Dynetics, Inc.
P. O. DRAWER B
HUNTSVILLE, ALABAMA 35805

TR-81-MSFC-33803-012

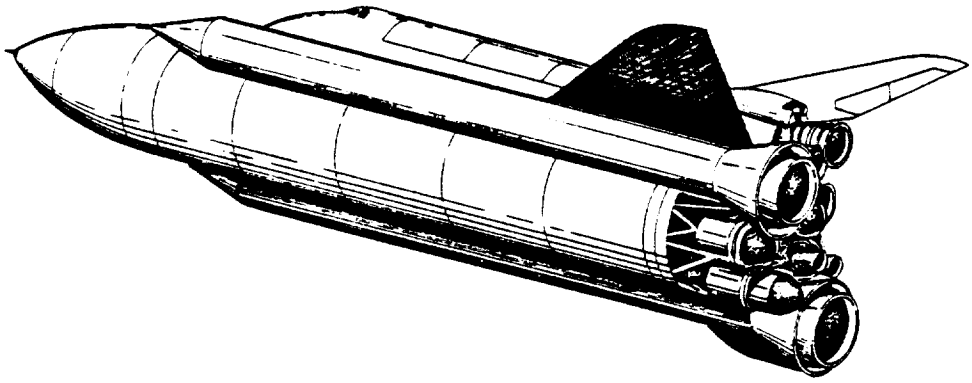
FINAL REPORT

MINIMUM HAMILTONIAN ASCENT SHUTTLE TRAJECTORY EVALUATION PROGRAM (MASTRE) (LBM MODIFICATION)

CONTRACT NAS8-33803

AUG 1981
RECEIVED
NASA STI FACILITY
ACCESS DEPT.

MAY 1981



PREPARED FOR:

GEORGE C. MARSHALL SPACE FLIGHT CENTER
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MARSHALL SPACE FLIGHT CENTER, AL

(NASA-CR-161628) MINIMUM HAMILTONIAN ASCENT SHUTTLE TRAJECTORY EVALUATION PROGRAM (MASTRE): LBM MODIFICATION Final Report (Dynetics, Inc., Huntsville, Ala.) 394 p N81-75604
Unclas
00/16 27333

FOREWARD

This report presents the results of work performed by Dynetics, Inc. while under contract to the System Analysis and Integration Laboratory of the Marshall Space Flight Center (NAS8-33803) to perform a research study entitled "Modification of 'MASTRE' Shuttle Ascent Trajectory Computer Program to Include the Liquid Boost Module". Technical coordination was provided by Mr. L. N. Lott of the System Performance Branch, Mission Analysis Division (EL-24).

TABLE OF CONTENTS

		Page
1.	INTRODUCTION	1-1
2.	PHYSICAL MODELS	2-1
	2.1 Environmental Definition	2-1
	2.2 Coordinate System	2-6
	2.3 Generalized Equations of Motion	2-9
	2.4 Body Related Forces	2-11
	2.4.1 Aerodynamic Forces	2-11
	2.4.2 Aerodynamic Moments	2-15
	2.4.3 Thrust Forces	2-18
	2.5 Mass Calculation	2-24
3.	THE FORWARD TRAJECTORY	3-1
	3.1 The Equations of Motion	3-1
	3.2 Trajectory Sequence Requirements	3-3
	3.2.1 Trajectory Initiation Equations	3-4
	3.2.2 Vertical Rise	3-5
	3.2.3 Roll to Flight Azimuth Command	3-6
	3.2.4 Booster Polynominal Attitude Control	3-7
	3.2.5 Optional Angle-of-Attack Attitude History for the Booster Phase	3-8
	3.2.6 Prime Contractor Attitude Angles θ_s, ψ, and σ	3-9
	3.2.7 Attitude Control During Min-H Phase	3-10
	3.3 Constraint Library	3-11
	3.4 Control Parameters	3-11

TABLE OF CONTENTS (Continued)

		Page
4.	THE BACKWARD TRAJECTORY	4-1
5.	INFLUENCE COEFFICIENTS	5-1
6.	OPTIMIZATION METHODOLOGY	6-1
	6.1 Steepest Descent Formulation	6-1
	6.2 Automatic Convergence Scheme	6-2
7.	REFERENCES	7-1
8.	USERS' MANUAL	8-1
9.	IMPLEMENTATION OF TRAJECTORY PROBLEMS ON THE MASTRE PROGRAM	9-1
	9.1 Thrust Events	9-1
	9.2 Attitude Control Options	9-2
	9.2.1 Angle of Attack/Sideslip Angle Control	9-4
	9.2.2 Attitude Polynominal Control	9-5
	9.2.3 Attitude Control Based Upon the Angles Theta (θ), Psi (ψ), and Sigma (σ)	9-6
	9.2.4 Attitude Control Using the Min-H Optimization Technique	9-7
	9.2.5 Post Launch Roll Maneuver	9-8
	9.3 Input Requirements for the Equations of Motion	9-9
	9.3.1 Aerodynamic Model Input	9-9
	9.3.2 Thrust Model Input	9-9
	9.3.3 Gravitational Model Input	9-12
	9.4 Integration Schemes	9-12
	9.5 Optimization Methodology	9-12

TABLE OF CONTENTS (Continued)

Page

9.5.1 Selection of the Payoff Function and the Terminal and Intermediate Constraints . . . 9-13

9.5.2 Selection of the Control Parameters 9-13

9.6 Selection Options 9-15

9.6.1 Jump-Start Option 9-16

9.6.2 Atmosphere Options 9-17

9.6.3 External Tank (ET) Propellant Jettison Weight Option 9-17

10. ITEMIZED DESCRIPTION OF INPUT VARIABLES 10-1

11. DETAILED OUTPUT DESCRIPTION 11-1

11.1 Initial Output 11-1

11.2 Trajectory Block Output 11-1

11.3 Iteration Output 11-9

11.4 Post-Processor Output 11-18

11.4.1 Output Table Post-Processor 11-18

11.4.2 Plotting Post-Processor (Demand Terminal Only) 11-22

12. EFFICIENT USE OF MASTRE 12-1

12.1 Body Axis Coordinate System 12-1

12.2 Attitude Control 12-2

12.3 Jump-Start 12-3

13. PROGRAMMER'S MANUAL 13-1

14. GENERAL PROGRAM FLOW 14-1

14.1 Initiation 14-1

TABLE OF CONTENTS (Concluded)

		Page
	14.2 Demand Terminal Communication	14-1
	14.3 Forward Trajectory Integration and Evaluation . .	14-1
	14.4 Backward Trajectory Integration	14-3
	14.5 Partial Derivative Calculation	14-3
	14.6 Parameter Update and Convergence Test	14-3
	14.7 Output Tables	14-4
15.	SUBROUTINE DESCRIPTION	15-1
	15.1 Non-System Subroutine Names	15-1
	15.2 Non-System Elements	15-6
	15.3 System Subroutine Names	15-6
16.	PROGRAM LISTING	16-1
	APPENDIX A - INTERPOLATION METHODS	A-1
	APPENDIX B - SOLUTION OF MOMENT BALANCE EQUATIONS . . .	B-1
	APPENDIX C - VELOCITY LOSS EQUATIONS	C-1
	APPENDIX D - PARTIAL DERIVATIVES OF THE PAYOFF AND CONSTRAINTS WITH RESPECT TO THE FINAL STATE	D-1
	APPENDIX E - FIRST PARTIAL DERIVATIVES OF GRAVITATIONAL ACCELERATION WITH RESPECT TO PLUMBLINE POSITION COORDINATES	E-1
	APPENDIX F - PARTIAL DERIVATIVES OF THE EQUATIONS OF MOTION WITH RESPECT TO THE STATE AND THE CONTROL	F-1
	APPENDIX G - PROGRAM LISTING	G-1

LIST OF ILLUSTRATIONS

Figure	Title	Page
2-1	The Earth - An Ellipsoid	2-2
2-2	Angle-of-Attack and Sideslip Angle Orientation	2-5
2-3	Reference Plumbline Coordinate System	2-6
2-4	Lift-Off Configuration	2-8
2-5	Attitude Angles	2-8
2-6	Aerodynamic Moment Orientation	2-16
2-7	Engine Deflection Angles for Fixed Orientation	2-21
2-8	Center of Gravity/Gimbal Plane Coordinate System	2-22
3-1	Prime Contractor Attitude Angle Orientation	3-10
3-2	Throttle Optimization Parameters	3-13
9-1	Typical Thrust Event Profilt	9-2
11-1	Trajectory Block Output	11-2
11-2	Trajectory Constraint Summary Table	11-9
11-3	Polynominal Coefficient Printout	11-10
11-4	COV Lambdas	11-10
11-5	Influence Coefficients Table	11-10
11-6	I-PSI-PSI Output	11-11
11-7	PSUBI, WIBT Output	11-11
11-8	Total I-SY-SY Matrix Output	11-11
11-9	WGNU Output	11-12
11-10	Control Step Variables Printout	11-12
11-11	DRHØ, WDS, GNU Optimization Parameters	11-12

LIST OF ILLUSTRATIONS (Concluded)

Figure	Title	Page
11-12	Parameter Update and Convergence Test Block	11-13
11-13	Control Variable Update Printout	11-14
11-14	Convergence Indicator Printout	11-15
11-15	Parameter Summary Table	11-15
11-16	Orbital Element Summary Table	11-16
11-17	Load Indicator Table	11-16
11-18	Weight Summary Table	11-17
11-19	NNAME Array	11-19
11-20	Variable Location Cross Reference	11-20
11-21	Table Module Output Example	11-20
11-22	Summary Output Table Example	11-21
11-23	Output Variable Dictionary	11-22
11-24	Examples of Plotting Post-Processor Output	11-23
12-1	Input Body Axis System	12-1
14-1	'MASTRE' Flow Diagram	14-2
15-1	Subroutine Cross Index	15-2
15-2	Overlay Structure	15-3

LIST OF TABLES

Table	Title	Page
2-1	Earth Model Constants	2-1
3-1	Payoff and Constraint Library	3-12
3-2	Parameter Library	3-13
9-1	Data Grouping	9-3
9-2	Constraint Codes	9-14
9-3	Optimization Codes	9-15
10-1	Input Variable Index	10-2
10-2	MASTRE Input Variables	10-3
10-3	MASTRE Input Variables	10-27
10-4	MASTRE Input Variables	10-31
10-5	MASTRE Input Variables	10-32
11-1	LPRINT Options	11-9

Section 1

INTRODUCTION

This final report documents all findings on Contract NAS8-33803 with MSFC entitled "Modification of 'MASTRE' Shuttle Ascent Trajectory Computer Program to Include the Liquid Boost Module".

Presented in this document is an efficient method for determining optimum Space Shuttle trajectories. Optimization is obtained by a steepest ascent algorithm using a minimum Hamiltonian (min-H) strategy in which the derivation has been expanded to encompass the effects of asymmetrical aerodynamics, moment balance control, and atmospheric effects. The algorithm also includes a means for optimizing control parameters, which allows great flexibility to the solution of many types of trajectory problems. This computer program is referred to as the Minimum Hamiltonian Ascent Shuttle Trajectory Evaluation (MASTRE) program.

The MASTRE program is an outgrowth of three previous NASA, Marshall Space Flight Center, ascent trajectory programs: RAGMOP (Reference 2), ROBOT (Reference 3), and Lifting ROBOT (Reference 4). The value of the minimum Hamiltonian strategy to determine optimum trajectories in a computationally fast and stable approach was realized when using the ROBOT program for vacuum, point mass simulations.

The development of the MASTRE program has been an evolutionary process which began with the ROBOT program developed by Dr. R. G. Gottlieb. This program was used for optimization of the upper stages of the Saturn rocket series. Due to the symmetric features of the Saturn, a near optimum attitude profile could be attained for the booster by using a gravity turn (or zero angle of attack) profile. The upper stage was optimized by nonatmospheric algorithms since its trajectory was significantly out of the atmosphere at both booster cutoff and orbit injection. However, the requirements of the Space Shuttle forced the implementation of atmospheric

optimization as well as the additional constraint of thrust vector control moment balance. The Lifting ROBOT program, also developed by Dr. Gottlieb, and the RAGMOP program, developed by a team assisted by Dr. Gottlieb and the author, were the first attempts at optimization within the atmosphere and the enforcement of Space Shuttle related constraints. The current MASTRE program combines the features of these latter programs and provides the additional adjoint equations required by the complex equations of motion of the Space Shuttle.

The Liquid Boost Module (LBM) modification to the program has been another evolutionary step in the program development. The LBM modification gives the program the capability of simulating additional propulsion systems which are in the planning stages. While the LBM modification has been of prime importance in this effort, other modifications have been made to the program to update the program capabilities to current trajectory shaping philosophy.

The formulation of the equations of motion in the MASTRE program includes the effects of the earth's "pear-shaped" gravitational model, the atmospheric perturbations (aerodynamic forces and moments and base axial force), and the calculation of moment balance thrust alignment. The additional effects of the atmospheric perturbations and moment balance in the adjoint equations of motion used in the steepest ascent algorithm considerably modify the original ROBOT formulation as programmed by Dr. R. G. Gottlieb in Reference 3.

The MASTRE program can simulate the trajectory of a multistage vehicle having up to fifteen thrust events, and the program can be used for both ground launched and orbital trajectories. The internal logic is based upon the flight profile of the Space Shuttle, and the constants and coordinate systems conform to the Space Shuttle standards. However, the program is general in nature and is not restricted to the Space Shuttle.

This report is divided into three parts. The first part of the report (Sections 2 through 6) presents a discussion of the equations of motion and a limited discussion of the optimization technique (see Reference 9 for a detailed explanation). The second part, referred to as the User's Manual (Sections 8 through 12) summarizes the input and output requirements. The third part, referred to as the Programmer's Manual (Sections 13 through 16) provides information required by a programmer for future modification. Appendices are provided for detailed presentation of the equations and to provide a program listing.

Documentation continuity has been maintained by coupling the previous documentation (Reference 1) with the current modifications.

Sample cases have been processed and results provided as required by the contract.

J. F. Lyons

J. F. Lyons

APPROVED:

T. A. Baumbach

T. A. Baumbach
Manager
Systems Analysis

Section 2

PHYSICAL MODELS

This section describes the assumptions and equations associated with determining the physical models of the environment and the launch vehicle.

2.1 ENVIRONMENTAL DEFINITION

The environmental model in the MASTRE program is based upon a rotating Fischer earth model (1969). Constants for the earth model are based on Reference 5 and are defined in Table 2-1. While the gravitational model assumes all four harmonics of the model, which results in a "pear-shaped" earth model, the calculation of earth-related parameters is based on a symmetrical ellipsoid.

Table 2-1. Earth Model Constants

PARAMETER	SYMBOL	VALUE
Earth gravitational constant	μ_e	3.986012E+14 m^3/sec^2
Earth rotation rate	ω_e	7.29211595E-05 rad/sec
Second harmonic of gravity	J	1.62405E-03
Third harmonic of gravity	H	-6.4E-06
Fourth harmonic of gravity	D	6.9125E-06
Earth radius at equator	R_E	6378160 meters
Gravity at sea level	g_o	9.80665 m/sec^2

The gravitational potential function used for the Fischer model is given by

$$U(R, \theta) = \frac{\mu_e}{R} \left[1 + \frac{J}{3} \left(\frac{R_e}{R}\right)^2 \left(1 - 3 \cos^2 \theta\right) + \frac{H}{5} \left(\frac{R_e}{R}\right)^3 \left(3 - 5 \cos^2 \theta\right) \cos \theta + \frac{D}{35} \left(\frac{R_e}{R}\right)^4 \left(3 - 30 \cos^2 \theta + 35 \cos^4 \theta\right) \right]$$

where R is the instantaneous radius.

The angle θ is defined as the geocentric colatitude of the instantaneous radius vector as shown in Figure 2-1.

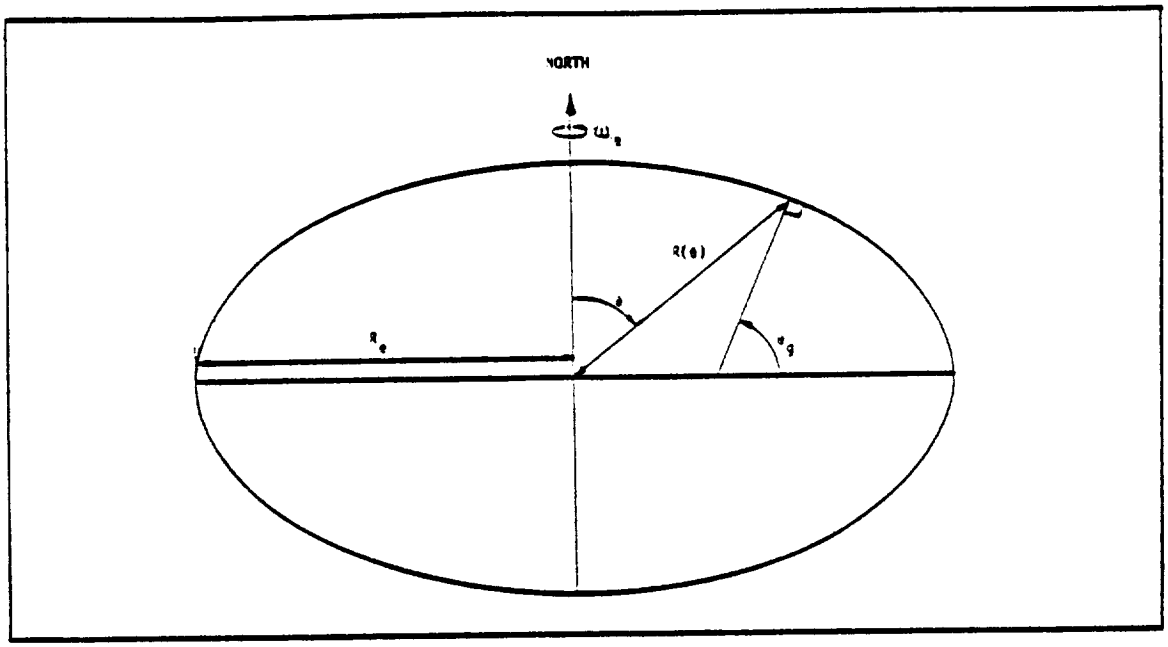


Figure 2-1. The Earth - An Ellipsoid

This angle is also used to define the instantaneous value of the earth radius on the ellipsoid as defined by the equation

$$R(\theta) = \frac{(1 - f)\, R_e}{\sqrt{(1 - f)^2 \sin^2 \theta + \cos^2 \theta}}$$

The instantaneous altitude (h) above the ellipsoid is approximated by the difference between the instantaneous radius and radius of the earth; thus

$$h = R - R(\theta).$$

The atmospheric parameters of density (ρ), speed of sound (V_{sos}), pressure (P), and coefficient of viscosity (μ) are determined as functions of the instantaneous altitude based upon one of six available atmospheric

models: the nominal, hot, and cold 1963 Patrick AFB Reference Atmospheres (PRA63, PHA63, and PCA63) (References 6 and 10) and the nominal, hot, and cold 1971 Vandenberg AFB Reference Atmospheres (VRA71, VHA71, and VCA71) (References 7 and 11). The calculation of these parameters is a function of the third order spline interpolation of the density, speed of sound, and pressure.

The primary reason for using the spline interpolation, which is used throughout the program, is that the method not only interpolates the function but also gives the partial derivative of the dependent variable with respect to the independent variable. Therefore, the spline interpolation of the density, speed of sound, and pressure also results in the partial derivatives of these parameters with respect to altitude. The coefficient of viscosity (used in calculation of Reynolds Number), temperature, and the partial derivatives of the coefficient of viscosity and temperature with respect to altitude, however, require use of the following equations:

$$\mu = 1.458E\text{-}08 \frac{T_n^{3/2}}{(T_n + 110.4)}$$

where T_n, the temperature is calculated by

$$T_n = \left(\frac{V_{sos}}{20.046707}\right)^2$$

and

$$\frac{\partial \mu}{\partial h} = \frac{\mu}{V_{sos}} \left(3 - \frac{2T_n}{(T_n = 110.4)}\right) \frac{\partial V_{sos}}{\partial h}$$

$$\frac{\partial T_n}{\partial h} = \frac{2T_n}{V_{sos}} \frac{\partial V_{sos}}{\partial h} .$$

It is assumed that the atmosphere rotates with the earth, thus, no additional perturbations are introduced into the equations of motion; however, a horizontal wind may be simulated by specifying the direction and magnitude of the wind as a function of altitude. The wind effects the simulation through the equation for relative velocity which accounts for the earth rotation and wind; thus,

$$\overline{V}_R = \overline{V}_I - \overline{\omega}_e \times \overline{R} + V_W \left(\sin A_{Z_W} \hat{E} + \cos A_{Z_W} \hat{N}\right)$$

where

$\overline{V}_I$ = inertial velocity vector

$\overline{R}$ = radius vector

$\overline{\omega}_e$ = angular momentum vector of the earth

V_W = velocity of the wind

A_{Z_W} = wind direction (azimuth)

$\hat{E} = \dfrac{\overline{\omega}_e \times \overline{R}}{|\overline{\omega}_e \times \overline{R}|}$ (local East)

$\hat{N} = \dfrac{\overline{R} \times \hat{E}}{|\overline{R}|}$. (local North)

Associated with the environment are several pertinent parameters which are calculated during the simulation. These parameters and equations are:

Dynamic pressure (Q) $\quad Q = \frac{1}{2} \rho V_R^2 \qquad V_R = |\overline{V}_R|$

Mach number (Mn) $\quad Mn = \dfrac{V_R}{V_{sos}}$

Reynolds number (Rn) $\quad Rn = \dfrac{\rho V_R \ell_{REF}}{\mu}$.

The relationship between the body axis and the relative velocity vector is represented by two angles, the angle of attack (α) and sideslip angle (β) (see Figure 2-2).

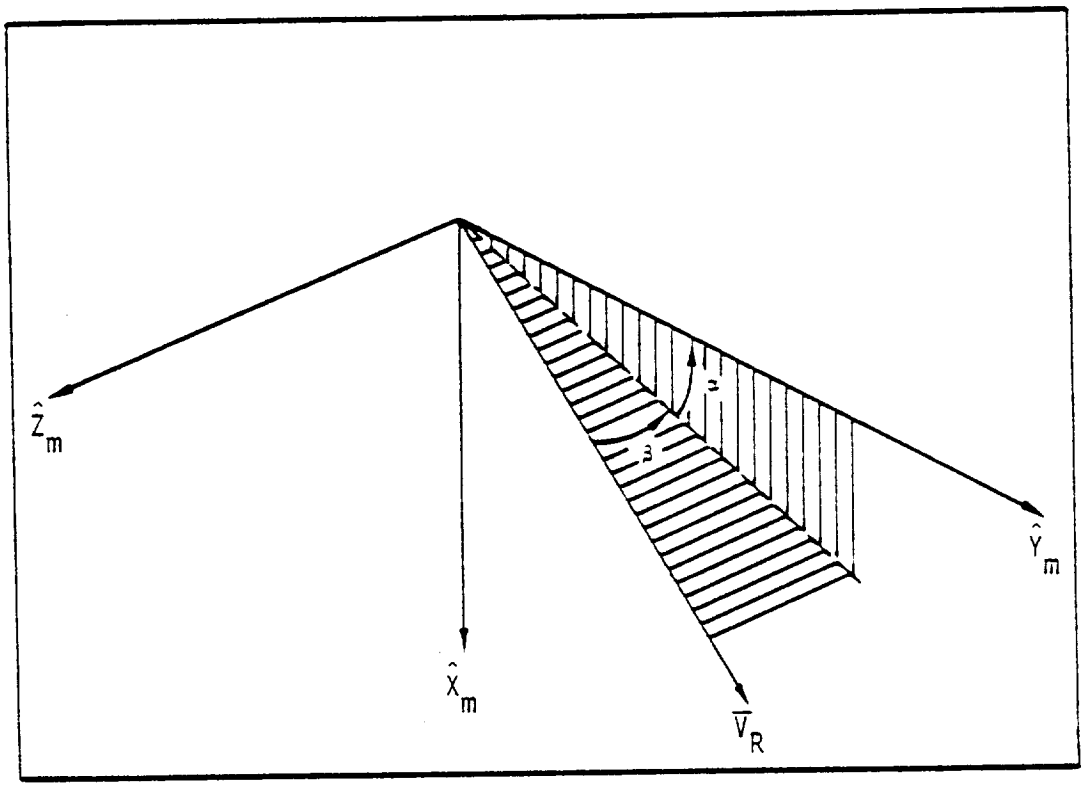


Figure 2-2. Angle-of-Attack and Sideslip Angle Orientation

The angle of attack defines the angle between the longitudinal axis and the projection of the relative velocity vector in the pitch plane. The sideslip angle defines the angle between the relative velocity vector and the pitch plane projection. These angles are evaluated by the following equations:

$$\alpha = \tan^{-1}\left(\frac{\hat{X}_M \cdot \overline{V}_R}{\hat{Y}_m \cdot \overline{V}_R}\right)$$

$$\beta = \sin^{-1}\left(\frac{\hat{Z}_m \cdot \overline{V}_R}{|\overline{V}_R|}\right).$$

2.2 COORDINATE SYSTEM

The primary coordinate system used in the MASTRE program is an inertial plumbline coordinate system which is earth centered and referenced to the launch site. The Y-axis of the system is parallel to the launch site geodetic latitude, the X-axis of the system is directed along the flight azimuth, and the Z-axis forms a right-hand set (Figure 2-3). Since the Y-axis is parallel to the launch site geodetic latitude, the plumbline system concurs with the inertial guidance and control platform established prior to launch based upon plumbline measurements. In this sense, the attitude angles, which measure the angular distance between the plumbline system and body system in pitch, yaw, and roll, are the same angles used by the guidance and control programs.

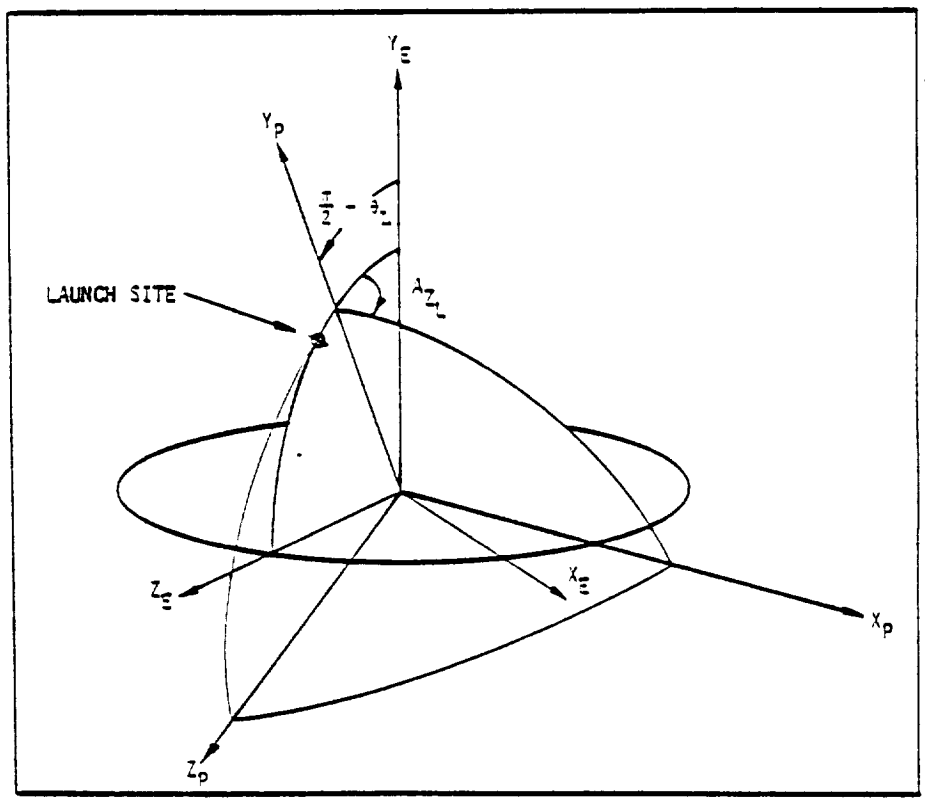


Figure 2-3. Reference Plumbline Coordinate System

The definition of earth related parameters are made through an inertial earth centered equatorial coordinate system defined where Y_E is through the north pole, Z_E is in the equatorial plane and through the launch meridian, and X_E forms a right-handed system (see Figure 2-3). This system is formed through the rotation of the launch azimuth (A_{Z_L}) and the launch geodetic latitude (θ_L), so that:

$$\overline{X}_E = [\theta_L - \tfrac{\pi}{2}]_1 \; [A_{Z_L} - \tfrac{\pi}{2}]_2 \; \overline{X}_P$$

or

$$\overline{X}_E = [A]^T \, \overline{X}_P \; .$$

$[\;]_i$, $i = 1, 2, 3$ represents an angular rotation matrix about the X, Y, Z axis, respectively.

