AT class microcomputers. The program is available on a 5¼-inch 360K byte floppy disk from the authors.

The flight vehicles used as examples in this report are not intended to represent any known or projected vehicle or level of technology. No inference should be made that the example vehicles typify current or projected capabilities.

This work was performed in the Analysis and Applications Branch, Advanced Propulsion Division of the WRDC (Wright Research Development Center) Aero Propulsion Laboratory. The work was done under in-house work unit 30120893. The work was begun in January 1987 and completed in July 1988.

The principal author was John L. Leingang. Co-authors were: Wayne A. Donaldson, Kenneth A. Watson, and 1Lt Louis R. Carreiro.



Accession For	
THIS CRA&I	<input checked="" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution	
Availability Codes	
Dist	Availability Codes
A-1	

## TABLE OF CONTENTS

<u>SECTION</u>	<u>PAGE</u>
1.0 INTRODUCTION AND SUMMARY	1
2.0 DEVELOPMENT AND DESCRIPTION OF PROGRAM ETO	2
2.1 Program Language and Compilers	2
2.2 Equations of Motion in a Plane	3
2.3 Aerodynamics	7
2.4 Propulsion	8
2.5 Flight Phases	8
2.6 Stopping Conditions	10
3.0 RUNNING PROGRAM ETO	16
3.1 Hardware Requirements	16
3.2 Using the Stand Alone.EXE Version of ETO	18
3.3 Running ETO within the QUICKBASIC 4.0 Environment	23
3.4 Input Data Files for a Single- or First-Stage Vehicle	23
3.5 Input Data Files for a Second-Stage Vehicle	32
3.6 Sample Input Data Files and Program Output	33
3.7 Sample Program Output	41
4.0 LISTING OF PROGRAM ETO	48
5.0 CROSS REFERENCED LISTING OF PROGRAM ETO VARIABLES	68
6.0 DEFINITION OF PROGRAM ETO VARIABLES	73

## LIST OF ILLUSTRATIONS

FIGURE		PAGE
2-1	Free Body Diagram of Flight Vehicle	11
2-2	Program ETO Block Diagram	12
2-3	Vehicle Flight Trajectory	13
2-4	Angle-of-Attack Command	14
2-5	Acceleration Command	15
3-1	Tabular Input Data	35

## LIST OF TABLES

TABLE		PAGE
1	Airbreather Specific Impulse	24
2	Airbreather Maximum Fuel Equivalence Ratio	25
3	Zero Lift Drag Coefficient	25
4	Friction Drag Increment as a Function of Attitude	25
5	Lift Curve Slope as a Function of Mach Number	25
6	Value of Drag Due to Lift Factor K, as a Function of Mach Number	26
7	Airbreather Inlet Capture Area Ratio vs Mach Number and Angle of Attack	26
8	Rocket Specific Impulse vs Altitude	26
9	Maximum Rocket Propellant Flow Rate vs Mach Number	26

## ABBREVIATIONS AND SYMBOLS

$A_c$	Inlet cowl area, $ft^2$
$A_o$	Inlet air capture area, $ft^2$
$A_o/A_c$	Inlet air capture ratio, $ft^2/ft^2$
$A_{Ref}$	Vehicle aerodynamic reference area, $ft^2$
$C_D$	Vehicle drag coefficient
$C_{D0}$	Vehicle zero lift drag coefficient
$\Delta C_D$	Vehicle drag increment due to friction
$C_L$	Vehicle lift coefficient
$C_{L\alpha}$	Vehicle lift curve slope, 1/deg
$D$	Vehicle drag, lbf
$dt$	Integration step, secs
$f/a$	Airbreather fuel-air ratio
$(f/a)_{STOICH}$	Airbreather stoichiometric fuel-air ratio
$F_c$	Vehicle centrifugal force relative to earth, lbf
$F_n$	Force component acting on vehicle normal to direction of motion, lbf
$F_v$	Force component acting on vehicle along axis of motion, lbf
$g$	Acceleration due to gravity, $ft/sec^2$
$g_c$	Acceleration constant, $32.174 \frac{lbm/ft}{lbf/sec^2}$
$h$	Height, feet
$h_o$	Vehicle altitude, feet
$I_{sp}$	Specific impulse, $\frac{lbf}{lbm/sec}$
$I_{spA}$	Airbreather specific impulse, $\frac{lbf}{lbm/sec}$



## ABBREVIATIONS AND SYMBOLS (Cont'd)

$I_{spR}$	Rocket specific impulse, $\frac{\text{lb}f}{\text{lbm}/\text{sec}}$
$K$	Vehicle aerodynamic drag due to lift constant
$L$	Vehicle lift, $\text{lb}f$
$M_e$	Mass of the earth, $5.98 \times 10^{27}$ grams
$M_0$	Flight Mach number
$q$	Atmospheric dynamic pressure, $1/2 \frac{\rho_0 v_0^2}{g_c}$ , $\text{lb}f/\text{ft}^2$
$q_{\text{cmd}}$	Vehicle commanded dynamic pressure, $\text{lb}f/\text{ft}^2$
$q_0 \text{ max}$	Vehicle commanded maximum dynamic pressure, $\text{lb}f/\text{ft}^2$
$q_0 \text{ min}$	Vehicle commanded minimum dynamic pressure, $\text{lb}f/\text{ft}^2$
$R$	Vehicle flight range, nautical miles
$R_e$	Earth radius, $6.23 \times 10^8$ cm, $2.0899 \times 10^7$ feet
$T$	Vehicle thrust, $\text{lb}f$
$t$	Flight time, secs
$v$	Velocity, $\text{ft}/\text{sec}$
$v_0$	Vehicle flight velocity, $\text{ft}/\text{sec}$
$v_n$	Vehicle velocity normal to flight path, $\text{ft}/\text{sec}$
$v_0 \text{ temp}$	Vehicle velocity at which $q_0$ is commanded to lower values to simulate a constant wall temperature flight path, $\text{ft}/\text{sec}$
$W$	Vehicle weight, $\text{lbm}$
$\dot{W}$	Vehicle rate of weight change, $\text{lbs}/\text{sec}$
$\dot{W}_{\text{AIR}}$	Airbreather propulsion air flow rate, $\text{lbm}/\text{sec}$
$\dot{W}_f$	Airbreather fuel flow, $\text{lbm}/\text{sec}$
$\dot{W}_p$	Rocket propellant flow, $\text{lbm}/\text{sec}$
$\alpha$	Vehicle angle of attack, deg

ABBREVIATIONS AND SYMBOLS (Concluded)

$\gamma$	Vehicle flight path angle, deg
$\phi$	Airbreather fuel equivalence ratio, $(f/a)/(f/a)_{\text{STOICH}}$
$\rho_0$	Atmospheric density, $\text{lbm/ft}^3$

## 1.0 INTRODUCTION AND SUMMARY

A two-degree-of-freedom (2-DOF) digital computer program called ETO (Earth-to-Orbit) has been written. Five equations of motion describing weight, velocity, flight path angle, altitude, and range of a vehicle are integrated by a second-order routine. The program assumes a spherical, nonrotating earth. The primary purpose of the program is to evaluate horizontal takeoff earth-to-orbit vehicles of single- or two-stage configurations. Options are available in the program to evaluate cruise vehicles of various types, aircraft, and ground or air launched missiles.

Because the vehicle motion is in a single plane, i.e., 2-DOF, simplified aerodynamics are used in the form of user-input coefficients for zero-lift drag, the lift curve slope, drag due to lift factor, and friction drag. Airbreathing propulsion performance is user input as specific impulse vs fuel equivalence ratio and flight Mach number. These data along with input values of inlet air capture area are sufficient to define airbreathing thrust at every integration point. Rocket thrust is user input as specific impulse vs altitude and propellant flow rate vs Mach number. Airbreathing and rocket thrust may be used separately or simultaneously in any phase of the flight.

A number of options are available for tailoring earth-to-orbit missions or cruise missions. For earth-to-orbit missions, options exist to climb to orbit at a commanded constant dynamic pressure, along a varying dynamic pressure path, or a ballistic ascent beginning at a prescribed velocity. For cruise aircraft, constant altitude cruise or maximum L/D cruise may be specified.

For ground-launched missiles, launch at a prescribed initial angle is used. For air launches, an initial velocity and angle are prescribed. Missiles may be single stage or two stage.

A major objective of the program was to achieve reasonable accuracy and very short run times on microcomputers. Run times of 1 to 3 minutes were sought.

## 2.0 DEVELOPMENT AND DESCRIPTION OF PROGRAM ETO

### 2.1 Program Language and Compilers

Initially, the program was written in Microsoft GW-BASIC<sup>\*</sup> with strict adherence to syntax that would permit it to run under the BASICA interpreter for debugging. Once a suitable version was obtained, it was compiled with the Microsoft BASIC Compiler, BASCOM. A little later in the program development, the Microway 87 BASIC compiler was obtained. It supported the Intel 8087/80287 floating point math chip, and program execution speed was improved greatly. Near the end of the code development cycle (a refinement phase), the Microsoft Quick BASIC 4.0 compiler and the Borland Turbo BASIC 1.0 compilers became available. They both support the 8087/80287 math chip, do not require line numbers in the codes, and support more structured programming. The code as listed in this report now conforms to the syntax requirements of the Microsoft Quick BASIC 4.0 compiler. It will not run under the BASIC interpreter. The integrated editing, debugging, and run environments of new Microsoft or Borland compilers greatly ease code development. They are available at prices as low as \$65.

The source code is about 45,000 bytes, and a stand-alone executable file of the code is about 85,000 bytes. On -XT or -AT compatible machines with the 8087/80287 floating point math chip, very short run times are possible. The following gives some results obtained for the example problem TEST1.DAT, an earth-to-orbit flight:

<u>Computer</u>	<u>Run Time, secs</u>
<u>Zenith Z248 (AT class)</u>	
8 MHz 80286/5 MHz 80287	38 secs
8 MHz 80286/no math chip	210 secs
<u>Sanyo 550 (Turbo XT class)</u>	
7.2 MHz V-20 /8 MHz 8087	60 secs
<u>ACER 1100</u>	
16 MHz 80386/16 MHz 80387	12 secs
<u>Vendex Turbo XT</u>	
8 MHz 8088/no math chip	916 secs

\* At time of publication, a Fortran version is now available. Contact authors for details.

## 2.2 Equations of Motion in a Plane

Figure 2-1\* is a free body diagram of a flight vehicle in motion about a spherical, nonrotating earth. The vehicle, of mass  $W/g_c$ , is at a distance  $(R_e + h_0)$  from the center of the earth of mass  $M_e$ . The vehicle is moving along a path at angle  $\gamma$  relative the earth. Vehicle thrust,  $T$ , acts along the axis of the vehicle. The vehicle is inclined to its axis of motion by its angle of attack,  $\alpha$ . Lift,  $L$ , is exerted in a direction normal to the axis of motion, and drag,  $D$ , is exerted along the axis of motion. The weight,  $W$ , and the centrifugal force,  $F_c$ , act oppositely and along the radius to the earth.

The motion of the vehicle is described by its acceleration. It is most convenient to resolve the acceleration along the motion axis (the velocity axis) and along a direction normal to the motion axis. The acceleration along the motion axis will be defined from Newton's law for acceleration:

$$\frac{W}{g_c} \frac{dv}{dt} = \Sigma F_v$$

Resolving all of the forces to their components along the motion axis gives

$$\frac{W}{g_c} \frac{dv}{dt} = T \cos \alpha - D - W \frac{g}{g_c} \sin \gamma + F_c \sin \gamma$$

The acceleration normal to the axis of motion will be given by Newton's law for acceleration:

$$\frac{W}{g_c} \frac{dv_n}{dt} = \Sigma F_n$$

The rate of change of flight path angle is

$$\frac{d\gamma}{dt} = \frac{dv_n}{dt} \cdot \frac{d\gamma}{dv_n}$$

---

\*Section 2 figures begin on page 11.

Since

$$\frac{dY}{dv_n} = \frac{1}{v_o}$$

We have

$$\frac{dY}{dt} = \frac{dv_n}{dt} \cdot \frac{1}{v_o}$$

and

$$\frac{dv_n}{dt} = v_o \frac{dY}{dt}$$

So that Newton's law gives

$$\frac{W}{g_c} v_o \frac{dY}{dt} = \sum F_n$$

The summation of normal forces gives

$$\frac{W}{g_c} v_o \frac{dY}{dt} = L + T \sin \alpha - W \frac{g}{g_c} \cos \gamma + F_c \cos \gamma$$

The above two expressions, one for the axial acceleration and one for the normal acceleration, are sufficient to describe the time-dependent motion of the vehicle but are not yet sufficient to allow solution.

An expression for the time rate of change of weight is needed, and since drag and lift are also dependent on altitude, an expression for the time rate of change of altitude is needed. The time rate of change of range is not required but is of

interest for cruising flight. The final system of five equations to be integrated becomes

Weight change:

$$\frac{dW}{dt} = - \dot{W}$$

Velocity change:

$$\frac{dv_o}{dt} = \frac{g_c}{W} \left[ T \cos \alpha - D - W \frac{g}{g_c} \sin \alpha + F_c \sin \gamma \right]$$

Flight path angle change:

$$\frac{d\gamma}{dt} = \frac{g_c}{v_o W} \left[ L + T \sin \alpha + F_c \cos \gamma - W \frac{g}{g_c} \cos \gamma \right]$$

Altitude change:

$$\frac{dh}{dt} = v_o \sin \gamma$$

Range increment:

$$\frac{dR}{dt} = v_o \cos \gamma$$

The first four equations are coupled and must be integrated by employing an algorithm for integrating first-order systems of differential equations. As was stated above, the fifth equation, which tracks the range, is of interest for cruising flights. We decided to employ a simple two-step integration scheme (second-order Runge-Kutta, sometimes called Heun's method), and we hoped that its accuracy would be compatible with the choice of large time-integration steps, i.e., 2 to 5 seconds, and the desire to keep program execution time short.

The overall strategy for solving the problem of a flight trajectory is to integrate the coupled equations of motion forward in time, enforcing constraints/limits on values of the variables until some predefined value of one of the variables is reached and triggers a stop to the problem. The solution to the trajectory is the time-ordered history of values of all of the relevant variables up to the stopping condition.

At each integration time step, current values are needed for

$\dot{W}$  = rate of weight change (usually propellant flow)

$W$  = weight

$T$  = thrust

$D$  = drag

$L$  = lift

$F_c$  = centrifugal force

$\alpha$  = vehicle angle of attack

$\gamma$  = vehicle flight path angle

$v_0$  = flight velocity

$g$  = acceleration due to gravity (varies with altitude)

Constraints are applied by defining limits for such variables as  $\alpha$  and  $\gamma$ . Thrust may be limited by commanded axial acceleration or by maximum fuel flow. Other constraints such as maximum and minimum flight dynamic pressure impose adjustments on  $\alpha$  and thrust. Basically, a trajectory can be flown by using two commands, angle-of-attack and axial acceleration ("stick and throttle"). The trajectory proceeds through a series of flight phases (up to five are used in ETO) which



enforce the problem constraints. The main program module enforces the constraints appropriate to each flight phase. At the end of each integration step, the program returns to the beginning of this main module and "falls through" to the current flight phase to obtain the appropriate angle-of-attack and acceleration commands. The flight proceeds through phases based on computed events and on inputs from the vehicle input data file.

The flow diagram of Figure 2-2 shows the basic program strategy in which lift, drag, and the flight conditions ( $h_0, v_0$ ) are defined; then the main module is entered to obtain angle-of-attack and acceleration commands, followed by generation of the current thrust value and finally going to the integration module. A runtime output step is taken, and the program returns to update lift, drag, and the flight condition ( $h_0, v_0$ ). This process is repeated until the problem is completed.

The remaining parts of Section 2 give more specifics on how aerodynamics, propulsion, flight phases, and stopping conditions are handled.

### 2.3 Aerodynamics

Because the program is two-degree-of-freedom, i.e., motion in a plane, only trim aerodynamics are used. The term "trim aerodynamics" means use of lift and drag coefficient values at which the sum of all moments about the vehicle are zero. A data table of  $C_{D0}$  vs Mach number, a data table of  $C_{L\alpha}$  and  $K$  vs Mach number and a table of friction increment to  $C_{D0}$  vs altitude defines the aerodynamics.

At a given Mach and altitude, the data tables are consulted for  $C_{D0}$ ,  $C_{L\alpha}$ ,  $K$  and  $\Delta C_D$  then used in the following equations:

$$\begin{aligned}C_L &= \alpha C_{L\alpha} \\L &= C_L q_0 A \text{ Ref} \\C_D &= C_{D0} + K C_L^2 + \Delta C_D \\D &= C_D q_0 A \text{ Ref}\end{aligned}$$

## 2.4 Propulsion

Airbreathing propulsion is described by tables of  $I_{sp}$  vs Mach number and equivalence ratio ( $\phi$ ), and a schedule of maximum  $\phi$  vs Mach number. This information, along with tables of inlet capture area ratio ( $A_o/A_c$ ) vs Mach and angle of attack, the inlet cowl area ( $A_c$ ), and stoichiometric fuel-air ratio are sufficient to establish thrust, fuel flow, and airflow at each integration time step. At each time step, the following steps are taken to obtain thrust and fuel flow:

$$A_o = \left( \frac{A_o}{A_c} \right) A_c$$

$$\dot{W}_{AIR} = \rho_o v_o A_o$$

at a specified  $\phi$ :

$$f/a = \phi (f/a)_{STOICH}$$

$$\dot{W}_f = (f/a) \dot{W}_{AIR}$$

$$T = \dot{W}_f I_{sp}$$

An iteration scheme is used to adjust  $\dot{W}_f$  by testing if  $T$  is greater or less than required to produce the commanded acceleration.

Rocket propulsion is described by tables of  $I_{sp}$  vs altitude and by propellant flow rate ( $\dot{W}_p$ ) vs Mach number:

$$T = \dot{W}_p I_{sp}$$

Rocket thrust is modulated in response to the acceleration command. If the airbreather and the rocket are operating simultaneously, the rocket is throttled first to reduce thrust to the commanded value.

## 2.5 Flight Phases

Figure 2-3 shows a typical succession of flight phases as discussed in the following paragraphs.

Phase I is the takeoff roll. The vehicle accelerates to the takeoff/rotation velocity at zero lift and zero commanded climb. Once the rotation velocity is achieved, control passes to Phase II.

Phase II is a climb at a commanded pull-up load factor and axial acceleration. After the rotation velocity is exceeded in Phase I, the vehicle is permitted to pitch up to a maximum angle of attack or a commanded load factor, whichever limit is controlling. The vehicle climbs and accelerates to a commanded maximum subsonic velocity which cannot be exceeded until a specified altitude is reached. Once this Mach and altitude threshold is reached, angle of attack is commanded to zero, and the vehicle accelerates on a ballistic path (no induced drag) through the transonic region until a commanded dynamic pressure is reached. This completes Phase II.

Phase III is a climb and acceleration at a commanded dynamic pressure,  $q_{com}$ . This value of  $q$  may be constant or may be made linearly variable with velocity, beginning at a specified velocity. A maximum and a minimum value of  $q$  is specified. A control algorithm generates angle-of-attack and acceleration commands at each integration step to keep the vehicle within the minimum and maximum  $q$  limits. The acceleration command is generated by examining whether  $q$  is greater or less than the commanded value and whether, in either case, it is getting larger or smaller. The algorithm for implementing this control is shown in Figure 2-4. A more detailed discussion is in Section 3.4. The output of this algorithm is a new value of  $\alpha$ , angle of attack, to be used in the next integration step. With proper choices for the adjustment coefficients, a relatively smooth ascent path can be achieved. Once the  $\alpha$  command is generated, an acceleration command is generated. The algorithm for this can be visualized from Figure 2-5. Basically, the strategy is that if  $q$  is greater than the maximum, then no acceleration is permitted. If  $q$  is less than the commanded value, then the maximum available acceleration is commanded. In the region where  $q$  is greater than commanded but less than maximum, a proportional reduction in acceleration is commanded as suggested by Figure 2-5. Acceleration along the commanded  $q$  path continues until a cruise command or a ballistic path command is made.

Phase IV generates a commanded zero acceleration to maintain a specified velocity at a commanded  $q$  (altitude) or to maintain velocity at maximum L/D. Maximum L/D cruise

is accomplished by finding the angle of attack that produces maximum L/D and then maintaining it to fuel exhaustion.

Phase V is a ballistic ascent and is normally commanded after having proceeded along a Phase III commanded  $q$  path. It is used primarily for earth-to-orbit flights. At a commanded velocity, the vehicle pulls up to a specified angle of attack, maintaining it until a specified flight path angle is achieved. Once the flight path angle is reached, the angle of attack is commanded to zero and the flight is ballistic, i.e., angle of attack equals zero from this point on. During Phase V an acceleration command different from that applied in Phase III may be used.

## 2.6 Stopping Conditions

Three stopping conditions are utilized: fuel exhaustion, achievement of commanded velocity, and achievement of the orbital condition. When the vehicle weight falls to the specified final value, the program stops and issues a message to that effect. For two-stage vehicles, the first stage is usually flown to a prescribed velocity or to fuel exhaustion. When either occurs, the program halts, issues a message that the first-stage flight has been completed and asks if you care to continue the problem with the second stage.

The orbital condition is defined by the centrifugal force becoming equal to the weight. When this occurs the program stops and advises that orbit has been achieved. Because the code is assuming a nonrotating earth, the orbital velocity required is always equivalent to be that of a polar orbit. Normally this will be slightly under 25,900 fps. The exact value depends on altitude. An Eastern equatorial orbit would be the minimum required orbital velocity and would be on the order of 24,500 fps. The user could specify a final velocity less than the polar orbit value that is appropriate to the orbit inclination of interest to him.