The body coordinate system is defined so that the Y_m-axis is along the longitudinal axis of the vehicle, the X_m-axis is in the pitch plane of the vehicle, and, in the case of the shuttle, points away from the orbiter tail, and the Z_m-axis forms a right-hand set (shown in the lift-off configuration portrayed in Figure 2-4). The body coordinate system is coupled to the plumbline system so that if all attitude angles are zero the two coordinate systems coincide.

The transformation matrix from the plumbline system to the body system is determined by an ordered rotation set of the attitude angles. The transformation matrix is defined:

$$\overline{X}_m = [\chi_R] \; [\chi_Y]_1 \; [-\chi_P]_3 \; \overline{X}_P$$

where

χ_P, χ_Y, χ_R are the pitch, yaw, and roll attitudes, respectively, (see Figure 2-5).

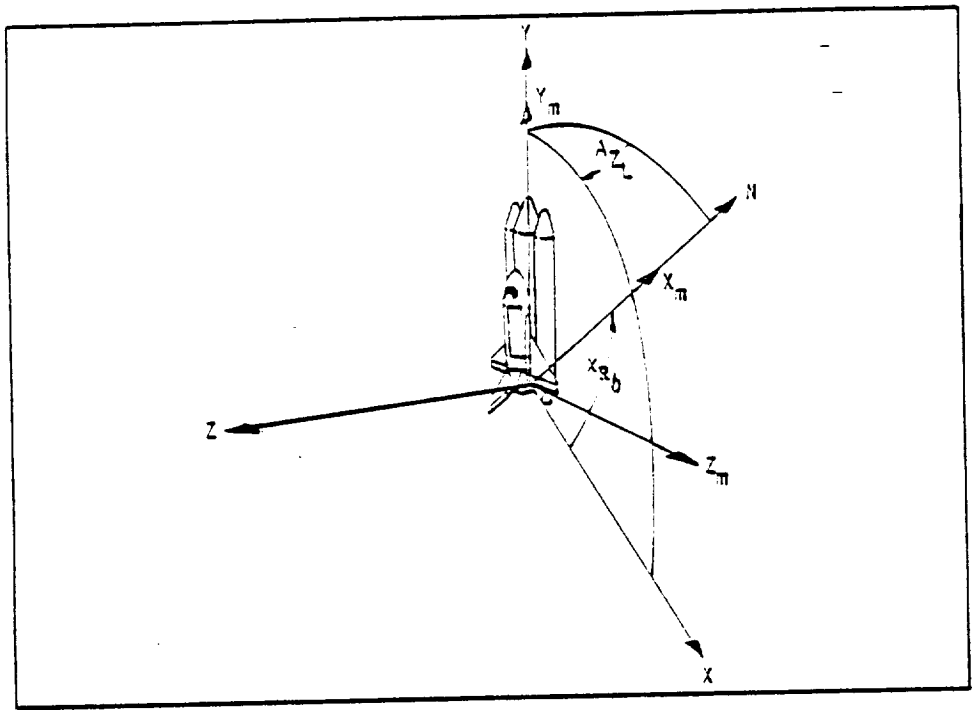


Figure 2-4. Lift-Off Configuration

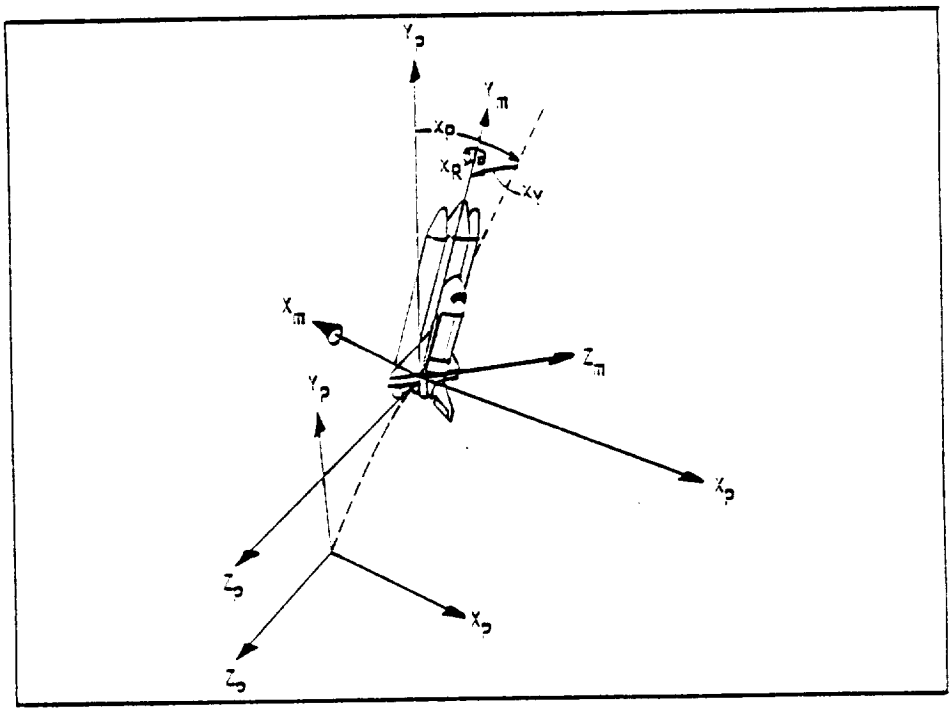


Figure 2-5. Attitude Angles

For further notation, let [C] represent this ordered set of rotation matrices, so that

$$\overline{X}_m = [C]\,\overline{X}_p \; .$$

2.3 GENERALIZED EQUATIONS OF MOTION

The equations of motion for the shuttle trajectory consist of a combination of accelerations associated with the body and the environment. In general, the equations of motion are written in the following form

$$\ddot{\overline{X}} = \frac{1}{M}\,[C]^T\,[\overline{T} + \overline{A}] + \overline{G}$$

where:

M = instantaneous mass of the vehicle

$[\,]^T$ = transpose of the C matrix

$\overline{T}$ = thrust force vector in the body system

$\overline{A}$ = aerodynamic force vector in the body system

$\overline{G}$ = gravitational acceleration in the plumbline system

$\ddot{\overline{X}}$ = total acceleration vector in the plumbline system.

The thrust and aerodynamic forces, which will be presented later in this section, are determined first in the body system for simplicity and then transformed to the plumbline system. These transformed accelerations are then combined with the gravitational accelerations to produce the total acceleration on the launch vehicle.

The calculation of the gravitational accelerations in the plumbline system is based upon determining the partial derivatives of the gravitational potential equation, presented in Subsection 2.1, with respect to the state position vector; therefore,

$$\overline{G} = \begin{bmatrix} G_X \\ G_Y \\ G_Z \end{bmatrix} = \begin{bmatrix} \frac{\partial u(R,\ \theta)}{\partial X} \\ \frac{\partial u(R,\ \theta)}{\partial Y} \\ \frac{\partial u(R,\ \theta)}{\partial Z} \end{bmatrix} = \frac{\partial u(R,\ \theta)}{\partial R} \begin{bmatrix} \frac{\partial R}{\partial X} \\ \frac{\partial R}{\partial Y} \\ \frac{\partial R}{\partial Z} \end{bmatrix} + \frac{\partial u(R,\ \theta)}{\partial \theta} \begin{bmatrix} \frac{\partial \theta}{\partial X} \\ \frac{\partial \theta}{\partial Y} \\ \frac{\partial \theta}{\partial Z} \end{bmatrix}$$

where θ is the geocentric colatitude.

This equation can be rearranged into the form:

$$\begin{bmatrix} G_X \\ G_Y \\ G_Z \end{bmatrix} = G_{11} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \frac{G_{T_o}}{|\overline{\omega}_e|} \begin{bmatrix} \omega_X \\ \omega_Y \\ \omega_Z \end{bmatrix}$$

where

$$G_{11} = -\frac{\mu_e}{R^3}\left[1 + J\left(\frac{R_e}{R}\right)^2 \left(1-5\cos^2\vartheta\right) + H\left(\frac{R_e}{R}\right)^3 \left(3-7\cos^2\vartheta\right)\cos\vartheta \right.$$

$$\left. + D\left(\frac{R_e}{R}\right)^4 \left(\frac{3}{7} - \left(6-9\cos^2\vartheta\right)\cos^2\vartheta\right)\right]$$

$$G_{T_o} = \frac{\mu_e}{R^2}\left[2J\left(\frac{R_e}{R}\right)^2 \cos\vartheta - H\left(\frac{R_e}{R}\right)^3 \left(\frac{3}{5} - 3\cos^2\vartheta\right)\right.$$

$$\left. + D\left(\frac{R_e}{R}\right)^4 \left(\frac{12}{7} - 4\cos^2\vartheta\right)\cos\vartheta\right]$$

For a spherical earth, where $J = H = D = 0$, the latter equations reduce to the following:

$$G_{11} = -\frac{\mu_e}{R^3}$$

$$G_{T_o} = 0 \ .$$

2.4 BODY RELATED FORCES

The thrust and aerodynamic forces are calculated in the body system before being transformed into the plumbline system. The body system is referenced to define all body related parameters associated with engine locations, center of gravity location, moment reference point, and thrust vector control requirements.

2.4.1 Aerodynamic Forces

The aerodynamic forces acting on the vehicle are resolved into axial, normal, and side components. These forces are determined by the product of the dynamic pressure (Q), the aerodynamic reference area (S), and corresponding values of the axial, normal, or side force coefficients, and base force components (F_{A_B} and F_{N_B}); thus,

$$F_A = Q\ S\ C_A + F_{AB} \qquad \text{(axial)}$$

$$F_N = Q\ S\ C_N + F_{N_B} \qquad \text{(normal)}$$

$$F_S = Q\ S\ C_S \qquad \text{(side).}$$

The aerodynamic coefficients C_A, C_N, and C_S used in these equations are composed of viscous and inviscid terms. The inviscid terms are evaluated whenever the mach number is less than 10.0 and are dependent functions of mach number (Mn), angle of attack (α), sideslip angle (β), and elevon deflection angle (δ) bias. For mach numbers greater than 10, the viscous terms are used and are dependent upon mach number and angle of attack.

The evaluation method for the inviscid terms depends on whether the first stage (complete configuration) or the second stage (post SRM separation) is being simulated. The first stage aerodynamics assume a four dimensional dependence on mach number, angle of attack, sideslip angle, and elevon deflection. During the first stage, the coefficients are evaluated using the following equations:

$$C_A = C_{A_1} + C_{A_2}\,\Delta\delta_{IN} + C_{A_3}\,\Delta\delta_{IN}^2 + C_{A_4}\,\Delta\delta_{OUT} + C_{A_5}\,\Delta\delta_{OUT}^2 + C_{A_6}\,\Delta\delta_{IN}\,\Delta\delta_{OUT} + \Delta C_A$$

$$C_N = C_{N_1} + C_{N_2}\,\Delta\delta_{IN} + C_{N_3}\,\Delta\delta_{IN}^2 + C_{N_4}\,\Delta\delta_{OUT} + C_{N_5}\,\Delta\delta_{OUT}^2 + C_{N_6}\,\Delta\delta_{IN}\,\Delta\delta_{OUT} + \Delta C_N$$

$$C_S = C_{S_1} + C_{S_2}\,\Delta\delta_{IN} + C_{S_3}\,\Delta\delta_{IN}^2 + C_{S_4}\,\Delta\delta_{OUT} + C_{S_5}\,\Delta\delta_{OUT}^2 + C_{S_6}\,\Delta\delta_{IN}\,\Delta\delta_{OUT}$$

where

C_{A_i}, C_{N_i}, and C_{S_i} are dependent functions of angle of attack, sideslip angle and mach number (C_{A_i} and C_{N_i} assume $\beta = 0$ data and C_{S_i} assumes $\alpha = 0$ data)

$\Delta\delta_{IN} = \delta_{IN_{SCH}} - \delta_{IN_{ACTUAL}}$ (inboard elevon deflection)

$\Delta\delta_{OUT} = \delta_{OUT_{SCH}} - \delta_{OUT_{ACTUAL}}$ (outboard elevon deflection)

$\Delta C_A, \Delta C_N$ = additional axial and normal forces based on minor configuration changes

$\delta_{IN_{SCH}}, \delta_{OUT_{SCH}}$ = scheduled history of inboard and outboard elevons

$\delta_{IN_{ACTUAL}}, \delta_{OUT_{ACTUAL}}$ = actual history of inboard and outboard elevons.

Inviscid aerodynamics for the second stage are evaluated using a linear relationship for variations in angle of attack or sideslip angle. Thus, for the second stage, the following equations are used:

$$C_A = C_{A_o} + C_{A_\alpha}\,\alpha$$

$$C_N = C_{N_o} + C_{N_\alpha}\,\alpha$$

$$C_S = C_{S_o} + C_{S_\beta}\,\beta$$

where

C_{A_o}, C_{N_o}, and C_{S_o} are mach number dependent values evaluated at α (or β) = 0.

$C_{A_\alpha}(Mn)$, $C_{N_\alpha}(Mn)$, $C_{S_\beta}(Mn)$ are the first derivatives of the coefficient with respect to α (or β).

Viscous terms only effect the axial and normal force coefficients and are represented by the following equations:

$$\left.\begin{aligned} C_A &= (C_{A_{MF}} - C_{A_{CF}}) \sin (9.009 (K_n - 0.01)) + C_{A_{CF}} \\ C_N &= (C_{N_{MF}} - C_{N_{CF}}) \sin (9.009 (K_n - 0.01)) + C_{N_{CF}} \end{aligned}\right\} \text{ for } .01 < K_n < 10.$$

If $K_n < 0.01$, then

$$C_A = C_{A_{CF}}$$

$$C_N = C_{N_{CF}}$$

and if $K_n > 10$, then

$$C_A = C_{A_{MF}}$$

$$C_N = C_{N_{MF}}$$

where

K_n = Knudsen number

$$K_n = 0.1407 \frac{\lambda}{V_R} \left(\frac{P}{\rho}\right)^{1/2}$$

λ = mean free path

$$\lambda = \frac{1}{(2)^{1/2} \pi N \sigma^2} \frac{R^*}{M_o} \frac{MT_m}{P}$$ (Using the constants as defined below, this equation reduces to $\lambda = 2.332353 \times 10^{-8}\ T_n$).

R^* = universal gas constant (8.31432×10^3 joules $(°K)^{-1}$ $(kg - mol)^{-1}$

M = mean molecular weight of air

M_o = molecular weight (at sea level) (28.9644 kg $(kg - mol)^{-1}$)

T_m = molecular scale temperature

T = kinetic temperature ($T = \frac{MT_m}{M_o}$)

= effective collision diameter of a mean air molecule (3.65×10^{-10} m)

N = Avogadro's number (6.02257×10^{26} (kg - mol)$^{-1}$)

$C_{A_{MF}}$ = free molecular flow axial force coefficient

$$C_{A_{MF}} = a_1 + b_1\alpha + c_1\alpha^2$$

$$a_1 = f_1(Mn),\ b_1 = f_2(Mn),\ c_1 = f_3(Mn)$$

$C_{A_{CF}}$ = continuum flow axial force coefficient

$$C_{A_{CF}} = a_2 + b_2\alpha + c_2\alpha^2$$

$$a_2 = f_4(Mn),\ b_2 = f_5(Mn),\ c_2 = f_6(Mn)$$

$C_{N_{MF}}$ = free molecular flow normal force coefficient

$$C_{N_{MF}} = a_3 + b_3\alpha + c_3\alpha^2$$

$$a_3 = f_7(Mn),\ b_3 = f_8(Mn),\ c_3 = f_9(Mn)$$

$C_{N_{CF}}$ = continuum flow normal force coefficient

$$C_{N_{CF}} = a_4 + b_4\alpha + c_4\alpha^2$$

$$a_4 = f_{10}(Mn),\ b_4 = f_{11}(Mn),\ c_4 = f_{12}(Mn)$$

α = angle of attack

f_i (Mn) = Mach number dependent tables for the ith coefficient.

The representation of the axial and normal base forces is also dependent on whether first or second stage is being simulated. First stage base forces are calculated as a function of altitude, angle of attack, sideslip angle, MPS throttle setting, and dynamic pressure. Second stage base forces are dependent on the number of MPS engines operating and altitude (MPS engine out in the first stage is not currently simulated). The current method of using the power on base effects for the first stage are represented in the following equations:

$$F_{AB} = \left\{ F_{AB_{TOT}(f = 1.09)} - F_{AB_{ORB}(f = 1.09)} + \left[F_{AB_{ORB}(f = 1.09)} (K_2 \Delta f^2 + K_1 \Delta f + 1) \right] \right\} (K_\beta \beta^2 + K_\alpha \alpha + 1) \frac{Q}{Q_{REF}} + \Delta F_{AB}$$

$$F_{NB} = F_{NB(f = 1.09)} (K_2 \Delta f^2 + K_1 \Delta f + 1)(K_\beta \beta^2 + K_\alpha \alpha + 1) \frac{Q}{Q_{REF}}$$

where

$F_{AB(f = 1.09)}$, $F_{NB(f = 1.09)}$ - base force components for 109% MPS thrust (altitude dependent) (subscripts are defined: TOT - total configuration, ORB - orbiter alone)

$f = 1.09 - f$

K_1, K_2, K_α, K_β - coefficients defined in aero data

Q_{REF} - reference dynamic pressure (altitude dependent)

ΔF_{AB} - additional base force due to configuration changes (LBM modification)

The total aerodynamic forces in the body axis system are defined in the following manner:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} -F_N \\ -F_A \\ F_S \end{bmatrix} .$$

2.4.2 Aerodynamic Moments

The aerodynamic moments, M_A, consist of pitch, yaw, and roll moments (see Figure 2-6) and are evaluated in the same manner as the aerodynamic forces. During the first stage simulation, the coefficients of these moments are evaluated by the following equations:

$$C_m = C_{m_1} + C_{m_2} \Delta\delta_{IN} + C_{m_3} \Delta\delta_{IN}^2 + C_{m_4} \Delta\delta_{OUT} + C_{m_5} \Delta\delta_{OUT}^2 + C_{m_6} \Delta\delta_{IN} \Delta\delta_{OUT} + \Delta C_m$$

$$C_n = C_{n_1} + C_{n_2} \Delta\delta_{IN} + C_{n_3} \Delta\delta_{IN}^2 + C_{n_4} \Delta\delta_{OUT} + C_{n_5} \Delta\delta_{OUT}^2 + C_{m_6} \Delta\delta_{IN} \Delta\delta_{OUT}$$

$$C_\ell = C_{\ell_1} + C_{\ell_2} \Delta\delta_{IN} + C_{\ell_3} \Delta\delta_{IN}^2 + C_{\ell_4} \Delta\delta_{OUT} + C_{\ell_5} \Delta\delta_{OUT}^2 + C_{\ell_6} \Delta\delta_{IN} \Delta\delta_{OUT}$$

where

C_{m_i}, C_{n_i}, C_{ℓ_i} are dependent functions of angle of attack, sideslip angle, and mach number (C_{m_i} assumes $\beta = 0$ data and C_{n_i} and C_{ℓ_i} assume $\alpha = 0$ data).

ΔC_m - additional pitch moment due to configuration changes.

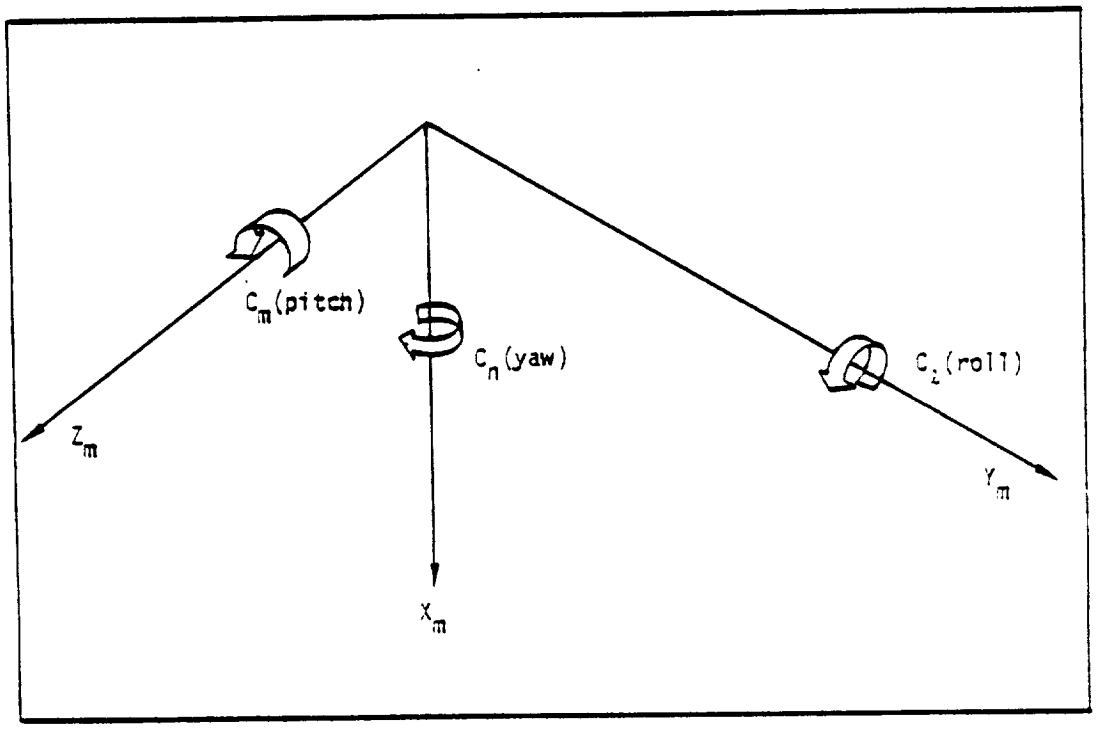


Figure 2-6. Aerodynamic Moment Orientation

Second stage moment coefficients are evaluated using a linear relationship, thus

$$C_m = C_{m_o} + C_{m_\alpha}\alpha$$

$$C_n = C_{n_o} + C_{n_\beta}\beta$$

$$C_\ell = C_{\ell_o} + C_{\ell_\beta}\beta$$

where

C_{m_o}, C_{n_o}, C_{ℓ_o} - mach number dependent values of pitch, yaw, and roll moment coefficients at α (or β) $= 0$

C_{m_α}, C_{n_β}, C_{ℓ_β} - mach numbers dependent values of the slope of the coefficients with respect to angle of attack or sideslip angle.

The pitch moment coefficient is also effected by viscous interaction, thus when mach number exceeds 10, the equation is changed to:

$$C_m = (C_{m_{MF}} - C_{m_{CF}}) \sin (9.009 (K_n - 0.01)) + C_{m_{CF}} \text{ for } .01 < K_n < 10$$

where for $K_n < .01$, $C_m = C_{m_{CF}}$; for $K_N > 10$ $C_m = C_{m_{MF}}$

$C_{m_{MF}}$ - free molecular flow pitch moment coefficient

$C_{m_{CF}}$ - continuum flow pitch moment coefficient.

The aerodynamic moments coefficients are translated from the reference point to the center of gravity by the equations:

$$C_{M_{Z_{CG}}} = C_m - C_N(Y_{CG} - Y_{REF})/\ell_{REF} + C_A(X_{CG} - X_{REF})/\ell_{REF}$$

$$C_{M_{Y_{CG}}} = C_\ell + C_S(X_{CG} - X_{REF})/\ell_{REF}$$

$$C_{M_{X_{CG}}} = C_n - C_S(Y_{CG} - Y_{REF})/\ell_{REF}.$$

These coefficients are used in conjunction with the dynamic pressure, the aerodynamic reference area, the reference length, and the base force contribution to determine the aerodynamic moments; thus,

$$\begin{bmatrix} M_{AX} \\ M_{AY} \\ M_{AZ} \end{bmatrix} = QS\ell_{REF} \begin{bmatrix} C_{M_{X_{CG}}} \\ C_{M_{Y_{CG}}} \\ C_{M_{Z_{CG}}} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \Delta M_Z \end{bmatrix}$$

where ΔM_Z is the additional pitch moment associated with the base force; ΔM_Z is evaluated based on the equations:

$$\Delta M_Z = \Delta M_{T=1.09}(K_2\Delta f^2 + K_1\Delta f + 1)(K_\beta\beta^2 + K_\alpha\alpha + 1)\frac{Q}{Q_{REF}}$$

during first stage simulation, and

$$\Delta M_Z = \text{function of altitude}$$

for second stage simulation.

2.4.3 Thrust Forces

Five propulsion systems are available for simulation in the program. Of these five, two systems, the Main Propulsion System (MPS) and the Solid Rocket Motors (SRM), are controllable via thrust vector control (TVC). The three other systems, the Liquid Boost Modules (LBM), the Orbital Maneuver System (OMS), and the Reaction Control System (RCS), are stationary. The controllable engines are used to counterbalance, or moment balance, the combination of aerodynamic moments and moments created by the fixed engine offsets. A three degree of freedom environment is maintained by the instantaneous commanded and response of the controlled engines, thus creating no vehicular angular rates.

The calculation of controllable engine thrust forces requires that the thrust vector gimbal angles be a function of the thrust magnitudes (fixed and controllable), the location of the engines with respect to the CG, the aerodynamic forces and moments about the CG, and any additional moment created by offset thrust components. This calculation is performed in the MASTRE programs using a three-dimensional moment balance scheme which nulls the moments in the body pitch, yaw, and roll direction. Three assumptions, required for the development of this scheme, are made which simplify the equations and provide a non-iterative analytic formulation. These assumptions are:

1. Controllable engines during the simulation will consist of either the MPS or the SRM, but not both.

2. All controllable engines are combined into two equivalent thrust engines equally spaced on either side of the pitch plane (defined by averaging the respective gimbal locations).

3. The pitch plane gimbal angles for each equivalent engine are variable, but the yaw plane gimbal angles are forced to be equivalent.

The total thrust vector is defined by the following equation:

$$\bar{T} = \sum_{i=1}^{5} \sum_{j=1}^{N_{ENG_i}} \begin{bmatrix} T_{X_{ji}} \\ T_{Y_{ji}} \\ T_{Z_{ji}} \end{bmatrix}$$

where

$T_{X_{ji}}$, $T_{Y_{ji}}$, $T_{Z_{ji}}$ - represent the body lateral, longitudinal, and side thrust forces for the ith propulsion system, and the jth engine.

N_{ENG_i} - number of engines associated with propulsion system i.

For the ith propulsion system, the jth fixed engine thrust force components are calculated via the following equation

$$\begin{bmatrix} T_{X_j} \\ T_{Y_j} \\ T_{Z_j} \end{bmatrix} = \frac{f_j \; T_{VAC_j} - A_{\epsilon_j} \; P}{\sqrt{1 + \tan^2\delta_{P_j} + \tan^2\delta_{Y_j}}} \begin{bmatrix} -\tan\delta_{P_j} \\ 1 \\ \tan\delta_{Y_j} \end{bmatrix}$$

where

f_j - throttle level (used for MPS engines only)

T_{VAC_j} - vacuum thrust of jth engine

A_{ε_j} - exit area

P - atmospheric pressure

δ_{P_j} - pitch gimbal angle defined in the pitch plane (XY) of the vehicle

δ_{Y_j} - yaw gimbal angle defined in the yaw plane (YZ) of the vehicle.

The total moment required to be balanced by controllable engines can now be defined, thus

$$\overline{M}_T = \begin{bmatrix} M_{AX} \\ M_{AY} \\ M_{AZ} \end{bmatrix} + \sum_{i=1}^{N_{FIXED}} \sum_{j=1}^{N_{ENG_i}} \begin{bmatrix} 0^* \\ 0^* \\ -T_{Y_{ji}}(X_{CG} - X_{GP_{ji}}) + T_{X_{ji}}(Y_{CG} - Y_{GP_{ji}}) \end{bmatrix}$$

where

N_{FIXED} - number of fixed engines

X_{CG}, Y_{CG} - center of gravity components dependent on instantaneous mass

$X_{GP_{ji}}$, $Y_{GP_{ji}}$ - gimbal location of jth engine in i^{th} propulsion system (see Figure 2-7).

The moment balance scheme used to calculate the controllable thrust vector components are developed in the following paragraphs.

---

*The assumption is made that the fixed engines are symmetric about all axes except pitch.

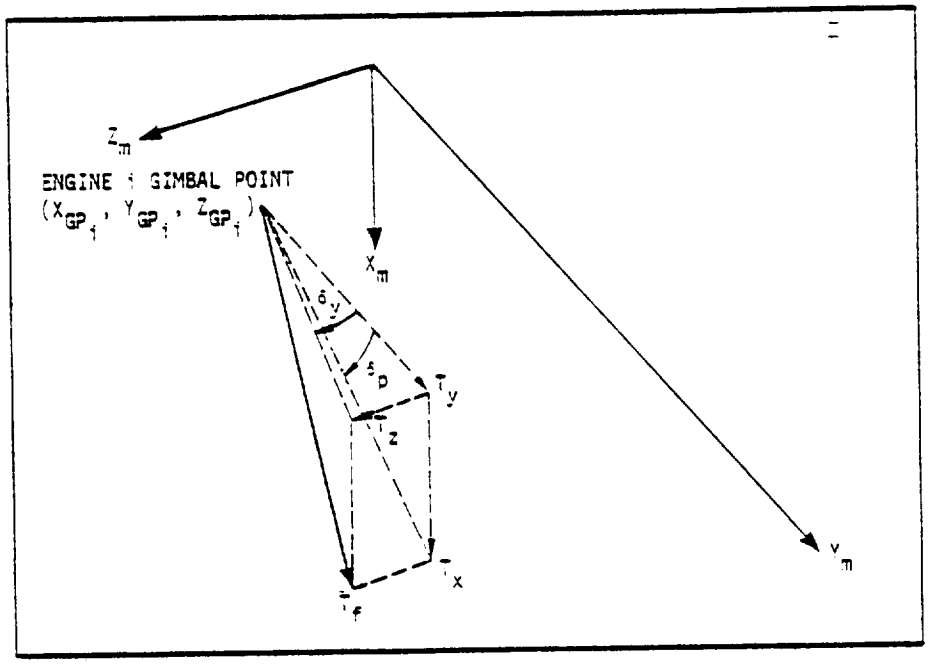


Figure 2-7. Engine Deflection Angles for Fixed Orientation

The moment balance problem is simplified by defining a coordinate system which relates the engine gimbal point and the center of gravity. Since the controllable engines are equally spaced with respect to the pitch and yaw planes, a plane containing the center of gravity and the gimbal points can be constructed (Figure 2-8). The coordinate system xyz is defined so that the origin is located at the center of gravity and the engine gimbal points are contained in the yz plane.

Pitch and yaw gimbal angles for each engine are defined as

$$\delta_{P_i} = \tan^{-1}\left(\frac{-T_{x_i}}{T_{y_i}}\right)$$

$$\delta_{Y_i} = \tan^{-1}\left(\frac{T_{z_i}}{T_{y_i}}\right)$$

where i denotes the first or second engine and T_x, T_y, T_z denotes the thrust components in the xyz system.

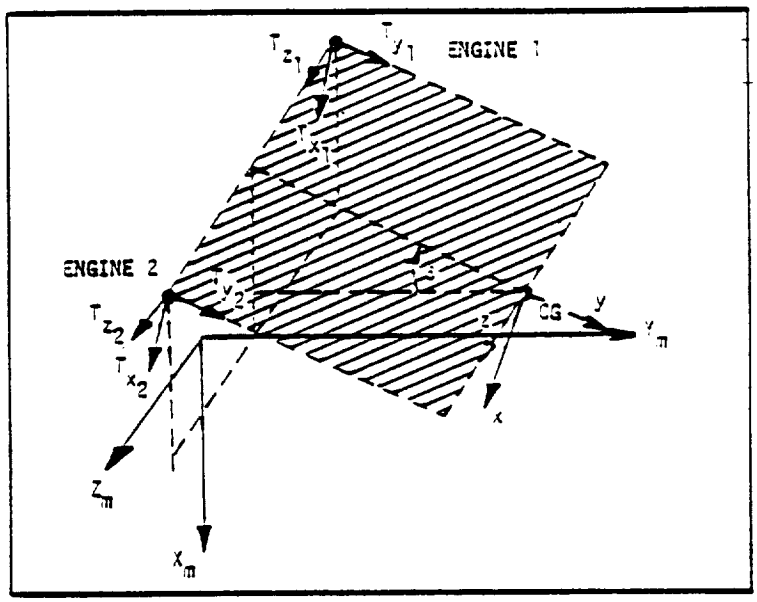


Figure 2-8. Center of Gravity/Gimbal Plane Coordinate System

The moment equations about each axis to be satisfied can be written

$$\left(T_{y_1} \tan \delta_{Y_1} + T_{y_2} \tan \delta_{Y_2}\right) dy + \left(T_{y_1} - T_{y_2}\right) dz = -M_x$$

$$\left(T_{y_1} \tan \delta_{P_1} - T_{y_2} \tan \delta_{P_2}\right) dz - \left(T_{y_1} \tan \delta_{Y_1} + T_{y_2} \tan \delta_{Y_2}\right) dx = -M_y$$

$$\left(T_{y_1} + T_{y_2}\right) dx + \left(T_{y_1} \tan \delta_{P_1} + T_{y_2} \tan \delta_{P_2}\right) = -M_z$$

where

M_x, M_y, M_z are the components of the compensating moment in the xyz coordinate system

$$dx = x_{GP_1} - x_{CG} = x_{GP_2} - x_{CG} = 0$$

$$dy = y_{GP_1} - y_{CG} = y_{GP_2} - y_{CG}$$

$$dz = z_{GP_2} - z_{CG} = -\left(z_{GP_1} - z_{CG}\right)$$

and the following substitutions have been made

$$T_{x_i} = T_{y_i} \tan \delta_{P_i}$$

$$T_{z_i} = T_{y_i} \tan \delta_{Y_i} \; .$$

The longitudinal component of thrust is defined based on the constraint

$$T_i = \sqrt{T_{x_i}^{\;2} + T_{y_i}^{\;2} + T_{z_i}^{\;2}}$$

and, through substitution, the equation for the longitudinal component is written

$$T_{y_i} = \frac{T_i}{\sqrt{1 + \tan^2 \delta_{P_i} + \tan^2 \delta_{Y_i}}}$$

In order to solve the moment equations for the four gimbal angles, the assumption is made that $\delta_{Y_1} = \delta_{Y_2} = \delta_Y$, thus, reducing the problem to three equations and three unknowns. The expansion of these equations is treated in Appendix B and results in the following equations in the body coordinate system

$$T_X = -\frac{fT}{2}\left[\frac{\tan \delta'_{P_1}}{\sqrt{1 + \tan^2 \sigma_{P_1} + \tan^2 \delta'_Y}} + \frac{\tan \delta'_{P_1}}{\sqrt{1 + \tan^2 \delta'_{P_2} + \tan^2 \delta'_Y}}\right]$$

$$T_Y = \frac{fT}{2}\left[\frac{1}{\sqrt{1 + \tan^2 \delta'_{P_1} + \tan^2 \delta'_Y}} + \frac{1}{\sqrt{1 + \tan^2 \delta'_{P_2} + \tan^2 \delta'_Y}}\right]$$

$$T_Z = T_Y \tan \delta'_Y$$

where

f = the throttle level (MPS only)

T = the total thrust of the controllable engines

$\delta'_{P_1} = \delta_{P_1} - \delta$

$\delta'_{P_2} = \delta_{P_2} - \delta$

$\delta'_Y = \delta_Y$

and

δ is the angle between the gimbal point and center of gravity plane and the vehicle yaw plane.

2.5 MASS CALCULATION

The calculation of the instantaneous mass, M, at any time, t, is determined by the summation of the integrated value of the propellant plus the payload, M_I, and the inert weight, M_A, which consists of the weight to be either jettisoned or dropped during the simulation. This procedure of not allowing the integrated mass to consider the jettisoned or dropped weights allows an easier reconstruction of the mass during the backward simulation of the adjoint equations of motion since the integrated mass is continuous throughout the trajectory.

The mass at any time, t, is given by the equation

$$M = M_A + M_I$$

where the inert mass (M_A) is determined as a function of the current thrust event so that

$$M_A = \left(\sum_{i=1}^{N} W_{d_i} + \sum_{k=1}^{J} W_{j_k} \right) / g_o = \sum_{i=1}^{K-1} (W_{d_i} + W_{j_i})/g_o$$

where

N = total number of drop weight events

J = total number of thrust events

K = current value of the thrust event

W_d = drop weight

W_j = jettison weight.

The term 'jettison' weight refers to weights jettisoned at the end of a thrust event where as 'drop' weights can occur any time during a thrust event. The option of both types of weight deletions gives the program flexibility for the thrust event type of logic being used.

Propellant flow rate is a function of the type of propulsion system being simulated. MPS flow rate is based on the relationship

$$\dot{m}_{MPS} = N_{ENG} \frac{\tau T_{VAC}}{g_o \, I_{sp}}$$

where N_{ENG} is the number of MPS engines, τ is the throttle setting, T_{VAC} is the vacuum thrust, and I_{sp} is the specific impulse. The uniqueness of the MPS flow rate is the dependence of specific impulse on time from launch and throttle setting. To simulate this dependence, the value of I_{sp} at 100% is evaluated as a function of time from launch and the throttle relationship is handled as a constant quadratic relationship, such that

$$I_{sp} = \frac{I_{sp_{100\%}}(t)}{I_{sp_{REF}}} \; (a \, \tau^2 + b\tau + c)$$

where

$I_{sp_{100\%}}(t)$ - specific impulse of 100% thrust evaluated at time t

$I_{sp_{REF}}$ - reference value of specific impulse

a,b,c - coefficients of quadratic relationship

τ - throttle setting.

The flow rate for the SRB and LBM systems is evaluated as a function of time from launch; thus

$$\dot{m}_{SRB} = N_{ENG}\dot{m}_{SRB}(t) \qquad \text{(input table)}$$

$$\dot{m}_{LBM} = N_{ENG}\dot{m}_{LBM}(t) \qquad \text{(input table).}$$

The simulation of dumping propellant from the interconnect system linking the OMS and RCS systems is performed by allowing the additional thrust component to be included in the RCS thrust calculation and the flow rate to be included in the OMS flow rate calculation. The equations for OMS and RCS flow rate are directly a function of user input, thus,

$$\dot{m}_{OMS} = N_{ENG}\dot{m}_{OMS_k} + \dot{m}_{DUMP_k}$$

$$\dot{m}_{RCS} = N_{ENG}\dot{m}_{RCS_k}$$

where:

$\dot{m}_{OMS}$, $\dot{m}_{RCS_k}$, $\dot{m}_{DUMP_k}$ - the OMS, RCS and dumped flow rates at thrust event k.

Section 3

THE FORWARD TRAJECTORY

This section describes the equations of motion used to evaluate the forward trajectory, a description of the forward trajectory flight phase options, the terminal and intermediate functions which may be selected as optimization parameters, and various options associated with the forward trajectory.

3.1 THE EQUATIONS OF MOTION

The general form of the equations of motion integrated in MASTRE are

$$\dot{P}_1 = \dot{W} = F_X/M + G_X$$

$$\dot{P}_2 = \dot{U} = F_Y/M + G_Y$$

$$\dot{P}_3 = \dot{V} = F_Z/M + G_Z$$

$$\dot{P}_4 = \dot{X} = W$$

$$\dot{P}_5 = \dot{Y} = U$$

$$\dot{P}_6 = \dot{Z} = V$$

$$\dot{P}_7 = -(\dot{m}_{SSME} + \dot{m}_{SRB} + \dot{m}_{LBM} + \dot{m}_{OMS} + \dot{m}_{RCS} + \dot{m}_{OTHER})$$

$$\dot{P}_8 = -\dot{m}_{SSME}$$

$$\dot{P}_9 = -\dot{m}_{SRB}$$

$$\dot{P}_{10} = -\dot{m}_{LBM}$$

$$\dot{P}_{11} = -\dot{m}_{OMS}$$

$$\dot{P}_{12} = -\dot{m}_{RCS}$$

$\dot{P}_{13} = -\dot{m}_{OTHER}$ (available for future propulsion system modification)

$$\dot{P}_{14} = 17700.(.0019403\ p)^{1/2}\left(\frac{V_R/.3048}{10000}\right)^{3.07}$$

$$\left(1. - .24\ \frac{460.}{(.24)(1.8)T_n + \frac{(V_R/.3048)^2}{50063.}}\right) \text{(stagnation heating indicator)}$$

$\dot{P}_{15} = T_{VAC}/M$ (ideal velocity)

$\dot{P}_{16} = -(T_{VAC} - T_S/M)$ (back pressure loss)

$\dot{P}_{17} = -(T_V - T_S/M)$ (gimbal loss)

$\dot{P}_{18} = -(\overline{V} \cdot \overline{G})$ (gravity loss)

$\dot{P}_{19} = -\frac{1}{M}[T_X(\hat{V} \cdot \hat{X}) + T_Y(\hat{V} \cdot \hat{Y}) + T_Z(\hat{V} \cdot \hat{Z}) - T_{VAC}]$ (turning loss)

$\dot{P}_{20} = -\frac{1}{M}[-F_N(\hat{V} \cdot \hat{X}) - F_A(\hat{V} \cdot \hat{Y}) + F_S(\hat{V} \cdot \hat{Z})]$ (aerodynamic loss)

$\dot{P}_{21} = Q - Q_{MAX}$ (evaluated only when $Q \geq Q_{MAX}$)

$\dot{P}_{22} = \dot{m}_{DUMP}$ (propellant dumped from interconnect system)

where

$$\begin{bmatrix} F_X \\ F_Y \\ F_Z \end{bmatrix} = [C]^T [\overline{T} + \overline{A}]$$

T_{VAC} = scalar sum of the vacuum thrust of all engines

T_S = scalar sum of the actual thrust of all engines

T_V = total resultant thrust

$\overline{V}$ = inertial velocity vector ('^' indicates unit vector)

$\hat{X},\hat{Y},\hat{Z}$ = vehicle axis unit vectors.

The first fourteen variables are always evaluated for every forward trajectory. Variables 15 through 22 are only integrated after a converged run has been achieved and during the final trajectory printout. The equations related to the losses (15 through 20) are developed in Appendix C.

3.2 TRAJECTORY SEQUENCE REQUIREMENTS

The flight profile of a ground launched trajectory is separated into a number of phases. The number of these phases is a function of the type of attitude control required for a particular mission profile. The sequence of these phases and the duration of each phase is optional. These phases are divided into the following categories:

- Trajectory initiation equations,
- Vertical rise,
- Roll to flight azimuth command,
- Booster polynominal attitude control,
- Booster angle-of-attack control,
- Prime contractor attitude angles θ, ψ, and σ, and
- Minimum Hamiltonian (min H) attitude control.

All trajectory simulations are required to use the initial equations and are restricted to using the min H attitude control during the terminal phase. All other phases are optional.

Jump start trajectories, started at a time other than liftoff, are restricted to the orbiter stage and can use either a fixed polynominal profile followed by the min H phase or the min H phase alone.

The following paragraphs explain the types of phases and the equations associated with each phase.

3.2.1 Trajectory Initiation Equations

The initiation of a ground launched trajectory requires the input values of the geodetic latitude (θ_L), launch azimuth (A_{Z_L}), and altitude (h_o) of the launch site. The geocentric colatitude of the launch site is calculated by the equation

$$\theta = \pi/2 - \tan^{-1} [(1 - f)^2 \tan \theta_L].$$

The radius of the launch site and the initial inertial velocity of the launch site is calculated by the equations:

$$R_o = R(\theta) + h_o$$

$$V_o = R_o \, \omega_e \sin \theta.$$

The initial plumbline position and velocity components are calculated by the equations

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \begin{bmatrix} -A_{Z_L} \end{bmatrix}_2 \begin{bmatrix} \theta_L + \theta - \pi/2 \end{bmatrix}_1 \begin{bmatrix} 0 \\ R_o \\ 0 \end{bmatrix}$$

and

$$\begin{bmatrix} W \\ U \\ V \end{bmatrix} = \begin{bmatrix} -A_{Z_L} \end{bmatrix}_2 \begin{bmatrix} \theta_L \end{bmatrix}_1 \begin{bmatrix} V_o \\ 0 \\ 0 \end{bmatrix}.$$

The initial value of the total mass to be integrated is determined by the difference between the initial mass, M_o, and the inert mass, M_A.

$$M_I = M_o - M_A.$$

With the exception of the separate propulsion flow rate values, the initial values of all other integrated variables are set to zero. Initial values of the propulsion flow rate are provided by input.

The initial values for a jump start trajectory also require the geodetic latitude of the launch site and launch azimuth along with the jump start time (from liftoff), the weight at the jump start, and the initial values of the state variables required in the equations of motion.

3.2.2 Vertical Rise

During the first portion of launch, the vehicle attitude maintains a vertical rise by defining the attitude angles which will allow vehicle orientation with respect to the launch tower. Since the Space Shuttle will assume a position on the pad such that the pitch plane is not in the flight plane, the inertial attitude angles, defined in Paragraph 2.2, no longer lie in the vehicle pitch and yaw axes defined on the launch pad. This discrepancy makes it difficult to relate to the vehicle motion before and during the roll maneuver to the flight plane. This problem has been solved by redefining the pitch, yaw, and roll attitudes in the instantaneous vehicle plane and transforming these angles to the inertial attitude angles of Paragraph 2.2. During the vertical rise maneuver, these redefined angles are defined by the following equations:

$$\chi_p' = \tan^{-1}\left(\frac{-Y_x}{Y_y}\right)$$

$$\chi_y' = \sin^{-1}(-Y_z)$$

where

$$\begin{bmatrix} Y_x \\ Y_y \\ Y_z \end{bmatrix} = \left[\chi'_{R_o} - A_{Z_L}\right]_2 \left[\theta_L\right]_1 \left[-\omega_e t\right]_2 \left[-\theta_L\right]_1 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$\chi'_{R_o} \quad - A_{Z_L} + \pi - A_{Z_{TAIL}}$

$A_{Z_{TAIL}}$ - azimuth of the orbiter tail.

The transformation to the inertial plumbline attitude angles is via the equations:

$$\chi_p = \tan^{-1}\left(\frac{Y_x}{Y_y}\right)$$

$$\chi_y = \sin^{-1}(Y_z)$$

$$\chi_R = \tan^{-1}\left(\frac{-X_Z}{Z_z}\right)$$

where

$$\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} = \left[-\chi'_y\right]_1 \left[\chi'_p\right]_3 \left[\chi'_R\right]_2 .$$

The vertical rise attitude is maintained until the initiation of the tilt-over command.

3.2.3 Roll to Flight Azimuth Command

Whenever the vehicle is launched with the Orbiter tail pointing out of the plane defined by the flight azimuth, a roll command maneuver is required to place the Orbiter tail (vehicle pitch plane) in either a cockpit up or down position in the flight azimuth plane. This maneuver can be initiated at any reasonable time, but usually as soon as possible.

The attitude angles for the roll command maneuver are based on independent pitch and yaw commands (required by the vertical rise command or other commands which will be discussed later) and roll command rate. During this maneuver, the redefined roll angle is updated via:

$$\chi'_R = \chi'_{R_o} + \dot{\chi}_R(t - t_R)$$

where

χ'_{R_o} - the initial value of roll

$\dot{\chi}_R$ - the commanded rate

t - instantaneous time from launch

t_R - roll initiation time.

The transformation equations of Paragraph 3.2.2 are used to define the inertial attitude angles.

3.2.4 Booster Polynominal Attitude Control

The termination of vertical rise is followed by a pitchover (or tiltover) phase which is commanded by a pitch polynominal as a function of time. The pitch and yaw attitude polynominals used in the simulation are of the general form

$$\chi = a_o + a_1 t + a_2 t^2 + \ldots + a_n t^n$$

where

$n \leq 6$ for pitch

$n \leq 4$ for yaw.

The order of the polynominal, as well as the time duration, is optional and can be maintained throughout the booster portion of the trajectory. However, if the time duration does not cover the entire booster portion, several optional forms of attitude philosophies can be used. These philosophies will be covered in the next section.

By inputting values of time and pitch angle or both pitch and yaw angles, plus the order of the polynominal(s) required, the program determines the coefficients of the requested order polynominal(s) in such a

manner that all points are forced to lie on the curve. Optimization of the pitch and yaw attitude angles can be performed when using a polynominal by treating input values of the pitch and yaw angles as independent parameters (as opposed to control variables used by the min H).

The MASTRE program allows the use of two such polynominals in the trajectory simulation.

If pitchover occurs prior to or during the roll maneuver, the values of pitch and yaw are assumed to be the redefined angles. In this case, the transformation equations of Paragraph 3.2.2 are used to define the inertial attitude angles.

3.2.5 Optional Angle-of-Attack Attitude History for the Booster Phase

Whenever the polynominal form of attitude control is not being used during the booster simulation, there are two attitude options. One of these options is based on restricting the attitude angles to mach number dependent values of angle of attack and sideslip angle.

For this option, the inertial attitude angles are defined by the following equations:

$$\chi_p = \tan^{-1}\left(\frac{Y'_X}{Y'_Y}\right)$$

$$\chi_y = \sin^{-1}\left(Y'_Z\right)$$

$$\chi_R = \tan^{-1}\left(\frac{-X'_Z}{Z'_Z}\right)$$

where

$$\begin{bmatrix} \hat{X}' \\ \hat{Y}' \\ \hat{Z}' \end{bmatrix} = [\alpha_c]_3 \, [\beta - \beta_c]_1 \, [-\alpha]_3 \begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix}$$

α_c, β_c - commanded values of angle of attack and sideslip angle

α, β - current values of angle of attack and sideslip angle

$\hat{X}$, $\hat{Y}$, $\hat{Z}$ - current direction cosine matrix.

3.2.6 Prime Contractor Attitude Angles θ_s, ψ, and σ

Another method of providing attitude commands during the booster phase is through the attitude angle histories supplied by the Space Shuttle prime contractor. These attitude angles differ from the attitude angles in MASTRE in both definition and order of rotation. Whereas the MASTRE program uses a pitch, yaw, and roll sequence in the transformation from the inertial system to the body axis, the prime contractor uses a yaw, pitch, and roll sequence. As shown on Figure 3-1, the yaw angle (ψ), or heading angle, defines the angle between the pitch plane and local north, the pitch angle (θ_s) is the angle between local horizontal and the vehicle longitudinal axis, and the roll angle (σ) is the angle between the vehicle Z axis and the pitch plane. These angles, which are input directly as tables, are converted to the attitude angles used by MASTRE. The equations required for this conversion are:

$$\chi_p = \tan^{-1}\left(\frac{Y_X}{Y_Y}\right)$$

$$\chi_Y + \sin^{-1}(Y_Z)$$

$$\chi_R = \tan^{-1}\left(\frac{-X_Z}{Z_Z}\right)$$

where

$$\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} = [\sigma]_2 \left[\theta_s - \frac{\pi}{2}\right]_3 [-\psi]_2 [\theta_L - \theta]_3 [A_{Z_L}]_2 \cdot$$

These angles are used from lift-off to the beginning of the min-H phase and are not structured to use parametric optimization.

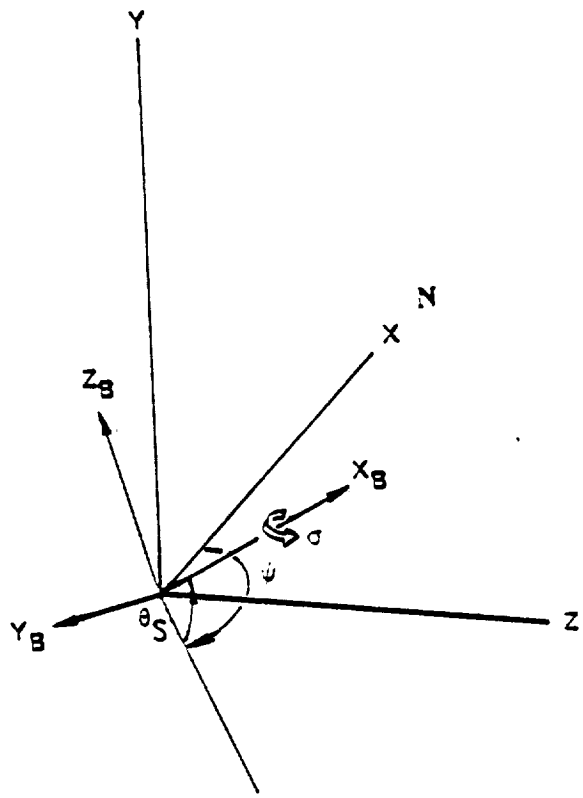


Figure 3-1. Prime Contractor Attitude Angle Orientation

3.2.7 Attitude Control During Min-H Phase

Following the booster simulation, the attitude of the vehicle can be determined by using the min-H technique. This method, used in conjunction with the steepest ascent algorithm, determines the optimum pitch or pitch and yaw attitude history which provides the maximization, or minimization, of some payoff function while enforcing intermediate and terminal constraints. The optimization criteria, based upon the minimization of the Hamiltonian along the trajectory, can be used for the total Orbiter trajectory.

The determination of the optimized trajectory is accomplished by an iteration process which involves the successive updating of an input attitude history. This updating is based upon the steepest ascent formulas presented in Section VI.

3.3 CONSTRAINT LIBRARY

The definition of the optimization problem requires the specification of a quantity to be maximized or minimized and a set of terminal or intermediate constraints to be satisfied. The MASTRE program has a library of 20 variables which can be treated (by choice of input) as either the payoff function of an intermediate or terminal constraint (Table 3-1). After selection of the payoff function, any physically realizable set of the remaining constraints may be selected and imposed at the terminal time or input-dependent intermediate times.

For the definition of the equations of the library functions as well as the partial derivatives with respect to the state variables, the reader should consult Appendix D.