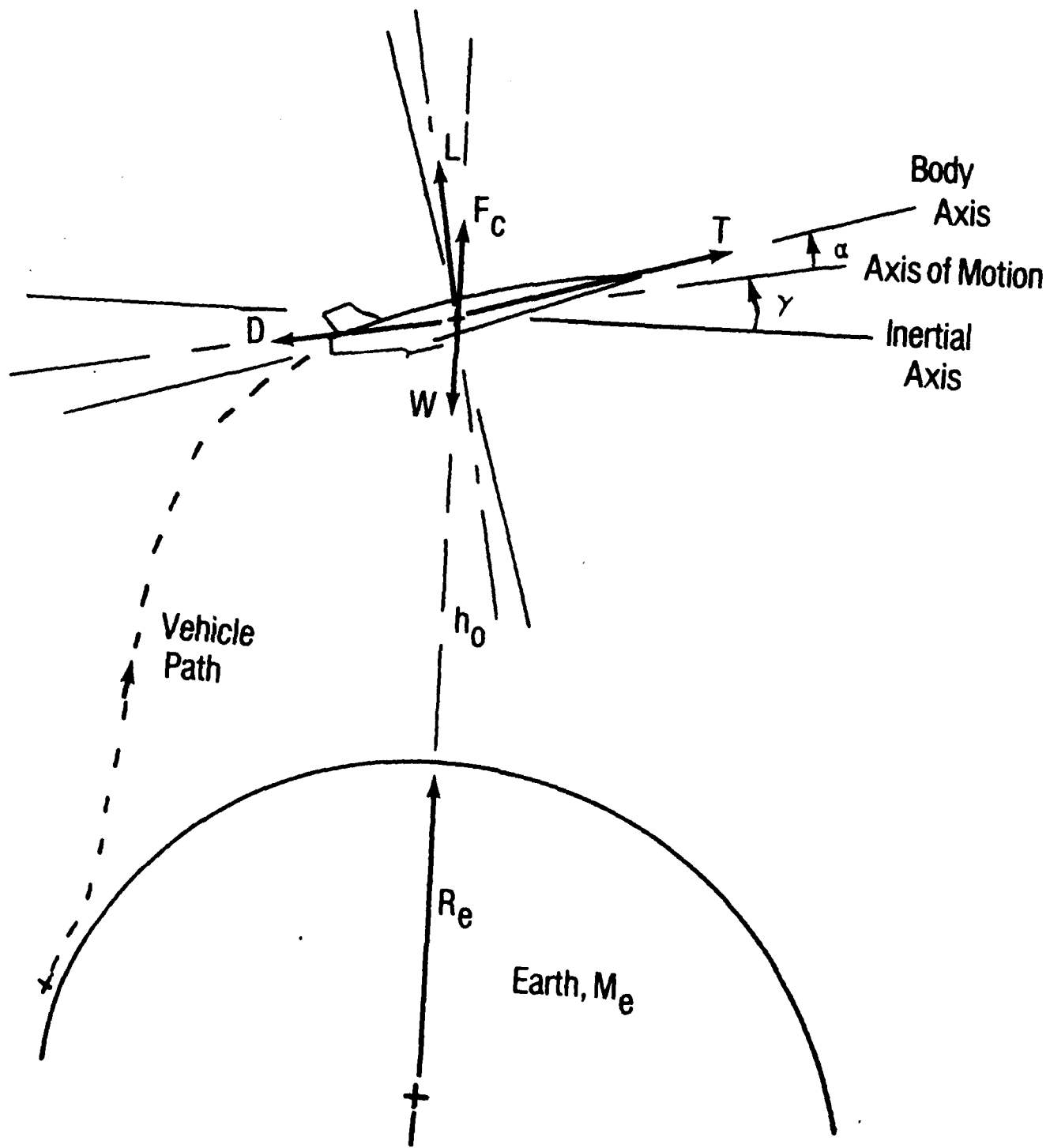


Figure 2-1. Free Body Diagram of Flight Vehicle

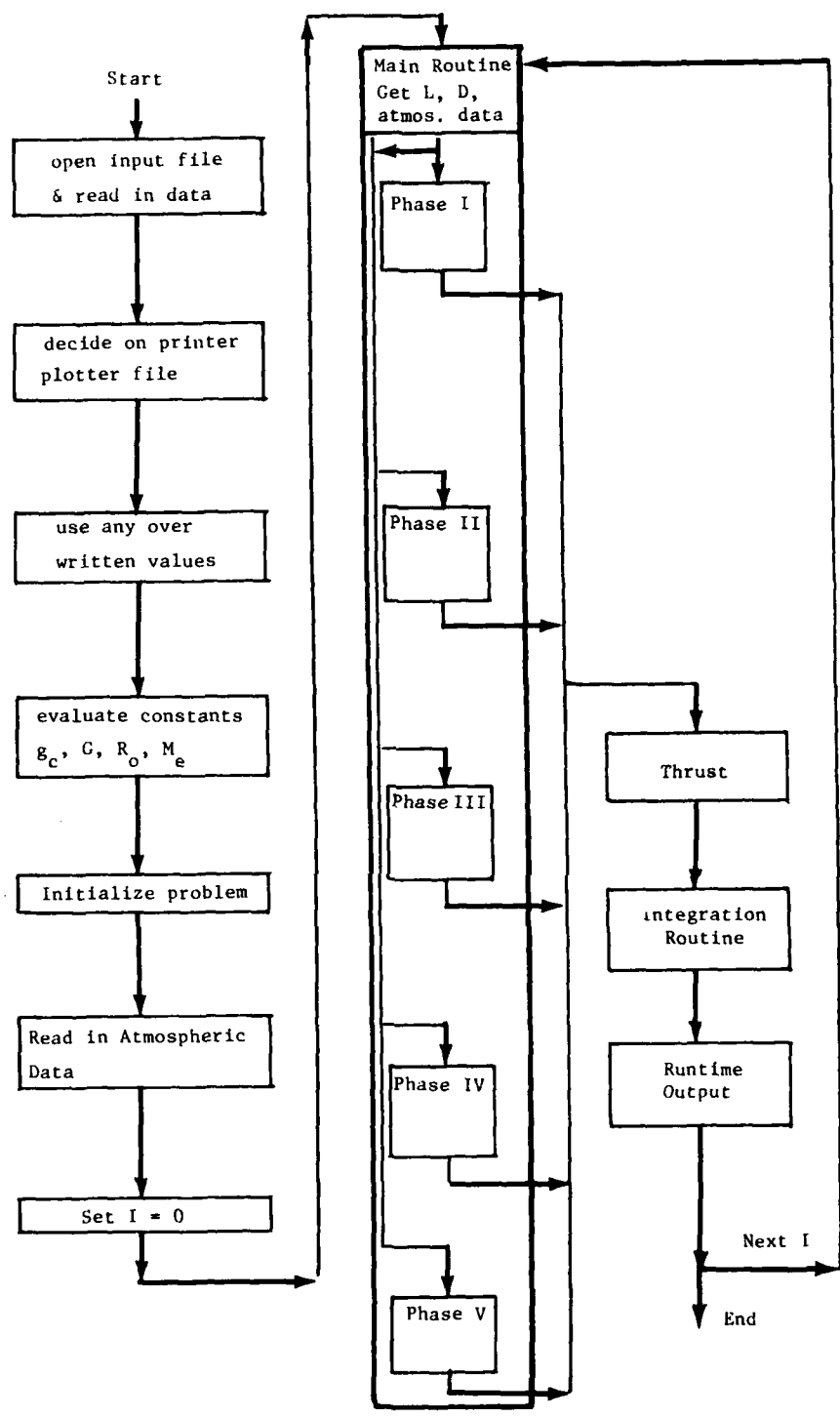


Figure 2-2. Program ETO Block Diagram

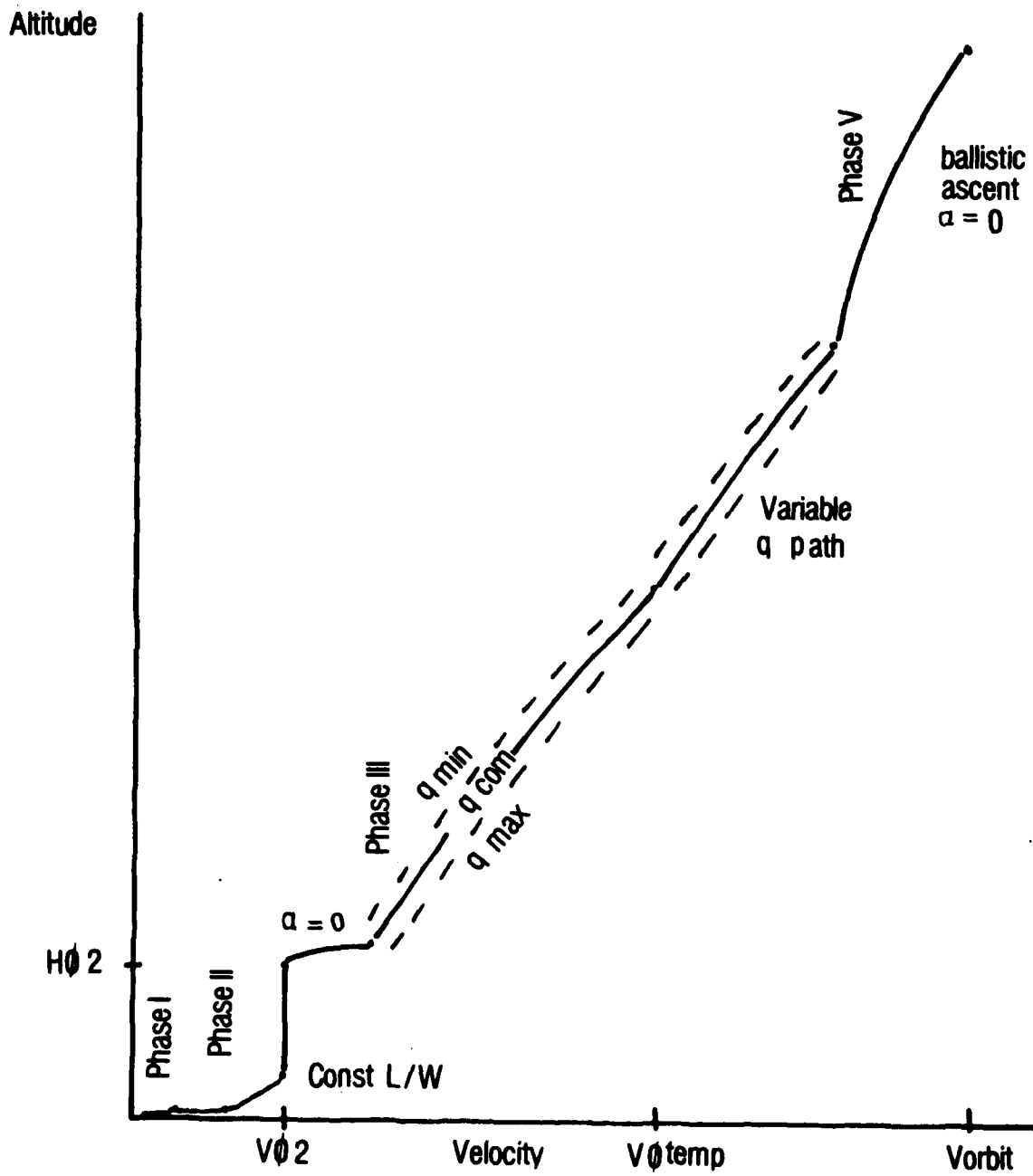


Figure 2-3. Vehicle Flight Trajectory

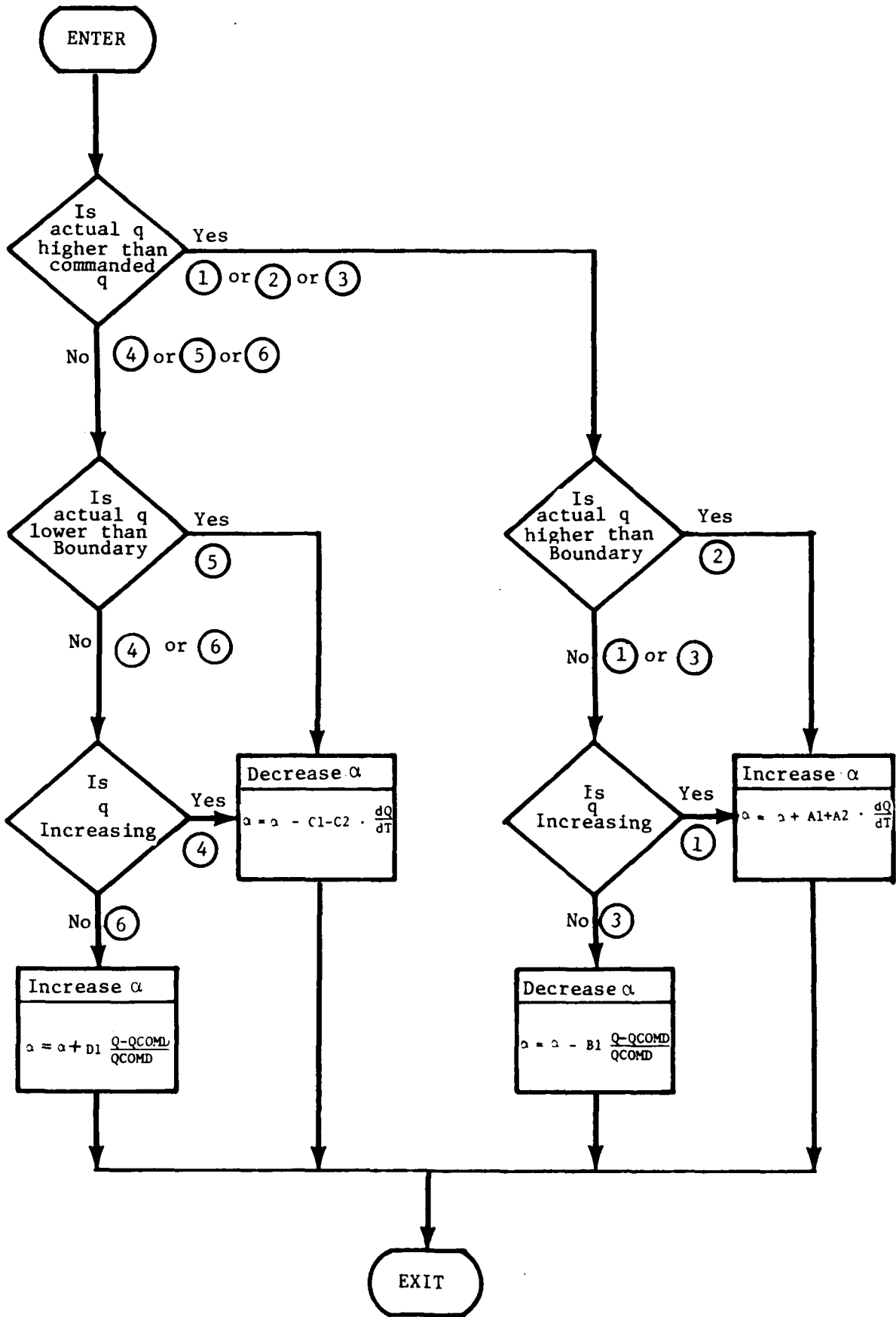


Figure 2-4. Angle-of-Attack Command



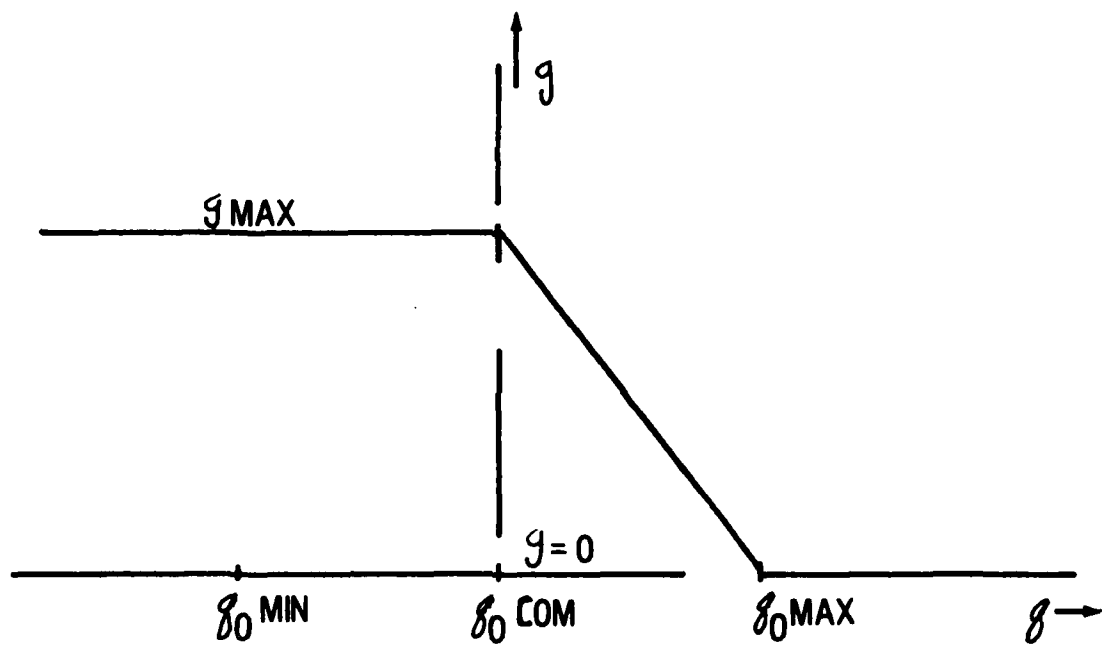


Figure 2-5. Acceleration Command

### 3.0 RUNNING PROGRAM ETO

#### 3.1 Hardware Requirements

The distribution disk is a 360K IBM/MS-DOS format 5 ¼-inch floppy disk which contains the following files:

ET013.BAS     (See Note Below)  
ET013.BAC  
ET013.EXE  
TEST1.DAT  
STG1.DAT  
STG2.DAT  
STG2BAL.DAT  
ETO.TAB  
GUID14.BAS  
ET0123.BAS  
ET0123.EXE

ET013.BAS is the program source code. ET013.BAC is a backup copy of the source code and ET013.EXE is a stand-alone executable version of the program. The designation ET013 reflects the version number of program ETO available at the time of this report. Significant upgrades will have higher numbers. The files with a .DAT suffix are sample input data files. Each data file illustrates a significant feature of the program. Each of these .DAT files is discussed in Section 3.6. The file ETO.TAB is an alphabetical listing with description of all the program variables. The file GUID14.BAS is an optional angle-of-attack module for flight Phase III which the user may substitute for GUID13 in the trajectory program (Section 2.5). ET0123.BAS and ET0123.EXE are the source code and executable file, respectively, which may be used to generate a file readable by LOTUS 123 of the program output file. This is included for users who may wish to use LOTUS 123 to generate plots of the program output.

NOTE: Since completion of this report FORTRAN versions are also available. Contact the authors for more information.

The minimum hardware equipment for running the stand-alone .EXE version of ETO is an IBM or compatible XT or AT microcomputer using MS.DOS 2.1 or higher. DOS versions earlier than 2.1 will not work properly. One hundred percent IBM compatibility does not appear to be required, however. A single disk drive (360K, 5¼-inch floppy disk) and 256K RAM are needed. It may be possible to run the program with less than 256K RAM but this has not been demonstrated. A math coprocessor, 8087/80287, is highly desirable for reducing run times but is not required. A printer is required for printed output of the time history of a flight. An Epson FX or compatible dot matrix printer is the best choice. With this printer letter-width paper can be used because this printer accepts the program commands to set compressed print. If your letter-width printer cannot respond to the program command for compressed print and cannot be set by a hardware switch, you will need to employ a wide-carriage printer.

The most convenient environment for running the program is from within the Microsoft Quick BASIC 4.0 editor/compiler. The computer will have to be 100 percent IBM compatible. The program was developed using this compiler, and the stand-alone .EXE version was generated by it. MS-DOS versions 2.11, 3.1, and 3.2 were used in the development. It does require more computer hardware resources, however. Microsoft advises that 320K RAM is needed. The minimum RAM used during development was 512K. It was satisfactory, but it is not known how close this is to the minimum needed for this program. Two 360K floppy disk drives are needed. A hard drive is much more convenient and permits quicker compilation. A RAM drive is also convenient. Quick BASIC 4.0 senses if a math coprocessor (8087/80287) is present and uses it if available. Version 4.0 of Quick BASIC is distinctly different from earlier versions. These earlier versions are not recommended.

Some limited development work was done using the Borland Turbo BASIC 1.0 editor/compiler. It does not require 100 percent IBM compatibility. It provides an integrated program development and compiler environment. Its choice over the Microsoft product is a matter of personal preference. To use the Borland compiler, the program variable AT would have to be changed to some other name. AT is a reserved work in Turbo BASIC. Some work would be needed in formatting the screen and printer output. Either the Microsoft or the Borland compiler will be required if the user desires to modify the code and recompile it.

### 3.2 Using the Stand-Alone .EXE Version of ETO

The first step should be to make a backup copy of the distribution disk. The backup copy should be used for running the program. We recommended that you copy your version of the MS-DOS COMMAND.COM file to the backup disk. The reason for this is that the program uses the BASIC "SHELL" command to temporarily exit to DOS to copy the input data file to the printer when the program prints out results. The program needs access to COMMAND.COM on the default drive in order for the "SHELL" command to work properly. MS-DOS 2.1 or greater is required.

Once the above has been accomplished, the program is ready to run. At the DOS prompt, type ET013 and press RETURN. The screen clears, and the following will be displayed:

IS THIS A SINGLE-STAGE (1), OR A TWO-STAGE VEHICLE (2)?

Respond by entering a 1 or a 2 and pressing the ENTER key. If you responded to the above with a 1, you will be prompted with:

SINGLE-STAGE FILE NAME?

Enter the name of the input file following the DOS file name convention, DRIVE: NAME.EXT. The input file does not need to be on the default drive.

If you had responded with a 2 to the first question, you would be asked the following:

FIRST-STAGE FILE NAME?

Again enter input file name as DRIVE: NAME.EXT and press RETURN key.

You will then be prompted with:

SECOND-STAGE FILE NAME?

Respond as above.

Next you will be asked:

"Do you want a listing of tabular input data (Y/N)?"

Normally you will not want this and will respond with a "no" (N).

If you respond with "yes" (Y), then the nine data tables in the first/single-stage input file are printed out. The printout is convenient for examining the data tables for accuracy.

The next prompt will be:

"Do you want printer output during run (Y/N)?"

As above, you will usually not want this. The printer output format is not particularly convenient and the program runs much more slowly in this mode. The principal use for this printout is diagnosis of problems early in the flight. The program can be stopped after the desired flight time has passed. The printer output interval is the same as the screen print interval, DELPRINT.

Whether you respond with "Y" or "N" the next screen message displays the value of the last item in the input file. If the screen shows a value different from what you intended, then the input file is in error. Also you will be prompted with the following:

"Do you want a plotter/printer file created (Y/N)?"

If you respond with "Y", then you will be prompted with:

"Enter the plotter/printer file name (DRIVE: NAME.EXT)".

After entering the desired drive and file name, press RETURN and the following will appear:

At the prompt below, enter a value for FREQ, the interval of plotter/printer output.

A value of 1 for FREQ means output at each time increment

A value of 2 means every other time increment, etc. As a guide to the amount of disk space required, use the following:

BYTES REQUIRED = (200) X (FLIGHT TIME)

-----  
(INTEGRATION INTERVAL) X (FREQ)

Very large files exceeding 200K may be generated. Exit the program by pressing CTRL C if disk space is insufficient, otherwise enter value for FREQ below:

In response, enter a value that is some integer multiple of the integration interval, e.g., a FREQ value of 20 means that output will be written to the disk every 20 integration intervals. If the integration interval is 5 seconds, then the disk file would contain the flight history in 100-second increments. After entering the desired value for FREQ and pressing ENTER, the screen clears and the problem solution begins. As the solution proceeds, the screen will display the following type data:

```
T=2000.0 PHASE 3, CLIMB AT COMMANDED Q*****
V0=15628.3 M0=15.1 ALPHA= 1.99          WDOTF= 20.05      RES1= 0.0
H0=129013.5      GAMMA= 0.10          WDOTA= 431.83     WDOTP= 0.0
R= 2304.32      Q0=1045.90          PHI= 1.590
ISP= 1357.00    ALPHANAX= 5.00          WT= 71861.9      L/W = 0.544
ISPEFF= 956.7   ALPHANIN= -5.00         FC= 25796.4
ISPMEAN=1431.8  ACCCOM = 0.27          LIPT= 39107.9    L/D = 4.967
ISPAVG= 2642.7  ACCBL-G'S= 0.27       DRAG= 7873.4
ISPA/B= 1357.0          THRUST= 27209.2     T/D = 3.456
ISPROC= 0.0          THRUSTCOM= 27209.2
                   THRUSTMAX= 42899.2     T/W = 0.379
```

The screen is updated at the frequency DELPRINT, a user input to the input data file. A DELPRINT value of 20 to 40 will insure that the speed of the solution is not limited by writing to the screen. The user should experiment with values of DELPRINT to find the values that are frequent enough to track the calculation progress without impeding the solution speed. The plotter/printer file that is generated has values of the following 25 variables recorded every  $FREQ \times DT$  seconds of the flight:

<u>Variable #</u>	<u>Variable</u>	<u>Units</u>
1	TIMEX	secs
2	VØ	ft/sec
3	HØ	feet
4	RANGE/6Ø76	nautical miles
5	ISP	secs
6	ISPEFF	secs
7	ISPMEAN	secs
8	WDOTF	lbs/sec
9	WDOTA	lbs/sec
10	PHI	fraction of stoichiometric
11	THRUST	lbs
12	THRUSTCOM	lbs
13	THRUSTMAX	lbs
14	QØ	lbs/ft <sup>2</sup>
15	ALPHA	degrees
16	GAMMA	degrees
17	WEIGHT	lbs
18	LIFT	lbs
19	DRAG	lbs
20	WDOTP	lbs/sec
21	RES1	not used
22	ISPAVG	secs
23	ACC	g's, accel/GA
24	ACCCOM	g's
25	THRUST/DRAG	lbs/lbs

One use of this file is for generating plots of different variables after run completion. The spreadsheet program LOTUS 1-2-3 can be used to generate graphics plots. The program ET0123 on the distribution disk can be used to build a file from this data that can be read by LOTUS.

At the end of the flight, you will be prompted for a printout of the plotter/printer file (presuming you responded with "Y" at start of problem). If you respond with "Y", then you will get an output giving the final run conditions, a copy of the input data files (first-stage and second-stage) and a time-ordered history of all of the above variables. This is the only opportunity you will have to generate this information from within the program. See the sample output at the end of this section. The program was written so that all printed output could fit on standard 8½-inch-wide sprocket-feed computer paper if compressed print (17 characters per inch) is supported by the printer. The code to turn on and turn off the compressed print is compatible with the Epson FX series dot matrix printers. If your printer cannot generate the 17-character/inch compressed print, you will have to use a standard wide carriage printer and wide paper. If your printer will not respond to the compressed print commands generated by the program but can be set by a printer switch to compressed print, you will be able to use 8½-inch paper.

If you are working with a two-stage vehicle, you will be prompted at completion of its first-stage flight that the second-stage input file has been read and the value that was read from the input file. If the file was read correctly and the stopping conditions of the first-stage were satisfactory, then the second-stage run can proceed. If you want to stop, press CTRL-C and the program will exit to DOS. Presuming you wish to proceed, you will need to respond to the prompt for a listing of the second-stage tabular data (nine input data tables). If the response is "Y" the tables will be listed and the program will then run to completion. An "N" response bypasses this printout and starts the second-stage problem. At the end of a second-stage run, you will be prompted for a printout of the plotter/printer file if you had requested one at the start of the problem.

Any time during the run, pressing CTRL-S or the BREAK key will suspend program execution. The program can be resumed from this point by pressing the spacebar. If you wish to terminate the program after suspending it, press the spacebar first to resume it, then press CTRL-C to terminate. The program may be terminated at any



time by pressing CTRL-C. If you had requested a plotter/printer file at the beginning of the run, you will be prompted for a printout of the run up to the point of termination.

### 3.3 Running ETO within the QUICKBASIC 4.0 Environment

The program was developed and tested using this compiler. It places the user in direct contact with the source code, allowing modifications to be made even within a run. Also, a "scratch space" can be utilized within the source code to temporarily overwrite/replace variables that were read in from the input data file. For example, between runs, fuel weight, launch weight, or other variables might be adjusted if program run results indicate the need. Values written into the "scratch space" later can be moved to the input data file. The Quick BASIC environment allows other files, such as an input data file to be loaded, modified, saved and remain resident, while the main program is still in memory.

To run the program in the Quick BASIC editor/compiler environment, you need to follow the compiler instructions for loading Quick BASIC and ET013.BAS. After selecting the RUN option and pressing ENTER, the prompts and required responses are exactly like those described previously in this section. Screen output and printer output are identical also. Be sure to use a copy of the distribution disk, not the original. Also, the MS-DOS COMMAND.COM for your computer should be on the disk or default drive. We have found that 100 percent IBM compatibility is needed to use the Quick BASIC editor/compiler environment.

### 3.4 Input Data Files for a Single- or First-Stage Vehicle

All of the information about the vehicle to be simulated is contained in an input data file (two files for a two-stage vehicle) which is/are opened by the program and the information read and stored in memory. The input file(s) consists of nine data tables plus additional vehicle information. An input file is normally less than 2,000 bytes, (about three quarters of a page).

Input File Information:

The first set of information is two lines of descriptive text about the data file, such as name, type of vehicle, etc. The program reads these two lines as string variables, so any ASCII characters may be used. Each line should be limited to 80 characters. Next follows nine data tables (Tables 1 through 9). See Figure 3-1\* and the input files at the end of Section 3.6.

Table 1. Airbreather Specific Impulse

This table is the airbreather specific impulse, ISPA. This table must be constructed as follows:

```
ISPA, 2
n (number of Mach args), m (number of phi args)
M1, M2, M3, ----- Mn
θ1, θ2, θ3, ----- θm

Isp1, Isp2, Isp3, ----- Ispn at θ1
Isp1, Isp2, Isp3, ----- Ispn at θ2
Isp1, Isp2, Isp3, ----- Ispn at θm
```

The above example is for a 2-D table. For a 1-D table, use the following format:

```
ISPA, 1
n, number of Mach args.
M1, M2, M3, ----- Mn
Isp1, Isp2, Isp3, ----- Ispn
```

The program uses the INPUT# statement so it expects all values to be separated by commas. If more than one line is required to list a group of variables, then the data can be continued on the next line. Do not use a comma as the last character in a line to be continued.

\*See page 35.

Table 2. Airbreather Maximum Fuel Equivalence Ratio

The next data field is the airbreather maximum allowable fuel equivalence ratio as a function of Mach number:

This is a 1-D table:

PHIMAX, 1

n, number of Mach args.

M<sub>1</sub>, M<sub>2</sub>, M<sub>3</sub>, ----- M<sub>n</sub>

Ømax<sub>1</sub>, Ømax<sub>2</sub>, Ømax<sub>3</sub>, ----- Ømax<sub>n</sub>

Following the same format as the above, the remainder of the tabular data follows:

Table 3. Zero Lift Drag Coefficient

CDØ, 1

n (number of Mach args)

M<sub>1</sub>, M<sub>2</sub>, M<sub>3</sub>, ----- M<sub>n</sub>

Table 4. Friction Drag Increment as a Function of Altitude

DELCD, 1

n (number of altitude args)

ho<sub>1</sub>, ho<sub>2</sub>, ho<sub>3</sub>, ----- ho<sub>n</sub>

ΔC<sub>D1</sub>, ΔC<sub>D2</sub>, ΔC<sub>D3</sub>, ----- ΔC<sub>Dn</sub>

Table 5. Lift Curve Slope as a Function of Mach Number

CLALPHA, 1

n (number of Mach args)

M<sub>1</sub>, M<sub>2</sub>, M<sub>3</sub>, ----- M<sub>n</sub>

C<sub>Lα1</sub>, C<sub>Lα2</sub>, C<sub>Lα3</sub>, ----- C<sub>Lαn</sub>

Table 6. Value of Drag Due to Lift Factor  $K$ , as a Function of Mach Number

$K$ , 1  
n (number of Mach args)  
 $M_1, M_2, M_3, \dots, M_n$   
 $K_1, K_2, K_3, \dots, K_n$

Table 7. Airbreather Inlet Capture Area Ratio vs Mach Number and Angle of Attack

This is most often a 2-D table. The example data file TEST1.DAT, however, shows how it is made effectively a 1-D table by making the Ao/Ac values invariant with alpha.

$A\emptyset AC$ , 2  
n (number of Mach args), m (number of  $\alpha$  args)  
 $M_1, M_2, M_3, \dots, M_n$   
 $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_m$   
 $A\emptyset AC_1, A\emptyset AC_2, A\emptyset AC_3, \dots, A\emptyset AC$  at  $\alpha_1$   
 $A\emptyset AC_1, A\emptyset AC_2, A\emptyset AC_3, \dots, A\emptyset AC$  at  $\alpha_2$   
.  
.  
 $A\emptyset AC_1, A\emptyset AC_2, A\emptyset AC_3, \dots, A\emptyset AC$  at  $\alpha_m$

Table 8. Rocket Specific Impulse vs Altitude

ISPR, 1  
n (number of altitudes)  
 $h_{o1}, h_{o2}, h_{o3}, \dots, h_{on}$   
 $Isp_{R1}, Isp_{R2}, Isp_{R3}, \dots, Isp_{Rn}$

This table allows the rocket performance to increase with altitude in accordance with the increasing nozzle pressure ratio.

Table 9. Maximum Rocket Propellant Flow Rate vs Mach Number

WDOTPMAX, 1  
n (number of Mach args)  
 $M_1, M_2, M_3, \dots, M_n$   
 $WDOTPMAX_1, WDOTPMAX_2, WDOTPMAX_3, \dots, WDOTPMAX_n$

Note that the data tables do not require uniform increments for the arguments. Note also that a step change in the value of a variable can be used, e.g., a rocket can be turned on at Mach 8 with a propellant flow of 600 lbs/sec by having `!DOTMAX` go from 0 to 600 between Mach 8 and 8.01. Linear interpolation is used between data points.

Following the tabular input values the following data items must be entered:

AC, AREF

AC is the airbreather inlet capture area in sq feet

AREF is the vehicle reference area in sq feet.

WLAUNCH, WFUEL, WFINAL, VFINAL, STAGE

WLAUNCH is launch weight, lbs

WFUEL is fuel plus rocket propellant, lbs

WFINAL is burnout weight of stage, lbs

VFINAL is the desired final velocity for this stage. For a single stage to orbit vehicle use a value beyond the polar orbit velocity, say 27,000 fps unless a smaller velocity is desired for a lower inclination orbit.

STAGE is 1 for a single-stage vehicle or the first-stage of a two-stage vehicle.

M0, H0, GAMMA, ALPHA, DT, DELPRINT, TINEX

M0 is the starting Mach number, a value of 0.001 is suggested instead of zero for a stationary launch. For an air launch, use the launch Mach number. For a low angle ground launch, use the Mach number at the end of the rail (assuming some cold launch system is used for the initial acceleration). H0 is the launch altitude in feet. Use 0.001 instead of a zero value.

GAMMA is the initial flight path angle, degrees. A value of 0.0 may be used for a horizontal launch.

ALPHA is the initial angle of attack, degrees. A value of 0.0 may be used for a horizontal launch.

DT is the integration interval, secs. The program is designed to use large values, i.e., 2 to 5 secs. If the flight seems to exhibit large, erratic angle of attack excursions in Phase III, then try smaller values of DT.

DELPRINT is the frequency at which integration steps are displayed to the screen. A value of 20 to 40 insures that program execution is not being controlled by the rate that the screen can be updated.

TIMEX is the starting value for the program time in seconds. 0.0 may be used to start the problem.

QCMD, QMAX, QMIN, ALPHAMAX, ALPHAMIN

QCMD is the commanded dynamic pressure during Phase III flight, psf.

QMAX and QMIN are values of  $q$  greater and lesser, respectively, than QCMD that the vehicle is limited to.

ALPHAMAX and ALPHAMIN are maximum and minimum values that are permitted during Phases III and IV.

VTAKEOFF, ALPHA2MAX

VTAKEOFF is the takeoff rotation speed in feet/sec.

ALPHA2MAX is the maximum angle of attack permitted during Phase II, the interval from rotation to Phase III flight. Normally this angle has its maximum just as Phase II begins. It becomes the vehicle rotation angle.

LOADFAC, ACCCMD, GAMMAMAX

LOADFAC is the maximum normal load factor permitted during the Phase II climb, in g's. The angle of attack is permitted to become as large as ALPHA2MAX in order to generate the maximum LOADFAC.

ACCCOMD is the commanded maximum axial acceleration, in g's, during Phases I, II, and III.

GAMMA4MAX is maximum flight path angle, in degrees, permitted during any flight phase except Phases IV and V.

#### FASTOIC

FASTOIC is the stoichiometric fuel-air ratio for the airbreathing propulsion.

#### VØTEMP, QØFINAL

VØTEMP is the flight velocity, in ft/sec during Phase III at which the vehicle begins to command to a final dynamic pressure QØFINAL. This variation is made linearly between VØTEMP and VFINAL. A combination of VØTEMP and QØFINAL is usually selected so the vehicle will ascend along an approximation of a constant vehicle skin temperature path. Enter a value of 0.0 for VØTEMP and the same value as QCOMD for QØFINAL if constant  $q$  climb to VFINAL is desired.

#### VØCRUISE, SWITCH4, GAMMA4MAX

VØCRUISE is the velocity, in ft/sec, that may be selected for a constant velocity cruise to fuel exhaustion. This is Phase IV. Enter a zero if there is no cruise phase.

SWITCH4 has a value of either 0 or 1. Zero selects cruise at constant altitude. The vehicle maintains the altitude it attained during the Phase III climb. A value of 1 selects cruise at maximum lift-to-drag ratio. Enter a zero if there is no cruise.

GAMMA4MAX is the maximum flight path angle (in degrees) permitted during a maximum L/D cruise (SWITCH4=1). A small value, in the range of 0.20 to 0.50 degrees is suggested. The value should be chosen so that the vehicle performs a fairly smooth climb/cruise only occasionally encountering the limiting value after experiencing some flight path oscillations early in the cruise. This variable has no

significance for a constant altitude cruise (SWITCH4=0) and a value of zero should be entered.

FLAG5, VØ5, ALPHA5MAX, GAMMA5, ACCCOMD5

These five variables are used to prescribe a pullup and ballistic ascent after completing part of the ascent after another phase. This is Phase V.

FLAG5 can have a value of 0 or 1. Zero means that a ballistic ascent is not going to be used. In this case, enter zeros for all the remaining variables on this line. If FLAG5 is set to 1, then enter values for the following:

VØ5 is the velocity, ft/sec, at which pullup is to begin. ALPHA5MAX is the commanded angle of attack, in degrees, to be employed during pullup.

GAMMA5 is the flight path angle, in degrees, to be achieved during the pullup. As soon as this flight path angle is achieved, angle of attack is commanded to zero to permit the flight to continue ballistically.

ACCCOMD5 is the axial acceleration, g's, commanded during Phase V. Thrust is reduced if required to avoid exceeding this commanded acceleration.

A1, A2, B1, C1, C2, D1, DELØ, VØ2, HØ2

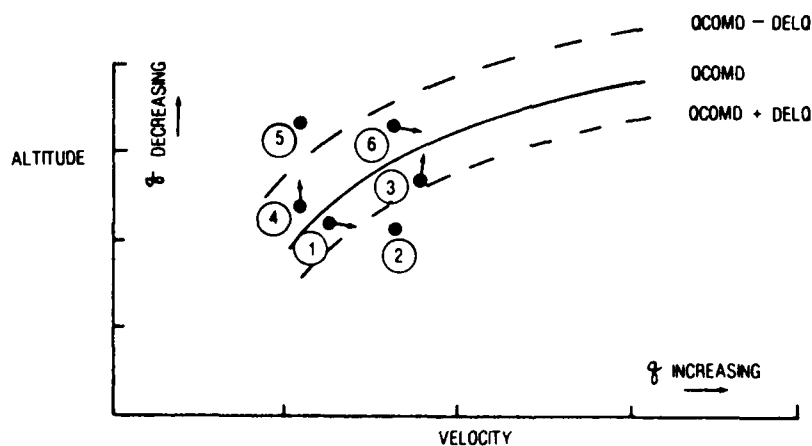
The first seven variables are adjustment coefficients used in the Phase III navigation algorithm, GUID13 (see Figure 2-4 for flow chart of control logic). These coefficients are used to set the sensitivity of angle of attack ( $\alpha$ ) while the vehicle is trying to follow a constant dynamic pressure ( $q$ ) path. Suggested values are, in order:

Ø.1, Ø.1, Ø.ØØ9, Ø.1, 2.Ø, Ø.1, 5Ø.Ø



As a further suggestion, set DELQ at about 5 percent of the commanded Phase III dynamic pressure, QCOMD. Some work has been done to implement a simpler Phase III algorithm, GUID14, (included on distribution disk as GUID14.BAS), which has constants "hardwired" into the code. Whether you are using GUID13 or GUID14, this is an area where some experimentation may be needed to obtain smooth constant q trajectory paths.

If the vehicle deviates from a smooth constant q trajectory, the following information will assist you in making the necessary changes. There are six possible deviations that may occur as shown in the following sketch:



Note that the value DELQ is used to determine if the vehicle reaches deviations 2 and 5. In order to correct each of these deviations (1 to 6),  $\alpha$  must be changed such that the vehicle returns to the desired q path. For each different deviation, there is a different  $\alpha$  sensitivity that is required to return to the desired dynamic pressure path. If the vehicle is having difficulty recovering from any of the deviations, identify the deviation condition that exists and slightly vary the corresponding adjustment coefficient.

The adjustment coefficients affect the angle of attack in the following manner.

<u>DESCRIPTION</u>	"q is"	<u>REACTION</u>	<u>CALCULATION</u>
(1) Higher than QCOMD & Increasing		Increase $\alpha$	} $\alpha = \alpha + A_1 + A_2 \cdot \frac{dq}{dt}$
(2) Higher than QCOMD & Outside DELQ		Increase $\alpha$	

<u>DESCRIPTION</u>	"q is"	<u>REACTION</u>	<u>CALCULATION</u>
(3) Higher than QCOMD & Decreasing		Decrease $\alpha$	$\alpha = \alpha - B1 \cdot \frac{Q-QCOMD}{QCOMD}$
(4) Lower than QCOMD & Decreasing		Decrease $\alpha$	$\alpha = \alpha - C1 - C2 \cdot \frac{dq}{dt}$
(5) Lower than QCOMD & Outside DELQ		Decrease $\alpha$	
(6) Lower than QCOMD & Increasing		Increase $\alpha$	$\alpha = \alpha + D1 \cdot \frac{Q-QCOMD}{QCOMD}$

Note that in conditions 3 and 6 the control logic is attempting to prevent overshooting QCOMD.

VØ2 and HØ2 are a velocity (ft/sec) and altitude (feet) pair which defines the end of the Phase II constant load factor climb. The vehicle is not permitted to exceed VØ2 until the altitude, HØ2 has been exceeded. For an aircraft, VØ2 values of 700 to 900 ft/sec are usually chosen. HØ2 can range from 10,000 feet to 30,000 feet. A combination of HØ2, VØ2 values is usually chosen that would be consistent with an optimum subsonic-transonic climb, such as would be prescribed by a Rutowski path optimization method.

### 3.5 Input Data Files for a Second-Stage Vehicle

The nine data tables described in the previous section are required and are used in the same manner for a second-stage vehicle.

The remaining values required are

AC, AREF

WEIGHT, WFUEL, WFINAL, VFINAL, STAGE

DT, DELPRINT

QØCOMD, QØMAX, QØMIN, ALPHAMAX, ALPHAMIN

LOADFAC, ACCCOMD, GAMMAMAX

FASTIOC

VØTEMP, QØFINAL

VØCRUISE, SWITCH4, GAMMA4MAX

FLAG5, VØ5, ALPHA5MAX, GAMMA5, ACCCOMD5

A1, A2, B1, C1, C2, D1, DELQ, VØ2, HØ2

Note that the variables VTAKOFF and ALPHA2MAX are not used, and since the flight is continuing from the previous stage, starting values of  $M_0$ ,  $H_0$ , GAMMA, ALPHA, and TIMEX are not input. When the second-stage data file is loaded, its values overwrite the first-stage data and current values of appropriate program variables. As an example, the current value of WEIGHT is replaced by the value of WEIGHT read from the second-stage file.

### 3.6 Sample Input Data Files and Program Output

This section describes the example input data files contained on the distribution disk. Each example illustrates features typically used in combination. The example input data files are completely arbitrary and are not intended to represent any known or projected vehicle or level of technology. No inference should be made that the example vehicles typify current or projected capabilities.

The file TEST1.DAT is the simplest example. It is a highly speculative single-stage-to-orbit horizontal takeoff vehicle having all-airbreathing propulsion to orbital velocity. The vehicle is commanded to fly to the orbital condition at a constant  $1,000 \text{ lb/ft}^2$  dynamic pressure. The vehicle has a gross takeoff weight of 100,000 lbs and has 60,000 lbs of fuel.

The file STG1.DAT and STG2.DAT define a two-stage vehicle intended to fly to orbit from a horizontal takeoff. The first stage uses airbreathing propulsion up to the Mach 8 staging point. The second stage operates with rocket propulsion from staging to orbit. The gross takeoff weight of the vehicle is 400,000 lbs, of which 150,000 lbs is first-stage fuel. The second stage weighs 150,000 lbs at staging, of which 120,000 lbs is propellant. The empty weight of the first stage is 100,000 lbs. The first stage accelerates along a constant  $1,000 \text{ lb/ft}^2$  dynamic pressure trajectory to the staging condition.

The file STG2BAL.DAT may be used as an alternative to STG2.DAT, above, as a second stage. This vehicle is identical to STG2.DAT but employs a pullup after staging and ballistic ascent to orbit.

The file CRUZ2.DAT is a Mach 4 cruise aircraft. It has a gross takeoff weight of 400,000 lbs, of which 170,000 lbs is fuel. It employs airbreathing propulsion in all flight phases. It cruises at maximum L/D.

A maximum L/D cruise may experience oscillations in altitude and flight path angle if the vehicle does not start the cruise phase at lift nearly equal to weight and at a very low flight path angle. The code includes an algorithm that smooths the climb/cruise flight path by reducing angle of attack whenever GAMMA4MAX is exceeded and whenever lift becomes less than weight. Also, if the flight path angle becomes negative, angle of attack is increased. This algorithm seems to produce fairly smooth climb/cruise paths once it has forced the vehicle to a fairly steady lift equal weight condition.

The program tests if the vehicle input data files have been read correctly. In starting the program, the following message will appear on the screen just above the prompt for creation of a plotter/printer file:

```
1ST/SINGLE (OR 2ND) STAGE INPUT FILE HAS BEEN READ. LAST ITEM IN INPUT FILE IS H02,  
VALUE READ FROM FILE IS 18000. IF THIS VALUE IS INCORRECT, INPUT FILE IS IN ERROR.
```

In all of the example vehicle files, the value of H02 is 18,000. The screen message above indicates that the program has read all of the data correctly. If you see an unexpected value, this means that the number of data items in the file exceeds the number that the program was directed to read. Check to see that the number of entries in the nine data tables correspond correctly with the declared number and that the data entered after the tabular input is correct. The problem will be found to be too many entries. A duplicate line of data or extra commas is the usual cause. An error of too many data items will not prevent the program from running, but erroneous results will immediately be apparent. The other type of error, not enough data items, will not permit the program to get as far as the prompt above. An error message "Input past end" and program termination will occur if this error is present. In this case the program is looking for data which is not present. A line of data items omitted, or a lacking comma are the usual causes. The program is quite exacting in its requirement that the input file correspond precisely to the declared number of items to be read. Values, separated by commas, must appear exactly in the prescribed format. If an option is not to be employed, or a zero value is planned for a variable, a numerical entry must be present (even a zero).