3.4 CONTROL PARAMETERS

The steepest ascent optimization, in addition to determining the control attitude tables during the min-H phase, can also simultaneously determine other parameters. These are defined in Table 3-2.

Variables 29, 30, and 31 are used when maximum dynamic pressure is required as a constraint. These variables define a ramp function (see Figure 3-2) for the thrust level requirements of the MPS propulsion system. Intermediate times in this ramp are determined by a user supplied throttle rate.

Table 3-1. Payoff and Constraint Library

LIBRARY NO.	SYMBOL	FUNCTION NAME	UNITS
1	M_f	Cutoff Weight	kg
2	V_I	Inertial velocity	m/sec
3	γ_I	Inertial flight-path angle	deg
4	R	Radius	m
5		Not used	
6		Not used	
7	V_E	Earth fixed velocity	m/sec
8	A_{ZE}	Earth relative heading angle	deg
9	γ_E	Earth fixed flight-path angle	deg
10	i	Inclination	deg
11	θ_N	Inertial descending node from launch	deg
12		Not used	
13		Not used	
14	ΔV	Orbital velocity including oblate effects (Reference 8)	m/sec
15	Q_{MAX}	Maximum dynamic pressure	lb/ft^2
16	H	Stagnation heating indicator	btu
17		Not used	
18	ΔV_{RTLS}	Relative velocity increment from linear relationship between relative velocity and range (RTLS terminal condition)	m/sec
19	ΔV_A	Velocity increment from linear relationship between ET propellant V available and range (RTLS initiation condition)	m/sec
20	R_{NG}	Range from the launch site	NMi

Table 3-2. Parameter Library

LIBRARY NO.	PARAMETER NAME	SYMBOL
1-15	Burn time for i^{th} thrust event	τ_i
16	Liftoff weight	W_{L_0}
17	Launch azimuth	A_{Z_L}
18-24	Booster pitch attitude	xp_B
25-28	Booster yaw attitude	xy_B
29	Time to start MPS throttle ramp	t_s
30	Time after maximum dynamic pressure	t_e
31	Value of throttle at bottom of ramp	f
32-34	Values of sideslip angle history	β_i

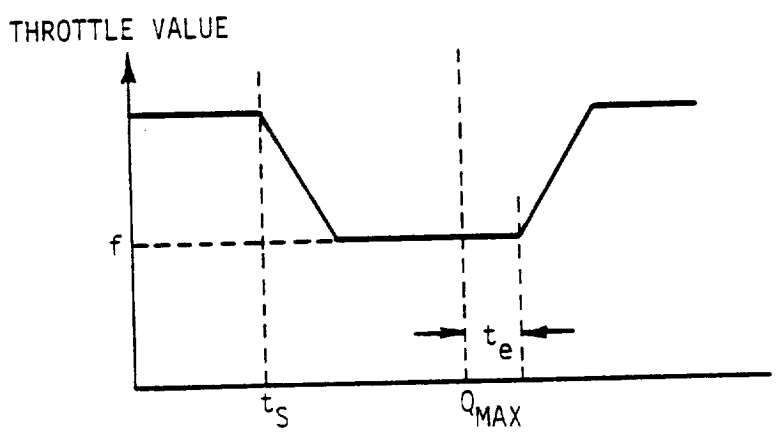


Figure 3-2. Throttle Optimization Parameters

Section 4

THE BACKWARD TRAJECTORY

During the min-H phase of the solution, influence coefficients are determined by the backward integration of the adjoint differential equations. The influence coefficients are used in the steepest ascent technique to determine the optimum set of control parameters based upon their relationship with respect to the active terminal and intermediate constraints. The construction of the adjoint equations requires one solution for each function being considered as either the payoff function or as a terminal or intermediate constraint. Since the adjoint equations can be determined numerically at the end point, the adjoint solutions proceed backward in time from the final time for the payoff and terminal constraints, and from the intermediate constraint times if applicable. The adjoint variables are also used to determine the impulse response functions which give the effect of changes in the payoff and constraints with respect to the pitch and yaw attitude angles.

The main difference between the min-H formulation and the regular steepest ascent formulation is the calculation of weighting matrix, W_a, which is used in the updating of the pitch and yaw attitude tables. The min-H formulation requires W_a to be a time-varying function which is composed of the second partials of the Hamiltonian function with respect to the pitch and yaw attitude angles. Using this weighting matrix allows the attitude history to follow an optimal path based upon the theory of calculus of variations. The reader should consult Reference 8 for a more detailed discussion of the min-H theory.

The notation traditionally used to describe the adjoint solution is introduced below:

- ϕ - The scalar payoff function
- ψ - an m x 1 matrix of constraints including both terminal and intermediate functions (the constraints are satisfied when $\psi = 0$)
- ν - an m x 1 matrix of constant Lagrangian multipliers associated with the constriants

Φ -The augmented scalar payoff function $\phi + \nu^T \psi$

λ_ϕ -A 7 x 1 matrix of particular adjoint solutions associated with the payoff function

λ_ψ -A 7 x m matrix particular adjoint solutions associated with the constraints

λ -A 7 x 1 matrix of adjoint solutions associated with the function Φ. When appearing without a subscript, λ is the equivalent of the Euler-Lagrange variables used in the calculus of variations and are formed as

$$\lambda = \lambda_\phi + \nu\lambda_\psi$$

The backward integration of the adjoint equations is essential since the boundary conditions at the terminal as well as the intermediate point are known. The boundary condition on the Euler-Lagrange variable, λ, are known by the equation

$$\lambda^T = \frac{\partial \Phi}{\partial p} \quad \text{(T refers to the transpose of the matrix } \lambda\text{)}$$

Therefore, the boundary conditions of λ_ϕ and λ_ψ are of the form

$$\lambda_\phi^T = \left.\frac{\partial \phi}{\partial p}\right|_{t = t_f}$$

and

$$\lambda_\psi^T = \left.\frac{\partial \psi}{\partial p}\right|_{t = \begin{cases} t_f, & \text{for terminal constraints} \\ t_i, & \text{for intermediate constraints} \end{cases}}$$

The partial derivatives of all functions which can be treated as the payoff function or constraints are given in Appendix E.

Defining the 7 x (m + 1) matrix $\psi_Z = [\psi_\Phi \vdots \lambda_\psi]$, the Euler-Lagrange or adjoint differential equations for the backward integration become

$$\dot{\lambda}_Z = F^T \lambda_Z$$

where the 7 x 7 matrix F contains the partial derivatives of the state equations of motion with respect to the state; therefore,

$$F^T = \left[\frac{\partial \dot{p}}{\partial p}\right]^T = \begin{bmatrix} \frac{\partial \dot{p}_1}{\partial W} & \frac{\partial \dot{p}_2}{\partial W} & \frac{\partial \dot{p}_3}{\partial W} & 1 & 0 & 0 & 0 \\ \frac{\partial \dot{p}_1}{\partial U} & \frac{\partial \dot{p}_2}{\partial U} & \frac{\partial \dot{p}_3}{\partial U} & 0 & 1 & 0 & 0 \\ \frac{\partial \dot{p}_1}{\partial V} & \frac{\partial \dot{p}_2}{\partial V} & \frac{\partial \dot{p}_3}{\partial V} & 0 & 0 & 1 & 0 \\ \frac{\partial \dot{p}_1}{\partial X} & \frac{\partial \dot{p}_2}{\partial X} & \frac{\partial \dot{p}_3}{\partial X} & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{p}_1}{\partial Y} & \frac{\partial \dot{p}_2}{\partial Y} & \frac{\partial \dot{p}_3}{\partial Y} & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{p}_1}{\partial Z} & \frac{\partial \dot{p}_2}{\partial Z} & \frac{\partial \dot{p}_3}{\partial Z} & 0 & 0 & 0 & 0 \\ \frac{\partial \dot{p}_1}{\partial M} & \frac{\partial \dot{p}_2}{\partial M} & \frac{\partial \dot{p}_3}{\partial M} & 0 & 0 & 0 & \frac{\partial \dot{p}_7^*}{\partial M} \end{bmatrix}$$

Since the MASTRE program includes the effects of the atmosphere in the adjoint equations the partial derivative equations are significantly more complex than with gravity and thrust alone. Rewriting the first three components of the equations of motion into a different form aids in the development of the partial derivative equations; thus,

$$\dot{\bar{p}} = \frac{1}{M}\left[\left(T_X - qSC_N\right)\hat{X} + \left(T_Y - qSC_A - F_{AB}\right)\hat{Y} + \left(T_Z + qSC_S\right)\hat{Z}\right] + \bar{G}$$

where the values in parentheses reflect the forces in the vehicle body system and the $\hat{X}$, $\hat{Y}$, $\hat{Z}$ unit vectors represent the direction cosines of the body axis in the reference coordinate system (launch plumbline). The general equations for the partial derivative equations become

---

*This term is used only when acceleration throttling is employed.

$$\frac{\partial \dot{\bar{D}}}{\partial \bar{W}} = \frac{1}{M}\left[\frac{\partial F_X}{\partial \bar{W}}\hat{X} + \frac{\partial F_Y}{\partial \bar{W}}\hat{Y} + \frac{\partial F_Z}{\partial \bar{W}}\hat{Z}\right]$$

$$\frac{\partial \dot{\bar{D}}}{\partial \bar{X}} = \frac{1}{M}\left[\frac{\partial F_X}{\partial \bar{X}}\hat{X} + \frac{\partial F_Y}{\partial \bar{X}}\hat{Y} + \frac{\partial F_Z}{\partial \bar{X}}\hat{Z}\right] - \frac{\partial}{\partial \bar{X}}\bar{G}$$

$$\frac{\partial \dot{\bar{D}}}{\partial M} = -\frac{1}{M^2}\left[F_X\hat{X} + F_Y\hat{Y} + F_Z\hat{Z}\right] + \frac{1}{M}\left[\frac{\partial F_X}{\partial M}\hat{X} + \frac{\partial F_Y}{\partial M}\hat{Y} + \frac{\partial F_Z}{\partial M}\hat{Z}\right]$$

$$\frac{\partial \dot{p}_7}{\partial M} = -a\left[\frac{I_{sp} - M\frac{\partial I_{sp}}{\partial M}}{I_{sp}^2}\right]\frac{\partial p_7}{\partial M} \quad \text{(used only when throttling)}$$

where

$\bar{W}$ = vector quantity representing W, U, and V

$\bar{X}$ = vector quantity representing, X, Y, and Z

M = instantaneous mass

F_X, F_Y, F_Z = vector representing thrust and aerodynamics forces in the body axis

$\bar{G}$ = gravity vector in the plumeline system

a = acceleration limit

I_{sp} = specific impulse

$$\frac{\partial I_{sp}}{\partial M} = \frac{g_o a}{T}(2a_1\tau + a_2)$$

τ - throttle level of MPS engines

T - thrust of MPS engines

a_1, a_2 - coefficients of MPS I_{sp} relationship.

The reader should consult Appendices E and F for more detailed descriptions of the partial derivatives.

During the backward integration, m + 1 sets of adjoint equations are integrated from the terminal point to the beginning of the thrust event which initiates the min-H optimization. During the integration, the plumbline state is reconstructed to calculate the F matrix by interpolation of stored values created in the forward trajectory.

The effect of the terminal and intermediate constraints with respect to the attitude control variables is monitored by defining the impulse response functions along the trajectory. The impulse response functions are of the form

$$G^T_{Z_p} = \frac{\partial\left(\lambda^T_Z \dot{p}\right)}{\partial X_p} = \lambda^T_Z \frac{\partial \dot{p}}{\partial X_p} \quad \text{for the pitch attitude}$$

$$G^T_{Z_Y} = \frac{\partial\left(\lambda^T_Z \dot{p}\right)}{\partial X_Y} = \lambda^T_Z \frac{\partial \dot{p}}{\partial X_Y} \quad \text{for the yaw attitude.}$$

The general equations for these partial derivatives are

$$\frac{\partial \dot{p}}{\partial X_p} = \frac{1}{M}\left[F_X \frac{\partial \hat{X}}{\partial X_p} + F_Y \frac{\partial \hat{Y}}{\partial X_p} + F_Z \frac{\partial \hat{Z}}{\partial X_p} + \frac{\partial F_X}{\partial X_p}\hat{X} + \frac{\partial F_Y}{\partial X_p}\hat{Y} + \frac{\partial F_Z}{\partial X_p}\hat{Z}\right]$$

and

$$\frac{\partial \dot{p}}{\partial X_Y} = \frac{1}{M}\left[F_X \frac{\partial \hat{X}}{\partial X_Y} + F_Y \frac{\partial \hat{Y}}{\partial X_Y} + F_Z \frac{\partial \hat{Z}}{\partial X_Y} + \frac{\partial F_X}{\partial X_Y}\hat{X} + \frac{\partial F_Y}{\partial X_Y}\hat{Y} + \frac{\partial F_Z}{\partial X_Y}\hat{Z}\right]$$

The reader should again consult Appendix F for the definition of the partial derivatives.

The product of the impulse functions, G_Z, which are denoted as

$$G_Z^T = \begin{bmatrix} G_{\partial_p} \\ \hline G_{\psi_p} \end{bmatrix} \quad \text{for pitch control only}$$

and

$$G_Z^T = \begin{bmatrix} G_{\partial_p} & G_{\partial_Y} \\ \hline G_{\psi_p} & G_{\psi_Y} \end{bmatrix} \quad \text{for pitch and yaw control}$$

and a time-varying weighting matrix, W_a^{-1}, is also calculated during the backward integration which is an $(m+1)$ by $(m+1)$ matrix of control variable integrals, so that

$$I_{ZZ}^a = \left[\begin{array}{c|c} I_{\partial\phi}^a & I_{\partial\psi}^a \\ \hline I_{\psi\phi}^a & I_{\psi\psi}^a \end{array}\right] = \int_{t_H}^{t_f} G_Z^T W_a^{-1} G_Z dt; \quad I_{ZZ}^a\left(t_f\right) = 0$$

If there are intermediate constraints, the above definition of I_{ZZ}^a may be used provided that, after the intermediate constraint time, the elements of G_Z corresponding to the intermediate constraints are zeroed.

Discussion of the formulation of time-varying weighting matrix, W_a^{-1}, which is also calculated during the backward trajectory, will be presented in Section VI.

Section 5

INFLUENCE COEFFICIENTS

A distinction is made between a control variable and a control parameter since a control variable refers to the attitude control determined during the min-H phase and a control parameter refers to all other parameters. Since the impulse response functions determine the partial derivative, or influence coefficients, of the payoff function and constraints with respect to the control variable, the influence coefficients of the control parameters are determined by either using numerical derivatives or by using analytical methods.

The numerical method requires the integration of the forward trajectory from the initial time, t_o, to the beginning of the min-H phase, t_H, varying the optimized parameter by some positive increment. Using the adjoint solution at that time (t_H) gives the influence coefficients of the payoff and constraints with respect to the state; the influence coefficients of the payoff and constraints with respect to the parameter are found by evaluating the change of state with respect to the parameter and using the chain rule. Thus, for any parameter, X, with a given increment, ΔX, the influence coefficient becomes

$$L_{Z_X} = \left[\frac{p^+ \left(t_H\right) - p^{NOM} \left(t_H\right)}{\Delta X} \right]^T \lambda_Z\left(t_H\right)$$

where p^+ and p^{NOM} refer to the plumbline state for the positive variation and the nominal, respectively. All parameters prior to the initiation of the min-H optimization, excluding launch azimuth and thrust event duration times, use this method for calculating the influence coefficients.

The influence coefficients for launch azimuth are determined by forward numerical differences which require the integration of the forward

trajectory to the terminal point based upon a given change in launch azimuth. This method is used since a variation in launch azimuth changes the orientation of the plumbline coordinates and thus the components are misaligned from the nominal trajectory. The influence coefficients for launch azimuth, A_Z, are determined by the equation:

$$L_{Z_{A_Z}} = \left[\frac{\phi^+ - \phi^{NOM}}{\Delta A_Z} \;\middle|\; \frac{\psi^+ - \psi^{NOM}}{\Delta A_Z} \right]$$

The influence coefficients for the thrust event duration time, τ_i, are calculated analytically based upon the change of the time derivatives of the state at the termination of each thrust event and during a weight drop event. At each thrust event and weight drop, one of the following equations is evaluated during the backward trajectory

$$L_{Z_{\tau}} = \Delta\dot{p}^T \lambda_Z \text{ for the thrust event}$$

and

$$L_{Z_{\tau_W}} = \Delta\dot{p}^T \lambda_Z \text{ for the weight drop}$$

where $\Delta\dot{p} = \left[\Delta\dot{p}^- - \dot{p}^+\right]$ is the discontinuity in the plumbline state derivatives.

When τ_i is a final time, $\dot{p}^+$ is set to zero. When τ_i is an intermediate constraint time, $\dot{p}^+$ is also set to zero, but only for those values of $\dot{p}$ pertinent to the multiplication of the column of λ_Z which correspond to the intermediate constraints.

Since the weight drop events are related to the thrust events by constant input time increments, the influence coefficients for the thrust events can be updated by the weight drop influence coefficients by addition; therefore,

$$L_{Z_{\tau_I}} = L_{Z_{\tau_i}} + L_{Z_{\tau_i}}$$ where $i = i^{th}$ thrust event and j = the number of weight drops in the ith event.

Also, in order for the duration times to be parameters, all times preceding a particular time parameter must reflect the change in a one-to-one relationship; therefore,

$$\frac{\partial t_j}{\partial \tau_i} = \frac{\partial t_i}{\partial \tau_i} = 1 \text{ for } j > i$$

where t_i and t_j reflect the time from time zero corresponding to the termination of the i and j thrust events. The influence coefficients for a given time, therefore, must reflect the additional influence coefficients which precede it;

$$L_{Z_{\tau_i}} = L_{Z_{\tau_i}} + \sum_{J=i+1}^{N} L_{Z_{\tau_j}}$$ where N is the total number of thrust events.

Grouping the control parameter influence coefficients into a matrix $L_Z = [L_\phi . L_\psi]$, the m+1 x m+1 parameter matrix I^b_{ZZ} can be formed as

$$I^b_{ZZ} = \begin{bmatrix} I^b_{\phi\phi} & I^b_{\phi\psi} \\ I^b_{\psi\phi} & I^b_{\psi\psi} \end{bmatrix} = L_Z^T W^{-1} L_Z$$

The weighting matrix, W_b, will be defined in Section VI.

Section 6

OPTIMIZATION METHODOLOGY

6.1 STEEPEST DESCENT FORMULATION

After calculating the influence coefficients, the steepest ascent formulae determine changes in the control parameters and variables which will simultaneously maximize (or minimize) the payoff function and enforce the terminal and intermediate constraints. Denoting the vector of active control parameters by b, and the vector of control variables by a, the changes in the controls are given by the formulae

$$\delta a = \pm W_a^{-1} \left(G_\phi - G_\psi \, I_{\psi\psi}^{-1} \, I_{\psi\phi} \right) E - W_a^{-1} \, G_\psi \, I_{\psi\psi}^{-1} \, k\psi$$

and

$$\delta b = \pm W_b^{-1} \left(L_\phi - L_\psi \, I_{\psi\psi}^{-1} \, I_{\psi\phi} \right) E - W_b^{-1} \, L_\psi \, I_{\psi\psi}^{-1} \, k\psi$$

where

$$I_{\psi\psi} = I_{\psi\psi}^{a} + I_{\psi\psi}^{b}$$

$$I_{\psi\phi} = I_{\psi\phi}^{a} + I_{\psi\phi}^{b} \; .$$

The plus or minus signs in the above equation denote that the payoff function ϕ is to be maximized or minimized, the constant $E(0 \leq E \leq 1)$ is chosen to aid convergence, the vector ψ reflects the violations of terminal or intermediate constraints, and k is a decimal fraction of the constraint violation to be removed. By limiting in the initial iterations the step taken by E and the amount of constraint to be removed by k, the probability of convergence and the stability of the scheme are enhanced.

The changes of the control parameters and variables are added to the present set and are used in the next forward trajectory. Changes in the payoff function resulting from new control can be reflected in the equation

$$d\phi = \pm \; I_{\phi\phi} - I_{\phi\psi} \, I_{\psi\psi}^{-1} \, I_{\psi\phi} \quad E - I_{\phi\psi} \, I_{\psi\psi}^{-1} \, k\psi$$

where

$$I_{\phi\phi} = I_{\phi\phi}^{a} + I_{\phi\phi}^{b}.$$

6.2 AUTOMATIC CONVERGENCE SCHEME

Determining the values of the parameters k, E, W_a^{-1}, and W_b^{-1} and defining a method for signifying convergence is the responsibility of the automatic convergence scheme. The logic for picking k and E is straightforward and is directly related to the iteration number. The first few iterations can be thought of as "restoration" steps since the value of E, the optimization coefficient, is set to zero. Unless otherwise specified by input, a starting value of k is set at 0.25. The second iteration uses a 0.5 for k, and the third iteration uses a full step by setting k = 1. By selecting a value of k in the region $0.5 \leq k \leq 1.$, the first and second iterations are eliminated and the third iteration procedure is used for the starting value. All iterations following the third iteration maintain k = 1. For the fourth iteration the optimization procedure starts by using one-half of the input value of E (preset value is 0.8). The fifth and subsequent iterations use the full value of E.

The choice of the weighting matrices W_a^{-1} and W_b^{-1} is also dependent on iteration number. On the first iteration, W_a^{-1} is chosen at each control variable time to be

$$W_a^{-1} = \frac{1}{\sum_{i=1}^{6} \frac{\partial^2 \dot{p}_i}{\partial \chi_p^2}} \qquad \text{for pitch only}$$

$$W_a^{-1} = \begin{bmatrix} \frac{1}{\sum_{i=1}^{6} \frac{\partial^2 \dot{p}_i}{\partial \chi_p^2}} & \frac{1}{\sum_{i=1}^{6} \frac{\partial^2 \dot{p}_i}{\partial \chi_Y \, \partial \chi_p}} \\ \frac{1}{\sum_{i=1}^{6} \frac{\partial^2 \dot{p}_i}{\partial \chi_p \, \partial \chi_Y}} & \frac{1}{\sum_{i=1}^{6} \frac{\partial^2 \dot{p}_i}{\partial \chi_Y^2}} \end{bmatrix} \qquad \text{for pitch and yaw}$$

and W_b^{-1} is chosen to be

$$W_b^{-1} = \begin{bmatrix} W_1 P_1 & 0 & \cdots & 0 \\ 0 & W_2 P_2 & \cdots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & W_n P_n \end{bmatrix}$$

where the $W_i (i = 1, n)$ is an input set of weighting numbers (preset equal to 1) for the n active parameters. On the first and subsequent iterations the P_i are chosen automatically so that the largest contribution of the ith parameter to the diagonal of $I^a_{\psi\psi}$ is equal to one. Denoting the influence coefficients of the ith parameter on the constraints by L^i_ψ, the $P_i{}^{th}$ scale factor is

$$P_i = \frac{1}{\max\limits_{j=1,n} \left(\frac{\left(L^i_{\psi_j}\right)^2}{I^a_{\psi_j \psi_j}} \right)}$$

For the second and third iterations the constant Lagrange multipliers on the constraints, υ, are defined by the equation

$$\upsilon = -I^{-1}_{\psi\psi} \, I_{\psi\psi}.$$

Thereafter, υ is formed by the equation

$$\upsilon = -I^{-1}_{\psi\psi} \, I_{\psi\phi} \mp I^{-1}_{\psi\psi} \, \psi$$

where the minus sign is used if maximizing and the plus sign is used if minimizing.

Once the Lagrange multipliers, υ, have been calculated, the min-H technique can begin on the optimization of the control variables. The Euler-Lagrange multipliers, λ, can now be formed as

$$\lambda = \lambda_{\phi} + \upsilon \lambda_{\psi}.$$

Along the trajectory the variational Hamiltonian H is found as

$$H = \lambda^T \dot{p} .$$

Through the min-H phase, the weighting matrix (W_a^{-1}) is set equal to the second partial derivative of the Hamiltonian with respect to control variables, therefore, at each point where the control variables are evaluated the following equation is also determined

$$W_a^{-1} = \mp H_{aa} = \mp \begin{bmatrix} \dfrac{\partial^2 H}{\partial X_p^{\,2}} & \dfrac{\partial^2 H}{\partial X_p \, \partial X_v} \\ \\ \dfrac{\partial^2 H}{\partial X_p \, \partial X_v} & \dfrac{\partial^2 H}{\partial X_v^{\,2}} \end{bmatrix}$$

where the minus sign is used if maximizing and the plus sign is used if minimizing.

The elements of H_{aa} are defined by the equations

$$\frac{\partial^2 H}{\partial X_p^{\,2}} = \lambda^T \frac{\partial^2 \dot{p}}{\partial X_p^{\,2}} = \lambda^T \frac{1}{M} \left\{ \left(\frac{\partial^2 F_X}{\partial X_p^{\,2}} - F_X \right) \hat{X} + \left(\frac{\partial^2 F_Y}{\partial X_p^{\,2}} - F_Y \right) \hat{Y} \right.$$

$$\left. + \left(\frac{\partial^2 F_Z}{\partial X_p^{\,2}} - F_Z \right) \hat{Z} + 2 \left(\frac{\partial F_X}{\partial X_p} \frac{\partial \hat{X}}{\partial X_p} + \frac{\partial F_Y}{\partial X_p} \frac{\partial \hat{Y}}{\partial X_p} + \frac{\partial F_Z}{\partial X_p} \frac{\partial \hat{Z}}{\partial X_p} \right) \right\}$$

$$\frac{\partial^2 H}{\partial x_p \, \partial x_Y} = \frac{\partial^2 H}{\partial x_Y \, \partial x_p} = \lambda^T \frac{\partial^2 \dot{p}}{\partial x_p \, \partial x_Y} = \lambda^T \frac{1}{m} \left\{ \frac{\partial^2 F_X}{\partial x_p \, \partial x_Y} \hat{X} + \frac{\partial^2 F_Y}{\partial x_p \, \partial x_Y} \hat{Y} \right.$$

$$+ \frac{\partial^2 F_Z}{\partial x_p \, \partial x_Y} \hat{Z} + \frac{\partial F_Y}{\partial x_p} \frac{\partial \hat{Y}}{\partial x_Y} + \frac{\partial F_Z}{\partial x_p} \frac{\partial \hat{Z}}{\partial x_Y} + \frac{\partial F_X}{\partial x_Y} \frac{\partial \hat{X}}{\partial x_p} + \frac{\partial F_Y}{\partial x_Y} \frac{\partial \hat{Y}}{\partial x_p}$$

$$\left. + \frac{\partial F_Z}{\partial x_Y} \frac{\partial \hat{Z}}{\partial x_p} + F_Y \frac{\partial^2 \hat{Y}}{\partial x_p \, \partial x_Y} + F_Z \frac{\partial^2 \hat{Z}}{\partial x_p \, \partial x_Y} \right\}$$

$$\frac{\partial^2 H}{\partial x_Y^2} = \lambda^T \frac{\partial^2 \dot{p}}{\partial x_Y^2} = \lambda^T \frac{1}{m} \left\{ \frac{\partial^2 F_X}{\partial x_Y^2} \hat{X} + \left(\frac{\partial^2 F_Y}{\partial x_Y^2} - F_Y \right) \hat{Y} + \left(\frac{\partial^2 F_Z}{\partial x_Y^2} - F_Z \right) \hat{Z} \right.$$

$$\left. + 2 \left(\frac{\partial F_Y}{\partial x_Y} \frac{\partial \hat{Y}}{\partial x_Y} + \frac{\partial F_Z}{\partial x_Y} \frac{\partial \hat{Z}}{\partial x_Y} \right) \right\}$$

More detailed explanations of the partial derivatives are given in Appendix F.