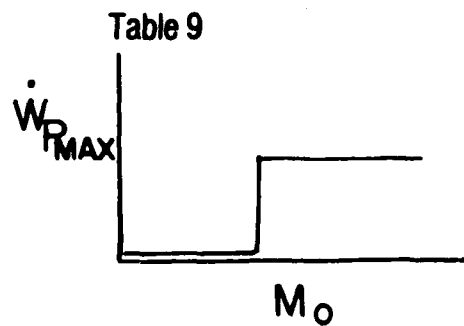
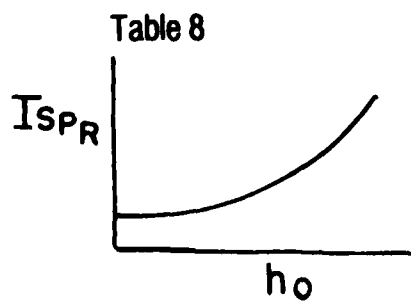
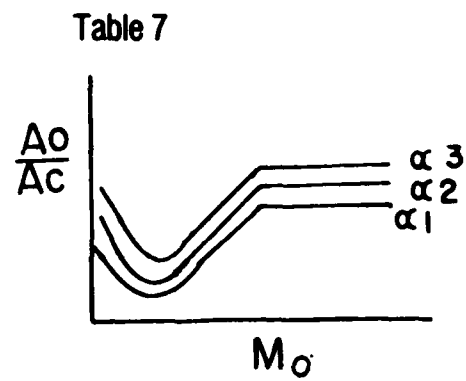
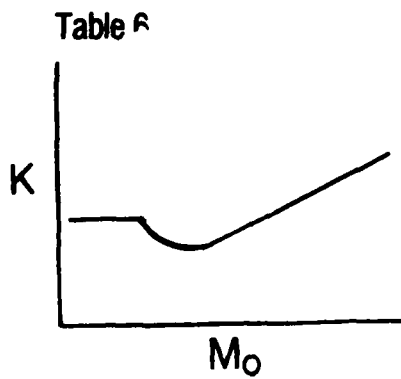
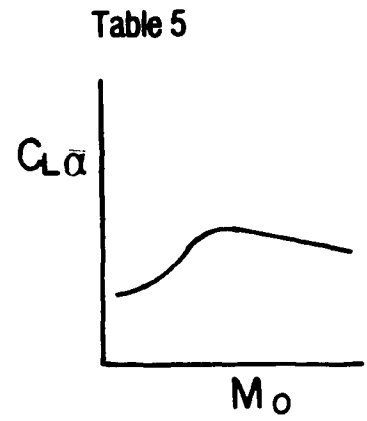
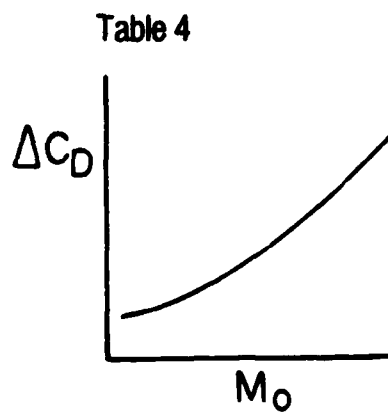
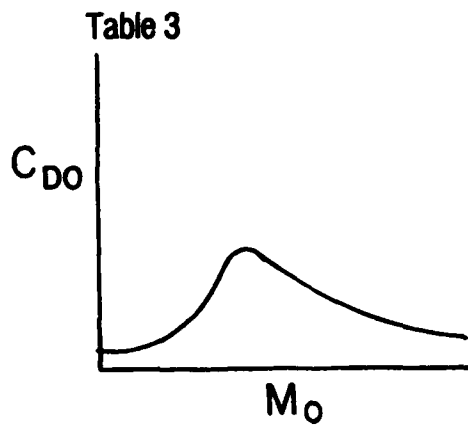
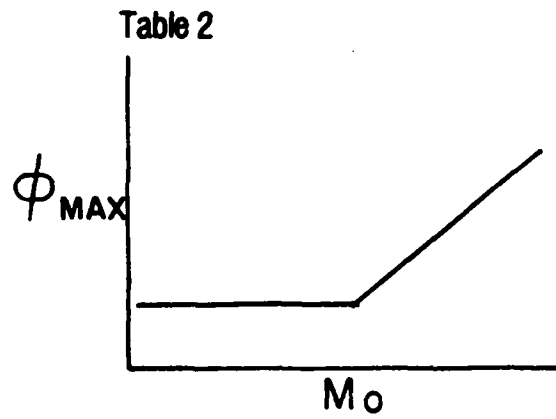
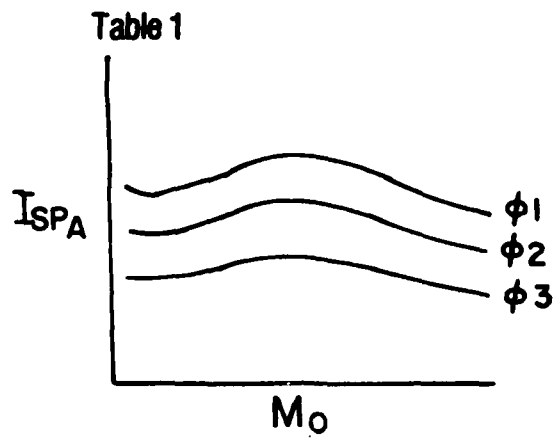


Figure 3-1. Tabular Input Data

Sample Input Data Files

INPUT DATA FILE "TEST1.DAT" 1 SEPT88

PROGRAM "ETO"

A/B TO ORBIT AT CONSTANT Q=1000

ISPA,2

12,3

0.0 , .5 , .9 , 1.5 , 2.0 , 2.5 , 3.0 , 4.0 , 8.0 , 12.0 , 20.0 , 28.0

0.1 , 1.0 , 10.0

4000, 4000,3500,3300,3650,3750,3700,3500,3000,1800,800,600

3000, 3000,3000,3300,3650,3750,3700,3500,3000,1800,800,600

500, 500, 500, 500, 500, 500, 500, 500, 600,500,400,400

PHIMAX,1

10

0.0 , 0.8 , 1.0 , 1.4 , 2.0 , 8.0 , 16.0 , 18.0 , 25.0 , 28.0

1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 3.0 , 3.0 , 5.0 , 5.0

CD0,1

15

0.0 , .4 , .8 , .9 , 1.0 , 1.1 , 1.2 , 1.5 , 2.0 , 3.0 , 5.0 , 10.0 , 15.0 , 20.0 , 25.0

0.0 , .010 , .010 , .010 , .023 , .022 , .018 , .016 , .013 , .008 , .004 , .002 , .0019 , .0017 , .0015

DELCD,1

5

0.0,50000.,100000.,150000.,300000.

0.0,.0006 ,.002 ,.001 ,.0030

CLALPHA,1

10

0.0, 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.040, .040,.039 ,.024, .020, .018,.017,.016, .010 ,.010

K,1

10

0.0, 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.400 , 0.400 ,.440,.730 ,.870,.970,1.025,1.090 ,1.745 ,1.745

A0AC,2

10,3

0.0 , 0.01, 0.1 , 0.5 , 0.8 , 1.0 , 10.0 , 15, 15.01 , 28.0

-10.0 , 0.0 , 10.0

80.0, 2.00, 0.50, 0.15, .15, .15 , .50, 1.0 , 1.0, 1.0

80.0, 2.00, 0.50, 0.15, .15, .15 , .50, 1.0 , 1.0, 1.0

80.0, 2.00, 0.50, 0.15, .15, .15 , .50, 1.0 , 1.0, 1.0

ISPR,1

4

0.0 , 30000.0 , 80000.0 , 350000.0

390 ,440 ,445 ,460

WDOTPMAX,1

4

0.0 , 8.0 , 8.01 , 28.0

0.0 , 0.0 , 0.0 , 0.0

100.0 , 1875.0

100000.,60000.,40000. ,27000 ,1

0.001 ,0.001 ,0.0 ,0.0 , 5.0 , 10.00 , 0.0

1000.0,1100.0,900.0, 5.0 , -5.0

375.0 , 5.0

2.00, 0.50 , 8.0

0.0292

0 , 1000

0 , 0 , 0

0 , 0 , 0 , 0

.1 , .1 , .009, .1, 2.0 , .1 , 50 , 900 , 18000

INPUT DATA FILE " STG1.DAT"/26JUNE88/1530/CONST Q TO ORBIT /PROGRAM"ETO2"  
ALL AIRBREATHING PROPULSION TO MACH 8 /FIRST STAGE

ISPA,2

13,3

0.0 , 0.5 , 1.0 , 1.5 , 2.0 , 2.5 , 3.0 , 4.0 , 5.0 , 5.1 , 8.0 , 15.0 , 26.0

0.1 , 1.0 , 10.0

4500 , 4400 , 4200 , 4000 , 3700 , 3500 , 3200 , 2800 , 2400 , 2400 , 2300 , 1000 , 600

4500 , 4400 , 4200 , 4000 , 3700 , 3500 , 3200 , 2800 , 2400 , 2400 , 2300 , 1000 , 500

1000 , 1000 , 1000 , 1000 , 1000 , 1000 , 1000 , 800 , 600 , 600 , 600 , 450 , 450

PHIMAX,1

12

0.0 , 0.8 , 1.0 , 1.4 , 2.0 , 5.0 , 5.1 , 8.0 , 16.0 , 18.0 , 25.0 , 28.0

1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 3.0 , 5.0 , 5.0 , 5.0

CD0,1

15

0.0 , .4 , .8 , .9 , 1.0 , 1.1 , 1.2 , 1.5 , 2.0 , 3.0 , 5.0 , 10.0 , 15.0 , 20.0 , 25.0

0.0 , .009 , .009 , .009 , .023 , .022 , .018 , .016 , .013 , .008 , .004 , .002 , .0019 , .0017 , .0015

DELCD,1

5

0.0 , 500000. , 1000000. , 1500000. , 2000000.

0.0 , .0006 , .002 , .001 , .0020

CLALPHA,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.043 , .043 , .0391 , .024 , .020 , .018 , .017 , .016 , .010 , .010

K,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.407 , 0.407 , .447 , .728 , .873 , .970 , 1.026 , 1.091 , 1.745 , 1.745

A0AC,2

9,3

0.0 , 0.01 , 0.1 , 0.5 , 0.8 , 1.0 , 10.0 , 25.0 , 28.0

-10.0 , 0.0 , 10.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

ISPR,1

4

0 , 30000 , 80000 , 350000

390 , 440 , 445 , 460

WDOTPMAX,1

4

0.0 , 8.0 , 8.01 , 28

0.0 , 0.0 , 0.0 , 0.0

400.0 , 5500.0

400000 , 1500000 , 2500000 , 8000 , 1

0.001 , 0.001 , 0.0 , 0.0 , 2.0 , 10.0 , 0.0

1500.0 , 1600.0 , 1400.0 , 5.0 , -5.0

390.0 , 15.0

2.00 , .50 , 8.0

0.0292

0 , 1000

0 , 0 , 0

0 , 0 , 0 , 0 , 0

.1 , .1 , .009 , .1 , 2.0 , .1 , 50.0 , 900 , 18000

INPUT DATA FILE "STG2.DAT"/26JUNE88/1530/CONSTANT Q TO ORBIT /PROGRAM"ETO2"  
ALL ROCKET SECOND STAGE

ISPA,2

13,3

0.0 , 0.5 , 1.0 , 1.5 , 2.0 , 2.5 , 3.0 , 4.0 , 5.0 , 5.1 , 8.0 , 15.0 , 26.0

0.1 , 1.0 , 10.0

4500, 4400, 4200, 4000, 3700, 3500, 3200, 2800, 2400, 2400, 2300, 1000, 600

4500, 4400, 4200, 4000, 3700, 3500, 3200, 2800, 2400, 2400, 2300, 1000, 500

1000, 1000, 1000, 1000, 1000, 1000, 1000, 800, 600, 600, 600, 450, 450

PHIMAX,1

12

0.0 , 0.8 , 1.0 , 1.4 , 2.0 , 5.0 , 5.1 , 8.0 , 16.0 , 18.0 , 25.0 , 28.0

0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0

CD0,1

15

0.0 , .4 , .8 , .9 , 1.0 , 1.1 , 1.2 , 1.5 , 2.0 , 3.0 , 5.0 , 10.0 , 15.0 , 20.0 , 25.0

0.0 , .009 , .009 , .023 , .022 , .018 , .016 , .013 , .008 , .004 , .002 , .0019 , .0017 , .0015

DELCD,1

5

0.0 , 50000.0 , 100000.0 , 150000.0 , 200000.0

0.0 , .0006 , .002 , .001 , .0020

CLALPHA,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.043 , .043 , .0391 , .024 , .020 , .018 , .017 , .016 , .010 , .010

K,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.407 , .407 , .447 , .728 , .873 , .970 , 1.026 , 1.091 , 1.745 , 1.745

A0AC,2

9,3

0.0 , 0.01 , 0.1 , 0.5 , 0.8 , 1.0 , 10.0 , 25.0 , 28.0

-10.0 , 0.0 , 10.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

ISPR,1

4

0 , 30000 , 80000 , 350000

390 , 440 , 445 , 460

WDOTPMAX,1

4

0.0 , 8.0 , 8.01 , 28

600 , 600 , 600 , 600

100.0 , 1500.0

150000 , 120000 , 30000.0 , 27000 , 2

2.0 , 10.0

1000.0 , 1100.0 , 900.0 , 5.0 , -5.0

2.00 , 1.00 , 8.0

0.0292

0 , 1000

0 , 0 , 0

0 , 0 , 0 , 0 , 0

.1 , .1 , .009 , .1 , 2.0 , .1 , 50.0 , 900 , 18000



INPUT DATA FILE "STG2BAL.DAT"/29DEC88/BALLISTIC ASCENT TO ORBIT/"ETO"

ALL ROCKET SECOND STAGE

ISPA,2

13,3

0.0 , 0.5 , 1.0 , 1.5 , 2.0 , 2.5 , 3.0 , 4.0 , 5.0 , 5.1 , 8.0 , 15.0 , 26.0

0.1 , 1.0 , 10.0

4500, 4400, 4200, 4000, 3700, 3500, 3200, 2800, 2400, 2400, 2300, 1000, 600

4500, 4400, 4200, 4000, 3700, 3500, 3200, 2800, 2400, 2400, 2300, 1000, 500

1000, 1000, 1000, 1000, 1000, 1000, 1000, 800, 600, 600, 600, 450, 450

PHIMAX,1

12

0.0 , 0.8 , 1.0 , 1.4 , 2.0 , 5.0 , 5.1 , 8.0 , 16.0 , 18.0 , 25.0 , 28.0

0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0

CD0,1

15

0.0 , .4 , .8 , .9 , 1.0 , 1.1 , 1.2 , 1.5 , 2.0 , 3.0 , 5.0 , 10.0 , 15.0 , 20.0 , 25.0

0.0 , .009 , .009 , .009 , .023 , .022 , .018 , .016 , .013 , .008 , .004 , .002 , .0019 , .0017 , .0015

DELCD,1

5

0.0, 50000. , 100000. , 150000. , 200000.

0.0 , .0006 , .002 , .001 , .0020

CLALPHA,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.043 , .043 , .0391 , .024 , .020 , .018 , .017 , .016 , .010 , .010

K,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.407 , .407 , .447 , .728 , .873 , .970 , 1.026 , 1.091 , 1.745 , 1.745

A0AC,2

9,3

0.0 , 0.01 , 0.1 , 0.5 , 0.8 , 1.0 , 10.0 , 25.0 , 28.0

-10.0 , 0.0 , 10.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

ISPR,1

4

0 , 30000 , 80000 , 350000

390 , 440 , 445 , 460

WDOTPMAX,1

4

0.0 , 8.0 , 8.01 , 28

600 , 600 , 600 , 600

100.0 , 1500.0

150000 , 120000 , 30000.0 , 27000 , 2

1.0 , 5.0

1000.0 , 1100.0 , 900.0 , 5.0 , -5.0

2.00 , 1.00 , 8.0

0.0292

0 , 1000

0 , 0 , 0

1 , 8000 , 10 , 1.5 , 2

.1 , .1 , .009 , .1 , 2.0 , .1 , 50.0 , 900 , 18000

INPUT DATA FILE " CRUZ2.DAT"/24AUG87/0045/M 4 CRUISE  
AB SINGLE STAGE

PROGRAM "ETO"

ISPA,2  
14,3

0.0 , 0.5 , 1.0 , 1.5 , 2.0 , 2.5 , 3.0 , 4.0 , 5.0 , 5.1 , 8.0 , 15.0,15.01,28.0  
0.1 , 1.0 , 10.0

4500, 4400,4200,4000,3700,3500,3200,2800,2400, 2400, 2300, 1000, 0.0 ,0.0  
4500, 4400,4200,4000,3700,3500,3200,2800,2400, 2400, 2300, 1000, 0.0 ,0.0  
1000, 1000,1000,1000,1000,1000,1000, 800, 600, 600, 600, 450 , 0.0 ,0.0

PHIMAX,1

11

0.0 ,0.8 ,1.0 ,1.4 ,2.0 , 5.0 , 5.1 , 8.0 ,15.0 ,15.01,28.0  
1.0 ,1.0 ,1.0 ,1.0 ,1.0 , 1.0 , 1.0 , 1.0 , 3.0 , 0.0 , 0.0

CD0,1

15

0.0 ,.4 ,.8 ,.9 ,1.0 ,1.1 ,1.2 ,1.5 ,2.0 ,3.0 ,5.0 ,10.0 ,15.0 ,20.0 ,25.0  
0.0 ,.009 ,.009 ,.009 ,.023 ,.022 ,.018 ,.016 ,.013 ,.008 ,.004 ,.002 ,.0019 ,.0017 ,.0015

DELCD,1

5

0.0,50000.,100000.,150000.,200000.  
0.0,.0006 ,.002 ,.001 ,.0020

CLALPHA,1

10

0.0, 1.5 ,2.0 ,3.0 , 4.0 , 5.0 ,6.0 ,8.0 ,15.0 ,25.0  
0.043, .043,.0391,.024, .020, .018,.017,.016, .010 ,.010

K,1

10

0.0, 1.5 ,2.0 ,3.0 , 4.0 , 5.0 ,6.0 ,8.0 ,15.0 ,25.0  
0.406 , 0.406,.446,.727,.873, .97 , 1.027,1.091,1.745,1.745

A0AC,2

10,3

0.0 , 0.01, 0.1 , 0.5 , 0.8 , 1.0 , 10.0, 15.0,15.01,28.0  
-10.0 ,0.0 ,10.0

80.0, 2.00, 0.50, 0.15, .15, .15 , .51,1.0 ,0.0, 0.0  
80.0, 2.00, 0.50, 0.15, .15, .15 , .51,1.0 ,0.0, 0.0  
80.0, 2.00, 0.50, 0.15, .15, .15 , .51,1.0 ,0.0, 0.0

ISPR,1

4

0, 30000, 80000, 350000  
390, 440, 445, 460

WDOTPMAX,1

4

0.0, 8.00, 8.01, 28.0  
0.0, 0.0, 600.0, 600.0

400.0 , 5500.0

400000 ,170000,230000 , 27000 , 1

0.001 ,0.001 ,0.0 ,0.0 , 2.0, 20.0 , 0.0  
800.0, 850.0, 750.0, 5.0 , -5.0

390.0 ,15.0

2.00, .50 , 8.0

0.0292

0

, 1000  
4000 , 1 , 0.5

0

, 0 , 0 , 0 , 0 , 0  
.1 ,.1 , .009 , .1 , 2.0 , .1 , 50.0,900 , 18000

### 3.7 Sample Program Output

The following sample output is for the two-stage vehicle defined by the data files STG1.DAT and STG2.DAT. The integration interval was 2 seconds. A FREQ value of 50 was used; thus the output is for every 100 seconds of flight.

01-29-1989

20:17:42

1ST/SINGLE STAGE INPUT DATA FILE: STG1.DAT  
2ND STAGE INPUT DATA FILE : STG2.DAT

PLOTTER/PRINTER FILE NAME : G:2STG.OUT

-----PROGRAM ETO2 RUN SUMMARY-----

LAUNCH WEIGHT = 400000.00 LBS.  
FINAL WEIGHT = 38045.15 LBS.  
FUEL/PROPELLANT REMAINING = 8045.15 LBS.

VEHICLE REACHED ORBITAL VELOCITY.

FLIGHT TIME = 3094.71 SECS.  
FINAL VELOCITY = 25858.04 FT/SEC  
FINAL ALTITUDE = 152301.95 FT  
FINAL Q0 = 1049.85 PSF

AVERAGE ISP = 1024.11 SEC  
MEAN ISP = 359.19 SEC

FINAL CONDITIONS ARE SHOWN BELOW:

TIME	V0	H0	R	ISP	ISPEFF	ISPMEAN	WDTF	WDOTA	PHI	THRUST	THRUSTCOM	THRUSTMAX
	00	ALPHA	GAMMA	WEIGHT	LIFT	DRAG	WDOTP	RES1	ISPAVG	ACCEL-G'S	ACCCOM	T/D
3094.71	25858.04	152301.95	4447.27	449.02	282.09	359.19	0.00	0.00	0.00	29886.49	29886.49	269409.94
3094.71	1049.85	-5.00	0.00	38045.15	-78738.77	10941.28	66.56	0.00	1024.11	0.50	0.50	2.73

INPUT DATA FILE " STG1.DAT"/26JUNE88/1530/CONST 0 TO ORBIT /PROGRAM"ETO2"  
ALL AIRBREATHING PROPULSION TO MACH 8 /FIRST STAGE

ISPA,2

13,3

0.0 , 0.5 , 1.0 , 1.5 , 2.0 , 2.5 , 3.0 , 4.0 , 5.0 , 5.1 , 8.0 , 15.0 , 28.0  
0.1 , 1.0 , 10.0

4500, 4400, 4200, 4000, 3700, 3500, 3200, 2800, 2400, 2400, 2300, 1000, 600

4500, 4400, 4200, 4000, 3700, 3500, 3200, 2800, 2400, 2400, 2300, 1000, 500

1000, 1000, 1000, 1000, 1000, 1000, 1000, 800, 600, 600, 600, 450, 450

PHIMAX,1

12

0.0 , 0.8 , 1.0 , 1.4 , 2.0 , 5.0 , 5.1 , 8.0 , 16.0 , 18.0 , 25.0 , 28.0

1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 1.0 , 3.0 , 5.0 , 5.0 , 5.0

CD0,1

15

0.0, .4 , .8 , .9 , 1.0 , 1.1 , 1.2 , 1.5 , 2.0 , 3.0 , 5.0 , 10.0 , 15.0 , 20.0 , 25.0

0.0, .0009, .0009, .0009, .023, .022, .018, .016, .013, .008, .004, .002, .0019, .0017, .0015

DELCD,1

5

0.0, 50000., 100000., 150000., 200000.

0.0, .0006 , .002 , .001 , .0020

CLALPHA,1

10

0.0, 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.043, .043, .0391, .024, .020, .018, .017, .016, .010 , .010

K,1

10

0.0, 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.407 , 0.407, .447, .728, .873 , .970 , 1.026, 1.091 , 1.745, 1.745

A0AC,2

9,3

0.0 , 0.01, 0.1 , 0.5 , 0.8 , 1.0 , 10.0 , 25.0 , 28.0

-10.0 , 0.0 , 10.0

80.0, 2.00, 0.50, 0.15, .15, .15 , .51, 1.0 , 1.0

80.0, 2.00, 0.50, 0.15, .15, .15 , .51, 1.0 , 1.0

80.0, 2.00, 0.50, 0.15, .15, .15 , .51, 1.0 , 1.0

ISFR,1

4

0 , 30000, 80000, 350000

390 , 440, 445, 460

WDOTPMAX,1

4

0.0 , 8.0 , 8.01 , 28

0.0 , 0.0 , 0.0 , 0.0

400.0 , 5500.0

400000 , 150000, 250000 , 8000 , 1

0.001 , 0.001 , 0.0 , 0.0 , 2.0, 10.0 , 0.0

1500.0, 1600.0, 1400.0, 5.0 , -5.0

190.0 , 15.0

2.00, .50 , 8.0

0.0292

0 , 1000

0 , 0 , 0

0 , 0 , 0 , 0

.1 , .1 , .009 , .1 , 2.0 , .1 , 50.0, 900 , 18000

INPUT DATA FILE "STG2.DAT"/26JUNE88/1530/CONSTANT Q TO ORBIT /PROGRAM"ET02"  
ALL ROCKET SECOND STAGE

ISPA,2

13,3

0.0 , 0.5 , 1.0 , 1.5 , 2.0 , 2.5 , 3.0 , 4.0 , 5.0 , 5.1 , 8.0 , 15.0 , 25.0

0.1 , 1.0 , 10.0

4500 , 4400 , 4200 , 4000 , 3700 , 3500 , 3200 , 2800 , 2400 , 2400 , 2300 , 1000 , 500

4500 , 4400 , 4200 , 4000 , 3700 , 3500 , 3200 , 2800 , 2400 , 2400 , 2300 , 1000 , 500

1000 , 1000 , 1000 , 1000 , 1000 , 1000 , 1000 , 800 , 600 , 600 , 600 , 450 , 450

PHIMAX,1

12

0.0 , 0.8 , 1.0 , 1.4 , 2.0 , 5.0 , 5.1 , 8.0 , 16.0 , 18.0 , 25.0 , 28.0

0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0 , 0.0

CD0,1

15

0.0 , .4 , .8 , .9 , 1.0 , 1.1 , 1.2 , 1.5 , 2.0 , 3.0 , 5.0 , 10.0 , 15.0 , 20.0 , 25.0

0.0 , .009 , .009 , .009 , .023 , .022 , .018 , .016 , .013 , .008 , .004 , .002 , .0019 , .0017 , .0015

DELCD,1

5

0.0 , 50000. , 100000. , 150000. , 200000.

0.0 , .0006 , .002 , .001 , .0020

CLALPHA,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.043 , .043 , .0391 , .024 , .020 , .018 , .017 , .016 , .010 , .010

K,1

10

0.0 , 1.5 , 2.0 , 3.0 , 4.0 , 5.0 , 6.0 , 8.0 , 15.0 , 25.0

0.407 , 0.407 , .447 , .728 , .873 , .970 , 1.026 , 1.091 , 1.745 , 1.745

A0AC,2

9,3

0.0 , 0.01 , 0.1 , 0.5 , 0.8 , 1.0 , 10.0 , 25.0 , 28.0

-10.0 , 0.0 , 10.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

80.0 , 2.00 , 0.50 , 0.15 , .15 , .15 , .51 , 1.0 , 1.0

ISPR,1

4

0 , 30000 , 80000 , 350000

390 , 440 , 445 , 460

WDOTPMAX,1

4

0.0 , 8.0 , 8.01 , 28

600 , 600 , 600 , 600

100.0 , 1500.0

150000 , 120000 , 30000.0 , 27000 , 2

2.0 , 10.0

1000.0 , 1100.0 , 900.0 , 5.0 , -5.0

2.00 , 1.00 , 8.0

0.0292

0 , 1000

0 , 0 , 0

0 , 0 , 0 , 0 , 0

.1 , .1 , .009 , .1 , 2.0 , .1 , 50.0 , 900 , 18000

6:25TG.OUT		01-29-1989			20:17:47						
TIME	H0	V0	00	R	WEIGHT	WDOTF	WDOTA	PHI	WDOTP	RES1	
100.00	2167.31	919.81	943.25	10.81	395217.31	27.84	3959.30	0.24	0.00	0.00	
200.00	10325.58	923.11	740.38	25.79	392033.19	34.37	3096.59	0.38	0.00	0.00	
300.00	25111.56	1474.98	1155.32	42.41	386615.31	87.87	3389.13	0.89	0.00	0.00	
400.00	48919.79	2677.66	1373.80	76.95	377626.69	85.89	2914.82	1.00	0.00	0.00	
500.00	63469.52	3311.88	1048.32	126.42	370790.59	58.73	2011.35	1.00	0.00	0.00	
600.00	63445.13	3982.33	1517.49	185.95	364502.31	48.92	2693.12	0.62	0.00	0.00	
700.00	59209.52	4018.83	1892.10	252.07	359204.19	58.11	3345.72	0.59	0.00	0.00	
800.00	65138.59	4133.29	1507.68	318.38	353262.19	76.99	2636.56	1.00	0.00	0.00	
900.00	73827.55	4732.20	1293.51	391.68	346526.59	62.55	2142.26	1.00	0.00	0.00	
1000.00	71043.86	5061.94	1694.84	473.45	340969.69	49.88	2744.32	0.62	0.00	0.00	
1100.00	73864.21	5152.98	1531.05	556.93	335650.50	71.63	2460.81	1.00	0.00	0.00	
1200.00	81434.77	5694.14	1296.90	646.38	329224.69	58.68	2009.63	1.00	0.00	0.00	
1300.00	78853.89	6063.88	1664.94	744.13	323766.00	47.49	2532.66	0.64	0.00	0.00	
1400.00	81387.32	6163.81	1523.14	844.11	318647.00	67.23	2302.55	1.00	0.00	0.00	
1500.00	88346.38	6666.24	1277.44	949.88	312657.19	54.91	1800.56	1.00	0.00	0.00	
1600.00	84914.17	7033.21	1674.76	1063.51	307359.59	46.72	2432.43	0.66	0.00	0.00	
1700.00	87016.58	7083.45	1536.60	1179.32	302497.91	62.01	2225.08	0.95	0.00	0.00	
1800.00	94202.17	7557.83	1244.34	1299.98	296783.19	51.52	1764.35	1.00	0.00	0.00	
1900.00	90241.23	7958.53	1664.84	1428.22	291588.31	45.45	2332.69	0.67	0.00	0.00	
2000.00	90952.39	7961.97	1610.27	1559.16	286874.41	45.55	2256.25	0.69	0.00	0.00	
2100.00	98501.49	8037.19	1149.67	1690.99	145552.09	0.00	0.00	0.00	48.61	0.00	
2200.00	107149.90	9381.72	1043.05	1831.56	129714.20	0.00	0.00	0.00	180.50	0.00	
2300.00	114917.40	11303.33	1044.63	2001.82	112933.60	0.00	0.00	0.00	155.76	0.00	
2400.00	121578.10	13201.00	1045.10	2203.46	98307.59	0.00	0.00	0.00	136.83	0.00	
2500.00	127428.40	15083.61	1045.54	2436.22	85585.84	0.00	0.00	0.00	118.53	0.00	
2600.00	132643.59	16948.62	1045.98	2699.81	74562.04	0.00	0.00	0.00	101.71	0.00	
2700.00	137355.70	18797.00	1046.30	2993.96	65021.18	0.00	0.00	0.00	88.34	0.00	
2800.00	141659.59	20632.69	1046.72	3318.43	56726.34	0.00	0.00	0.00	76.86	0.00	
2900.00	145592.50	22444.70	1047.52	3672.94	49525.49	0.00	0.00	0.00	66.85	0.00	
3000.00	149177.20	24219.52	1048.93	4056.97	43287.50	0.00	0.00	0.00	57.79	0.00	
3064.00	151315.20	25332.92	1049.71	4317.94	39731.11	0.00	0.00	0.00	53.43	0.00	
3088.00	152095.20	25757.05	1050.59	4422.21	38426.70	0.00	0.00	0.00	57.71	0.00	

TIME	H0	V0	ISP	ISPAVG	ISPEFF	ISPMEAN	THRUST	THRUSTCOM	THRUSTMAX
100.00	2167.31	919.81	4267.90	4366.57	-8.01	2395.35	118834.70	118834.70	493417.81
200.00	10325.58	923.11	4256.79	4326.29	-7.06	1484.12	146289.70	146289.70	384901.41
300.00	25111.56	1474.98	4018.96	4270.74	2206.68	1436.25	353138.81	353138.81	397726.50
400.00	48919.79	2677.66	3340.07	4022.78	987.47	1544.59	284205.00	388000.31	284205.00
500.00	63469.52	3311.88	3031.27	3824.78	1114.59	1435.55	178031.30	297046.31	178031.30
600.00	63445.13	3982.33	2754.19	3661.37	1909.01	1396.94	134734.59	134734.59	216586.50
700.00	59209.52	4018.83	2739.10	3541.64	-1.64	1225.65	159171.50	159171.50	267596.91
800.00	65138.59	4133.29	2691.80	3438.99	954.57	1096.76	207235.59	224654.30	207235.59
900.00	73827.55	4732.20	2455.13	3329.78	985.60	1075.43	153578.00	270389.81	153578.00
1000.00	71043.86	5061.94	2396.17	3242.90	-1.14	1035.78	119516.50	119516.50	192015.00
1100.00	73864.21	5152.98	2393.30	3172.89	778.30	963.33	171421.41	171421.41	171972.00
1200.00	81434.77	5694.14	2375.17	3101.29	804.02	950.75	139378.00	255637.00	139378.00
1300.00	78853.09	6063.08	2361.79	3048.72	-0.93	933.30	112153.40	112153.40	174663.30
1400.00	81387.32	6163.81	2358.61	3005.47	767.07	885.57	158579.70	162870.00	158579.70
1500.00	88346.38	6666.24	2342.01	2960.55	756.51	877.86	128605.40	242200.30	128605.40
1600.00	84914.17	7033.21	2328.56	2924.71	-0.80	867.59	108781.30	108781.30	165390.59
1700.00	87016.58	7083.45	2327.15	2894.98	606.75	828.21	144298.00	144298.00	151200.41
1800.00	94202.17	7557.83	2311.78	2863.13	714.66	821.95	119100.60	229469.70	119100.60
1900.00	90241.23	7958.53	2284.02	2836.14	-0.73	818.16	103804.70	103804.70	158641.00
2000.00	98952.39	7961.97	2284.09	2813.14	-0.71	784.04	104043.70	104043.70	153434.30
2100.00	98501.49	8037.19	446.03	1267.74	0.00	349.79	21682.32	21682.32	267616.69
2200.00	107149.90	9381.72	446.51	1219.60	403.64	349.35	80593.04	80593.04	267905.00
2300.00	114917.40	11303.33	446.94	1174.43	396.78	352.20	69616.70	69616.70	268163.91
2400.00	121578.10	13201.00	447.31	1139.17	389.84	354.16	61206.68	61206.68	268385.91
2500.00	127428.40	15083.61	447.63	1111.18	388.37	355.60	53056.02	53056.02	268580.91
2600.00	132643.59	16948.62	447.92	1088.71	390.65	356.69	45559.97	45559.97	268754.81
2700.00	137355.70	18797.00	448.19	1070.46	389.59	357.57	39592.23	39592.23	268911.81
2800.00	141659.59	20632.69	448.43	1055.43	387.12	358.25	34465.09	34465.09	269055.31
2900.00	145592.50	22444.70	448.64	1042.96	382.40	359.76	29991.06	29991.06	269186.41
3000.00	149177.20	24219.52	448.84	1032.57	376.35	359.11	25940.01	25940.01	269305.91
3064.00	151315.20	25332.92	449.96	1026.80	366.53	359.24	23989.11	23989.11	269377.19
3088.00	152095.20	25757.05	449.81	1024.72	326.06	359.27	25911.53	25911.53	269407.19



TIME	H0	V0	Q0	ALPHA	GAMMA	LIFT	DRAG	ACCEL-G'S	ACCDOM	T/D
100.00	2167.31	919.81	943.25	3.54	3.71	790434.38	95841.65	-0.00	0.00	1.24
200.00	10325.58	923.11	740.38	4.48	6.94	784066.38	98597.94	-0.00	0.00	1.48
300.00	25111.56	1474.98	1155.32	0.00	8.22	0.00	105590.90	0.50	0.50	3.34
400.00	48919.79	2677.66	1373.80	-5.00	4.65	-1039890.00	168502.59	0.23	0.50	1.89
500.00	63469.52	3311.88	1848.32	-5.00	1.31	-643249.00	103528.70	0.18	0.50	1.72
600.00	63445.13	3982.33	1517.49	0.61	-1.18	99953.04	57373.41	0.31	0.23	2.35
700.00	59209.52	4018.83	1892.10	5.00	0.12	1024911.00	157792.30	-0.00	0.00	1.01
800.00	65138.59	4133.29	1507.68	5.00	1.47	806796.13	124345.50	0.21	0.26	1.67
900.00	73827.55	4732.20	1293.51	-5.00	0.22	-650093.00	96261.41	0.16	0.50	1.60
1000.00	71043.86	5061.94	1694.84	5.00	-0.16	829109.50	119989.30	-0.00	0.00	1.02
1100.00	73864.21	5152.98	1531.05	5.00	0.77	745473.78	108474.18	0.16	0.17	1.55
1200.00	81434.77	5694.14	1296.90	-5.00	0.24	-612717.31	90220.30	0.14	0.50	1.54
1300.00	78853.09	6063.08	1664.94	5.00	-0.15	773595.38	112590.30	-0.00	0.00	1.02
1400.00	81387.32	6163.81	1523.14	5.00	0.61	705775.69	103127.90	0.16	0.18	1.54
1500.00	88346.38	6666.24	1277.44	-5.00	0.12	-583474.13	85801.66	0.13	0.50	1.50
1600.00	84914.17	7033.21	1674.76	5.00	-0.19	755966.88	109393.10	-0.00	0.00	0.99
1700.00	87016.58	7083.45	1536.60	5.00	0.52	692738.81	100652.40	0.12	0.13	1.43
1800.00	94202.17	7557.83	1244.34	-5.00	0.15	-553352.38	80989.87	0.13	0.50	1.47
1900.00	90241.23	7958.53	1664.04	5.00	-0.30	728801.80	104898.00	-0.00	0.00	0.99
2000.00	90952.39	7961.97	1610.27	5.00	0.40	705265.69	101686.50	-0.00	0.00	1.02
2100.00	98501.49	8037.19	1149.67	5.00	0.58	137066.80	20130.63	0.00	0.00	1.03
2200.00	107149.90	9381.72	1043.05	0.10	0.53	2361.28	6410.00	0.57	0.57	12.57
2300.00	114917.40	11303.33	1044.63	1.89	0.36	39263.29	7128.71	0.55	0.55	9.77
2400.00	121578.10	13201.00	1045.10	2.39	0.27	44215.36	7434.94	0.55	0.55	8.27
2500.00	127426.40	15083.61	1045.54	2.26	0.21	36739.28	6737.90	0.55	0.54	7.97
2600.00	132643.59	16948.62	1045.98	1.49	0.17	23357.93	5622.90	0.54	0.54	6.10
2700.00	137355.70	18797.00	1046.30	0.97	0.14	15198.83	5024.11	0.54	0.54	7.88
2800.00	141659.59	20632.69	1046.72	0.44	0.11	6621.42	4584.90	0.53	0.53	7.52
2900.00	145592.50	22444.70	1047.52	-0.14	0.10	-2148.31	4318.83	0.53	0.52	6.94
3000.00	149177.20	24219.52	1048.97	0.20	0.08	3120.03	4120.82	0.51	0.51	6.29
3064.00	151315.20	25332.92	1049.71	0.87	0.07	13643.94	4274.48	0.50	0.50	5.61
3098.80	152095.20	25757.05	1050.59	3.34	0.07	52615.54	7138.62	0.49	0.49	3.63

#### 4.0 LISTING OF PROGRAM ETO

In the following program listing, the leftmost column of numbers are sequential program line numbers that were generated by the file cross referencing code used to generate the program listing. They are not part of the program. These line numbers are used with the cross referenced listing of program variables in Section 5.0.

```

1      ' 1615/14JAN89   ET013.BAS (Date of last update)
2      '
3      ' *****
4      ' *
5      ' *
6      ' *           ET013.BAS
7      ' *           DEVELOPMENT VERSION OF "ET0.BAS"
8      ' *           *****
9      '
10     '           WRITTEN BY:
11     '           JOHN L. LEINGANG
12     '           WAYNE A. DONALDSON
13     '           KENNETH A. WATSON
14     '           2LT LOUIS R. CARREIRO
15     '
16     '           AFWAL/POPA
17     '           WPAFB,OHIO
18     '
19     '           Written during period January 1986 to February 1987
20     '
21     '           This is a 2-DOF trajectory program for calculating the earth-to-orbit
22     '           performance of single stage or two stage horizontal takeoff vehicles.
23     '           The program assumes a spherical non-rotating earth.A 2nd order integ-
24     '           ration is used for the equations governing weight,velocity,flight path
25     '           angle,altitude,and range.
26     '
27     '           Rocket-only,airbreathing-only,and simultaneous rocket and airbreath-
28     '           ing propulsion may be used in any part of the trajectory.
29     '
30     '           Trajectory options are:
31     '           * Cruise at constant Mach number and altitude
32     '           * Cruise at constant Mach number and constant L/D
33     '           * Climb to orbit at constant Q
34     '           * Climb to orbit along a velocity-biased Q path
35     '           * Climb to orbit ballistically,beginning at a commanded velocity
36     '
37     '           Algorithms for table loading and lookup developed by Ken Watson.
38     '           Algorithm for constant Q0 path developed by Wayne Donaldson.
39     '
40     ' *****
41     '
42     ' CLS
43     ' WIDTH LPRINT 80
44     ' DIM I1ST(50), J1ST(50), T$(50), IO(50), AT(1000)
45     ' DIM TG(3), GA(3)
46     ' DIM DELW(4), DELV(4), DELGAM(4), DELH(4), DELR(4)
47     ' DIM RHOB(10), GLMB(10), HB(10), TMB(10)
48     ' INPUT "IS THIS A SINGLE-STAGE(1),OR A 2-STAGE VEHICLE (2)? ", NSTAGES
49     ' IF NSTAGES = 2 GOTO 460
50     ' INPUT "SINGLE STAGE FILE NAME"; IFN$
51     '                                     GOTO 470
52     ' INPUT "FIRST STAGE FILE NAME"; IFN$
53     ' INPUT "SECOND STAGE FILE NAME"; IFN2$
54     '
55     ' 470      OPEN "I", #1, IFN$: IL = 0: IFILE = 1
56     '          GOTO 480
57     '
58     ' 480      LINE INPUT #1, TITL1$
59     '          LINE INPUT #1, TITL2$
60     '
61     '          GOSUB 16000: 'Input airbreather Isp vs. Mach and PHI as table #1
62     '          GOSUB 16000: 'Input airbreather PHImax vs. Mach as table #2
63     '          GOSUB 16000: 'Input Cdo vs.Mach as table #3
64     '          GOSUB 16000: 'Input delta Cd vs. Alt. as table #4
65     '          GOSUB 16000: 'Input Clalpha vs. Mach as table #5
66     '          GOSUB 16000: 'Input K vs. Mach as table #6
67     '          GOSUB 16000: 'Input airbreather Ao/Ac vs. Mach and alpha as table #7
68     '          GOSUB 16000: 'Input rocket Isp vs. Altitude as table #8
69     '          GOSUB 16000: 'Input rocket WDOTPMAX vs.Mach as table #9
70     '
71     '          INPUT "Do you want a listing of tabular input data(Y/N)"; WTIN$
72     '          IF WTIN$ = "Y" OR WTIN$ = "y" THEN 630 ELSE 640
73     '          FOR ITAB = 1 TO 9: GOSUB 16230: NEXT ITAB
74     '          640

```

```

72     INPUT "Do you want printer output during run(Y/N)"; PRCOM$
73     IF PRCOM$ = "Y" OR PRCOM$ = "y" THEN PRCOM = 1 ELSE PRCOM = 0
74     IF PRCOM = 1 THEN GOTO 680 ELSE GOTO 700
75 680  "' LPRINT CHR$(27); "B"; CHR$(3); : 'Select condensed print (SR15 printer)
76     LPRINT CHR$(27); CHR$(33); CHR$(4); : 'Select condensed print(Epson printer)
77     GOTO 710
78 700  "' LPRINT CHR$(27); "B"; CHR$(1); : 'Select standard print (SR15 printer)
79     LPRINT CHR$(27); CHR$(33); CHR$(0); : 'Select standard print (Epson printer)
80     .....
81 710  INPUT #1, AC, AREF
82     INPUT #1, WLAUNCH, WFUEL, WFINAL, VFINAL, STAGE
83     INPUT #1, M0, H0, GAMMA, ALPHA, DT, DELPRINT, TIMEX
84     INPUT #1, Q0COMD, Q0MAX, Q0MIN, ALPHAMAX, ALPHAMIN
85     INPUT #1, VTAKOFF, ALPHA2MAX
86     INPUT #1, LOADFAC, ACCCOMD, GAMMAMAX
87     INPUT #1, FASTOIC
88     INPUT #1, V0TEMP, Q0FINAL
89     INPUT #1, V0CRUISE, SWITCH4, GAMMA4MAX
90     INPUT #1, FLAG5, V05, ALPHA5MAX, GAMMA5, ACCCOMD5
91     INPUT #1, A1, A2, B1, C1, C2, D1, DELQ, V02, H02
92     .
93     PRINT
94     PRINT "1ST/SINGLE STAGE INPUT FILE HAS BEEN READ."
95     PRINT "LAST ITEM IN INPUT FILE IS H02,VALUE READ FROM FILE IS"; H02
96     PRINT "IF THIS VALUE IS INCORRECT,INPUT FILE IS IN ERROR."
97     PRINT
98     INPUT "Do you want a plotter/printer file created? (Y/N)"; PLOT$
99     IF PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 1020
100    INPUT "Enter the plotter/printer file name (DRIVE:NAME.EXT)   ", PFN$
101    OPEN "O", #2, PFN$
102    VARNUM = 25
103    WRITE #2, VARNUM
104    PRINT
105    PRINT
106    PRINT "At the prompt below,enter a value for FREQ,the interval of plotter/"
107    PRINT "printer output."
108    PRINT "A value of 1 for FREQ means output at each time increment  "
109    PRINT "A value of 2 means every other time increment,etc. As a guide to  "
110    PRINT "the amount of disk space required,use the following:"
111    PRINT
112    PRINT "  BYTES REQUIRED = (200) X (FLIGHT TIME)"
113    PRINT "  -----"
114    PRINT "                (INTEGRATION INTERVAL) X (FREQ)"
115    PRINT
116    PRINT "  Very large files exceeding 200K may be generated.Exit the  "
117    PRINT "  program by pressing CTRL C if disk space is insufficient,"
118    PRINT "  otherwise enter value for FREQ below:"
119    INPUT "                ", FREQ
120 1020  STARTIME$ = TIME$: START = TIMER
121     IF EOF(1) THEN GOTO 1022
122 1022  CLOSE #1
123     .....
124     ' Use this area to the line below as a "scratch" space to
125     ' temporarily override values read from the 1st/single stage
126     ' input file.
127
128
129  '*** DT = .1
130  '*** DELPRINT = 1
131
132
133  '-----
134  CLS :                'Clears screen
135
136  '----- EVALUATE CONSTANTS -----
137
138  GC = 32.174
139  MEARTH = (5.98E+27) / 454:                'Earth mass
140  REARTH = (6.38E+08) / (2.54 * 12):        'Earth radius
141  BIGG = (6.67E-11) * (1 / 4.4482) * (.4536 / .3048) ^ 2: 'Universal gravitation const
142
143  FC = (WEIGHT / GC) * (V0) ^ 2 / (H0 + REARTH): 'Centrifugal force

```

```
143     ANG = 57.3: 'Degrees per radian
144     '
145     '
146     '----- INITIALIZE THE PROBLEM -----'
147     '
148     IF TIMEX = 0! THEN WEIGHT = WLAUNCH
149     COUNT = 0
150     PRCOUNT = 0
151     I = 0
152     '-----Load data for atmosphere routine -----'
153     FOR J1 = 1 TO 10
154     '
155     READ HB(J1): READ RH0B(J1): READ TMB(J1): READ GLMB(J1)
156     NEXT J1
157     '
158 2540 DATA 0,2.3769E-3,518.7,-3.5662E-3
159     DATA 36089,7.0612E-4,390,0
160     DATA 65617,1.7082E-4,390,5.4864E-4
161     DATA 104986,2.5661E-5,411.6,1.5362E-3
162     DATA 154199,2.7698E-6,487.2,0
163     DATA 170604,1.4735E-6,487.2,-1.0973E-3
164     DATA 200131,4.8719E-7,454.8,-2.1946E-3
165     DATA 259186,3.8826E-8,325.2,0
166     DATA 291153,6.1508E-9,325.2,1.6953E-3
167     DATA 323003,9.6511E-10,374.2,2.8343E-3
168     RESTORE 2540
169     '
170     '$PAGE
```

```

171 '*****
172 '          MAIN PROGRAM ROUTINE
173 '*****
174 '
175 '-----Get current values from input data:
176 ITAB = 3: XARG = M0: GOSUB 16400: CD0 = VALI: 'Look up Cd0
177 ITAB = 4: XARG = H0: GOSUB 16400: DELCD = VALI: 'Look up delCd friction
178 ITAB = 5: XARG = M0: GOSUB 16400: CLALPHA = VALI: 'Look up Clalpha
179 ITAB = 6: XARG = M0: GOSUB 16400: K = VALI: 'LOOK UP K
180 ITAB = 7: XARG = M0: YARG = ALPHA: GOSUB 16400: A0AC = VALI: 'LOOK UP A0AC
181 '
182 '----- CALCULATE CURRENT LIFT AND DRAG -----
183 '
184 CL = ALPHA * CLALPHA
185 LIFT = CL * Q0 * AREF
186 CD = CD0 + K * CL ^ 2 + DELCD
187 DRAG = CD * Q0 * AREF
188 '
189 '
190 FOR I = 0 TO 1000000!
191 '.....
192 3100 A$ = INKEY$: IF A$ = CHR$(3) THEN GOTO 13260: 'Intercepts CTRL-C entered
193 '          from keyboard to halt execution and exit program
194 Q$ = INKEY$: IF Q$ = CHR$(19) THEN GOTO 3108 ELSE GOTO 3118: 'Intercepts
195 '          CTRL-S to halt execution until space bar is pressed
196 3108 '
197 3110 Q$ = INKEY$
198 IF Q$ <> CHR$(32) THEN GOTO 3110: 'Resumes if space bar is pressed
199 '
200 3118 '
201 GOSUB 13500: ' Go to atmosphere subroutine and return here
202 '
203 GA = BIGG * MEARTH / (H0 + REARTH) ^ 2 * GC: 'Gravitational acceleration
204 '-----Reduce integration interval near end of flight
205 IF WEIGHT - WFINAL < .01 * WFINAL THEN DT = .1
206 IF V0 < 24000 THEN GOTO 3700
207 DV = CINT((26200 - V0) / 1000)
208 DT = DV
209 IF DT = 0 THEN DT = .1
210 3700 '
211 GOTO 4900
212 4900 '
213 '
214 '----- FLIGHT PHASE COMMANDS -----
215 '
216 PHASE = 1
217 ' Takeoff Roll at zero alpha until VTAKEOFF is reached
218 IF V0 >= VTAKEOFF OR TIMEX > TTAKEOFF THEN 6500: 'Test to see if
219 '          Phase 1 is completed
220 ALPHA = 0!: GAMMA = 0!: LIFT = WEIGHT: H0 = 0!
221 ACCCOM = ACCCOMD
222 GOSUB 14000: 'Go to thrust subroutine and return here
223 ALPHA = 0!: GAMMA = 0!: LIFT = WEIGHT: H0 = 0!
224 '
225 '
226 TTAKEOFF = TTAKEOFF + DT: A$ = "PHASE 1,TAKEOFF ROLL *****"
227 GOSUB 15800: ' Go to integration subroutine then return here
228 GOTO 12000: ' Go to runtime output section
229 6500 '-----
230 PHASE = 2: 'Commanded climb at constant load factor. Required ALPHA is
231 '          calculated, but is limited to ALPHAMAX. Maximum velocity allowed
232 '          is V02 fps. ALPHA is commanded to zero above H02 ft. This phase
233 '          is maintained until V02 fps, H02 ft, and 0.95*Q0COMD are reached.
234 '
235 IF TIMEX > TTAKEOFF + TQ0 + DT THEN 6860: 'Test if Phase 2 is
236 '          completed
237 IF V0 > V02 AND H0 > H02 AND Q0 > .95 * Q0COMD THEN GOTO 6860
238 ALPHA = WEIGHT * LOADFAC / (CLALPHA * Q0 * AREF)
239 IF ALPHA > ALPHA2MAX THEN ALPHA = ALPHA2MAX
240 ACCCOM = ACCCOMD
241 IF V0 > V02 AND H0 < H02 THEN ACCCOM = 0