On each iteration a normalized total influence coefficient for each parameter L_i is formed as

$$L_i = \frac{L_{\phi_i} + \nu \, L_{\psi_i}}{\max\limits_{j=1,n} \left(\left| L_{\phi_i} \right|, \left| \upsilon_j \, L_{\psi_{j_i}} \right| \right)} \quad .$$

Prior to the sixth iteration, the input values of W_i are used in the construction of W_b^{-1}. For the sixth and subsequent iterations, each W_i is altered according to the following logic

W_i **unchanged if** $\left| L_i \right| < .005$

W_i **unchanged if** $\left| L_{i_{\text{present}}} \right| < \left(1 - \frac{E}{2} \right) \left| L_{i_{\text{last}}} \right|$

otherwise

$$W_i = 2\,W_i \text{ if } \left|L_{i_{present}} - L_{i_{last}}\right| < \left|L_{i_{present}}\right|$$

$$W_i = W_i/2 \text{ if } \left|L_{i_{present}} - L_{i_{last}}\right| \geq \left|L_{i_{present}}\right| .$$

This dynamic updating of W_i will generally ensure smooth convergence of the parameters. The relative magnitude of the W_i on a converged run can be used as a guide in picking input values of W_i for a similar trajectory and more rapid convergence.

Convergence of the trajectory is determined by a sequence of tests on the control parameters, the control variables, the total influence coefficients, and the iteration number. The tests are:

- The number of iterations has to exceed five
- The absolute value of the total influence coefficients for the ith parameter, $|L_i| < 0.005$.
- The change in a thrust event duration time, $|\Delta\tau| \leq 0.5$
- The change in liftoff weight, $|\Delta W_{LO}| \leq 100.$ kg
- The change in launch azimuth, $|\Delta A_z| \leq .005\,|A_{z_{i-1}}|$ (i-1 denotes the previous value)
- The change in booster pitch attitude, $|\Delta\chi_P| \leq .01\,|\chi_{P_{i-1}}|$ } used in the attitude polynominals in the booster
- The change in booster yaw attitude, $|\Delta_{\chi Y}| \leq .01\,|\chi_{Y_{i-1}}|$ } used in the attitude polynominals in the booster
- The change in throttle parameters, $|\Delta t_Y| \leq .01\,|t_{i-1}|$

  $|\Delta f_i| \leq .01\,|f_{i-1}|$
- The change in beta history parameters, $|\Delta\beta_j| \leq .01\,|\beta_{j-1}|$

- The test for the control variables is

$$\left.\begin{array}{l} \left|d_{\chi_P}\right|_{max} \\ \text{or} \\ \left|d_{\chi_Y}\right|_{max} \end{array}\right\} = \left|H_{aa}^{-1} H_a\right| < .005$$

where $\left|d_{\chi P}\right|_{max}$ and $\left|d_{\chi Y}\right|_{max}$ are the maximum increments in pitch and yaw, and H_a is the matrix of the first partial derivatives of the Hamiltonian with respect to the control variables.

If any test does not meet the tolerances, the trajectory has not converged and the iteration process will continue. If all tests meet the tolerances, the converged trajectory is printed and the case is terminated. In most cases, the tolerances for the test are preset and cannot be changed by input.

Section 7

REFERENCES

1. Lyons, J. T., "Minimum Hamiltonian Ascent Shuttle Trajectory Evaluation Program (MASTRE) - Engineering Manual," Northrop Services, Inc. Technical Report No. TR-275-1628, June 1976.

2. Lyons, J. T., Woltsz, W. S., Abercrombie, G. E., and Gottlieb, R. G., "Rocket Ascent G-Limited Moment Balanced Optimization Program (RAGMOP)," NASA CR-129000, August 1972.

3. Gottlieb, R. G., "ROBOT-Apollo and AAP Preliminary Mission Profile Program, Part I - Mathematical Formulation," NASA CR-61223, May 1968.

4. Gottlieb, R. G., "Lifting ROBOT - Space Shuttle Preliminary Mission Profile Optimization Program," Dynamics Research Corp., Report R-120U, May 29, 1971.

5. "Natural Environments and Physical Standards for the Apollo Program and the Apollo Applications Program," Office of Manned Space Flight, M-DE 8020.008C, SE 015-011-1B, Washington, D.C., July 1969.

6. Smith, O. E., and Weidner, D. K., "A Reference Atmosphere for Patrick AFB, Florida, Annual (1963) Revision," NASA TM-X-53139, September 1964.

7. Carter, E. A., and Brown, F. C., "A Reference Atmosphere for Vandenberg AFB, California, Annual (1971) Version," May 10, 1974, NASA TMX-64590.

8. Whipple, P. H., "A Modified Perigee and Apogee Velocity Expression to Reduce Radius Errors Caused by the Earth's Asphericity," TM-69-1025-1, Bellcomm, Inc., July 14, 1969.

9. Gottlieb, R. G., "Rapid Convergence to Optimum Solutions Using a Min-H Strategy," AIAA J., Vol. 5, No. 2, February 1967.

10. Daniels, G. E., "Terrestrial Environment (Climatic) Criteria Guidelines for Use in Aerospace Vehicle Development, 1973 Revision," July 5, 1973, NASA TMX-64757.

11. Johnson, D. L., "Hot, Cold, and Annual Reference Atmospheres for Edwards Air Force Base, California (1974 Version)," July 1975, NASA TMX-64941.

12. "Users' Guide for ASCII Marshall Interactive Planning System (MIPS) 3.0", John Jaap and Ron Hall, Document is resident on the MSFC Computer System.

Section 8

USERS' MANUAL

This manual has been prepared to aid the users of the Minimum Hamiltonian Ascent Shuttle Trajectory Evaluation (MASTRE) program. Presented herein is general information concerning the inputs required for simulating various trajectory problems, detailed descriptions of the input variables and the input format. Program output options and associated formats are discussed in the context of a specific example.

Section 9

IMPLEMENTATION OF TRAJECTORY PROBLEMS ON THE MASTRE PROGRAM

The implementation of a trajectory problem on the MASTRE program necessitates that the user understand the basic input mechanics and be aware of the various available options. Through proper option selection, the MASTRE program can be used to simulate a variety of trajectory problems varying from simple point mass trajectories to complex atmospheric moment balanced trajectories. This flexibility stresses the importance of understanding the available options. This section contains a discussion of the program options and explains the methods of setting up a trajectory problem.

9.1 THRUST EVENTS

The trajectory profile is simulated and setup in the MASTRE program as a succession of up to 15 thrust events of which one or more describe a complete physical stage. A thrust event is a period of time characterized by a continuous thrust profile (constant, zero, or varying in a continuous fashion). Each thrust event is initiated at the termination of a previous event except, of course, for the first event which initiates the trajectory.

Figure 9-1 shows a typical thrust event profile for the space shuttle. In this example, the first stage (orbiter and booster) has three thrust events and the second stage (orbiter alone) has five thrust events. These times, as shown by the figure, reflect major events in the trajectory simulation. The uncircled numbers represent the thrust events and the circled numbers are referred to as "picket" numbers. The "picket" number is analogous to a picket fence in that there is always one more picket than there are boards. A great deal of generality is gained by allowing many of the input values to be dependent on either the number of the thrust event or the picket, thus reducing the number of time dependent inputs.

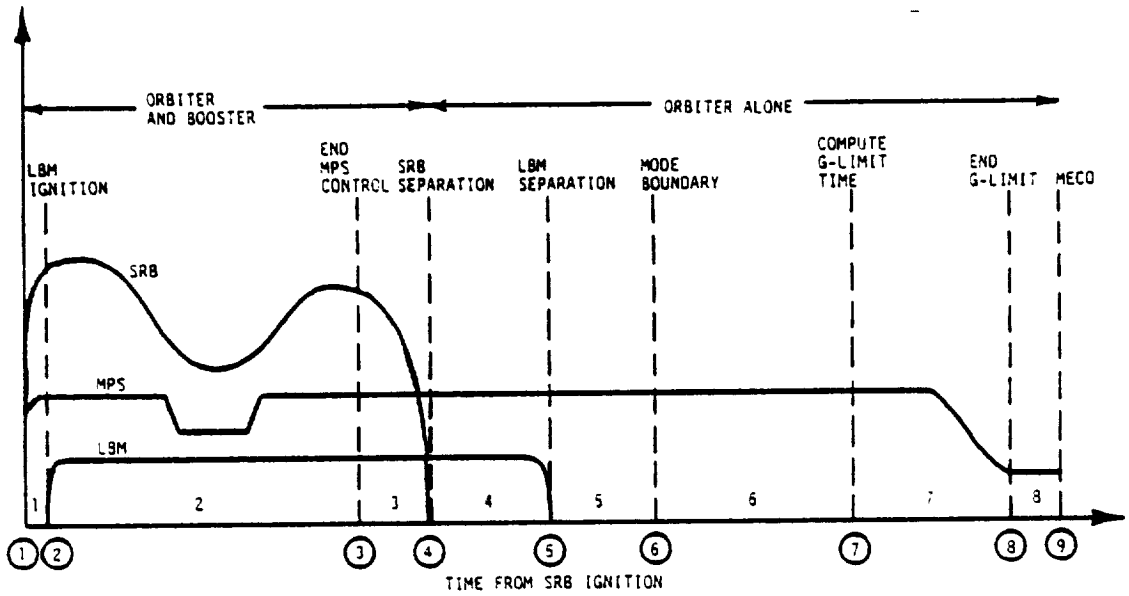


Figure 9-1. Typical Thrust Event Profile

Input associated with the thrust event or picket describe in detail the characteristics of a particular period of time. However, since some data are applicable to more than one thrust event, a distinction is made between data associated with the entire trajectory, stage dependent data, and thrust event related data. Table 9-1 provides a breakdown of the input variables into these input classifications.

9.2 ATTITUDE CONTROL OPTIONS

The complexity of the first stage aerodynamic model prevents the use of Min-H optimization scheme in the determination of the first stage attitude. Therefore, additional attitude control options are available to use during simulation of the first stage (of course, these options are not restrictive to the first stage and can be used whenever a particular orientation is required; i.e., the turnaround maneuver during the return

Table 9-1. Data Grouping

DATA ASSOCIATED WITH:

ENTIRE TRAJECTORY	STAGE	THRUST EVENT
Lift-Off Weight	Aerodynamic Coefficient Tables	Vacuum Thrust Level **
Flight Azimuth	Base Force Tables	Mass Flowrate**
Longitude of Launch	Center-of-Gravity Tables	Acceleration Limit
Latitude of Launch	Aerodynamic Reference Length	Number of Engines
Engine Data	Attitude Polynominal Order	Throttle Option Flag (MPS only)
Wind Tables	Attitude Angle Tables	Jettison Weight***
Optimized Parameter Selection Flag	Number of Thrust Events per Stage	Drop Weight***
Constraint Definition Codes (see Subsection 9.5)	Control Option Switch	Time Duration of Event
Initial State Requirements (if Jump Start)	Aerodynamic Moment Reference Point Locations	Time from Picket to Weight Drop Event
MPS Reference Specific Impulse Time	Solid Rocket Booster (SRB) Data*	Number of Weight Drops per Thrust Event
Liquid Boost Module (LBM) Data	Angle-of-Attack Data*	Integration Step Size for Forward Integration
Initial Time	Time of Vertical Rise Termination*	Integration Step Size for Backward Integration
Jump Start Flag Case No.	Time of Roll Maneuver Initiation*	Print Interval
Integration Selection Flag (Backward and Forward)	Time of Tilt-Over Maneuver Termination*	MPS Throttle
Postprocessor Option Flag		Aerodynamic Reference Area
Print Option Flag		
Attitude Control Option Flag		
Optimization Parameters		
Earth Model Parameters		
Integration Limits		
Weighting Factors on Parameters		

*Booster Stage Only

**OMS and RCS Engines Only

***Jettison refers to weight dropped only at the end of a thrust event, whereas drop weight can be dropped at any time.

to launch site (RTLS) mission). The following three attitude control options are available:

- Attitude control based on time dependent n^{th} order polynominal equations.
- Attitude control based on Mach number dependent tables of angle of attack and sideslip angle.
- Attitude control based on attitude angle tables used by the Space Shuttle prime contractor, Boost Reference Coordinate System, [pitch (θ) and yaw (ψ) angles are functions of earth-fixed velocity and roll (σ) angle is a function of time from launch].

Using the second method, a near optimum attitude profile can be attained for either a portion of or the entire (excluding the lift-off sequence) booster trajectory. Optimization requires the parametric variation of the angles used in the definition of the polynominal expression(s). Use of the first or third method precludes optimization of the attitude history. The input variable 'IFACT' is used as a flag to indicate which option is desired.

9.2.1 Angle of Attack/Sideslip Angle Control

Angle of attack/sideslip angle control (IFACT = 2) requires that the booster trajectory be subdivided into a minimum of three attitude phases:

1. A vertical rise phase where attitude is determined referenced to the Earth model.
2. A tilt over maneuver consisting of attitude polynominals (in most cases linear pitch and yaw functions) which are used to force the relative velocity into the desired flight plane. (This portion of the trajectory can be optimized by the variation of the angles used to define the polynominals).
3. Continuation of the booster simulation based on the angle of attack/sideslip angle history.

The first phase requires only the input of the end time of the vertical rise maneuver (represented as the input mnemonic TLIFT) and requires no additional input. The second phase begins at the end of the vertical

rise manuever and ends at the time TTILT. Other inputs for this phase are described in Paragraph 9.2.2. The third phase begins at TTILT and terminates at either booster separation or prior to a thrust event specified by the input flag 'IPOLY' (see Paragraph 9.2.2). During this phase, the angle of attack table is input by the array TALFP, the sideslip table is input by the array TBETA, and the Mach number table is input by the array TMACH.

9.2.2 Attitude Polynominal Control

Two sets of attitude polynominals of the form

$$X = a_0 + a_2 t + a_2 t^2 + \ldots + a_n t^n$$

are available for a variety of uses. As discussed in the previous paragraph, the tiltover maneuver (phase 2) uses a polynominal expression, in most cases linear, to describe the attitude history during this maneuver. Input requirements for this maneuver are the end of the tiltover time (TTILT), the order of the pitch and yaw polynominal expressions (input as the first value of KP and KY, respectively), values of pitch and yaw attitude angles which define the initial guess for the polynominal expression (CPTBL and CYTBL, respectively), and table values of incremental time from the beginning of the tiltover maneuver; (TTBL(1-15) corresponds to the pitch values and TTBL(16-30) corresponds to the yaw values). Continuity of the attitude angles between the vertical rise maneuver and the tiltover maneuver is maintained by internally setting the first values of CPTBL and CYTBL to the current values.

Optimization of the tiltover maneuver polynominal expression (except for the first values CPTBL(1) and CYTBL(1)) can be obtained by setting corresponding values of the parameter optimization indicator, KDB, to one (see Paragraph 9.5.2). For user convenience, the method of parameterizing the attitude angles in the construction of the polynominal expression is used rather than the polynominal coefficients.

A value of IFACT equal to one indicates that booster attitude following the vertical rise phase will be determined via an nth order polynominal expression. Input for this option is the same as for the tiltover maneuver with the exception of the tiltover termination time (TTILT). Continuity of the attitude profile is again maintained by restricting the first values of the attitude angle tables to be equal to the terminal vertical rise attitude. Optimization of these polynominals is handled in the same manner as the tiltover maneuver polynominal.

During the latter part of the booster trajectory or prior to the beginning of the min-H optimization, additional pitch and yaw polynominal expressions can be used in conjunction with the attitude control as specified by IFACT. The input variable IPOLY is used to indicate an additional expression starting at the beginning of the IPOLY thrust event. The order of the polynominals are input as the second values of the input variables KP and KY. Placement of the input in the attitude angle and time tables is based on the first value of the KP and KY array such that the pitch attitude angles are CPTBL (KP(1) + 2) through CPTBL (10), the yaw attitude angles are CYTBL (KY(1) + 2) through CYTBL (15), the time tables are TTBL (KP(1) + 2) through TTBL (15) for pitch and TTBL (KY(1) + 15) through TTBL (30) for yaw. These polynominals can be optimized in the same manner as before by proper selection of the KDB array. For this option, attitude continuity between the beginning of this attitude control and the ending of the previous method can be maintained by placing a zero in the KDB array corresponding to the first attitude angle of the additional polynominal. However, continuity will only be maintained if the polynominal is to be optimized (therefore, if the corresponding KDB array for both the first and second attitude angles are zero, continuity will not be enforced).

9.2.3 Attitude Control Based Upon the Angles Theta (θ), Psi (ψ), and Sigma (σ)

A value of IFACT = 3 implies that the attitude control from lift-off to booster separation will be a function of the Boost Reference Coordinate System attitude angles theta (θ) (pitch), psi (ψ) (yaw), and sigma (σ)

(roll). The angles theta and psi are input as tabular functions of earth-fixed velocity and the sigma angle is input as a tabular function of time from launch. The input mnemonics are CPTBL (1-30) for the theta angle, CYTBL (1-30) for the psi angle, and SIGTAB (1-20) for the sigma angle. The independent earth-fixed velocity table is input TTBL (1-30) and the independent time table is TTABLE (1-30). When using this option neither the vertical rise maneuver nor the tiltover maneuver are required. No optimization parameters are available when using this option.

9.2.4 Attitude Control Using the Min-H Optimization Technique

Any time following first stage the min-H optimization technique can be used to define the vehicle attitude. Initiation of this method is denoted by the input variable "MINH" which indicates the thrust event that min-H optimization will begin. The attitude history during the min-H phase can be divided into three independent segments of which each segment is specified by the initiation and terminal picket number (NBGCT and NENDCT, respectively), the number of table points in the segment (NP), and the table values of time and attitude angles. During this phase the user can select either in-plane (pitch plane only) or pitch and yaw control by setting the input values of KWTA equal to 2 or 3 respectively for each segment. For example, if pitch and yaw min-H optimization is used on the trajectory presented in Figure 9-1, the input would be: MINH = 4, KWTA = 3, 3, 3, NBGCT = 4, 5, 6, and NENDCT = 5, 6, 9. The number of points in each attitude angle-versus-time table for each segment cannot exceed 29 points and must be an odd number of points (>1) due to the use of Simpson's Rule in the evaluation of the backward trajectory. Referring to Figure 9-1, if the number of table points in the first burn is 15 and 29 points are required for the second and third segments, the input would be: NP = 15, 29, 29.

Values of the independent time table and dependent pitch and yaw attitude tables are input through the arrays TOBL, CPOTBL, and CYOTBL. For each segment there are 29 points available to be used where TOBL, CPOTBL, CYOTBL (1-29) are for the first segment, TOBL, CPOTBL, CYOTBL (31-59) are for the second segment, and TOBL, CPOTBL, CYOTBL (61-89) are for the third segment. Although the user may require up to 29 points for a given segment, all 29 points in the attitude angle array are not required as input. The reason is that the program, by linear interpolation, initially determines the attitude values at discrete points based upon the estimated values given in the input regardless of the number of points input. After the initial iteration, however, the optimization algorithm continually alters both the independent and dependent tables and stores the required number (NP) of updated variables. For the above example, the input would be of the form:

TOBL = 123.77, 208., CPOTBL = 60., 70., CYOTBL = 5., 6.,
TOBL(31) = 208., 265., CPOTBL(31) = 70., 85., CYOTBL(31) = 6., 7.,
TOBL(61) = 265., 480., CPOTBL(61) = 85., 105., CYOTBL(61) = 7., 10.,.

After the first iteration, the above linear tables are adjusted and stored according to the number of points specified by the input array NP.

9.2.5 Post Launch Roll Maneuver

All of the booster attitude options, except for the θ, ψ, and σ option (IFACT = 3.), can use a roll maneuver which simulates the post-launch rotation of the vehicle so that the orbiter tail fin is in the flight plane. This maneuver begins at the input time TCHIR and ends at a time defined by the angular requirement (the angle between the initial placement, defined by the input value FAZ, and final placement of the orbiter tail fin) divided by the roll rate (CHRDOT). If this maneuver is not required, the value of CHRDOT should be zero. At the termination of this maneuver, the orbiter tail fin is pointed in either a local up or down position based upon the value of the flag IHEAD (IHEAD = 0, up; IHEAD = 1, down).

9.3 INPUT REQUIREMENTS FOR THE EQUATIONS OF MOTION

Although the MASTRE program provides a three degree of-freedom simulation, the aerodynamic, thrust, and gravity modeling reflects the complex equations of motion of the Space Shuttle. This paragraph provides an overview of the input requirements for these models.

9.3.1 Aerodynamic Model Input

The aerodynamic variation between the orbiter-ET configuration and booster configuration have forced the use of two sets of aerodynamic input. Because of the large amount of data, the program uses the automated aerodynamic data base for the majority of the aerodynamic input (see Section 10). This feature minimizes data handling by the user and allows current access to the latest aerodynamic data. The Namelist array AEROIN can also be used for aerodynamic input. Since all variables in the AEROIN array are on the aerodynamic data base, all data input via the Namelist array will update the data base information. The only exception to this is the first stage aerodynamic coefficient tables which are not in the AEROIN array and therefore cannot be altered by user input. The input mnemonics, the array sizes, input variables descriptions, and preset values are discussed in Section 10.

9.3.2 Thrust Model Input

Six propulsion systems can be simulated in the MASTRE program. These systems are compatible with either current or future Space Shuttle propulsion systems. Current systems are the Solid Rocket Motors (SRM), the Main Propulsion System (MPS), the Orbital Maneuver System (OMS), and the Reaction Control System (RCS). The Liquid Booster Module (LBM) is a future system designed to augment the booster configuration thus providing an increase in the Space Shuttle payload-to-orbit capability. To accommodate future propulsion systems, additional space has been reserved to facilitate coding.

An attempt has been made in the programming of these propulsion systems to generalize the input requirements but because of the uniqueness of each system, a totally general input was impossible. However, four input variable arrays do define, in a general nature, certain information concerning all propulsion systems. These input arrays are ENGDAT, NSYST, TNE and PROP.

The variable ENGDAT is defined as a double dimensional array, ENGDAT(8, 15), where the first dimension consists of eight data items associated with a particular propulsion system and the last dimension allows up to 15 propulsion systems. The first eight data items are defined as follows:

ENGDAT(1,I) = longitudinal location (X coordinate) of engine i
ENGDAT(2,I) = lateral location (Y coordinate) of engine i
ENGDAT(3,I) = vertical location (Z coordinate) of engine i
ENGDAT(4,I) = pitch cant angle of engine i
ENGDAT(5,I) = yaw cant angle of engine i
ENGDAT(6,I) = vacuum thrust of engine i at 100% (reference) power level
ENGDAT(7,I) = propellant flowrate of engine i at 100% (reference) power level
ENGDAT(8,I) = nozzle exit area of engine i.

The integer array NSYST relates the Space Shuttle propulsion systems to the ENGDAT input. The values of NSYST represent:

1 - MPS
2 - SRM
3 - LBM
4 - OMS
5 - RCS
6 - (open for future system).

The fifteen locations of the NSYST array have a one-to-one matching with the fifteen locations of ENGDAT.

The TNE variable array is a double dimensioned array, TNE(6,15), where the first dimension relates to the six propulsion systems and the latter dimensions to the thrust event. The array is used to specify the number of engines of that type that will be used. For example, an input of TNE(1,3) = 3., 2., 2., 0., 0., 0., indicates that in the third thrust event three MPSs, two SRMs, and two LBMs will be simulated. The number used in the TNE array should be less than or equal to the number of propulsion systems of that type as indicated by the NSYST and ENGDAT arrays. The PROP input array defines the amount of propellant allotted to each system. This array allows for the integration and printout of the separate weight time history of each system.

Section 10 defines the input requirements for all propulsion systems and in most cases is straightforward. The variation in MPS propulsion system input, however, might be confusing to the user. The variable array MPSFLG is a thrust event dependent flag used for denoting the type of MPS throttling history to be used. A value of one indicates that the throttle history will be constant during the thrust event. Associated with constant throttle is the input array THROT.

A MPSFLG value of two indicates that the throttle history will vary, in a linear fashion, based on the tabular arrays TIMMPS and FTBL. TIMMPS is the time from SRB ignition and FTBL is the corresponding thrust throttle value (109% is input as 1.09).

A MPSFLG value of four (MPSFLG = 3 is currently not available) indicates that an acceleration limit is anticipated and will be simulated whenever the total acceleration reaches the specified limit. The acceleration limit is input by the thrust event dependent array GLIM. Before the acceleration limit is reached, the input array THROT is used to specify the throttle level.

There are several MPS inputs which are used regardless of the value of MPSFLG. These are TAUALT and TAUTBL which are tables used to determine altitude dependent levels of minimum throttle; TAULIM, TIMTAU, and

NTAU which are used in the throttle optimization; and TIMISP and TRFISP which are tables used to define the specific impulse as a function of time from SRB ignition for the 100% or reference power level.

9.3.3 Gravitation Model Input

All gravitation model input is preset by the program, but can be changed via the namelist 'INPUT' array. Section 10 describes the input parameters.

9.4 INTEGRATION SCHEMES

The MASTRE program has three integration methods available which are selected by the input variable KIND. If KIND = 1, an Adams-Moulton fourth-order scheme is used which has the ability to select its own stepsize. However, the initialization of this integration requires four Runge-Kutta steps using a fixed stepsize. These stepsizes are input by the user as STEP; therefore, the user should carefully select the value of STEP. The user should avoid the use of small print intervals since maximum stepsize is controlled by the print interval, input PRINT. PRINT should be even multiples of STEP. The minimum stepsize is based upon the input variable HMN.

If KIND = 2 or 3, a fixed stepsize fourth-order method is used with the stepsize based on the value of the input array STEP. A Runge-Kutta integration is used when KIND = 2, and fixed step Adams-Moulton integration is used when KIND = 3.

9.5 OPTIMIZATION METHODOLOGY

The flow logic of the MASTRE program assumes that every problem simulated will determine the maximum or minimum of some payoff function subject to a set of terminal and intermediate constraints by the optimization of the attitude angles during the min-H phase and the optimization of certain trajectory parameters. The user can select a variety of constraints and parameters by proper selection of input flags.

9.5.1 Selection of the Payoff Function and the Terminal and Intermediate Constraints

A large library of parameters can be used either as the payoff function, terminal constraints, or intermediate constraints (see Table 9-2). The user specifies which parameters will be used by placing the library code number in the input arrays KCDPHI and KCDRES. The first value of the KCDPHI array corresponds to the payoff function (KCDPHI(1)) and latter values (KCDPHI(2-10)) correspond to terminal constraints. In conjunction with KCDPHI(2-10) the required values of the constraints are input into the array PSIREQ(1-9).

The input array KCDRES is used to identify intermediate constraints. These intermediate constraints are enforced at the termination of the thrust events specified by the array NVRST. Corresponding to KCDRES, the required values of the constraints are placed in PSIRST. A maximum of four intermediate constraints are allowed.

The following information should be noted by the user when using the constraints of Table 9-2:

- If the orbital velocity constraint (code number 14) is used, radius (code number 4) must be included as an additional constraint and the value of PSIREQ or PSIRST for the orbital velocity constraint must reflect the radius 180 degrees away from initial orbit (to specify the orbit requirements). After initial calculations, the value of PSIREQ or PSIRST is set to zero.

- The linear relationships in constraint codes 18 and 19 are defined by the input VRINT and VSLOPE. VRINT is the zero range intercept and VSLOPE is the slope of the desired curve. In both constraints a value of zero in PSIREQ or PSIRST indicates that the solutions on the constraint curve (VRINT(1) and VSLOPE(1)) are for terminal constraints and VRINT(2) and VSLOPE(2) are for intermediate constraints.