```

```

242     IF H0 > H02 THEN ALPHA = 0
243     TQ0 = TQ0 + DT
244     B$ = "PHASE 2,CLIMB AT CONSTANT L/W*****"
245     GOSUB 14000: 'Go to thrust subroutine and return here
246     GOSUB 15000: 'Go to integration subroutine then return here
247     GOTO 12000: 'Go to runtime output section
248
249
250 6860 '-----
251
252     PHASE = 3: 'Commanded climb at constant Q0. An angle of attack (ALPHA)
253     'is computed that will correct the flight path angle (GAMMA) to
254     'seek the commanded Q0COM.
255     'This is GUID13 version. 29SEPT86. THIS ALGORITHM DEVELOPED BY
256     'WAYNE DONALDSON
257
258
259     Q0COM = Q0CMD
260     IF V0 < V0TEMP THEN GOTO 7100
261         IF FLAGS > 0 AND TSET5 = 1 THEN GOTO 8300
262     Q0COM = Q0CMD - (Q0CMD - Q0FINAL) * (V0 - V0TEMP) / (26200 - V0TEMP)
263         Q0CMD is reduced for V0 > V0TEMP
264 7100     IF Q0 < Q0COM THEN GOTO 7320
265         IF Q0 < Q0LD THEN GOTO 7220
266
267         RATEDQ = (Q0 - Q0LD) / DT
268 7180     ALPHA = ALPHA + A1 + RATEDQ * A2
269         GOTO 7500
270 7220     RATEDELQ = (Q0 - Q0COM) / DT
271         IF Q0 - Q0COM > DELQ THEN RATEDQ = (Q0LD - Q0) / DT
272         IF Q0 - Q0COM > DELQ THEN GOTO 7180
273         ALPHA = ALPHA - RATEDELQ * B1
274         GOTO 7500
275 7320     IF Q0 > Q0LD THEN GOTO 7420
276
277         RATEDQ = (Q0 - Q0LD) / DT
278 7380     ALPHA = ALPHA - C1 + RATEDQ * C2
279         GOTO 7500
280 7420     RATEDELQ = (Q0 - Q0COM) / DT
281         IF Q0COM - Q0 > DELQ THEN RATEDQ = (Q0LD - Q0) / DT
282         IF Q0COM - Q0 > DELQ THEN GOTO 7380
283         ALPHA = ALPHA + RATEDELQ * D1
284 7500     Q0LD = Q0
285         IF ALPHA > ALPHAMAX THEN ALPHA = ALPHAMAX
286         IF ALPHA < ALPHAMIN THEN ALPHA = ALPHAMIN
287
288     '-----
289     'CALCULATE ACCELERATION COMMAND FOR PHASE 3
290     '-----
291     IF Q0 <= Q0COM GOTO 7660 ELSE GOTO 7780
292 7660 '.....'For Q0 < Q0COM, maximum available acceleration is commanded
293
294     ACCCOM = ACCCOMD
295     GOTO 7980
296
297
298 7780 '.....'For Q0 > Q0MAX, acceleration is reduced linearly to zero for Q0
299     'between Q0COM and Q0MAX and is held to zero when
300     'Q0MAX is exceeded
301
302     ACCPERQ = ACCCOMD / (Q0COM - Q0MAX)
303     ACCCOM = ACCCOMD - ACCPERQ * (Q0COM - Q0)
304
305     IF Q0 > Q0MAX THEN ACCCOM = 0
306     GOTO 7980
307
308 7980 '.....
309
310     Q0OLD = Q0
311     RH0OLD = RH0
312     H0OLD = H0
313     IF V0CRUISE > 0 AND V0 >= V0CRUISE THEN GOTO 8100

```

```

314             IF FLAGS > 0 AND V0 > V05 THEN GOTO 8300
315             C$ = "PHASE 3,CLIMB AT COMMANDED Q*****"
316     GOSUB 14000: 'Go to THRUST subroutine and return here
317     GOSUB 15800: 'Goto integration subroutine and return here
318     GOTO 12000: 'Go to runtime output section
319
320 '-----
321 8180 PHASE = 4: 'Cruise at commanded V0CRUISE and Q0COM
322             ACCCOM = 0
323             IF SWITCH4 = 0 THEN GOTO 8260
324     IF FLAGCRUISE = 1 THEN GOTO 8250
325     '---
326     FOR ALPHA = 0 TO ALPHAMAX STEP .1
327             ITAB = 3: XARG = M0: GOSUB 16400: CD0 = VALI: 'Look up Cd0
328             ITAB = 4: XARG = H0: GOSUB 16400: DELCD = VALI: 'Look up delCd friction
329             ITAB = 5: XARG = M0: GOSUB 16400: CLALPHA = VALI: 'Look up Clalpha
330             ITAB = 6: XARG = M0: GOSUB 16400: K = VALI: 'LOOK UP K
331     CL = ALPHA * CLALPHA
332     CD = CD0 + K * CL ^ 2 + DELCD
333     LOVRD = CL / CD: PRINT ALPHA; LOVRD
334     IF LOVRD < LOVRDOLD THEN GOTO 8225
335     LOVRDOLD = LOVRD
336     NEXT ALPHA
337
338 8225 '-----
339     ALPHACRUISE = ALPHA - .1
340
341     FLAGCRUISE = 1
342 8250 ALPHA = ALPHACRUISE
343     IF GAMMA > GAMMA4MAX THEN ALPHA = 0!
344     IF LIFT < WEIGHT THEN ALPHA = 0!
345     IF GAMMA < 0! THEN ALPHA = ALPHAMAX
346
347 8260 '-----
348             D$ = "PHASE 4,CRUISE AT CONSTANT ALTITUDE *****"
349     IF SWITCH4 = 1 THEN D$ = "PHASE 4,CRUISE AT CONSTANT L/D *****"
350     GOSUB 14000: 'Go to thrust subroutine and return here
351     GOSUB 15800: 'Goto integration subroutine and return here
352     GOTO 12000: 'Go to runtime output section
353 8300 '-----
354     PHASE = 5: ' At V05 begin pullup to flight path angle GAMMAS then
355             continue ballistically at commanded acceleration ACCCOMD5
356
357             ACCCOM = ACCCOMD5
358             ALPHA = ALPHASMAX: ALPHAMAX = ALPHASMAX
359             IF TSET5 = 1 THEN ALPHA = 0
360             IF TSET5 = 1 THEN GOTO 8500
361             IF GAMMA >= GAMMA5 THEN TSET5 = 1 ELSE TSET5 = 0
362
363 8500     E$ = "PHASE 5,PULLUP AND BALLISTIC ASCENT"
364     GOSUB 14000: 'Go to thrust subroutine and return here
365     GOSUB 15800: 'Go to integration subroutine
366     GOTO 12000: 'Go to runtime output section
367
368 12000 '-----
369             ----- RUNTIME OUTPUT SECTION -----
370
371
372     ISPEFF = WEIGHT * (V0 - V00) / (GC * (WEIGHT0 - WEIGHT))
373     ITCUM = ITCUM + ISPEFF * (WEIGHT0 - WEIGHT)
374     ISPMEAN = ITCUM / (WLAUNCH - WEIGHT)
375
376     ITSUM = ITSUM + ISP * (WEIGHT0 - WEIGHT)
377     ISPAVG = ITSUM / (WLAUNCH - WEIGHT)
378     ACC = (V0 - V00) / (DT * GA)
379
380
381     'First test for weight or velocity stopping condition:
382     VORBIT = REARTH * SQRT(GC / (REARTH + H0))
383     IF V0 >= VORBIT GOTO 13020
384     IF WEIGHT <= WFINAL GOTO 13140
385     IF V0 >= VFINAL AND STAGE = 1 THEN GOTO 13190

```



```

386
387
388     COUNT = COUNT + 1
389     IF COUNT <= (DELPRINT - 1) THEN GOTO 12980
390     CLS
391     IF PHASE = 1 THEN X$ = A$ ELSE IF PHASE = 2 THEN X$ = B$ ELSE IF PHASE = 3 THEN
391 X$ = C$ ELSE IF PHASE = 4 THEN X$ = D$ ELSE IF PHASE = 5 THEN X$ = E$
392     PRINT USING "T=####.# \          \"; TIMEX; X$
393     PRINT USING "V0=#####.# M0=##.# ALPHA=###.##          WDOTF=#####.##          RES1=###
393 ##.#"; V0; M0; ALPHA; WDOTF; RES1
394     PRINT USING "H0=#####.#          GAMMA=###.##          WDOTA=#####.##          WDOTP=##
394 ##.#"; H0; GAMMA; WDOTA; WDOTP
395     PRINT USING "R=#####.##          Q0=#####.##          PHI=##.### "; RANGE / 607
395 6; Q0; PHI
396     PRINT USING "ISP=#####.##          ALPHAMAX=###.##          WT=#####.##          L/W =###
396 .###"; ISP; ALPHAMAX; WEIGHT; LIFT / WEIGHT
397     PRINT USING "ISPEFF=#####.##          ALPHAMIN=###.##          FC=#####.##"; ISPEFF;
397 ALPHAMIN; FC
398     PRINT USING "ISPMEAN=#####.##          ACCCOM =###.##          LIFT=#####.##          L/D =###
398 .###"; ISPMEAN; ACCCOM; LIFT; LIFT / DRAG
399     PRINT USING "ISPAVG=#####.##          ACCEL-G'S=###.##          DRAG=#####.##"; ISPAVG;
399 ACC; DRAG
400     PRINT USING "ISPA/B=#####.##          THRUST=#####.##          T/D =###
400 .###"; ISPA; THRUST; THRUST / DRAG
401     PRINT USING "ISPROC=#####.##          THRUSTCOM=#####.##"; ISPR;
401 THRUSTCOM
402     PRINT USING "          THRUSTMAX=#####.##          T/W =###
402 .###"; THRUSTMAX; THRUST / WEIGHT
403
404     COUNT = 0!
405
406
407     IF PRCOM <> 1 GOTO 12980
408
409     IF PRCOUNT < 1 GOTO 12660
410     IF 2 <= PRCOUNT <= 14 GOTO 12780
411 12660 LPRINT CHR$(12)
412     IF PRCOUNT < 1 THEN LPRINT TAB(5); TITL1$, TAB(5); TITL2$, DATE$, TIME$
413     WIDTH LPRINT 136
414     LPRINT TAB(5); "TIME"; TAB(15); "V0"; TAB(25); "H0"; TAB(35); "R"; TAB(45); "ISP";
414 TAB(55); "ISPEFF"; TAB(65); "ISPMEAN"; TAB(75); "WDOTF"; TAB(85); "WDOTA"; TAB(97); "PHI"
414 ; TAB(105); "THRUST"; TAB(112); "THRUSTCOM"; TAB(122); "THRUSTMAX"
415     LPRINT TAB(15); "Q0"; TAB(25); "ALPHA"; TAB(35); "GAMMA"; TAB(45); "WEIGHT"; TAB(55
415 ); "LIFT"; TAB(65); "DRAG"; TAB(75); "WDOTP"; TAB(85); "RES1"; TAB(94); "ISPAVG"; TAB(102
415 ); "ACCEL-G'S"; TAB(115); "ACCCOM"; TAB(128); "T/D"
416
417 12780 LPRINT
418     PRCOUNT = PRCOUNT + 1
419     WIDTH LPRINT 136
420
421     LPRINT USING "#####.##"; TIMEX; V0; H0; RANGE / 6076; ISP; ISPEFF; ISPMEAN;
421 WDOTF; WDOTA; PHI; THRUST; THRUSTCOM; THRUSTMAX
422     LPRINT USING "#####.##"; TIMEX; Q0; ALPHA; GAMMA; WEIGHT; LIFT; DRAG; WDOTP;
422 RES1; ISPAVG; ACC; ACCCOM; THRUST / DRAG
423     WIDTH LPRINT 80
424
425     IF PRCOUNT >= 16 THEN PRCOUNT = 0
426
427 12980 NEXT I
428
429

```

```

430 13020 '----- TEST FOR STOPPING CONDITIONS -----
431
432
433     PRINT USING "REACHED ORBITAL VELOCITY=#####.# FPS AT TIME=#####.#SECS. ,WEIGHT=
433 #####.# LBS."; V0; TIMEX; WEIGHT
434     IF PRCOM = 1 THEN LPRINT USING "REACHED ORBITAL VELOCITY=#####.# FPS AT TIME=##
434 ###.#SECS.,WEIGHT= #####.# LBS."; V0; TIMEX; WEIGHT
435     GOTO 13220
436 13140     PRINT USING "FUEL EXHAUSTED AT TIME= #####.#SECS.,WEIGHT= #####.#,VELOCITY=####
436 ##.# FPS."; TIMEX; WEIGHT; V0
437     IF PRCOM = 1 THEN LPRINT USING "FUEL EXHAUSTED AT TIME= #####.#SECS.,WEIGHT= ###
437 ###.#,VELOCITY=#####.#FPS."; TIMEX; WEIGHT; V0
438     GOTO 13220
439 13190     PRINT USING "REACHED 1ST STAGE VELOCITY=#####.# FPS AT TIME=#####.#SECS. ,WEIGH
439 T= #####.# LBS."; V0; TIMEX; WEIGHT
440     IF PRCOM = 1 THEN LPRINT USING "REACHED 1ST STAGE VELOCITY=#####.# FPS AT TIME=
440 #####.#SECS.,WEIGHT= #####.# LBS."; V0; TIMEX; WEIGHT
441     GOTO 13220
442
443 13220     IF NSTAGES = 1 THEN GOTO 13250
444 13240     IF NSTAGES = 2 AND STAGE = 1 THEN GOSUB 18000: 'Reads 2nd stage data
445             IF TIMEX = TIMEX2 THEN GOTO 3100
446 13250     IF PRCOM = 0 AND PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 13300
447
448 13260     'LPRINT CHR$(27); "B"; CHR$(1); : 'Select standard print (SR15 printer)
449     LPRINT CHR$(27); CHR$(33); CHR$(0); : 'Select standard print (Epson printer)
450     WIDTH LPRINT 80
451     IF PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 13300
452     INPUT "Do you want printer output of run(Y/N)"; PLOT$
453     IF PLOT$ = "N" OR PLOT$ = "n" THEN GOTO 13300
454     CLOSE 2
455     IF PLOT$ = "Y" OR PLOT$ = "y" THEN GOSUB 20000: 'Goes to printer output subroutine
455
456 13300 '---
457     PRINT TIME$
458     PRINT STARTTIME$; FINISH = TIMER
459     PRINT "PROGRAM TOOK"; FINISH - START;
460     PRINT "SECONDS"
461     END
462
463     '-----End of main routine -----
464
465
466
467
468     '$PAGE

```

```

469 *****
470 *****
471 |
472 |                               SUBROUTINES
473 |
474 |*****
475 |*****
476 |
477 13500 *****
478 |           SUBROUTINE TO DEFINE ATMOSPHERIC CONDITIONS
479 |           USING
480 |           U.S. STANDARD ATMOSPHERE, 1962
481 |           4FEB87 VERSION
482 |*****
483 |
484 |----- DATA -----
485 |FOR J1 = 1 TO 10
486 |READ HB(J1):READ RH0B(J1):READ TMB(J1):READ GLMB(J1)
487 |NEXT J1
488 |
489 13540 'DATA 0,2.3769E-3,518.7,-3.5662E-3
490 'DATA 36089,7.0612E-4,390,0
491 'DATA 65617,1.7082E-4,390,5.4864E-4
492 'DATA 104986,2.5661E-5,411.6,1.5362E-3
493 'DATA 154199,2.7698E-6,487.2,0
494 'DATA 170604,1.4735E-6,487.2,-1.0973E-3
495 'DATA 200131,4.8719E-7,454.8,-2.1946E-3
496 'DATA 259186,3.8826E-8,325.2,0
497 'DATA 291153,6.1508E-9,325.2,1.6953E-3
498 'DATA 323003,9.6511E-10,374.2,2.8343E-3
499 |RESTORE 13540
500 |----- SET CONSTANTS -----
501 |           N1 = 10
502 |           GC = 32.17405
503 |           RE = 2.0899E+07: ' radius of earth
504 |           CPCV = 1.4: ' ratio of specific heats
505 |----- CONVERT TO GEOPOTENTIAL ALTITUDE -----
506 |H = (RE * H0) / (RE + H0): ' geopotential altitude
507 |----- FINDS ALTITUDE BASE POINT -----
508 |FOR I1 = 2 TO N1
509 |HDEL = H - HB(I1)
510 |IF HDEL <= 0 GOTO 13740
511 |NEXT I1
512 |
513 |
514 13740 |----- CALCULATE TEMPERATURE AND DENSITY -----
515 |IF GLMB(I1 - 1) <> 0 GOTO 13840: 'checks to see if temp changes with alt.
516 |R0 = RH0B(I1 - 1) * EXP(-(H - HB(I1 - 1)) * GC / (1716.483 * TMB(I1 - 1)))
517 |T0 = TMB(I1 - 1): ' temp doesnt change with alt
518 |GOTO 13880
519 13840 |T0 = TMB(I1 - 1) + GLMB(I1 - 1) * (H - HB(I1 - 1)): 'temp changes with alt
520 |R0 = RH0B(I1 - 1) * EXP(-(! + GC / (1716.483 * GLMB(I1 - 1))) * LOG(T0 / TMB(I1 -
520 |)))
521 13880 |----- AMBIENT CONDITIONS -----
522 |RH0 = R0 * GC: ' converts density to lbm/cu.ft
523 |P0 = RH0 * T0 * 53.28 / 144: 'pressure in psia
524 |
525 |C0 = 49.01 * SQR(T0): ' ft/sec
526 |V0 = M0 * C0
527 |Q0 = RH0 * (V0 * 2) / (2 * GC): ' lbf/sqft
528 |FC = (WEIGHT / GC) * (V0 * 2) / (H0 + REARTH)
529 |G0 = (P0 * V0) / (53.28 * T0): ' lbm/(sec*sqrin)
530 |RETURN
531 |
532 |$PAGE

```

```

533 14000 '*****
534 '          THRUST CALCULATION SUBROUTINE
535 '*****
536 'First update lift and drag:
537 '
538 ITAB = 3: XARG = M0: GOSUB 16400: CDD = VALI: 'Look up Cdd
539 ITAB = 4: XARG = H0: GOSUB 16400: DELCD = VALI: 'Look up delCd friction
540 ITAB = 5: XARG = M0: GOSUB 16400: CLALPHA = VALI: 'Look up Clalpha
541 ITAB = 6: XARG = M0: GOSUB 16400: K = VALI: 'LOOK UP K
542 CL = ALPHA * CLALPHA
543 LIFT = CL * Q0 * AREF
544 CD = CDD + K * CL ^ 2 + DELCD
545 DRAG = CD * Q0 * AREF
546 '
547 '-----
548 ' Calculate commanded THRUST value
549 '
550 THRUSTCOM = (WEIGHT / (GC * COS(ALPHA / ANG))) * (ACCCOM * GA + GA * SIN(GAMMA /
550 ANG)) + DRAG / (COS(ALPHA / ANG))
551 '-----
552 ''AIRBREATHER THRUST-----
553 ITAB = 7: XARG = M0: YARG = ALPHA: GOSUB 16400: A0AC = VALI: 'Look up A0/AC
554 ITAB = 2: XARG = M0: GOSUB 16400: PHIMAX = VALI: 'Look up PHImax
555 PHI = PHIMAX
556 ITAB = 1: XARG = M0: YARG = PHI: GOSUB 16400: ISPA = VALI: 'Look up airbreathing I
556 sp
557
558 A0 = A0AC * AC
559 IF PHI = 0 THEN A0 = 0
560 WDOTA = G0 * A0 * 144
561 FAMAX = FASTOIC * PHI
562
563 WDOTFMAX = FAMAX * WDOTA
564 THRUSTAMAX = ISPA * WDOTFMAX
565 ''ROCKET THRUST-----
566 ''
567 ITAB = 8: XARG = H0: GOSUB 16400: ISPR = VALI: 'Look up rocket Isp
568 ITAB = 9: XARG = M0: GOSUB 16400: WDOTPMAX = VALI: 'Rocket propellant flow
569
570 THRUSTRMAX = ISPR * WDOTPMAX
571 THRUSTMAX = THRUSTAMAX + THRUSTRMAX
572 THRUST = THRUSTMAX
573 WDOTF = WDOTFMAX
574 WDOTP = WDOTPMAX
575 WDOT = WDOTFMAX + WDOTPMAX
576 ISP = THRUST / WDOT
577
578 IF THRUST < THRUSTCOM GOTO 15660: '-----
579 '
580 '-----
581 '' For THRUSTMAX > THRUSTCOM reduce rocket to zero if needed and
582 ' reduce airbreathing thrust as needed to obtain THRUSTCOM
583
584 IF WDOTPMAX = 0 THEN GOTO 15260: '-----
585 '
586 DELTHRUST = THRUSTMAX - THRUSTCOM
587 IF DELTHRUST > THRUSTRMAX THEN WDOTP = 0
588 IF WDOTP = 0 THEN GOTO 15260: '-----
589 ''''Throttle rocket to reduce thrust
590 THRUSTR = THRUSTCOM - THRUSTAMAX
591 WDOTP = THRUSTR / ISPR
592 WDOTF = WDOTFMAX
593 WDOT = WDOTF + WDOTP
594 THRUST = THRUSTAMAX + THRUSTR
595 ISP = THRUST / WDOT
596 GOTO 15660: '-----
597 15260 IF THRUST = 0! THEN THRUST = 1!
598 IF A0AC = 0 OR PHIMAX = 0 OR ISPA = 0 THEN GOTO 15660: '-----
599 PHI = PHIMAX
600 ''
601 15340 '