9.5.2 Selection of the Control Parameters

In addition to the min-H optimization of the attitude control, several trajectory parameters can be optimized using the steepest ascent algorithm. The specification of these parameters is input by using the

Table 9-2. Constraint Codes

LIBRARY NO.	SYMBOL	FUNCTION NAME	UNITS
1	M_f	Cutoff Weight	kg
2	V_I	Inertial velocity	m/sec
3	γ_I	Inertial flight-path angle	deg
4	R	Radius	m
5		Not used	
6		Not used	
7	V_E	Earth fixed velocity	m/sec
8	A_{ZE}	Earth relative heading angle	deg
9	γ_E	Earth fixed flight-path angle	deg
10	i	Inclination	deg
11	θ_N	Inertial descending node from launch	deg
12		Not used	
13		Not used	
14	ΔV	Orbital velocity including oblate effects (Reference 8)	m/sec
15	Q_{MAX}	Maximum dynamic pressure	lb/ft^2
16	H	Stagnation heating indicator	btu
17		Not used	
18	ΔV_{RTLS}	Relative velocity increment from linear relationship between relative velocity and range (RTLS terminal condition)	m/sec
19	ΔV_A	Velocity increment from linear relationship between ET propellant V available and range (RTLS initiation condition)	m/sec
20	R_{NG}	Range from the launch site	NMi

locations of the input array KDB. Each non-zero location of this array signifies to the algorithm that a corresponding parameter will be optimized. Table 9-3 shows the parameters assigned to the KDB array.

Table 9-3. Optimization Codes

LIBRARY NO.	PARAMETER NAME	SYMBOL
1-15	Burn time for i^{th} thrust event	τ_i
16	Liftoff weight	W_{L_0}
17	Launch azimuth	A_{Z_L}
18-24	Booster pitch attitude	XP_B
25-28	Booster yaw attitude	XY_B
29	Time to start MPS throttle ramp	t_s
30	Time after maximum dynamic pressure	t_e
31	Value of throttle at bottom of ramp	f
32-34	Values of sideslip angle history	β_i

9.6 SIMULATION OPTIONS

The following paragraphs discuss several simulation options available to the user.

9.6.1 Jump-Start Option

Reinitialization of the orbiter trajectory at the beginning of a given thrust event is signified by the input variable JUMP. Nominally, the value of JUMP will be equal to one which specifies that the trajectory will be initialized on the ground. In this case, the normal ground-to-insertion logic will be maintained. No additional state vector input is required except geodetic latitude (ALAT), longitude (ALONG), altitude (ALTLS), and launch azimuth (AZ).

If JUMP > 1, the trajectory will be initiated at the beginning of the thrust event specified by JUMP. In this event, the additional input array VIV, the starting time TZERO, and the starting weight W01 must reflect the state. The trajectory cannot be jump-started in the booster phase.

A special option using the jump-start logic has been coded to implement the current Space Shuttle shaping philosophy. This involves shaping an initial trajectory to meet pseudo shaping constraints and the determining the abort-once-around (AOA) trajectory, the nominal trajectory, and the return-to-launch-site (RTLS) trajectory from a point on the initial trajectory referred to as the mode boundary point. The shaping of these trajectories can be accomplished in one run by using the input variables ISTART and JSTART. The variable ISTART signifies to the program that state conditions including time and weight will be stored at the beginning of thrust events ISTART(1) and ISTART(2). The variable JSTART indicates the value of the thrust event in ISTART(1) or ISTART(2) which will be used in the next case. If ISTART = 7, 10, for the first case and JSTART = 10, the state conditions at the beginning of the 10th thrust events are stored and placed in the proper position (VIV, TZERO, W01) for the second case. Therefore, no state input is required for the second case. In the second case, a value of JSTART = 7 indicates that the next case will begin at the 7th thrust event of the initial case. In all preceding cases, the value of JUMP is set equal to the value of JSTART of the prior case.

9.6.2 Atmosphere Options

Nominal, hot, and cold atmospheric subroutines are available for both the Eastern and Western Test Ranges.

9.6.3 External Tank (ET) Propellant Jettison Weight Option

When using the ISTART option of Paragraph 9.3.1 for a post-MECO burn, the total ET jettison weight directly after MECO is not known because of the uncertainty of the amount of unused propellant. This option determines the correct jettison weight for the JETET thrust event by using weights in the weight summary output subroutine SUMARY. The equation for the calculation is:

$$WTJET(JETET) = W_{MECO} - W_{OMS/IG} + W_{MPS/TRAPPED} + W_{FPR}$$

where

WJJET - calculated jettison weight occuring at the end of the JETET thrust event (JETET is input)

WMECO - injected weight at MECO

$W_{OMS/IG}$ - orbiter weight at OMS ignition

$W_{MPS/TRAPPED}$ - trapped MPS propellant (orbiter and SSME)

W_{FPR} - flight performance reserve (orbiter and SSME).

Section 10

ITEMIZED DESCRIPTION OF INPUT VARIABLES

The large quantity of input required to simulate a Space Shuttle trajectory has dictated the use of multiple namelist arrays and access to the Shuttle aerodynamic data base. Namelist arrays have been defined which subdivide the data into three types: one - trajectory data, exclusive of aerodynamic data, located in namelist INPUT; two - aerodynamic data located in namelist AEROIN; and three - data associated with the table output post-processor. Due to the length of the first stage aerodynamic coefficient tables, these tables have been omitted from the regular areodynamic data namelist array. Table 10-1 provides an input variable index which gives the location of the variable with respect to the namelist array. A further subdivision places each variable into categories which provide additional user ease in location of the variable. Tables 10-2, 10-3, 10-4 and 10-5 breakdown the input variables into namelist arrays and categories providing input mnemonics, units, description, and preset values.

Table 10-1. Input Variable Index

ALAT	A-13	CNALP	B-2	GZERO	A-11	NBETA	A-7	RATIO	A-3	TOBL	A-7
ALONG	A-13	CNBETA	B-2	H	A-11	NBGCT	A-15	RE	A-11	TOL	A-15
ALTBAS	B-3	CNBO	B-2	HEAD	A-1	NCASE	C	S	A-8	TOMACH	A-8
ALTLS	A-13	CNO	B-2	HMN	A-12	NCOAST	A-14	SIGTAB	A-7	TRFISP	A-3
ALTTBL	A-9	CPOTBL	A-7	ICONSW	A-14	NENDCT	A-15	SPRADB	A-7	TTABLE	A-7
AYL	A-12	CPTBL	A-7	IFACT	A-7	NMAX	A-14	SRID	C	TTBL	A-7
AZ	A-13	CXA	D	IHEAD	A-7	NOBASE	A-8	SRMAE	A-4	TTILT	A-10
AZWTBL	A-9	CYA	D	IPOLY	A-7	NOEVNT	A-6	SRMFTB	A-4	TXCG	A-9
BAXIAL	B-3	CYBETA	B-2	IPR	A-3	NOTAB	C	SRMTTB	A-4	TXMAC	B-1
BEU	A-12	CYO	B-2	IRTAOA	A-14	NOWD	A-6	SRMWDT	A-4	TZCG	A-9
BHMN	A-12	CYOTBL	A-7	IRTLS	A-14	NP	A-15	STEP	A-12	TZERO	A-10
BKFABT	B-3	CYTBL	A-7	ISTART	A-14	NSYST	A-2	SYRADB	A-7	VAEROD	B-4
BKFTHR	B-3	DATE	C	ITOL	A-8	NTABLE	A-14	TALFP	A-7	VIV	A-13
BNORM	B-3	DELALT	A-8	JETET	A-14	NTALPH	B-1	TALPH	B-1	VRCUT	A-14
BPIT	B-3	DELCA	A-8	JSTART	A-14	NTAU	A-3	TALYH	B-1	VRINT	A-14
BSTEP	A-12	DELCM	A-8	JUMP	A-13	NTBETA	B-1	TAUALT	A-3	VSLOPE	A-14
BYL	A-12	DELCN	A-8	KCDPHI	A-14	NTMACH	B-1	TAULIM	A-3	WDDUMP	A-5
CAA	D	DELFAB	A-8	KCDRES	A-14	NVRST	A-14	TAUT	A-6	WDLBS	A-6
CAALP	B-2	DELVG	A-3	KDB	A-14	NWIND	A-9	TAUTBL	A-3	WDOMS	A-5
CASE	A-1	DJ	A-11	KDT	A-14	OFFICE	C	TAUW	A-6	WDRCS	A-5
CAO	B-2	DPZ	A-15	KIND	A-12	OMEGA	A-11	TBDWT	A-9	WDTLBM	A-4
CBAXIL	A-8	DTZ	A-10	KINDB	A-12	ORBASE	B-3	TBETA	A-7	WIBT	A-15
CHRDOT	A-7	ELEVON	B-4	KP	A-7	PNM	B-2	TCHIR	A-10	WOI	A-13
CHVEL	A-14	EMACH	B-4	KRDER	A-12	PRINT	A-6	THRLBM	A-4	WTBL	A-9
CJ	A-11	ENGDAT	A-2	KRDERB	A-12	PROP	A-2	THROT	A-3	WTJET	A-6
CLA	D	EU	A-12	KWTA	A-7	PSFTM	A-11	TIMISP	A-3	XJEXT	A-15
CLBETA	B-2	FAZ	A-13	KY	A-7	PSIREQ	A-14	TIMLBM	A-4	XLEN	A-8
CLO	B-2	FLAT	A-11	LAST	A-1	PSIRST	A-14	TIMMPS	A-3	XREF	A-8
CMA	D	FOMS	A-5	LPRINT	A-14	PTKG	A-11	TIMTAU	A-3	ZREF	A-8
CMALP	B-2	FPRFAC	A-15	MINH	A-15	PTN	A-11	TITLE	C		
CMO	B-2	FRCS	A-5	MISION	A-1	QMAX	A-14	TLIFT	A-10		
CMUE	A-11	FTBL	A-3	MPSFLG	A-3	QREFT	B-3	TMACH	A-7		
CNA	D	GLIM	A-3	MSWCH	A-14	QY	A-15	TNE	A-2		

A - NAMELIST 'INPUT'; B - NAMELIST 'AEROIN' OR DATA BASE; C - NAMELIST 'INPUT2';
D - DATA BASE ONLY.

Table 10-2. MASTRE Input Variables

Category A-1. General Input — 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
CASE	1	--	Identification number for the case. For output use only.	1.
HEAD	60 spaces	--	60 space Hollerith field used for identification of each page of trajectory printout. (Must be input in the form HEAD = 'TRAJECTORY TITLE'.)	blank spaces
LAST	1	--	If LAST = 1, additional cases will follow. LAST = 0 indicates the last case.	0
MISION	6 spaces	--	Identification name of mission being simulated. Must be one of the following Hollerith fields: 'NOM^^^', - nominal mission 'AOA^^^', - abort once around mission 'RTLS^^', - return to launch site mission 'POST^N', - post nominal mission 'POST^A', - post AOA mission 'VAR^IY', - variable IY mission	'AOA^^^'

Table 10-2. MASTRE Input Variables

Category A-2. General Propulsion Input 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
ENGDAT	8, 15	Variable	Specifies general information concerning each engine being simulated (maximum 15 engines). The type of propulsion system is specified by the corresponding value of NSYST. The matrix ENGDAT(I,J) is defined such that J indicates the jth engine and I denotes the parameter as defined by the following: ENGDAT(1,J) = X location (longitudinal) of gimbal point (m) ENGDAT(2,J) = Y location (lateral) of gimbal point (m) ENGDAT(3,J) = Z location (vertical) of gimbal point (m) ENGDAT(4,J) = pitch cant angle (deg) ENGDAT(5,J) = yaw cant angle (deg) ENGDAT(6,J) = vacuum thrust @ 100% (lbs) ENGDAT(7,J) = flowrate @ 100% (lbs/sec) ENGDAT(8,J) = exit area (m^2)	120 *0.
NSYST	15	--	Corresponds to the ENGDAT array to specify the propulsion system associated with each engine. The values must be one of the following: NSYST = 1, MPS propulsion system NSYST = 2, SRM propulsion system NSYST = 3, LBM propulsion system NSYST = 4, OMS propulsion system	15 *0.

Table 10-2. MASTRE Input Variable

Category A-2. General Propulsion Input (Continued) 'INPUT'NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
			NSYST = 5, RCS propulsion system NSYST = 6, space available for future system.	
PROP	6	lbs	Total amount of loaded propellant per propulsion system	6 * 0.
TNE	(6,15)	-	Specifies the number of engines of each propulsion system per thrust event. TNE(1,ITHR) = number of MPS engines TNE(2,ITHR) = number of SRM engines TNE(3,ITHR) = number of LBM engines TNE(4,ITHR) = number of OMS engines TNE(5,ITHR) = number of RCS engines TNE(6,ITHR) = number of future system engines NOTE: Using the propulsion systems as defined in these examples, the simulation of an MPS engine failure (TNE(1,ITHR) = 2) assumes the data stored in ENGDAT(I,1) to be for the failed engine. Therefore, a top, lower left or lower right engine failure can be simulated by storing the engine data for the failed engine in ENGDAT(I,1).	90 * 0.

Table 10-2. MASTRE Input Variables

Category A-3. MPS Data 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
DELVG	1	m/sec	ΔV for geometry reserves (currently not used)	0.
FTBL	15	-	Table of MPS throttle level (100% is input as 1.0). Input as a function of time table TIMMPS.	15 *0.
GLIM	15		Maximum allowable value of acceleration for each thrust event.	15 *.1E20
IPR	1	-	(not currently used)	-
MPSFLG	15	-	Flag indicating the type of throttling per thrust event MPSFLG = 1, constant throttle level using THROT. MPSFLG = 2, variable throttle level as a function of tables TIMMPS and FTBL (1st stage only) MPSFLG = 3, not used MPSFLG = 4, throttle based on acceleration limit GLIM. Uses THROT when acceleration throttling is not required.	15 *0
NTAU	1	-	Indicator representing the value of the FTBL/TIMMPS tables when the dynamic pressure throttling maneuver will begin. Used if values of FTBL and TIMMPS are used as optimization variables.	0
RATIO	1	-	Mixture ratio (used for load indicator calculation)	0.
TAUALT	5	m	Independent altitude table used for minimum throttle level table (TAUTBL).	5 * 0.

Table 10-2. MASTRE Input Variables

Category A-3. MPS Data (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
TAULIM	1	sec^{-1}	Maximum rate of throttle used with NTAU	0.1
TAUTBL	5	--	Minimum throttle level table and in conjunction with altitude table TAUALT (used only during the booster phase, set equal .65 during orbiter phase).	5 *0/
THRØT	15	--	Throttle level as a function of thrust event.	15 *0.
TIMISP	20	sec	Independent time table for reference specific impulse table TRFISP	20 *0.
TIMMPS	15	sec	Independent time table for throttle table FTBL	15 *0.
TIMTAU	1	sec	Time used to define MPS throttle change point used as an optimization parameter.	0.
TRFISP	20	sec	Reference (100%) specific impulse table used in conjunction with TIMISP time table.	20 *0.

Table 10-2. MASTRE Input Variables

Category A-4. SRM and LBM Data

'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
SRMAE	2	--	SRM exit area and rate of change of exit area SRMAE(1) = exit area at time = 0. (m^2) SRMAE(2) = rate of change of exit area with respect to time (m^2/sec).	2 *0.
SRMFTB	250	lbs	SRM single engine vacuum thrust table used in conjunction with time table SRMTTB.	250 *0.
SRMTTB	250	sec	Independent time table used for SRMFTB and SRMWDT.	250 *0.
SRMWDT	250	lbs/sec	SRM single engine flowrate table used in conjunction with time table SRMTTB.	250 *0.
THRLBM	30	lbs	LBM single engine vacuum thrust table used in conjunction with time table TIMLBM.	30 *0.
TIMLBM	30	sec	Independent time table used for THRLBM and WDTLBM tables.	30 *0.
WDTLBM	30	lbs/sec	LBM single engine flowrate table used in conjunction with time table TIMLBM.	30 *0.

Table 10-2. MASTRE Input Variables

Category A-5. OMS and RCS Data

'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
FØMS	15	lbs	OMS single engine vacuum thrust, input per thrust event.	15 *0.
FRCS	15	lbs	RCS single engine vacuum thrust, input per thrust event.	15 *0.
WDDUMP	15	lbs/sec	Flowrate of total propellant dumped through the OMS/RCS interconnect system per thrust event.	15 *0.
WDØMS	15	lbs/sec	OMS single engine flowrate per thrust event.	15 *0.
WDRCS	15	lbs/sec	RCS single engine flowrate per thrust event.	15 *0.

Table 10-2. MASTRE Input Variables

Category A-6. Thrust Event Dependent Input 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
NØEVNT	5	--	Number of thrust events in each stage, for example, if there are two stages, with five thrust events in the first stage and two in the second, input NØEVNT = 5, 2.	5 * 0
NØWD	15	--	Denotes "picket" number from which TAUW is defined.	15 * 0
PRINT	15	sec	Print time increment for each thrust event during the converged trajectory printout. Also used as the maximum step size for the variable step Adams-Moulton integration scheme.	15 * 10.
TAUT	15	sec	Time duration of each thrust event. If the thrust event time is being optimized, this value is used as the initial estimate.	15 * 0.
TAUW	15	sec	Elapsed time between a thrust event and a weight drop event.	15 * 0.
WDLBS	15	lbm	Weight dropped during a weight drop event. Only applicable during min-H optimization.	15 * 0.
WTJET	15	lbm	Jettison weight per thrust event. The weight is assumed jettisoned at the end of the corresponding thrust event.	15 * 0.

Table 10-2. MASTRE Input Variables

Category A-7. Attitude Control

'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
CHRDØT	1	deg/sec	Roll rate required to place the orbiter tail fin into the flight plane (measured positive counterclockwise).	0.
CPØTBL	91	deg	Orbiter pitch attitude tables which correspond to the time points of TØBL.	91 * 0.
CPTBL	30	deg	Values of pitch attitude (x_p) to be used in forming the nominal (first guess) trajectory for the booster stage. If IPØLY is used (denoting an additional polynominal), the first value of the x_p tables and time tables which correspond to the additional polynominal is defined by CPTBL (KB(1) + 2) and TTBL (KB(1) + 2), respectively.	30 * 0
CYØTBL	91	deg	Orbiter yaw attitude table which corresponds to the time points of TØBL (same time points as CPØTBL).	91 * 0.
CYTBL	30	deg	Same as CPTBL except corresponding to the x_y attitude history (dependent on TTBL(16-30)). If IPOLY is used, the first value of the x_y tables which correspond to the additional polynominal is CYTBL(KY(1) + 2) and the time table is TTBL(KY(1) + 17).	30 * 0.
IFACT	1	--	Flag denoting the type of attitude control used during the booster phase of the trajectory. IFACT = 1, Polynominals based on the values of CPTBL and CYTBL with orders specified by KP and KY are used throughout the booster phase.	1

Table 10-2. MASTRE Input Variables

Category A-7. Attitude Control (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
		--	IFACT = 2, Attitude after the tiltover maneuver (which assumes a ploynominal attitude control between begin and end tiltover maneuver) is determined based on a Mach dependent angle of attack and sideslip angle tables (TMACH is used for Mach tables, TALFP is used for angle of attack table, and TBETA is used for sideslip angle table). IFACT = 3, Attitude from liftoff to the beginning of the min-H optimization is determined based on the prime contractor attitude angles θ (pitch), ψ (yaw), and σ (roll). θ and ψ are functions of earth fixed velocity and σ is a function of time. (CPTBL is used for θ table, CYTBL is used for ψ table, TTBL is used for earth-fixed velocity table, SIGTAB is used for σ table, and TTABL is used for time table.	
IHEAD	1	--	Flag to indicate either a heads up or heads down attitude after the initial roll maneuver. IHEAD = 0, heads up IHEAD $\neq$ 1, heads down.	1
IPOLY	1	--	Denotes the thrust event where a split polynominal will begin. KP(2) and KY(2) should be used for this option. Can be used in either the booster or orbiter phase before min-H initialization. The values of TTBL, CPTBL, and CYTBL are used if the orbiter phase might be simulated.	20

Table 10-2. MASTRE Input Variables

Category A-7. Attitude Control (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
KP	2	--	Order of the booster pitch polynominal. If IPØLY is used the second value is used for the additional polynominal.	2 * 0
KWTA	4	--	Control option switch per min-H attitude segment = 2, pitch attitude only is optimized in min-H = 3, pitch and yaw attitude are optimized in min-H.	4 * 2
KY	2	--	Same as KP except yaw.	2 * 0
NBETA	1	--	Specifies the first value of the TBETA table that will be in the optimization. Must have a value if KDB(32-23) are used.	0
SIGTAB	20	deg	Roll attitude table which corresponds to prime contractor sigma attitude angle, dependent on time table TTABLE.	20 * 0
SPRADB	1	deg	Perturbation value of x_p used to determine booster attitude influence coefficients.	.1
SYRADB	1	deg	Same as SPRADB but for yaw.	.1
TALFP	25	deg	Angle-of-attack control history table.	25 * 0.
TBETA	25	deg	Sideslip angle control history table.	25 * 0.
TMACH	25	deg	Independent Mach table for TALFP and TBETA.	25 * 0.

Table 10-2. MASTRE Input Variables

Category A-7. Attitude Control (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
TØBL	91	sec	Time points for the control tables measured from zero. The control tables must start in the locations (TØBL(1), TØBL(31), or TØBL(61).	91 * 0.
TTABLE	20	sec	Independent time table for SIGTAB.	20 * 0.
TTBL	30	sec	Times referenced from TLIFT which correspond to the x_p, and x_y tables, CPTBL and CYTBL. For IFACT < 3, TTBL(1-15) corresponds to x_p and TTBL(16-30) corresponds to x_y. If IFACT = 3, TTBL(1-30) is the independent relative velocity table for CPTBL and CYTBL (which represent θ and ψ).	30 * 0.

Table 10-2. MASTRE Input Variables

Category A-8. Extra Aero Data

'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
CBAXIL	1	nt	Constant value of base axial force after thrust event denoted by NØBASE.	0.
DELALT	25	ft	Independent altitude table used for incremental base force table DELFAB.	25 * 0.
DELCA	20	--	Incremental axial force coefficient table input as a function of Mach table TØMACH. Used when simulating the LBM or other added configurations.	20 * 0.
DELCM	20	--	Incremental pitch moment coefficient table dependent on Mach table TØMACH.	20 * 0.
DELCN	20	--	Incremental normal force coefficient table dependent on Mach table TØMACH.	20 * 0.
DELFAB	25	lbs	Incremental base force table dependent on altitude table DELALT. Used only when ITØL ≠ 0.	25 * 0.
ITØL	15	--	Flag to indicate that incremental aerodynamic tables will be used. ITØL = 0, not used ITØL ≠ 0, used.	15 * 0.
NØBASE	1	--	Beginning thrust event where the base axial force will be forced to a constant value (CBAXIL).	20
S	15	m^2	Aerodynamic reference area for each thrust event.	15 * 0.
TØMACH	20	--	Independent Mach number table used in conjunction with DELCA, DELCM, DELCN.	20 * 0

Table 10-2. MASTRE Input Variables

Category A-8. Extra Aero Data (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
XLEN	2	m	Aerodynamic reference length for each stage.	2 * 0.
XREF	2	m	Longitudinal coordinates of the aerodynamic reference point for each stage (see Figure 12-1).	2 * 0.
ZREF	2	m	Vertical coordinates of the aerodynamic reference point for each stage (see Figure 12-1).	2 * 0.

Table 10-2. MASTRE Input Variables

Category A-9. Center of Gravity and Wind Data 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
ALTTBL	75	--	Tabular values of altitude which correspond to AZWTBL and WTBL (independent of stage). If altitude exceeds last value of ALTTBL table, wind velocity and direction are set to zero.	75 * 0.
AZWTBL	75	deg	Tabular values of wind azimuth angles which represent the direction from which the wind is blowing (measured positive clockwise from north).	75 * 0.
NWIND	1	--	Indicates the number of points used in the wind tables. If NWIND = 0, no wind tables are used.	0
TBDWT	15, 2	lbm	Total delta weight values which correspond to the center-of-gravity locations of TXCG and TZCG. Up to 15 points may be used for each stage. The second index denotes two tables, one for the first stage and one for the second stage.	30 * 0.
TXCG	15, 2	m	Longitudinal coordinates of the center of gravity (per stage).	30 * 0.
TZCG	15, 2	m	Vertical coordinates of the center of gravity (per stage).	30 * 0.
WTBL	75	m/sec	Tabular values of wind velocity magnitude.	75 * 0.

Table 10-2. MASTRE Input Variables

Category A-10. Time Input

'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
DTZ	1	sec	Time from ground reference release (GRR) to lift-off. GRR is the point in the countdown at which the launch inertial coordinate system is established. Must be input for jump-start as well as for ground-launch trajectories (if required).	0.
TCHIR	1	sec	Time which signifies the beginning of the rolling maneuver which rolls the orbiter tail fin from an initial azimuth (FAZ) into the flight plane in a "heads up" or "heads down" position. Termination time is determined by dividing the roll angle required by the roll rate (CHRDØT).	0.
TLIFT	1	sec	Time which signifies the end of the lift-off phase and the beginning of the X_p and X_y attitude control program. Referenced to TZERØ.	8.
TTILT	1	sec	Time which signifies the end of tiltover phase. This input is only needed when IFACT < 3. The tiltover maneuver is accomplished by using a predetermined X_p polynominal between TLIFT and TTILT. The tiltover can be optimized by use of the variable KDB.	0.
TZERØ	1	sec	Time at the start of the trajectory. For a jump-start, TZERØ is the time at the initiation of the trajectory.	0.

Table 10-2. MASTRE Input Variables

Category A-11. Constants 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
CJ	1	--	First coefficient in the Fischer ellipsoid gravitational expansion.	.00162405
CMUE	1	m^3/sec^2	Product of the universal gravitational constant and the mass of the earth.	2.9860120E+14
DJ	1	--	Third coefficient in the Fischer ellipsoid gravitational expansion.	.69125E-05
FLAT	1	--	Flattening coefficient of the Fischer model.	1./298.25
GZERØ	1	m/sec^2	Gravitational acceleration of the earth at the equator.	9.80665
II	1	--	Second coefficient in the Fischer elliposid gravitational expansion.	-.64E-05
ØMEGA	1	rad/sec	Angular rotational velocity of the earth.	.72921159E-04
PSFTM	1	--	Conversion from lbs/ft^2 to N/m^2.	47.8802589
PTKG	1	--	Conversion from pounds mass to kilograms.	.45359237
PTN	1	--	Conversion from pounds force to Newtons.	4.44822162
RE	1	m	Radius of the earth at the equator.	6378160.