```

```
602          ITAB = 1: XARG = M0: YARG = PHI: GOSUB 16400: ISPA = VALI: 'Look up Isp
603
604          THRUST = ISPA * WDOTF
605          WDOTF = WDOTF * THRUSTCOM / THRUST
606
607          PHI = (WDOTF / WDOTA) / FASTOIC
608
609          IF ABS(THRUST - THRUSTCOM) < 200 GOTO 15560
610
611          GOTO 15340: ' Returns to try new PHI value
612 15560 '
613          WDOTF = PHI * FASTOIC * WDOTA
614          THRUST = ISPA * WDOTF
615          WDOT = WDOTF
616          ISP = ISPA
617 15660 '
618          IF WDOTF = 0 THEN ISPA = 0
619          IF WDOTP = 0 THEN ISPR = 0
620
621          RETURN
622
623
624          $PAGE
```

```

625 15800 *****
626 '          INTEGRATION SUBROUTINE
627 '          2ND ORDER R-K INTEGRATION (HEUN'S METHOD)
628 '*****
629 IF PLOTS = "N" OR PLOTS = "n" THEN GOTO 15920
630 '-----CREATE/ADD TO PLOTTER FILE-----
631 IF FREQC < FREQ THEN GOTO 15910
632 FREQC = 0
633 WRITE #2, TIMEX, V0, H0, RANGE / 6076, ISP, ISPEFF, ISPMEAN, WDOTF, WDOTA, PHI,
633 THRUST, THRUSTCOM, THRUSTMAX, Q0, ALPHA, GAMMA, WEIGHT, LIFT, DRAG, WDOTP, RES1, ISPAVG,
633 ACC, ACCCOM, THRUST / DRAG
634 15910 FREQC = FREQC + 1
635 '-----
636 15920 '
637 '
638 '-----Set "old" values of integration variables-----
639 TIMEX0 = TIMEX
640 WEIGHT0 = WEIGHT
641 V00 = V0
642 GAMMA0 = GAMMA
643 H00 = H0
644 RANG0 = RANGE
645 '
646 '-----Start integration-----
647 FOR N = 1 TO 2
648 '
649 '
650 '-----Weight increment-----
651 DELW(N) = (-WDOT) * DT / 1
652 '-----Velocity increment-----
653 IF TIMEX <= DT THEN WEIGHT = WLAUNCH
654 DELV(N) = (GC / WEIGHT) * ((THRUST * COS(ALPHA / ANG) - DRAG - WEIGHT * (GA / GC) *
654 SIN(GAMMA / ANG))) * DT / 1
655 '-----Flight path angle increment-----
656 IF V0 < DT THEN GOTO 15940
657 DELGAM(N) = (GC / (WEIGHT * V0)) * ((LIFT + THRUST * SIN(ALPHA / ANG) + FC * (GA /
657 GC) * COS(GAMMA / ANG) - WEIGHT * (GA / GC) * COS(GAMMA / ANG))) * DT / 1
658 '-----Altitude increment-----
659 15940 DELH(N) = (V0 * SIN(GAMMA / ANG)) * DT / 1
660 '-----Range increment-----
661 DELR(N) = (V0 * COS(GAMMA / ANG)) * DT / 1
662 '
663 '-----Go back for 2nd pass thru equations-----
664 TIMEX = TIMEX + DT / 2
665 WEIGHT = WEIGHT + DELW(N)
666 V0 = V0 + DELV(N)
667 GAMMA = GAMMA + DELGAM(N)
668 H0 = H0 + DELH(N)
669 RANGE = RANGE + DELR(N)
670 NEXT N
671 '-----After two passes :
672 '-----Add increments as below of two-----
673 '          integration steps to "old" values to
674 '          obtain "new" values of integration
675 '          variables at "new" time
676 TIMEX = TIMEX0 + DT
677 WEIGHT = WEIGHT0 + (DELW(1) + DELW(2)) / 2
678 V0 = V00 + (DELV(1) + DELV(2)) / 2
679 M0 = V0 / C0
680 GAMMA = GAMMA0 + (DELGAM(1) + DELGAM(2)) / 2
681 H0 = H00 + (DELH(1) + DELH(2)) / 2
682 RANGE = RANG0 + (DELR(1) + DELR(2)) / 2
683 Q0OLD = Q0: H0OLD = H0: GAMMAOLD = GAMMA: RH0OLD = RH0
684 '
685 '-----End of integration routine-----
686 RETURN
687 '
688 '$PAGE

```

```

689 16000 *****
690      '
691          TABLE LOADER SUBROUTINE
692          THIS SECTION PROVIDED BY KEN WATSON
693      '
694      IL = IL + 1: IF IL = 1 THEN ILTI = 0: 'Initialize table location index
695      I1ST(IL) = 2: J1ST(IL) = 2
696      IO(IL) = ILTI + 1: ITO = IO(IL): 'Set table origin
697      INPUT #1, T$(IL), AT(ITO): 'Read table title & type
698      IF AT(ITO) = 1 THEN 16090 ELSE 16150
699 16090      '
700          1-D TABLE
701      IX = ITO + 1: INPUT #1, AT(IX): 'Input number of points
702      IY = IX + AT(IX)
703      FOR I = 1 TO AT(IX): INPUT #1, AT(I + IX): NEXT I: 'Input x
704      FOR I = 1 TO AT(IX): INPUT #1, AT(I + IY): NEXT I: 'Input y
705      ILTI = AT(IX) + IY: RETURN: 'Set index of last point
706 16150      '
707          2-D TABLE
708      IX = ITO + 2: INPUT #1, AT(ITO + 1), AT(IX): 'Input N and M
709      IY = IX + AT(ITO + 1): IZ = IX + AT(IX)
710      FOR I = 1 TO AT(ITO + 1): INPUT #1, AT(I + IX): NEXT I: 'Input x
711      FOR I = 1 TO AT(IX): INPUT #1, AT(I + IY): NEXT I: 'Input y
712      FOR J = 1 TO AT(IX): FOR I = 1 TO AT(ITO + 1): 'Input z
713      K = I + AT(ITO + 1) * J + IZ: INPUT #1, AT(K): NEXT I: NEXT J
714      ILTI = AT(ITO + 1) * (1 + AT(IX)) + IZ: RETURN
715      '
716      '$PAGE

```

```
717 16230 *****
718      '          TABLE PRINTER SUBROUTINE
719      '          PROVIDED BY KEN WATSON
720      '*****
721      LPRINT : LPRINT : LPRINT "      "; T$(ITAB): ITO = IO(ITAB)
722      LPRINT "      "; AT(ITO); "-D TABLE", "STARTS AT "; ITO: LPRINT
723      IF AT(ITO) = 1 THEN 16290 ELSE 16340
724 16290 IX = ITO + 1: IY = IX + AT(IX)
725      LPRINT TAB(10); "X"; TAB(25); "Y"
726      FOR I = 1 TO AT(IX)
727      LPRINT USING "#####.####"; AT(I + IX), AT(I + IY): NEXT I
728      RETURN
729 16340 IX = ITO + 2: IY = IX + AT(ITO + 1): IZ = IX + AT(ITO + 2)
730      LPRINT TAB(10); "X"; TAB(25); "Y"; TAB(35); "Z"
731      FOR J = 1 TO AT(ITO + 2): FOR I = 1 TO AT(ITO + 1)
732      K = I + J * AT(ITO + 1) + IZ
733      LPRINT USING "#####.####"; AT(I + IX), AT(J + IY), AT(K)
734      NEXT I: NEXT J: RETURN
735      '
736      '
737      '
738      '$PAGE
```



```

739 16400 '*****
740      '          TABLE INTERPOLATION SUBROUTINE
741      '          PROVIDED BY KEN WATSON
742      '*****
743      ITO = IO(ITAB): IF AT(ITO) = 1 THEN 16440 ELSE 16590
744 16440 '          1-D table
745      IX = ITO + 1: N = AT(IX): IY = IX + N
746      IF XARG <= AT(IX + 1) THEN 16470 ELSE 16490
747 16470 I = 2
748 16480 I2 = I + IX: I1 = I2 - 1: J2 = I + IY: J1 = J2 - 1: GOTO 16570
749 16490 IF XARG >= AT(IX + N) THEN 16500 ELSE 16510
750 16500 I = N: GOTO 16480
751 16510 I = I1ST(ITAB)
752 16520 I2 = I + IX: I1 = I2 - 1: J2 = I + IY: J1 = J2 - 1
753      IF XARG <= AT(I2) THEN 16540 ELSE 16560
754 16540 IF XARG > AT(I1) THEN 16570 ELSE 16550
755 16550 I = I - 1: GOTO 16520
756 16560 I = I + 1: GOTO 16520
757 16570 VAL1 = AT(J1) + (AT(J2) - AT(J1)) * ((XARG - AT(I1)) / (AT(I2) - AT(I1)))
758      I1ST(ITAB) = I: RETURN
759      '
760      '
761      '
762 16590 '          2-D table
763      IX = ITO + 2: N = AT(ITO + 1): M = AT(ITO + 2): IY = IX + N: IZ = IX + M
764      IF XARG <= AT(1 + IX) THEN 16630 ELSE 16620: 'Locate I1 & I2
765 16620 IF XARG >= AT(N + IX) THEN 16640 ELSE 16650
766 16630 I = 2: GOTO 16710
767 16640 I = N: GOTO 16710
768 16650 I = I1ST(ITAB)
769 16660 I2 = I + IX: I1 = I2 - 1
770      IF XARG <= AT(I2) THEN 16680 ELSE 16700
771 16680 IF XARG > AT(I1) THEN 16710 ELSE 16690
772 16690 I = I - 1: GOTO 16660
773 16700 I = I + 1: GOTO 16660
774 16710 I1ST(ITAB) = I
775      I2 = I + IX: I1 = I2 - 1
776      IF YARG <= AT(1 + IY) THEN 16750 ELSE 16740: 'Locate J1 & J2
777 16740 IF YARG >= AT(M + IY) THEN 16760 ELSE 16770
778 16750 J = 2: GOTO 16830
779 16760 J = M: GOTO 16830
780 16770 J = J1ST(ITAB)
781 16780 J2 = J + IY: J1 = J2 - 1
782      IF YARG <= AT(J2) THEN 16800 ELSE 16820
783 16800 IF YARG > AT(J1) THEN 16830 ELSE 16810
784 16810 J = J - 1: GOTO 16780
785 16820 J = J + 1: GOTO 16780
786 16830 J1ST(ITAB) = J
787      J2 = J + IY: J1 = J2 - 1
788      K11 = I - 1 + N * (J - 1) + IZ: K21 = I + N * (J - 1) + IZ
789      K12 = I - 1 + N * J + IZ: K22 = I + N * J + IZ
790      DXI = (XARG - AT(I1)) / (AT(I2) - AT(I1)): 'Have I1,I2,J1,J2,K11,K21,K12,K22
791      DYJ = (YARG - AT(J1)) / (AT(J2) - AT(J1))
792      VAL1 = AT(K11) + (AT(K21) - AT(K11)) * DXI
793      VAL2 = AT(K12) + (AT(K22) - AT(K12)) * DXI
794      VALI = VAL1 + (VAL2 - VAL1) * DYJ
795      RETURN
796      '
797      '
798      '$PAGE

```

```

799 18000 *****
800 '          SUBROUTINE TO LOAD SECOND STAGE DATA FILE
801 '*****
802     OPEN "I", #1, IFN2$: IL = 0: IFILE = 1
803 '
804     LINE INPUT #1, TITL1$
805     LINE INPUT #1, TITL2$
806 '
807     GOSUB 16000: 'Input airbreather Isp vs. Mach and PHI as table #1
808     GOSUB 16000: 'Input airbreather PHImax vs. Mach as table #2
809     GOSUB 16000: 'Input Cdo vs. Mach as table #3
810     GOSUB 16000: 'Input delta Cd vs. Alt. as table #4
811     GOSUB 16000: 'Input Clalpha vs. Mach as table #5
812     GOSUB 16000: 'INPUT K VS. MACH AS TABLE #6
813     GOSUB 16000: 'Input airbreather Ao/Ac vs. Mach and alpha as table #7
814     GOSUB 16000: 'Input rocket Isp vs. Altitude as table #8
815     GOSUB 16000: 'Input rocket WDOTPMAX vs. Mach as table #9
816 '
817     INPUT #1, AC, AREF
818     INPUT #1, WEIGHT, WFUEL, WFINAL, VFINAL, STAGE
819     INPUT #1, DT, DELPRINT
820     INPUT #1, Q0COMD, Q0MAX, Q0MIN, ALPHAMAX, ALPHAMIN
821 '
822     INPUT #1, LOADFAC, ACCCOMD, GAMMAMAX
823     INPUT #1, FASTOIC
824     INPUT #1, V0TEMP, Q0FINAL
825     INPUT #1, V0CRUISE, SWITCH4, GAMMA4MAX
826     INPUT #1, FLAGS, V05, ALPHASMAX, GAMMA5, ACCCOMD5
827     INPUT #1, A1, A2, B1, C1, C2, D1, DELQ, V02, H02
828 '
829     PRINT
830     PRINT "INPUT FILE FOR 2ND STAGE HAS BEEN READ."
831     PRINT "LAST ITEM IN 2ND STAGE INPUT FILE IS H02, VALUE READ FROM FILE IS"; H02
832     PRINT "IF THIS VALUE IS INCORRECT THEN INPUT FILE IS IN ERROR."
833     PRINT
834     INPUT "Ready to run 2nd stage:
834     Do you want a listing of 2nd stage tabular input data(Y/N)"; WTIN$
835     IF WTIN$ = "Y" OR WTIN$ = "y" THEN 18230 ELSE 18240
836 18230 FOR ITAB = 1 TO 9: GOSUB 16230: NEXT ITAB
837 18240     TIMEX2 = TIMEX
838     IF EOF(1) THEN GOTO 18250
839 18250     CLOSE #1
840     RETURN: 'Returns to main routine
841 '
842 '
843     '$PAGE

```

```

844 20000 'PRIN.BAS *****
845 '          POST-RUN PRINTER OUTPUT SUBROUTINE
846 '*****
847 ' 8MAY87/0030/PROGRAM "ETOTAB.BAS",MERGED FOR USE WITH PROGRAM "ETO"
848 N = 55: 'Number of lines printed per page
849 WIDTH LPRINT 136
850 '
851 LPRINT TAB(40); DATE$; TAB(60); TIME$
852 LPRINT
853 LPRINT TAB(10); "1ST/SINGLE STAGE INPUT DATA FILE: "; IFN$
854 LPRINT TAB(10); "2ND STAGE INPUT DATA FILE      : "; IFN2$
855 LPRINT
856 LPRINT TAB(10); "PLOTTER/PRINTER FILE NAME      : "; PFN$
857 LPRINT
858 LPRINT
859 LPRINT TAB(10); "-----PROGRAM ETO2 RUN SUMMARY-----"
860 LPRINT USING "          LAUNCH WEIGHT = #####.## LBS."; WLAUNCH
861 LPRINT USING "          FINAL WEIGHT = #####.## LBS."; WEIGHT
862 LPRINT USING "FUEL/PROPELLANT REMAINING =#####.## LBS."; WEIGHT - WFINAL
863 LPRINT
864 IF V0 >= VORBIT THEN LPRINT TAB(10); "VEHICLE REACHED ORBITAL VELOCITY."
865 IF WEIGHT <= WFINAL THEN LPRINT TAB(10); "RUN TERMINATED BY FUEL EXHAUSTION"
866 LPRINT
867 LPRINT USING "FLIGHT TIME      = #####.## SECS."; TIMEX
868 LPRINT USING "FINAL VELOCITY = #####.## FT/SEC"; V0
869 LPRINT USING "FINAL ALTITUDE = #####.## FT"; H0
870 LPRINT USING "FINAL Q0       = #####.## PSF"; Q0
871 LPRINT
872 LPRINT USING "AVERAGE ISP      = #####.## SEC "; ISPAVG
873 LPRINT USING "MEAN ISP        = #####.## SEC "; ISPMEAN
874 '
875 LPRINT
876 LPRINT
877 LPRINT
878 LPRINT "FINAL CONDITIONS ARE SHOWN BELOW:"
879 LPRINT
880 WIDTH LPRINT 136
881 ''' LPRINT CHR$(27); "B"; CHR$(3); 'Selects condensed print(SR15 printer)
882 LPRINT CHR$(27); CHR$(33); CHR$(4); : 'Selects condensed print(Epson printer)
883 LPRINT TAB(5); "TIME"; TAB(15); "V0"; TAB(25); "H0"; TAB(35); "R"; TAB(45); "ISP";
883 TAB(55); "ISPEFF"; TAB(65); "ISPMEAN"; TAB(75); "WDOTF"; TAB(85); "WDOTA"; TAB(97); "PHI"
883 ; TAB(105); "THRUST"; TAB(112); "THRUSTCOM"; TAB(122); "THRUSTMAX"
884 LPRINT TAB(15); "Q0"; TAB(25); "ALPHA"; TAB(35); "GAMMA"; TAB(45); "WEIGHT"; TAB(55
884 ); "LIFT"; TAB(65); "DRAG"; TAB(75); "WDOTP"; TAB(85); "RES1"; TAB(94); "ISPAVG"; TAB(102
884 ); "ACCEL-G'S"; TAB(115); "ACCCOM"; TAB(128); "T/D"
885 LPRINT
886 LPRINT USING "#####.##"; TIMEX; V0; H0; RANGE / 6076; ISP; ISPEFF; ISPMEAN;
886 WDOTF; WDOTA; PHI; THRUST; THRUSTCOM; THRUSTMAX
887 LPRINT USING "#####.##"; TIMEX; Q0; ALPHA; GAMMA; WEIGHT; LIFT; DRAG; WDOTP;
887 RES1; ISPAVG; ACC; ACCCOM; THRUST / DRAG
888 WIDTH LPRINT 80
889 LPRINT CHR$(12); 'Form feed
890 'LPRINT CHR$(27) "B" CHR$( 1) : 'Resets to standard print(SR15 printer)
891 LPRINT CHR$(27); CHR$(33); CHR$(0); : 'Resets to standard print(Epson printer)
892 '
893 CMD1$ = "COPY" + CHR$(32) + IFN$ + CHR$(32) + "PRN"
894 PRINT CMD1$
895 SHELL CMD1$
896 LPRINT CHR$(12); 'Form feed
897 IF NSTAGES = 1 THEN GOTO 20820
898 CMD2$ = "COPY" + CHR$(32) + IFN2$ + CHR$(32) + "PRN"
899 PRINT CMD2$
900 SHELL CMD2$
901 LPRINT CHR$(12)
902 20820 WIDTH LPRINT 136
903 '''LPRINT CHR$(27) "B" CHR$(3) : 'Selects condensed print(SR15 printer)
904 LPRINT CHR$(27); CHR$(33); CHR$(4); : 'Selects condensed print(Epson printer)
905 '
906 '
907 DIM V(25)
908 COUNT = 0

```

```

909     LPRINT TAB(10); PFNS; TAB(40); DATE$; TAB(60); TIME$
910     GOSUB 22460
911     '
912     OPEN "I", 2, PFNS: INPUT #2, VARNUM
913 21140 FOR I = 1 TO 25
914         INPUT #2, V(I)
915         IF EOF(2) THEN GOTO 21340
916     NEXT I
917     COUNT = COUNT + 1
918     LPRINT USING "#####.##"; V(1); V(3); V(2); V(14); V(4); V(17); V(8); V(9); V(10)
919 ); V(20); V(21)
920     IF COUNT >= N THEN LPRINT CHR$(12)
921     IF COUNT >= N THEN COUNT = 0
922     IF COUNT = 0 THEN GOSUB 22460
923     GOTO 21140
924 21340 CLOSE 2
925     '.....'
926     GOTO 21700: 'SKIPS OVER 2ND FIELD FOR PROGRAM "ETO"
927     LPRINT CHR$(12): 'FORM FEED
928     COUNT = 0
929     GOSUB 22500
930 21480 OPEN "I", 2, PFNS: INPUT #2, VARNUM
931     FOR I = 1 TO 25
932         INPUT #2, V(I)
933         IF EOF(2) THEN GOTO 21680
934     NEXT I
935     COUNT = COUNT + 1
936     LPRINT USING "#####.##"; V(1); V(3); V(2); V(8); V(9); V(10); V(29); V(28); V(2
937 )
938     IF COUNT >= N THEN LPRINT CHR$(12)
939     IF COUNT >= N THEN COUNT = 0
940     IF COUNT = 0 THEN GOSUB 22500
941     GOTO 21480
942 21680 CLOSE 2
943 21700 '.....'
944     LPRINT CHR$(12): 'FORM FEED
945     COUNT = 0
946     GOSUB 22540
947 21800 OPEN "I", 2, PFNS: INPUT #2, VARNUM
948     FOR I = 1 TO 25
949         INPUT #2, V(I)
950         IF EOF(2) THEN GOTO 22000
951     NEXT I
952     COUNT = COUNT + 1
953     LPRINT USING "#####.##"; V(1); V(3); V(2); V(5); V(22); V(6); V(7); V(11); V(12)
954 ); V(13)
955     IF COUNT >= N THEN LPRINT CHR$(12)
956     IF COUNT >= N THEN COUNT = 0
957     IF COUNT = 0 THEN GOSUB 22540
958     GOTO 21800
959 22000 CLOSE 2
960     '.....'
961     LPRINT CHR$(12): 'FORM FEED
962     COUNT = 0
963     GOSUB 22580
964 22120 OPEN "I", 2, PFNS: INPUT #2, VARNUM
965     FOR I = 1 TO 25
966         INPUT #2, V(I)
967         IF EOF(2) THEN GOTO 22320
968     NEXT I
969     COUNT = COUNT + 1
970     LPRINT USING "#####.##"; V(1); V(3); V(2); V(14); V(15); V(16); V(18); V(19);
971 V(23); V(24); V(25)
972     IF COUNT >= N THEN LPRINT CHR$(12)
973     IF COUNT >= N THEN COUNT = 0
974     IF COUNT = 0 THEN GOSUB 22580
975     GOTO 22120
976 22320 CLOSE 2
977     '.....'
978     ' ' LPRINT CHR$(27); "B"; CHR$(1): 'Resets to standard print
979     LPRINT CHR$(27); CHR$(33); CHR$(0); : ' Resets to standard print(Epson printer)

```

```
977      WIDTH LPRINT 80
978      LPRINT CHR$(12): 'Form feed
979      GOTO 22999
980 22460 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "Q0"; TAB(54); "R";
980 TAB(66); "WEIGHT"; TAB(78); "WDOTF "; TAB(90); " WDOTA "; TAB(102); " PHI "; TAB(114)
980 ; " WDOTP "; TAB(125); " RES1 "
981      RETURN
982 22500 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "WDOTF"; TAB(54); "WD
982 OTA"; TAB(66); "PHI"; TAB(78); "WDOTFROC"; TAB(90); "WDOTLOX"; TAB(102); "WDOTP"
983      RETURN
984 22540 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "ISP"; TAB(54); "ISPA
984 VG"; TAB(66); "ISPEFF"; TAB(78); "ISPMEAN"; TAB(90); "THRUST"; TAB(102); "THRUSTCOM";
984 TAB(114); "THRUSTMAX"
985      RETURN
986 22580 LPRINT TAB(6); "TIME"; TAB(18); "H0"; TAB(30); "V0"; TAB(42); "Q0"; TAB(54); "ALPHA
986 "; TAB(66); "GAMMA"; TAB(78); "LIFT"; TAB(90); "DRAG"; TAB(102); "ACCEL-G'S"; TAB(114); "
986 ACCCOM"; TAB(129); "T/D"
987      RETURN
988 22999 RETURN
989
990
```

## 5.0 CROSS REFERENCED LISTING OF PROGRAM ETO VARIABLES

Each variable used in the program is listed alphabetically along with the sequential program line numbers for each occurrence. The line numbers are consistent with the program listing in Section 4.0.