Table 10-2. MASTRE Input Variables

Category A-12. Integration Data 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
AYL	1	--	Used for error check in integration. The preset value should be used (no input required).	.002
BEU	1	--	Same as EU except for backward trajectory.	.00002
BHMN	1	sec	Same as HMN except for backward trajectory.	.5
BSTEP	15	sec	Integration stepsize increment for backward simulation per thrust event.	15 * 16.
BYL	1	--	Same as AYL except for backward trajectory.	.00004
EU	1	--	Upper bound on integration stepsize control algorithm. The preset value should be used (no input required).	.00001
HMN	1	sec	Minimum stepsize for integration. The preset value should be used (no input required).	.25
KIND	1	--	Integration method indicator. KIND = 1, 2, 3 denotes variable stepsize Adams-Moulton, fixed step Runge-Kutta, or fixed step Adams-Moulton, respectively.	1
KINDB	1	--	Same as KIND except for backward trajectory.	3
KRDER	1	--	Order of differences used in the integration. The preset value should be used (no input required).	3
KRDERB	1	--	Same as KRDER except for backward trajectory.	3
STEP	15	sec	Integration stepsize increment for each event when Runge-Kutta or fixed stepsize Adams-Moulton integration is used. For the variable step Adams-Moulton STEP is the initialization stepsize.	1. for 1st thrust event 8. thereafter

Table 10-2. MASTRE Input Variables

Category A-13. Launch and Initial Conditions 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
ALAT	1	deg	Launch site geodetic latitude.	28.608422
ALØNG	1	deg	Launch site longitude (measured positive West of Greenwich).	80.6041334
ALTLS	1	m	Altitude of the launch site above the spheriod.	0.
AZ	1	deg	Launch azimuth (measured positive clockwise of north).	90.
FAZ	1	deg	Initial pointing azimuth of the orbiter tail fin (measured clockwise from north).	0.
JUMP	1	--	Jump-start indicator. Denotes the number of the thrust event at which the start is to occur. When using the jump-start option, the array VIV must be input which contains the initial state at the beginning of thrust event JUMP.	1
VIV	25	variable	Initial state for jump-start option.	25 * 0.
WØ1	1	lbm	Lift-off weight at TZERØ.	0.

Table 10-2. MASTRE Input Variables

Category A-14. Options 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
FPRFAC	1	--	Factor multiplied by terminal characteristic velocity to calculate the delta velocity for flight performance reserves (FPR)	.0085
ICØNSW	1	--	Indicates which thrust event will terminate SRB TVC and begin liquid engine TVC.	20
IRTAØA	1	--	Indicates which thrust event is denoted as 'mode boundary' initiation. Used only if NVRST does not denote thrust event where mode boundary occurs.	0
IRTLS	1	--	Currently not used.	0
ISTART	2	--	Indicates that the state variables at the beginning of the thrust event denoted by ISTART(1) or ISTART(2) will be stored for subsequent cases (but only in the case of convergence of the first case).	2 * 0.
JETET	1	--	The number of thrust event where the calculated value of the external tank jettison weight will be used to modify WTJET (JETET).	0.
JSTART	1	--	Used in conjunction with the ISTART option to indicate which thrust event data will be used in the next case. For example, if in the first case ISTART = 7, 10, and the next case is required to start at the 10th thrust event, then JSTART = 10, is used in the input of the first case. (NOTE: unless the values of ISTART are changed in the preceding case, the values of the first case are retained and can be revised by proper choice of JSTART).	2 * 0.

Table 10-2. MASTRE Input Variables

Category A-14. Options (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
KCDPHI	10	--	Terminal payoff and constraint codes. KCDPHI(1) denotes the payoff desired; ICDPHI(2-10) are the code numbers of the terminal constraints desired.	10 * 0
KCDRES	6, 4	--	Intermediate constraint codes for up to four immediate constraints.	24 * 0
KDB	40	--	Parameter optimization indicator. KDB(1) through KDB(15) corresponds to times TAUT(1) through TAUT(15). KDB(16) corresponds to lift-off weight WØ1. KDB(17) corresponds to flight azimuth AZ. KDB(18) through KDB(24) correspond to booster pitch attitude CPTBL(2) through CPTBL(8). KDB(25) through KDB(28) correspond to booster yaw attitude CYTBL(2) through CYTBL(5). KDB(29) through KDB(31) correspond to throttle parameters TIMMPS(NTAU), TIMTAU, and FTBL(NTAU + 1), respectively. KDB(32) through KDB(34) correspond to sideslip control parameters TBETA(NBETA), TBETA(NBETA + 1), and TBETA (NBETA + 2), respectively.	40 * 0
KDT	15	--	Currently not used.	15 * 0
LPRINT	1	--	Print option indicator (see Section 11).	0
MSWCH	15	--	Currently not used.	15 * 0

Table 10-2. MASTRE Input Variables

Category A-14. Options (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
NCØAST	1	--	Indicator of thrust event where coast event begins (only performed during converged trajectory printout).	0
NMAX	1	--	Maximum number of trajectory simulations allowed. (This does not include the trajectories required to determine influence coefficients.)	0
NTABLE	1	--	Indicator for output tables: NTABLE = 0, no tables are required. NTABLE ≠ 0, tables are required.	0
NVRST	4	--	Indicates that at the termination of the NVRST thrust events, intermediate constraints (denoted by KCDRES) are to be imposed. Must be zero if no intermediate constraints are desired.	4 * 0
PSIREQ	10	variable depending on constraints	Desired values of the end constraints which correspond to the code numbers of KCDPHI(2-10).	10 * 0.
PSIRST	6, 4	variable depending on constraints	Desired values of the intermediate constraints in the same order as KCDRES. (PSIRST(1, 1) through PSIRST(6, 1) correspond to NVRST(1); PSIRST(2, 1) through PSIRST(6, 2) correspond to NVRST(2); etc.	24 * 0.
QMAX	1	lb/ft^2	Value of dynamic pressure for calculation of dynamic pressure penalty function.	0.
VRCUT	1	m/sec	ΔV allotment for thrust event (currently not used).	0.

Table 10-2. MASTRE Input Variables

Category A-14. Options (Continued) 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
VRINT	2	m/sec	Used in constraint options 18 and 19 as the intercept point on the relative velocity axis (18) or the ΔV available axis (19).	2 * 0.
VSLØPE	2	m/sec/ n mi	Used with VRINT as the slope of the linear curves used in options 18 and 19.	2 * 0.

Table 10-2. MASTRE Input Variables

Category A-15. Optimization Data 'INPUT' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
DP2	1	--	Decimal fraction of constraint error to remove on first iteration. (If initial guesses are known to be good, setting DP2 = 1. will speed convergence.)	.25
MINH	1	--	Denotes the thrust event where the min-H optimization will begin.	0
NBGCT	4	--	j^{th} control table begins at the "picket" number corresponding to NBGCT(J).	4 * 0
NENDCT	4	--	j^{th} control table ends at the "picket" number corresponding to NENDCT(J).	4 * 0
NP	4	--	The number of points in a control table (must be an odd number of points $\leq$ 29).	4 * 0
QY	1	--	Decimal fraction of optimization step used during optimization process.	.8
TØL	1	--	Convergence tolerance on the value of the normalized total derivatives of the payoff with respect to the parameters (output as "NEW PCØN").	0.025
WIBT	15	--	Input value of weight matrix corresponding to the optimization parameters being used.	15 * 1.
XJEXT	1	--	If maximizing the payoff function XJEXT = +1; if minimizing the XJEXT = -1.	1.

Table 10-3. MASTRE Input Variables

Category B-1. First Stage Aero (Additional)

'AEROIN' NAMELIST DATA OR DATA BASE

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
NTALPH	2	--	Number of angle of attack dependent points for the C_A, C_N, C_M tables and the C_y, C_n, C_l tables, respectively.	2 * 0
NTBETA	2	--	Number of sideslip angle dependent points for the C_A, C_N, C_M tables and the C_y, C_n, C_l tables, respectively.	2 * 0
NTMACH	1	--	Number of Mach number dependent points for both tables.	0
TALPH	11, 2	deg	Angle of attack values used for dependent values for above tables.	22 * 0.
TALYH	11, 2	deg	Sideslip angle values used for dependent values for above tables.	22 * 0.
TXMAC	28	--	Mach number values for dependent values for above tables.	28 * 0.

Figure 10-3. MASTRE Input Variables

Category B-2. Second Stage Aero

'AEROIN' NAMELIST DATA OR DATA BASE

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
CAALP	30	deg	Coefficient table of slope of axial force with respect to angle of attack.	30 * 0.
CAO	30	--	Coefficient table of zero angle of attack axial force.	30 * 0.
CLBETA	30	deg	Coefficient table of slope of rolling moment with respect to sideslip angle.	30 * 0.
CLO	30	--	Coefficient table of zero sideslip angle rolling moment.	30 * 0.
CMALP	30	deg	Coefficient table of slope of pitch moment with respect to angle of attack.	30 * 0.
CMØ	30	--	Coefficient table of zero angle of attack pitch moment.	30 * 0.
CNALP	30	deg	Coefficient table of slope of normal force with respect to angle of attack.	30 * 0.
CNBETA	30	deg	Coefficient table of slope of yawing moment with respect to sideslip angle.	30 * 0.
CNBØ	30	--	Coefficient table of zero sideslip angle yawing moment.	30 * 0.
CNØ	30	--	Coefficient table of zero angle of attack normal force	30 * 0.
PNM	30	--	Mach number tables used for above tables.	30 * 0.
CYBETA	30	deg	Coefficient table of slope of side force with respect to sideslip angle.	30 * 0.
CYO	30	--	Coefficient table of zero sideslip angle side force.	30 * 0.

Table 10-3. MASTRE Input Variables

Category B-3. Base Drag Data

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	'AEROIN' NAMELIST DATA OR DATA BASE PRESET VALUE
ALTBAS	50, 3	m	Independent altitude tables for base force and moment tables. ALTBAS(1-50, 1) - first stage data ALTBAS(1-50, 2) - second stage data (3 engines) ALTBAS(1-50, 3) - second stage data (2 engines).	150 * 0.
BAXIAL	50, 3	lbs	Base axial force tables dependent on ALTBAS.	150 * 0.
BKFABT	50, 2	--	Base force coefficient tables for alpha/beta effects. BKFABT(1,1) is the beginning of the K_α coefficient table and BKFABT(1,2) is the beginning of the K_β coefficient table (see Subsection 2.4.1).	100 * 0.
BKFTHR	50, 2	--	Base force coefficient tables for SSME throttle effects. BKFTHR(1,1) is the beginning of the K_1 coefficient tables and BKFTHR(1,2) is the beginning of the K_2 coefficient tables (see Subsection 2.4.1).	100 * 0.
BNØRM	50, 3	lbs	Base normal force tables dependent on ALTBAS.	150 * 0.
BPIT	50, 3	ft/lbs	Base pitching moment tables dependent on ALTBAS.	150 * 0.
ØRBASE	50	lbf	Orbiter axial base force for 109% MPS throttle level.	50 * 0.
QREFT	50	lbf/ft^2	Reference dynamic pressure (booster stage only) dependent on ALTBAS (1-50, 1) table.	50 * 0.

Table 10-3. MASTRE Input Variables

Category B-4. Additional Data

'AEROIN' NAMELIST DATA OR DATA BASE

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
ELEVON	25, 4	deg	Inboard and outboard elevon deflection angles. ELEVON(1-25, 1) - inboard (schedule) ELEVON(1-25, 2) - outboard (schedule ELEVON(1-25, 3) - inboard (actual) ELEVON(1-25, 4) - outboard (actual).	100 * 0.
EMACH	25, 2	ft/sec --	Independent tables for ELEVON. EMACH (1-25, 1) - earth relative velocity tables for scheduled data EMACH (1-25, 2) - earth relative mach number tables for actual data	50 * 0.
VAERØD	10, 19	--	Coefficients of viscous interaction data.	190 * 0.

Table 10-4. MASTRE Input Variables

Category C. Output Tables Data

'INPUT2' NAMELIST DATA

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
DATE	1 (12 spaces)	--	Hollerith field for calendar date (input as DATE = '05/01/81').	blanks
NCASE	1	--	Case number desired or output.	0
NØTAB	1	--	(Not used).	--
ØFFICE	1 (12 spaces)	--	Hollerith field identification (input as OFFICE = 'MSFC/EL24').	blank
SRID	60 (6 spaces each)		Hollerith field for optional thrust event and weight drop event notations. One notation requires two Hollerith fields. The variables located in SRID(1-30) replace the header "THRUST␣EVENT" and SRID(31-60) replaces the header "WEIGHT␣DRØP". (Input as SRID = 'SPECIA', 'L␣EVNT'.)	blank
TITLE	1 (48 spaces)		Hollerith field for publication title. (Input as TITLE = 'LBM TEST SIMULATIØN'.)	blank

Table 10-5. MASTRE Input Variables

Category D. First Stage Data

DATA BASE ONLY

VARIABLE INPUT NAME	DIMENSION	UNITS	DESCRIPTION	PRESET VALUE
CAA	15, 11, 6	--	Axial force coefficient table assuming zero sideslip angle.	990 * 0.
CLA	15, 11, 6	--	Rolling moment coefficient table assuming zero angle of attack.	990 * 0.
CMA	15, 11, 6	--	Pitching moment coefficient table assuming zero sideslip angle.	990 * 0.
CNA	15, 11, 6	--	Normal force coefficient table assuming zero sideslip angle.	990 * 0.
CXA	15, 11, 6	--	Yawing moment coefficient table assuming zero angle of attack.	990 * 0.
CYA	15, 11, 6	--	Side force coefficient table assuming zero angle of attack.	990 * 0.

Section 11

DETAILED OUTPUT DESCRIPTION

The output for the MASTRE program is described in this section. The output is divided into the following groups:

1. Initiation output;
2. Trajectory block output;
3. Iteration output; and
4. Postprocessor output.

11.1 INITIAL OUTPUT

Initial output is the printout of the variables input via the Namelist arrays 'AERØIN' and 'INPUT'. This output describes the current status of these variables in memory. In a multiple run job, one consisting of more than one trajectory simulation, the Namelist array 'AERØIN' output will only appear on the first trajectory simulation (the assumption has been made that the aerodynamic data will not change for multiple runs.)

11.2 TRAJECTORY BLOCK OUTPUT

Parameters associated with equations of motion, as well as other pertinent data, are printed in the trajectory block output. Printing of the block output occurs on the first (or nominal) and converged trajectories and is based on either describing discrete events or print interval. A page of output, as shown in Figure 11-1, consists of two blocks of print. The caption at the upper left hand corner of the block identifies the significance of the particular printout. If a caption is not shown, either a print interval has been completed or a thrust event has been terminated. Discrete events are described by the following captions:

MASTER CASE = 1.0000 LBM(350K)+SRB-MOD(ITC-121-781-WTR(52 F)/MSN4-WIN/P-AOA/ PAGE 1 (AERO - JUN 79 BTV) (ATMOS- VRA/II)

IGNITION

TIME	.00000000	VSUBE	.38146973-005	GAME	-.30037164-007	AZE	.89999999+002	THRSI	.11287039+007	XMLB	.48439989+007
R	.63714681+007	VSUBI	.38335190+003	GAMI	-.29766114-015	AZI	.89999999+002	GCLAT	.34401763+002	LONG	-.12072528+003
ALT	.15806250+003	VSUBR	.13158405+001	GAMR	.12328325-009	AZR	-.20236708+002	GDLAT	.34581389+002	NODE	.89999999+002
ZB-X	.19975062+005	XB-Y	.63714367+007	YB-Z	.39634295-004	ZUB-W	.76064191-006	XDB-U	.00000000	YDB-V	-.38335190+003
CHIP	.17655535-030	CHIY	-.39756934-015	CHIR	.89999996+002	TIMP	.00000000	LTIMP	.34401763+002	LNGMP	-.12072528+003
THERI	.90000000+002	PSIRI	.18000000+003	PHIRI	.89999996+002	INCL	.34401762+002	RANGE	.42040172+000	RNGAN	.69941137-002
THMPS	.11287039+007	THSRM	.00000000	THLBM	.00000000	THOMS	.00000000	THRCS	.00000000	TH	.00000000
TVMPS	.14100000+007	TVSRM	.00000000	TVLBM	.00000000	TVOMS	.00000000	TVRCS	.00000000	TVAC	.00000000
ISPMP	.45320475+003	ISPSR	.00000000	ISPLB	.00000000	ISPOM	.00000000	ISPRC	.00000000	ISP	.00000000
WDMPS	.31111765+004	WDSRM	.00000000	WDLBM	.00000000	WDOMS	.00000000	WDRCS	.00000000	WDOT	.00000000
MPS F	.15760470+007	SRM F	.22325240+007	LBM F	.34999999+006	OMS F	.25064000+005	RCS F	.75079999+004	FUEL	.00000000
MPS D	.21117031+005	SRM D	.31250000-001	LBM D	.78125000-002	OMS D	.48828125-003	RCS D	.12207031-003	SPENT	.00000000
DPMPS	-.11999187+002	DPSRM	.00000000	DPLBM	.00000000	DPOMS	.00000000	DPRCS	.00000000	DELP	.00000000
DYMPS	.00000000	DYSRM	.00000000	DYLBM	.00000000	DYOMS	.00000000	DYRCS	.00000000	DELY	.00000000
TAU	.10000000+001	SRMAE	.10744825+002	PDOME	.29748739+002	SRMET	-.29332215+006	ORBET	.58664337+006	CH VL	.00000000
IN L	.00000000	GV L	.00000000	DR L	.00000000	BKP L	.00000000	GIM L	.00000000	ID VL	.00000000
MACH	.38717563-002	ALPHA	.00000000	BETA	.00000000	FAA	-.29348314+001	FAS	-.87852214-001	FAN	.56663772+001
Q	.21918673-001	QALPH	.00000000	QBETA	.00000000	FAB	-.95048811+001	FMB	.14122959+003	FNB	-.19791254+001
DRAG	-.29348314+001	LIFT	.56663772+001	VWIND	.13158418+001	AZW	.33976313+003	HEAT	.00000000	HRATE	.44507322-006
PRESS	.20887392+004	RHO	.12122558+001	TEMP	.28741165+003	PEYNO	.29267413+007	DELPC	-.11999187+002	DELYC	.00000000
EMACH	.11224444-007	INBD	.10000000+002	OUTBD	.90000000+001	AXACC	.22764258+000	NRACC	.48382250-001	LTACC	-.18136299-007
XCG	-.51220499+003	ZCG	-.16811033+002	OPEN	.00000000						

LIFT-OFF

TIME	.25236010+000	VSUBE	.70545925-002	GAME	.55598225+002	AZE	.17994703+003	THRSI	.48619039+007	XMLB	.48415760+007
R	.63714681+007	VSUBI	.38335190+003	GAMI	.86996429-003	AZI	.90000595+002	GCLAT	.34401763+002	LONG	-.12072528+003
ALT	.15806250+003	VSUBR	.13121144+001	GAMR	.25417261+000	AZR	-.20296916+002	GDLAT	.34581389+002	NODE	.89999998+002
ZB-X	.19975062+005	XB-Y	.63714367+007	YB-Z	-.96742684+002	ZUB-W	.76064191-006	XDB-U	.00000000	YDB-V	-.38335190+003
CHIP	.38825130-018	CHIY	-.86809239-003	CHIR	.89999996+002	TIMP	.16985575-004	LTIMP	.34401763+002	LNGMP	-.12072528+003
THERI	.90000000+002	PSIRI	.17999913+003	PHIRI	.89999996+002	INCL	.34401762+002	RANGE	.42040172+000	RNGAN	.69941137-002
THMPS	.11287039+007	THSRM	.37332000+007	THLBM	.00000000	THOMS	.00000000	THRCS	.00000000	TH	.00000000
TVMPS	.14100000+007	TVSRM	.42163518+007	TVLBM	.00000000	TVOMS	.00000000	TVRCS	.00000000	TVAC	.00000000
ISPMP	.45320914+003	ISPSR	.26507040+003	ISPLB	.00000000	ISPOM	.00000000	ISPRC	.00000000	ISP	.00000000
WDMPS	.31111464+004	WDSRM	.15906535+005	WDLBM	.00000000	WDOMS	.00000000	WDRCS	.00000000	WDOT	.00000000
MPS F	.15752618+007	SRM F	.22308862+007	LBM F	.34999999+006	OMS F	.25064000+005	RCS F	.75079999+004	FUEL	.00000000
MPS D	.21902172+005	SRM D	.16378437+004	LBM D	.78125000-002	OMS D	.48828125-003	RCS D	.12207031-003	SPENT	.00000000
DPMPS	-.11999187+002	DPSRM	-.15854960+001	DPLBM	.00000000	DPOMS	.00000000	DPRCS	.00000000	DELP	.00000000
DYMPS	.00000000	DYSRM	.29831932-006	DYLBM	.00000000	DYOMS	.00000000	DYRCS	.00000000	DELY	.00000000
TAU	.10000000+001	SRMAE	.10744825+002	PDOME	.45770673+002	SRMET	.58007191+006	ORBET	-.11601447+007	CH VL	.00000000
IN L	.00000000	GV L	.00000000	DR L	.00000000	BKP L	.00000000	GIM L	.00000000	ID VL	.00000000
MACH	.38607927-002	ALPHA	.00000000	BETA	.00000000	FAA	-.29719875+001	FAS	-.87355378-001	FAN	.56231391+001
Q	.21794714-001	QALPH	.00000000	QBETA	.00000000	FAB	-.95048811+001	FMB	.14122959+003	FNB	-.19791254+001
DRAG	-.29719875+001	LIFT	.56231391+001	VWIND	.13158418+001	AZW	.33976313+003	HEAT	.11183086-008	HRATE	.44121534-008
PRESS	.20887392+004	RHO	.12122558+001	TEMP	.28741165+003	PEYNO	.29184537+007	DELPC	-.39953141+001	DELYC	.23027544-006
EMACH	.20757579-004	INBD	.10000000+002	OUTBD	.90000000+001	AXACC	.99853250+000	NRACC	.69740928-001	LTACC	-.14029592-007
XCG	-.51223169+003	ZCG	-.16819535+002	OPEN	.00000000						

Figure 11-1. Trajectory Block Output

IGNITIØN	-	Initiation of the trajectory from launch site or jump-start conditions
LIFTØFF	-	Initial movement of the vehicle from launch site when total acceleration, less gravity, is equal to one
THRUST_EVENT	-	Beginning of a thrust event
BEGIN_CØAST	-	Beginning of a coast thrust event (specified by NCØAST)
END CØAST	-	Coast thrust event termination
BEGIN_TILT	-	Beginning of tilt over maneuver
END_TILT	-	Tilt-over maneuver termination
SRB_SEPARATE	-	Separation of the Solid Rocket Motors
BEGIN_MINH	-	Min-H control has been initiated
LBM_IGNITIØN	-	Begin simulation of Liquid Boost Module engines
LBM_SEPARATE	-	Separation of Liquid Boost Module
MAX_DYN_PRESS	-	Maximum dynamic pressure
BEGIN_RØLL	-	Begin initial roll maneuver to place the orbiter pitch plane in the flight plane
END_RØLL	-	Roll maneuver termination
INTERMEDIATE	-	Time point on the trajectory where the intermediate constraints are evaluated
RTLS/AØA	-	Special intermediate point denoting mode boundary
G-LIMIT	-	Begin continuous throttle based on acceleration limit
MPS_THRØTTLE	-	Discrete changes in main propulsion system throttle history (booster stage only)
WEIGHT DRØP	-	Weight drop event
INJECTIØN	-	Terminal trajectory constraints are imposed at this time

The block printout consists of 22 lines of output having the following labeled mnemonics:

Line 1

TIME - instantaneous time (sec)
VSUBE - earth-fixed velocity (m/sec)
GAME - earth-fixed flight path angle (deg)
AZE - earth-fixed azimuth (deg)
THRST - total vehicle thrust (lbf)
XMLB - total weight (lbm)

Line 2

R - radius from center of earth (m)
VSUBI - inertial velocity (m/sec)
GAMI - inertial flight path angle (deg)
AZI - inertial azimuth (deg)
GCLAT - geocentric latitude (deg)
LØNG - longitude (deg) (positive east of Greenwich)

LINE 3

ALT - altitude above ellipsoid (m)
VSUBR - relative velocity (m/sec)
GAMR - relative flight path angle (deg)
GDLAT - geodetic latitude (deg)
NØDE - inertial descending node (deg)

LINE 4

Z8-X - Shuttle 8 Z position coordinate, MASTRE X coordinate (m)
X8-4 - Shuttle 8 X position coordinate, MASTRE Y coordinate (m)
Y8-Z - Shuttle 8 Y position coordinate, MASTRE Z coordinate (m)
ZD8-W - Shuttle 8 Z velocity coordinate, MASTRE W coordinate (m/sec)
XD8-U - Shuttle 8 X velocity coordinate, MASTRE U coordinate (m/sec)
YD8 V - Shuttle 8 Y velocity coordinate, MASTRE V coordinate (m/sec)

LINE 5

CHIP - pitch attitude angle (deg)

CHIY - yaw attitude angle (deg)

CHIR - roll attitude angle (deg)

TIMP - time to impact point (sec)

LTIMP - geocentric latitude of instantaneous (vacuum) impact point (deg)

LNGMP - longitude of instantaneous (vacuum) impact point (deg)

LINE 6

THERI - pitch attitude angle in Boost Reference coordinate system (deg)

PSIRI - yaw attitude angle in Boost Reference coordinate system (deg)

PHIRI - roll attitude angle in Boost Reference coordinate system (deg)

INCL - inclination (deg)

RANGE - relative range from the launch site to vehicle subpoint (n.mi)

RNGAN - inertial range angle between the launch site and the vehicle radius vector (deg)

Lines 7, 8, 9, 10, 11, 12, 13, and 14 provide data for all of the six propulsion systems; columns 1 through 6 represent the output of the MPS, SRB, LBM, OMS, RCS, and any other additional system. The lines of output are defined.

LINE 7

TH___ - total thrust of the propulsion system (lbf)

LINE 8

TV___ - vacuum thrust (lbf)

Line 9

ISP__ - specific impulse (sec)

Line 10

WD___ - fuel flowrate (lbm/sec)

Line 11

___F - fuel on board (lbm)

Line 12

___D - depleted fuel (lbm)

Line 13

DP___ - pitch gimbal angle (deg)

Line 14

DY___ - yaw gimbal angle (deg)

Line 15

TAU - throttle setting of MPS engines
SRMAE - SRM exit area (m^2)
PDØME - LOX bulkhead pressure indicator (lbf/m^2)
SRMET - SRM-ET shear load indicator (lbf)
ØRBET - Orbiter-ET attack load indicator (lbf)
CH_VL - Characteristic velocity (m/sec)

Line 16

TN_L - turning losses (m/sec)
GV_L - gravity losses (m/sec)
DR_L - drag losses (m/sec)

BKP_L - back pressure losses (m/sec)

GIM_L - gimbal losses (m/sec)

ID_VL - ideal velocity (m/sec)

Line 17

MACH - Mach number

ALPHA - angle of attack (deg)

BETA - sideslip angle (deg)

FAA - aerodynamic axial force (lbf)

FAS - aerodynamic side force (lbf)

FAN - aerodynamic normal force (lbf)

Line 18

Q - dynamic pressure (lbf/ft^2)

QALPH - product of Q and angle of attack ($lbf\text{-}deg/ft^2$)

QBETA - product of Q and sideslip angle ($lbf\text{-}deg/ft^2$)

FAB - axial component of base force (lbf)

FMB - base moment (lbf-m)

FNB - normal component of base force (lbf)

Line 19

DRAG - aerodynamic drag force (lbf)

LIFT - aerodynamic lift force (lbf)

VWIND - magnitude of wind velocity (m/sec)

AZW - wind direction (deg)

HEAT - stagnation heating indicator (BTU)

HRATE - stagnation heating rate (BTU/sec)

Line 20

PRESS - atmospheric pressure (lbs/ft^2)

RHØ - atmospheric density ($slugs/ft^3$)

TEMP - atmospheric temperature (K°)

REYNO - Reynolds number

DELPC - pitch misalignment of total thrust vector (deg)

DELYC - yaw misalignment of total thrust vector (deg)

Line 21

EMACH - Mach number based on earth fixed velocity

INBD - inboard elevon setting (deg)

OUTBD - outboard elevon setting (deg)

AXACC - total axial acceleration (less gravity) (g's)

NRACC - total normal acceleration (less gravity) (g's)

LTACC - total lateral acceleration (less gravity) (g's)

Line 22

XCG - longitudinal component of center of gravity (in)

ZCG - normal component of center of gravity (in)

QPEN - dynamic pressure penalty function

11.3 ITERATION OUTPUT

Following the trajectory output are several variables, vectors, matrices, and other pertinent data associated with the iteration and optimization process. The input variable LPRINT, however, limits the amount of printout obtained. Table 11-1 lists the output obtained using the LPRINT options.