A\$	192	192	226	391								
A0	558	559	560									
A0AC	180	553	558	598								
AI	91	268	827									
A2	91	268	827									
AC	81	558	817									
ACC	378	399	422	633	887							
ACCCOM	221	240	241	294	303	305	322	357	398	422	550	633
	887											
ACCCOMD	86	221	240	294	302	303	822					
ACCCOMD5	90	357	826									
ACCPERQ	302	303										
ALPHA	83	180	184	220	223	238	239	239	242	268	268	273
	273	278	278	283	283	285	285	286	286	326	331	333
	336	339	342	343	344	345	358	359	393	422	542	550
	550	553	633	654	657	887						
ALPHA2MAX	85	239	239									
ALPHA5MAX	90	358	358	826								
ALPHACRUISE	339	342										
ALPHAMAX	84	285	285	326	345	358	396	820				
ALPHAMIN	84	286	286	397	820							
ANG	143	550	550	550	654	654	657	657	657	659	661	
AREF	81	185	187	238	543	545	817					
AT	41	697	698	700	701	702	702	703	703	704	706	706
	707	707	708	708	709	709	710	710	711	711	712	712
	722	723	724	726	727	727	729	729	731	731	732	733
	733	733	743	745	746	749	753	754	757	757	757	757
	757	757	763	763	764	765	770	771	776	777	782	783
	790	790	790	791	791	791	792	792	792	793	793	793
B\$	244	391										
B1	91	273	827									
BIGG	141	203										
C\$	315	391										
C0	525	526	679									
C1	91	278	827									
C2	91	278	827									
CD	186	187	332	333	544	545						
CD0	176	186	327	332	538	544						
CL	184	185	186	331	332	333	542	543	544			
CLALPHA	178	184	238	329	331	540	542					
CMD1\$	893	894	895									
CMD2\$	898	899	900									
COUNT	149	388	388	389	404	908	917	917	919	920	920	921
	927	934	934	936	937	937	938	943	950	950	952	953
	953	954	959	966	966	968	969	969	970			
CPCV	504											
D\$	348	349	391									
D1	91	283	827									
DELCD	177	186	328	332	539	544						
DELGAM	43	657	667	680	680							
DELH	43	659	668	681	681							
DELPRINT	83	389	819									
DELQ	91	271	272	281	282	827						
DELR	43	661	669	682	682							
DELTHRUST	586	587										
DELV	43	654	666	678	678							
DELW	43	651	665	677	677							
DRAG	187	398	399	400	422	422	545	550	633	633	654	887
	887											
DT	83	205	208	209	209	226	235	243	267	270	271	277
	280	281	378	651	653	654	656	657	659	661	664	676
	819											
DV	207	208										
DXI	790	792	793									
DYJ	791	794										
E	139	140	141	503								
E\$	363	391										
FAMAX	561	563										
FASTOIC	87	561	607	613	823							
FC	142	397	528	657								
FINISH	458	459										
FLAGS	90	261	314	826								

FLAGCRUISE	324	341										
FREQ	119	631										
FREQC	631	632	634	634								
G0	529	560										
GA	42	203	378	550	550	654	657	657				
GAMMA	83	220	223	343	345	361	394	422	550	633	642	654
	657	657	659	661	667	667	680	683	887			
GAMMA0	642	680										
GAMMA4MAX	89	343	825									
GAMMA5	90	361	826									
GAMMAMAX	86	822										
GAMMAOLD	683											
GC	138	142	203	372	382	502	516	520	522	527	528	550
	654	654	657	657	657							
GLMB	44	155	515	519	520							
H	506	509	516	519								
H0	83	142	177	203	220	223	237	241	242	312	328	382
	394	421	506	506	528	539	567	633	643	668	668	681
	683	869	886									
H00	643	681										
H02	91	95	237	241	242	827	831					
H0OLD	312	683										
HB	44	155	509	516	519							
HOEL	509	510										
I	151	190	427	702	702	702	703	703	703	708	708	708
	709	709	709	710	711	711	726	727	727	727	731	732
	733	734	747	748	748	750	751	752	752	755	755	756
	756	758	766	767	768	769	772	772	773	773	774	775
	788	788	789	789	913	914	916	930	931	933	946	947
	949	962	963	965								
I1	508	509	511	515	516	516	516	517	519	519	519	520
	520	520	748	752	754	757	757	769	771	775	790	790
I1ST	41	695	751	758	768	774						
I2	748	748	752	752	753	757	769	769	770	775	775	790
IFILE	52	802										
IFN\$	47	49	52	853	893							
IFN2\$	50	802	854	898								
IL	52	694	694	694	695	695	696	696	697	802		
ILTI	694	696	704	712								
IO	41	696	696	721	743							
ISP	376	396	421	576	595	616	633	886				
ISPA	400	556	564	598	602	604	614	616	618			
ISPAVG	377	399	422	633	872	887						
ISPEFF	372	373	397	421	633	886						
ISPMEAN	374	398	421	633	873	886						
ISPR	401	567	570	591	619							
ITAB	70	70	176	177	178	179	180	327	328	329	330	538
	539	540	541	553	554	556	567	568	602	721	721	743
	751	758	768	774	780	786	836	836				
ITCUM	373	373	374									
ITO	696	697	698	700	706	706	707	708	710	711	712	721
	722	722	723	724	729	729	729	731	731	732	743	743
	745	763	763	763								
ITSUM	376	376	377									
IX	700	700	701	701	702	702	703	704	706	706	707	707
	707	708	709	710	712	724	724	724	726	727	729	729
	729	733	745	745	745	746	748	749	752	763	763	763
	764	765	769	775								
IY	701	703	704	707	709	724	727	729	733	745	748	752
	763	776	777	781	787							
IZ	707	711	712	729	732	763	788	788	789	789		
J	710	711	711	731	732	733	734	778	779	780	781	784
	784	785	785	786	787	788	788	789	789			
J1	153	155	155	155	155	156	748	752	757	757	781	783
	787	791	791									
J1ST	41	695	780	786								
J2	748	748	752	752	757	781	781	782	787	787	791	
K	179	186	330	332	541	544	711	711	732	733		
K11	788	792	792									
K12	789	793	793									
K21	788	792										
K22	789	793										



LIFT	185	220	223	344	396	398	398	422	543	633	657	887
LOADFAC	86	238	822									
LOVRD	333	333	334	335								
LOVRDOLD	334	335										
M	763	763	777	779								
M0	83	176	178	179	180	327	329	330	393	526	538	540
	541	553	554	556	568	602	679					
MEARTH	139	203										
N	647	651	654	657	659	661	665	666	667	668	669	670
	745	745	749	750	763	763	765	767	788	788	789	789
	848	919	920	936	937	952	953	968	969			
N1	501	508										
NSTAGES	45	46	443	444	897							
P0	523	529										
PFNS	100	101	856	909	912	929	945	961				
PHASE	216	230	252	321	354	391	391	391	391	391		
PHI	395	421	555	556	559	561	599	602	607	613	633	886
PHIMAX	554	555	598	599								
PLOT\$	98	99	99	446	446	451	451	452	453	453	455	455
	629	629										
PRCOM	73	73	74	407	434	437	440	446				
PRCOM\$	72	73	73									
PRCOUNT	150	409	410	412	418	418	425	425				
Q\$	194	194	197	198								
Q0	185	187	237	238	264	265	267	270	271	271	272	275
	277	280	281	281	282	284	291	303	305	310	395	422
	527	543	545	633	683	870	887					
Q0COM	259	262	264	270	271	272	280	281	282	291	302	303
Q0CMD	84	237	259	262	262	820						
Q0FINAL	88	262	824									
Q0MAX	84	302	305	820								
Q0MIN	84	820										
Q0OLD	310	683										
QOLD	265	267	271	275	277	281	284					
R0	516	520	522									
RANGE	395	421	633	644	669	669	682	886				
RANGE0	644	682										
RATEDELQ	270	273	280	283								
RATEDQ	267	268	271	277	278	281						
RE	503	506	506									
REARTH	140	142	203	382	382	528						
RES1	393	422	633	887								
RH0	311	522	523	527	683							
RH0B	44	155	516	520								
RH0OLD	311	683										
STAGE	82	385	444	818								
START	120	459										
STARTTIME\$	120	458										
SWITCH4	89	323	349	825								
T\$	41	697	721									
T0	517	519	520	523	525	529						
TG	42											
THRUST	400	400	402	421	422	572	576	578	594	595	597	597
	604	605	609	614	633	633	654	657	886	887		
THRUSTAMAX	564	571	590	594								
THRUSTCOM	401	421	550	578	586	590	605	609	633	886		
THRUSTMAX	402	421	571	572	586	633	886					
THRUSTR	590	591	594									
THRUSTRMAX	570	571	587									
TIMEX	83	148	218	235	392	421	422	433	434	436	437	439
	440	445	633	639	653	664	664	676	837	867	886	887
TIMEX0	639	676										
TIMEX2	445	837										
TITL1\$	55	412	804									
TITL2\$	56	412	805									
TMB	44	155	516	517	519	520						
TQ0	235	243	243									
TSET5	261	359	360	361	361							
TTAKEOFF	218	226	226	235								
V	907	914	918	918	918	918	918	918	918	918	918	918
	918	931	935	935	935	935	935	935	935	935	935	947
	951	951	951	951	951	951	951	951	951	951	963	967

	967	967	967	967	967	967	967	967	967	967	967		
V0	142	206	207	218	237	241	260	262	313	314	372	378	
	383	385	393	421	433	434	436	437	439	440	526	527	
	528	529	633	641	656	657	659	661	666	666	678	679	
	864	868	886										
V00	372	378	641	678									
V02	91	237	241	827									
V05	90	314	826										
V0CRUISE	89	313	313	825									
V0TEMP	88	260	262	262	824								
VAL1	792	794	794										
VAL2	793	794											
VALI	176	177	178	179	180	327	328	329	330	538	539	540	
	541	553	554	556	567	568	602	757	794				
VARNUM	102	103	912	929	945	961							
VFINAL	82	385	818										
VORBIT	382	383	864										
VTAKEOFF	85	218											
WDOT	575	576	593	595	615	651							
WDOTA	394	421	560	563	607	613	633	886					
WDOTF	393	421	573	592	593	604	605	605	607	613	614	615	
	618	633	886										
WDOTFMAX	563	564	573	575	592								
WDOTP	394	422	574	587	588	591	593	619	633	887			
WDOTPMAX	568	570	574	575	584								
WEIGHT	142	148	205	220	223	238	344	372	372	373	374	376	
	377	384	396	396	402	422	433	434	436	437	439	440	
	528	550	633	640	653	654	654	657	657	665	665	677	
	818	861	862	865	887								
WEIGHT0	372	373	376	640	677								
WFINAL	82	205	205	384	818	862	865						
WFUEL	82	818											
WLAUNCH	82	148	374	377	653	860							
WTIN\$	68	69	69	834	835	835							
X\$	391	391	391	391	391	392							
XARG	176	177	178	179	180	327	328	329	330	538	539	540	
	541	553	554	556	567	568	602	746	749	753	754	757	
	764	765	770	771	790								
YARG	180	553	556	602	776	777	782	783	791				

## 6.0 DEFINITION OF PROGRAM LTO VARIABLES

The program variables in this section are listed alphabetically by the following groupings:

- Simple Real Variables
- Simple String Variables
- Array Real Variables
- Array String Variables

VARIABLE LISTING FOR PROGRAM "ETO"

Simple Real Variables

A0	Air capture streamtube,sq.ft.	Internal
ABAC	Air capture/cowl area ratio.Values are input as table 6 of input data file.	Input
A1	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 0.1	Input
A2	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 0.1	Input
AC	Air inlet cowl area,sq.ft.	Input
ACC	Axial acceleration,g's.Output is labeled ACCEL-G'S.	Output
ACCCOM	Commanded axial acceleration,g's	Output
ACCCOMD	Maximum axial acceleration,g's	Input
ACCCOMD5	Commanded acceleration in Phase 5,g's	Input
ACCPERQ	Variable used to adjust acceleration in Phase 3,constant q path.	Internal
ALPHA	Angle of attack,degrees	Output
ALPHA2MAX	Maximum angle of attack in Phase 2,degrees	Input
ALPHA5MAX	Maximum angle of attack in Phase 5,degree	Input
ALPHACRUISE	Angle of attack for maximum L/D cruise in Phase 4	Internal
ALPHAMAX	Maximum angle of attack in Phase 1 thru 4,degrees.	Input
ALPHAMIN	Minimum angle of attack,degrees	Input
ANG	57.3 degrees/radian	Internal
AREF	Vehicle lift and drag reference area,sq.ft.	Input
B1	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 0.009	Input
BIGG	Universal gravitational constant	Internal
C0	Velocity of sound,ft./sec.	Internal
C1	Adjustment "knob" for controlling Phase 3 constant q path. Presently set to 0.1	Input
C2	Adjustment "knob" for controlling Phase 3 constant q path.Presently set to 2.0	Input
CD	Vehicle drag coefficient	Internal
CD0	Zero lift drag coefficient.Values are input as table 3 of input data file.	Input
CL	Lift coefficient	Internal
CLALPHA	Lift coefficient/degree.Values are input as table 5 of input data file.	Input
COUNT	Loop counter for controlling screen and printer interval.	Internal
CPCV	Air specific heat ratio.Not used	Internal
D1	Adjustment "knob" for controlling Phase 3	Input

constant q path. Presently set to 0.1

DELCD	Friction increment of drag. Values are input as table 4 of input data file.	Input
DELPRINT	Print and screen output interval	Input
DELQ	Adjustment "knob" for controlling Phase 3 constant q path. A value equal to 5% of Q0COMD is suggested.	Input
DELTHRUST	Iteration tolerance on commanded thrust	Internal
DRAG	Vehicle drag, lbf.	Output
DT	Trajectory integration interval, sec.	Input
DV	Reduced integration interval used near end of flight, secs.	Internal
DXI	Variable in table interpolation routine.	Internal
DYJ	Variable in table interpolation routine.	Internal
FAMAX	Maximum fuel/air ratio	Internal
FASTOIC	Stoichiometric fuel/air ratio	Input
FC	Vehicle centrifugal force, lbf.	Internal
FINISH	Clock time at end of problem, hr/min/sec	Output
FLAGS	Selector for Phase 5, ballistic ascent. Enter 0 if not used, 1 if ballistic phase is used.	Input
FLAGCRUISE	If vehicle is to cruise at maximum L/D, this variable is reset from zero to a value of 1 to bypass optimum alpha search once it is made.	Internal
FREQ	Frequency of output to plotter file	Input at prompt
FREQC	Plotter file counter variable	Internal
G0	Freestream specific airflow, lbf/(sec-sq in)	Internal
GA	Gravitational acceleration, ft/sec-sec	Internal
GAMMA	Vehicle flight path angle, degrees	Internal
GAMMA0	"Old" flight path angle at start of integration interval, degrees	Internal
GAMMA4MAX	Maximum flight path angle permitted during Phase 4 maximum L/D cruise, degrees. A small value, less than 0.5 degrees is suggested.	Input
GAMMA5	Flight path angle to be achieved before ballistic path begins, degrees	Input
GAMMAMAX	Maximum allowed flight path angle for all phases except 4 and 5, degrees.	Input
GAMMAOLD	Same as GAMMA0, degrees	Internal
GC	32.174 lbf ft per sec sq/lbf	Internal
H	Geopotential altitude, ft.	Internal
H0	Vehicle flight altitude, ft. An initial value is input. Values during run are output.	Input and Output
H00	"Old" value of flight altitude at start of integration interval, ft.	Internal

H02	Phase 2 altitude to be achieved before V02 can be exceeded,ft.	Input
H00LD	Same as H00	Internal
HDEL	Altitude variable used in atmosphere routine,ft.	Internal
I	Integration loop counter and index variable	Internal
I1	Counter and index variable	Internal
I2	Counter and index variable	Internal
IFILE	Input file designator	Internal
IL	Input file designator	Internal
ILTI	Index variable in table loader	Internal
ISP	Specific impulse,secs.Sum of airbreathing and rocket thrust divided by sum of airbreathing fuel and rocket propellant flows.	Output
ISPA	Specific impulse of airbreather,sec. Values are input as table 1 of input data file.	Input and Output
ISPAVG	Running average specific impulse since start of flight,secs.	Output
ISPEFF	Instantaneous effective specific impulse defined by $(\text{weight/gc}) \times (\text{delta velocity/delta weight})$ .	Output
ISPMEAN	Running mean value of ISPEFF since start of flight.	Output
ISPR	Rocket specific impulse,secs.Values are input as table 7 of input data file.	Input and Output
ITAB	Input table number.Eight tables are currently used in input data file.	Input
ITCUM	Cumulative total effective impulse since start of flight,lb.secs.	Internal
ITO	Input data table variable.	Internal
ITSUM	Cumulative total propulsive impulse since start of flight,lb.secs.	Internal
IX	Variable used in table loader and lookup.	Internal
IY	Variable used in table loader and lookup.	Internal
IZ	Variable used in table loader and lookup.	Internal
J	Variable used in table loader and lookup.	Internal
J1	Variable used in table loader and lookup.	Internal
J2	Variable used in table loader and lookup.	Internal
K	Ratio of specific heats.Also a variable used in table loader and lookup.	Internal
K11	Variable used in table interpolation.	Internal
K12	Variable used in table interpolation.	Internal
K21	Variable used in table interpolation.	Internal
K22	Variable used in table interpolation	Internal
LIFT	Vehicle lift, lbf.	Output

LOADFAC	Maximum vertical acceleration allowed during Phase 2 climb,g's.	Input
LOVRD	L/D value during Phase 4 cruise alpha search	Internal
LOVRDOLD	"Old" L/D value used in Phase 4 cruise alpha search	Internal
M	Variable used in table interpolation	Internal
M0	Flight Mach number.Input a small starting value,say .001.	Input and Output
M00	"Old" value of Mach number	Internal
MEARTH	Mass of the earth	Internal
N	Integration counter for equations of motion	Internal
N1	Variable in atmosphere routine	Internal
NSTAGES	Number of vehicle stages,1 or 2.	Input at prompt
P0	Atmospheric static pressure,psia.	Internal
PHASE	Flight phase designator.Values are 1 thru 5.	Internal
PHI	Airbreathing fuel equivalence ratio.	Output
PHIMAX	Maximum airbreathing fuel equivalence ratio.Values are input as table 2 of input data file.	Input
PRCOM	Command for printer output during run. It is set to 1 by responding "Y" at prompt for printer output decision.	Internal
PRCOUNT	Counter for designating number of output points to be printed per page.	Internal
Q0	Flight dynamic pressure,lb/sq.ft.	Output
Q0COM	Calculated dynamic pressure command	Internal
Q0COMD	Commanded dynamic pressure at start of Phase 3,lb/sq.ft.	Input
Q0FINAL	Desired final dynamic pressure at orbital or final condition,lb/sq.ft.	Input
Q0MAX	Maximum dynamic pressure allowed during Phase 3,lb/sq.ft.	Input
Q0MIN	Minimum dynamic pressure allowed during Phase 3,lb/sq.ft.	Input
Q0OLD	"Old" value of dynamic pressure,lb/sq.ft.	Internal
Q0LD1	Same as Q0OLD.	Internal
R0	Density variable in atmosphere routine.	Internal
RANGE	Flight range,nautical miles.	Output
RANGE0	"Old" value of range,nm.	Internal
RATEDELQ	Commanded rate of change of dynamic pressure in Phase 3,psf/sec.	Internal
RATEQ	Current rate of change of dynamic pressure,psf/sec.	Internal
RE	Earth radius	Internal
REARTH	Same as RE	Internal

RES1	Not used.	
RH0	Atmospheric density, lbm./cu.ft.	Internal
RH0B	Density variable used in atmosphere routine	Internal
RH0OLD	"Old" value of density, lbm./cu.ft.	Internal
STAGE	Identifier in data table for first stage or second stage. Enter 1 or 2 in input data file.	Input
START	Clock time at start of problem, hr/min/sec	Output
SWITCH4	Used for Phase 4 cruise. Zero selects constant altitude cruise. 1 selects constant L/D cruise.	Input
T0	Static temperature of atmosphere, degrees-R.	Internal
THRUST	Vehicle thrust, lbf.	Output
THRUSTAMAX	Maximum available airbreathing thrust, lbf.	Internal
THRUSTCOM	Commanded thrust, lbf.	Output
THRUSTMAX	Maximum total thrust available, lbf.	Output
THRUSTR	Rocket thrust, lbf.	Internal
THRUSTRMAX	Maximum rocket thrust available, lbf.	Internal
TIMEX	Flight time since takeoff, secs. Input small initial value, say .001.	Output and Input
TIMEX0	"Old" value of time, secs.	Internal
TIMEX2	Flight time at start of 2nd stage, secs.	Internal
TQ0	Time variable for testing if Phase 2 is completed.	Internal
TSET5	Time variable in Phase 5 to indicate that commanded flight path angle has been reached. When value becomes 1, ballistic ascent can begin.	Internal
TTAKEOFF	Time variable used to test if takeoff is completed.	Internal
V	Identifier for printer output variables	Internal
V0	Vehicle flight velocity, ft./sec.	Output
V00	"Old" value of flight velocity, ft./sec.	Internal
V02	Maximum velocity permitted in Phase 2 until H02 altitude is reached, ft./sec.	Input
V05	Velocity at which Phase 5 pullup to ballistic ascent is to begin, ft./sec.	Input
V0CRUISE	Commanded Phase 4 cruise velocity, ft./sec.	Input
V0TEMP	Velocity at which commanded dynamic pressure is made a function of velocity, ft./sec. This variable is used with Q0FINAL to approximate a temperature limited ascent path.	Input
VAL1	Variable in table interpolation.	Internal
VAL2	Variable in table interpolation.	Internal
VALI	variable returned from table interpolation	Internal
VARNUM	Number of output variables written to plotter file.	Internal



VFINAL	Commanded final flight velocity,ft./sec. For a single stage going to orbit,enter a value beyond orbital velocity,say 27000. For a single stage suborbital vehicle, enter the desired final velocity. For a 2-stage vehicle,enter the desired staging velocity in the 1st stage data table and enter 27000 or a desired sub- orbital velocity in the 2nd stage data table.	Input
VORBIT	Orbital velocity,ft./sec.	Output
VTAKEOFF	Commanded takeoff velocity,ft./sec.	Input
WDOT	Total airbreathing fuel and rocket prop- ellant flow rate,lbm./sec.	Internal
WDOTA	Propulsive air flow,lbm./sec.	Output
WDOTF	Airbreather fuel flow,lbm./sec.	Output
WDOTFMAX	Maximum airbreather fuel flow,lbm./sec.	Internal
WDOTP	Rocket propellant flow rate,lbm./sec.	Output
WDOTPMAX	Maximum rocket propellant flow rate,lbm./sec. Values are input as table 8 of input data file.	Input
WEIGHT	Vehicle weight,lbm.	Output
WEIGHT0	"Old" value of vehicle weight,lbm.	Internal
WFINAL	Vehicle final weight,lbm.	Input
WFUEL	Vehicle fuel+propellant weight,lbm. Value not required,enter zero if desired.	Input
WLAUNCH	Starting weight of vehicle,lbm.	Input
XARG	Table lookup variable	Internal
YARG	Table lookup variable	Internal

#### Simple String Variables

A\$	Phase 1,takeoff roll	Output
B\$	Phase 2,climb at constant L/W	Output
C\$	Phase 3,climb at commanded Q	Output
CMD1\$	Command string to print 1st stage input file to printer.	Internal
CMD2\$	Command string to print 2nd stage input file to printer.	Internal
D\$	Phase 4,cruise at $V_0$ cruise	Output
E\$	Phase 5,pullup and ballistic ascent	Output
IFN\$	Input data file name (single stage or two stage) read from screen at prompt.	Input at prompt
IFN2\$	Input data file name for 2nd stage.Name is read from screen at prompt.	Input at prompt
PFN\$	Plotter file name entered at prompt	Input at prompt
PLOTS\$	Set to "Y" or "N" in response to prompt for decision to create plotter file.	Input at prompt
PRCOM\$	Set to "Y" or "N" in response to prompt	Input at

	for decision for printer output during run.	prompt
Q\$	Used to halt/resume program	Internal
STARTTIME\$	Clock time at start of program,hr/min/sec	Output
TITL1\$	Text of first line of input data file.	Input
TITL2\$	Text of second line of input data file.	Input
WTIN\$	Set to "Y" or "N" in response to prompt for listing of tabular data in input file.	Input at prompt
X\$	Variable which is set to A\$ thru E\$ as each flight Phase(1 thru 5) is reached.	Internal

Array Real Variables

AT	Sets number of points in tables.	Internal
DELGAM	Change in flight path in one integration pass, degrees.	Internal
DELH	Change in altitude in one integration pass,feet.	Internal
DELR	Change in range in one integration pass, feet.	Internal
DELV	Change in velocity in one integration pass,ft./sec.	Internal
DELW	Change in weight in one integration pass, lbm.	Internal
GA	Not used as array variable.	Internal
GLMB	Variable used in atmosphere routine.	Internal
HB	Variable used in atmosphere routine.	Internal
IIST	Variable used in table loader.	Internal
IO	Variable used in table loader.	Internal
JIST	Variable used in table loader.	Internal
RH0B	Air density variable used in atmosphere routine.	Internal
TG	Not used.	Internal
TMB	Temperature variable used in atmosphere routine.	Internal

Array String Variables

T\$	Table title used in table loader.	Internal
-----	-----------------------------------	----------