Table 11-1. LPRINT Options

OUTPUT GROUP	LPRINT=2	LPRINT=1	LPRINT=0
Trajectory Block Output	All trajectories are printed giving discrete point printouts on the nonconverged trajectories and full printout on the converged trajectory.	Only first and converged trajectories are printed, the first giving only discrete points and the converged giving the full printout.	Same as LPRINT=1
Iteration Output	All formats are printed. Control variable printout consists of all points in control table.	Same as LPRINT=2, except control variable printout is printed full on the first trajectory and first and last points thereafter.	Formats 1, 2, 3, 4, 10, 12 (treated the same as LPRINT=1), and 13.

To separate the various output formats, a number will be used to agree with the nomenclature used in Table 11-1. The output formats are:

1. Trajectory Summary Table. Directly after the trajectory printout, a trajectory summary table is printed which presents the payoff and the errors in the desired equality constraints in the present and previous trajectories. Also presented are the amounts of change requested from the algorithm, the amounts of change obtained, and the percentages of predictability (see Figure 11-2).

MASTRE CASE= 1.0001 LBM(350K)+SRB-MOF1(TC-121-78)-WTR(52 F)/MSN4-WIN/P-AOA/

	MASS	VEL	GAMI	R	INCL	NODE	VELE	Q MAX
OLD	.14582+006	.79919+002	-.51998+000	.38233+005	-.10356+001	.70082+000	.85446+002	-.20838+002
NEW	.14742+006	.59191+002	-.42947+000	.27990+005	-.77839+000	.52664+000	.63960+002	-.15556+002
ASK	.15960+004	-.19980+002	.12999+000	-.95581+004	.25890+000	-.17520+000	-.21362+002	.52096+001
GOT	.16043+004	-.20728+002	.90505-001	-.10243+005	.25721+000	-.17418+000	-.21487+002	.52819+001
PCNT	.10052+003	.10374+003	.69623+002	.10717+003	.99347+002	.99414+002	.10059+003	.10139+003

Figure 11-2. Trajectory Constraint Summary Table

2. Polynominal Coefficients. The determined coefficients required to define the booster pitch and yaw attitude are printed (see Figure 11-3).

```
POLYNOMIAL COEFFICIENTS
AP -.46228877-003  .26674916-001  .00000000  .00000000  .00000000
    .00000000      .00000000      .00000000  .00000000  .00000000
    .00000000      .00000000      .00000000  .00000000  .00000000
    .00000000      .00000000      .00000000  .00000000  .00000000
AY  .74526929-007 -.50519455-002  .00000000  .00000000  .00000000
    .00000000      .00000000      .00000000  .00000000  .00000000
    .00000000      .00000000      .00000000  .00000000  .00000000
    .00000000      .00000000      .00000000  .00000000  .00000000
```

Figure 11-3. Polynominal Coefficient Printout

3. COV Lambdas. The initiation time of the min-H phase is printed along with the seven variables associated with the Euler-Lagrangian multiplers of the calculus of variations. These variables, except for the time, are determined after the second iteration and are zero prior to this iteration (see Figure 11-4).

```
COV LAMBDAS
    .123770+003   .355632+002   .248401+002   .624147+001   .974007-003   .104533+000   .360407-002   .177780+000
```

Figure 11-4. COV Lambdas

4. Influence Coefficients. This block of output provides the user with the partial derivatives, or influence coefficients, of the payoff and the constraints with respect to the control parameter sited in the KDB array. The size of this output block varies as a function of the number of equality constraints (left-to-right on the page) and the number of equality parameters (top-to-bottom) (see Figure 11-5).

```
INFLUENCE COEFFICIENTS
-.10098101+004  .29744164+002  .21507260-001  .28098602+003  .12861876-001 -.51530530-002  .11940711+002  .00000000
-.10098101+004  .30387598+002  .14729061-001  .31344432+002  .13127604-001 -.53652113-002  .00000000       .00000000
 .51523334-001  .11153100+003 -.38314379+000 -.36700819+005  .22359980+000 -.26768133+000  .88338764+002  .25920334+002
 .15457000+000 -.45646711+002 -.12490132-001  .70867886+004  .82378609+000 -.97051291+000 -.33763089+002 -.64069366+001
```

Figure 11-5. Influence Coefficients Table

5. I-PSI-PSI. This output block provides the printout of the optimization weighting matrix, W_a, presented in the Engineering manual, which is based on integrated values of the impulse functions. The output presents a square matrix (although the first row, except for the first element, is not shown) based upon the number of constraints plus the payoff (see Figure 11-6).

```
I PSI PSI
.00000000
.00000000   .29408676+002  -.38711282+000  -.11037954+005  -.52144438-001   .29508381-001   .94979728+001   .00000000
.00000000  -.38711282+000   .84628330-002   .14361662+003  -.14067661-003   .95409015-004  -.70314177-001   .00000000
.00000000  -.11037954+005   .14361662+003   .42303917+007  -.20717193+001   .15011922+001  -.36134353+004   .00000000
.00000000  -.52144438-001  -.14067661-003  -.20717193+001   .76681243-002  -.46279164-002   .55022724-002   .00000000
.00000000   .29508381-001   .95409015-004   .15011922+001  -.46279164-002   .29912259-002  -.49865041-002   .00000000
.00000000   .94979728+001  -.70314177-001  -.36134353+004   .55022724-002  -.49865041-002   .58377409+001   .00000000
.00000000   .00000000       .00000000       .00000000       .00000000       .00000000       .00000000       .00000000
```

Figure 11-6. I-PSI-PSI Output

6. PSUBI, WIBT. This output block provides the elements of the weighting matrix, W_b, presented in the Engineering manual. PSUBI shows the calculated weighting on each control parameter in the order it appears in the KDB array, and WIBT shows the input and updated values of the weighting of each parameter before calculation (see Figure 11-7).

```
PSUBI
  .33240837-001   .31848042-001   .74806789-003   .31757520-002

WIBT
  .10000000+001   .10000000+001   .10000000+001   .10000000+001
```

Figure 11-7. PSUBI, WIBT Output

7. Total I-SY-SY Matrix. This output block presents the matrix product of the influence coefficients and the weighting matrix before inversion of the matrix. As in the printout of the I-PSI-PSI, the matrix is square. Only the first element of the first row is shown, with the size dependent on the number of constraints plus the payoff (see Figure 11-8).

```
TOTAL I SY SY MATRIX
 .66372195+005
-.19757179+004   .10414846+003  -.38174980+000  -.14819181+005  -.12748553+000   .13757573+000   .33568756+002   .30913771+001
-.11956472+001  -.38174980+000   .85954295-002   .15407023+003  -.22208709-003   .20442654-003  -.85757809-001  -.71751080-002
-.10437826+005  -.14819181+005   .15407023+003   .54001546+007   .10462669+002  -.13045413+002  -.66870957+004  -.85583076+003
-.85351069+000  -.12748553+000  -.22208709-003   .10462669+002   .98716528-002  -.72161352-002  -.62945309-001  -.12425796-001
 .34503319+000   .13757573+000   .20442654-003  -.13045413+002  -.72161352-002   .60378529-002   .79340311-001   .14556473-001
-.40082621+003   .33568756+002  -.85757809-001  -.66870957+004  -.62945310-001   .79340311-001   .20035166+002   .23998804+001
-.21459626-002   .30913771+001  -.71751081-002  -.85583076+003  -.12425796-001   .14556473-001   .23998804+001   .63296193+000
```

Figure 11-8. Total I-SY-SY Matrix Output

8. WGNU. This output block presents a parameter which is an indicator, or measure, of the overall improvement of the payoff function on an iteration-by-iteration process when the constraint violations are small (see Figure 11-9).

```
WGNU   .66036029+005
```

Figure 11-9. WGNU Output

9. XKAY, E1. This output block provides the current values of the control step variables where XKAY, or k, is the restoration control step variable and E1, or E, is the optimization control step variable (see Figure 11-10).

```
XKAY   .50000000+000 E1   .00000000
```

Figure 11-10. Control Step Variables Printout

10. DRHØ, WDS, GNU. The variables are defined in the following manner (see Figure 11-11):

DRHØ - equality constraint errors used in parameter update equation,

WDS - product of the inverse of the total I-SY-SY matrix and the DRHØ vector,

GNU - Lagrangian multipliers on the constraints used in the optimization procedure (represent the partial derivative of the payoff function with respect to each constraint).

```
DRHO,WDS,GNU
 -.19979628+002  .24471857-001  .32377020+002
  .12999397+000  .58436892+001  .28652670+002
 -.95581450+004 -.34551662-003  .84110510-001
  .25890040+000  .23576525+001 -.15154921+003
 -.17520456+000  .19619523+001 -.70903114+003
 -.21361561+002 -.23551227+000  .75632157+000
  .52095852+001  .69114078+000 -.41794800+002
```

Figure 11-11. DRHØ, WDS, GNU Optimization Parameters

11. Parameter Update and Convergence Test Block. The following variables are printed in this block (see Figure 11-12):

DPAR - parameter update requirement (one per parameter),

ØLD_PCØN - post convergence factor for parameters (one per parameter),

NEW_PCØN - present convergence factor for parameters (one per parameter),

P_SCALE - convergence parameter normalization factor (one per parameter),

TAUT(I) - if thrust event time is being optimized, prints updated value,

WØ1 - if lift-off weight is being optimized, prints updated value,

AZ - if launch azimuth is being optimized, prints updated value,

CPTBL(I) - if the booster pitch attitude is being optimized, prints updated values,

CYTBL(I) - if the booster yaw attitude is being optimized, prints updated values,

THROTTLE PAR(1) - if MPS throttle down time is being optimized, prints updated values,

THROTTLE PAR(2) - if time after Q_{MAX} is being optimized, prints updated values,

THROTTLE PAR(3) - if throttle level is being optimized, prints updated values,

SIDESLIP PAR(I) - if sideslip angles are being optimized, prints updated values.

DPAR	OLD PCON	NEW PCON	P SCALE
-.25622747+001	.00000000	.31043011-004	.99929474+003
.96516606+000	.00000000	-.89653301-002	.99929474+003
.21562244+000	.00000000	-.31038729-001	.37489059+004
.18616141-001	.00000000	-.69508561-001	.15355764+004

TAUT(7)= .40337726+002
TAUT(12)= .97165165+002
CPTBL(2)= .15715622+002
CYTBL(2)= -.29813839+001

Figure 11-12. Parameter Update and Convergence Test Block

12. Control Variable Update. The updated and previous values of the control variables and the increments determined by the algorithm are printed in this output block. These angles and increments are printed in degrees and should be used for next guess values if the trajectory does not converge. If LPRINT is equal to either zero or one, the complete list of tables are printed after the first trajectory and a compact list showing only the first and last value are printed thereafter. If LPRINT is equal 2, the complete list is printed after every trajectory (see Figure 11-13).

TIME	NEW CP	OLD CP	NEW CY	OLD CY	DCP	DCY	
.1237701+003	.6307767+002	.6460001+002	.5406029+001	.5400001+001	-.1522338+001	.6027954-002	1
.1282352+003	.6378446+002	.6526961+002	.5387649+001	.5462654+001	-.1485149+001	-.7500502-001	2
.1327004+003	.6450722+002	.6593921+002	.5373739+001	.5525306+001	-.1431994+001	-.1515670+000	3
.1371656+003	.6523974+002	.6660881+002	.5376095+001	.5587959+001	-.1369073+001	-.2118638+000	4
.1416308+003	.6597884+002	.6727842+002	.5399978+001	.5650612+001	-.1299582+001	-.2506343+000	5
.1460960+003	.6672357+002	.6794802+002	.5444642+001	.5713265+001	-.1224450+001	-.2686230+000	6
.1505612+003	.6747192+002	.6861762+002	.5502051+001	.5775918+001	-.1145704+001	-.2738663+000	7
.1550264+003	.6822225+002	.6928722+002	.5568055+001	.5838571+001	-.1064979+001	-.2705152+000	8
.1594916+003	.6897426+002	.6995683+002	.5638285+001	.5901224+001	-.9825692+000	-.2629383+000	9
.1639568+003	.6972721+002	.7062643+002	.5710824+001	.5963876+001	-.8992154+000	-.2530522+000	10
.1684220+003	.7048049+002	.7129603+002	.5784360+001	.6026529+001	-.8155367+000	-.2421696+000	11
.1728872+003	.7123379+002	.7196563+002	.5858283+001	.6089182+001	-.7318458+000	-.2308992+000	12
.1773524+003	.7198695+002	.7263524+002	.5932319+001	.6151835+001	-.6482879+000	-.2195157+000	13
.1818176+003	.7273977+002	.7330484+002	.6006325+001	.6214488+001	-.5650654+000	-.2081625+000	14
.1862828+003	.7349206+002	.7397444+002	.6080269+001	.6277141+001	-.4823827+000	-.1968717+000	15
.1907479+003	.7424348+002	.7464404+002	.6154134+001	.6339794+001	-.4005672+000	-.1856597+000	16
.1952131+003	.7499457+002	.7531365+002	.6227927+001	.6402446+001	-.3190799+000	-.1745191+000	17
.1996783+003	.7574574+002	.7598325+002	.6301652+001	.6465099+001	-.2375073+000	-.1634471+000	18
.2041435+003	.7649649+002	.7665285+002	.6375319+001	.6527752+001	-.1563610+000	-.1524335+000	19
.2086087+003	.7621767+002	.7732245+002	.6448907+001	.6590405+001	-.1104781+001	-.1414983+000	20
.2130739+003	.7695528+002	.7799205+002	.6522459+001	.6653058+001	-.1036774+001	-.1305985+000	21
.2175391+003	.7769102+002	.7866166+002	.6595992+001	.6715711+001	-.9706423+000	-.1197187+000	22
.2220043+003	.7842512+002	.7933126+002	.6669455+001	.6778364+001	-.9061390+000	-.1089082+000	23
.2264695+003	.7915765+002	.9000086+002	.6742861+001	.6841016+001	-.8430120+000	-.9815539-001	24
.2309347+003	.7988946+002	.8067047+002	.6816221+001	.6903669+001	-.7810084+000	-.8744874-001	25
.2353999+003	.8062020+002	.8134007+002	.6889534+001	.6966322+001	-.7198716+000	-.7678803-001	26
.2398651+003	.8097332+002	.8170000+002	.6929144+001	.7000000+001	-.7266801+000	-.7085598-001	27
.2443303+003	.8088855+002	.8170000+002	.6929631+001	.7000000+001	-.8114505+000	-.7036889-001	28
.2487955+003	.8080459+002	.8170000+002	.6930132+001	.7000000+001	-.8954121+000	-.6986766-001	29
.2487954+003	.8588361+002	.8296257+002	.7401524+001	.7093031+001	.2921043+001	.3084924+000	31
.2572668+003	.8669581+002	.8393531+002	.7473024+001	.7164707+001	.2760496+001	.3083173+000	32
.2657383+003	.8750502+002	.8490906+002	.7545067+001	.7236383+001	.2596963+001	.3086834+000	33
.2742097+003	.8831143+002	.8588081+002	.7617674+001	.7308059+001	.2430621+001	.3096145+000	34
.2826811+003	.8911513+002	.8685355+002	.7690869+001	.7379735+001	.2261577+001	.3111340+000	35
.2911525+003	.8991618+002	.8782630+002	.7764676+001	.7451411+001	.2089886+001	.3132650+000	36
.2996240+003	.9071463+002	.8879905+002	.7839118+001	.7523087+001	.1915588+001	.3160307+000	37
.3080954+003	.9151050+002	.8977179+002	.7914217+001	.7594764+001	.1738704+001	.3194540+000	38
.3165668+003	.9230378+002	.9074454+002	.7989997+001	.7666440+001	.1559239+001	.3235575+000	39
.3250383+003	.9309450+002	.9171728+002	.8066480+001	.7738116+001	.1377221+001	.3283639+000	40
.3335097+003	.9388285+002	.9269003+002	.8143687+001	.7809792+001	.1192820+001	.3338955+000	41
.3419811+003	.9466837+002	.9366278+002	.8221642+001	.7881468+001	.1005597+001	.3401743+000	42
.3504525+003	.9545034+002	.9463552+002	.8300366+001	.7953144+001	.8148180+000	.3472225+000	43
.3589240+003	.9622829+002	.9560827+002	.8379881+001	.8024820+001	.6200243+000	.3550616+000	44
.3673954+003	.9700272+002	.9658102+002	.8460209+001	.8096496+001	.4217039+000	.3637131+000	45
.3758668+003	.9777307+002	.9755376+002	.8541370+001	.8168172+001	.2193106+000	.3731984+000	46
.3843383+003	.9853858+002	.9852651+002	.8623387+001	.8239848+001	.1206581-001	.3835384+000	47
.3928097+003	.9929824+002	.9949926+002	.8706278+001	.8311524+001	-.2010213+000	.3947539+000	48
.4012811+003	.1000446+003	.1004720+003	.8790066+001	.8383200+001	-.4274022+000	.4068653+000	49
.4097525+003	.1007837+003	.1014447+003	.8874769+001	.8454876+001	-.6610174+000	.4198929+000	50
.4182240+003	.1015417+003	.1024175+003	.8960409+001	.8526552+001	-.8758013+000	.4338566+000	51
.4266954+003	.1023095+003	.1033902+003	.9047004+001	.8598228+001	-.1080704+001	.4487759+000	52
.4351668+003	.1030843+003	.1043630+003	.9134575+001	.8669904+001	-.1278681+001	.4646703+000	53
.4436382+003	.1038637+003	.1053357+003	.9223139+001	.8741581+001	-.1472000+001	.4815586+000	54
.4521097+003	.1046459+003	.1063085+003	.9312716+001	.8813257+001	-.1662600+001	.4994597+000	55
.4605811+003	.1054293+003	.1072812+003	.9403324+001	.8884933+001	-.1851909+001	.5183917+000	56
.4690525+003	.1062131+003	.1082540+003	.9494982+001	.8956609+001	-.2040864+001	.5383728+000	57
.4775240+003	.1069967+003	.1092267+003	.9587705+001	.9028285+001	-.2229991+001	.5594206+000	58
.4859954+003	.1077798+003	.1101995+003	.9681513+001	.9099961+001	-.2419693+001	.5815519+000	59

Figure 11-13. Control Variable Update Printout

13. Convergence Indicator. Directly after the control variable update printout is an output that shows the maximum increment (or decrement) required in pitch and yaw attitude (in radians). When the maximum values are less than 0.005 and all the other convergence tests are met, the logical word BETCØN is output as a T, or true, which indicates that convergence has occurred. If BETCØN is equal to F, or false, the convergence tests have not been met and the iteration process continues. The fourth word on this line is KAT which is used to indicate that an ill-conditioned matrix has been encountered during the determination of the W_a weighting matrix in the backward trajectory. KAT = 1 is the ill-condition indicator and KAT = 0 means that no problems have been encountered (see Figure 11-14).

```
DEL CHIP MAX=  .2209+001   DEL CHIY MAX=  .1722+001   BETCON=  F  KAT=  0
```

Figure 11-14. Convergence Indicator Printout

14. Summary Tables. At the conclusion of the converged trajectory, the parameter summary table (Figure 11-15), the orbital element summary table (Figure 11-16), the load indicator table (Figure 11-17), and the weight summary table (Figure 11-18) are printed.

```
PARAMETERS
  TAUT
        .50000000+001  .45000000+002  .58770000+002  .15000000+002  .41230000+002  .40895400+002  .28507395+002  .80000000+001
        .16070000+002  .80000000+001  .10300000+003  .10473379+003  .60000000+001  .00000000     .00000000
   WD1
        .48439989+007
    A2
        .18000000+003
  CPTBL
       -.26487196-001  .16193096+002  .00000000     .00000000     .00000000     .00000000     .00000000     .00000000
  CYTBL
        .42700785-005 -.33112448+001  .00000000     .00000000     .00000000
  THROTTLE PARAMETERS
        .00000000     .40000000+001  .10000000+001
  SIDESLIP CONTROL
       -.23561945-001 -.17976891-001 -.52359878-002
```

Figure 11-15. Parameter Summary Table

LBM(350K)+SRB-MOF1(TC-121-78)-WTR(52 F)/MSN4-WIN/P-AOA/

TIME	.48013655+003 SEC	RANGE	.14031816+007 M	RANGE ANGLE	.12422180+002 DEG
INERTIAL VELOCITY	.77338599+004 M/SEC	RADIUS	.64660500+007 M	FLIGHT PATH ANGLE	.64943012+000 DEG
INCL	.76183999+002 DEG	DES. NODE	-.29829153+001 DEG	FLIGHT AZIMUTH	.18000000+003 DEG
GDLAT	.22120536+002 DEG	GCLAT	.21986606+002 DEG	LONG	-.12320688+003 DEG
INERTIAL AZIMUTH	-.17332885+003 DEG	C1	.50140927+011 M**2/SEC		
WEIGHT	.33463186+006 LBS	C3	-.63141969+008 M**2/SEC**2		

KEPLERIAN ORBITAL PARAMETERS

ECNTRICY	.79364951-001	APOGEE RADIUS	.68137906+007 M	PERIGEE RADIUS	.58117642+007 M
APOGEE HEIGHT	.23522172+003 NM	PERIGEE HEIGHT	-.30582927+003 NM		

Figure 11-16. Orbital Element Summary Table

LOAD INDICATOR	FIRST STAGE MINIMUM TIME	FIRST STAGE MINIMUM VALUE	FIRST STAGE MAXIMUM TIME	FIRST STAGE MAXIMUM VALUE	SECOND STAGE MINIMUM TIME	SECOND STAGE MINIMUM VALUE	SECOND STAGE MAXIMUM TIME	SECOND STAGE MAXIMUM VALUE	UNITS
Q ALPHA	48.00	-2105.85	85.00	934.69	163.77	-13.27	123.77	211.47	LB-DEG/FT**2
Q BETA	18.00	-338.49	13.00	35.63	123.77	-293.90	123.77	.00	LB-DEG/FT**2
Q	.00	.00	59.75	649.99	322.40	.02	123.77	53.88	LB/FT**2
LONG ACC	.00	.00	103.00	2.66	123.77	1.02	438.94	2.97	G'S
LAT ACC	79.50	.00	18.00	.01	466.94	.00	123.77	.01	G'S
NORM ACC	.00	.00	123.77	.21	123.77	.21	385.40	.53	G'S
TOTAL ACC	.00	.00	103.00	2.66	123.77	1.05	474.14	3.00	G'S
STAG HEAT	123.77	.00	123.77	.00	480.14	.00	480.14	.00	BTU/FT**2

Figure 11-17. Load Indicator Table

LBM(350K)+SRB-MOF1(TC-121-78)-WTR(52 F)/MSN4-WIN/RTLS/

	NOMINAL		AOA		RTLS	
PAYLOAD	32234.		32234.		32234.	
OMS KIT	0.		0.		0.	
MISSION KIT	1608.		1608.		1608.	
PERSONNEL	2796.		2796.		2796.	
ORBITER - EMPTY	141041.		141041.		141041.	
- NONPROPULSIVE CONSUMABLES	4259.		4259.		4259.	
- USABLE RCS (RCS LOADED)	5890.	(7508.)	4632.		4632.	
- RCS RESERVES	828.		524.		524.	
- RCS RESIDUALS	579.		579.		579.	
- USABLE OMS (OMS LOADED)	23027.	(25064.)	10170.		0.	
- OMS PROPELLANT (DEORBIT)	0.	(.0 F/S)	0.	(.0 F/S)		
- OMS RESERVES	1238.		289.		289.	
- OMS RESIDUALS	799.		799.		799.	
SSME EMPTY	20484.		20484.		20484.	
ORBITER AT ORBIT INSERTION		234783.		219415.		
MECO OMS MANEUVER - OMS PROPELLANT	0.		0.			
- RCS PROPELLANT	0.		0.			
TRAPPED MPS PROPELLANT - ORBITER	1337.		2268.		2268.	
- SSME	1383.		1383.		1383.	
FLIGHT PERFORMANCE RESERVE - ORBITER	2365.		1461.		1671.	
- SSME	0.		0.		0.	
ORBITER AT OMS IGNITION		239868.		224527.		214567.
RCS PROPELLANT	211.		211.		211.	
EXTERNAL TANK - EMPTY	73495.		73495.		73495.	
- RESIDUALS	4378.		4480.		4480.	
FLIGHT PERFORMANCE RESERVE - ET	3518.		2363.		2067.	
FUEL BIAS	1047.		1100.	MARGIN	1100.	
EXCESS MPS PROPELLANT (MPS PROP LOADED)	10534.	(1597164.)	8284.	(8781.)	9087.	
EXPELLED MPS PROPELLANT	230.		154.		154.	
INJECTED WEIGHT AT MECO		333280.		314615.		305161.
PROPELLANT BURNED-AOA/RTLS TO MECO						
MPS	788461.		791683.		790967.	
OMS	0.		11490.		10536.	
RCS	0.		1562.		1562.	
OMS PROPELLANT DUMPED (RCS BURN)	0.		2316.		13440.	
EXPELLED MPS PROPELLANT	0.		76.		76.	
WEIGHT AT AOA/RTLS		1121742.				
MPS PROPELLANT BURNED-LBM SEPARATION TO RTLS/AOA	96246.					
WEIGHT AT LBM SEPARATION		1217988.				
LBM INERT WEIGHT	32653.					
PROPELLANT BURNED-SRM SEPARATION TO LBM SEPARATION						
MPS	277387.					
LBM	142461.					
ET+ORBITER AFTER SRM SEPARATION		1670489.				
SRM INERT WEIGHT	344521.					
SRM RESIDUAL PROPELLANT	0.					
INERTS CONSUMED	0.					
PROPELLANT BURNED-LBM IGNITION TO SRM SEPARATION						
MPS	373314.					
LBM	207360.					
SRM	2122344.					
LBM IGNITION (T+5 SEC)		4718027.				
PROPELLANT BURNED-LIFTOFF TO LBM IGNITION						
MPS	15062.					
SRM	108486.					
LIFTOFF (T/W=1.0)		4841576.				
SRB PROPELLANTS	1634.					
SRB INERTS	4.					
MPS PROPELLANT	785.					
SRB IGNITION COMMAND (T-0)		4843999.				
MPS THRUST BUILDUP + SRM IGNITION DELAY	12426.					
OVERFILL,DRAINBACK,AND BOILOFF	8641.					
NONPROPULSIVE CONSUMABLES	51.					
PRELAUNCH WEIGHT (T-5 MIN)		4865167.				

Figure 11-18. Weight Summary Table

The variables of Figure 11-16 are defined as:

TIME - time (sec)

RANGE - relative range (m)

RANGE_ANGLE - relative range angle (deg)

INERTIAL VELOCITY - inertial velocity (m/sec)

RADIUS - radius (m)

FLIGHT_PATH_ANGLE - inertial flight-path angle (deg)

INCL - inclination (deg)

DES. NØDE - descending node (deg)

FLIGHT AZIMUTH - launch azimuth (deg)

GDLAT - geodetic latitude (deg)

GCLAT - geocentric latitude (deg)

LØNG - longitude (deg)

INERTIAL_AZIMUTH - inertial azimuth (deg)

C1 - angular momentum magnitude (m^2/sec)

WEIGHT - mass (lbm)

C3 - vis viva integral (twice the energy) (m^2/sec^2)

ECNTRICY - orbital eccentricity (unitless)

APØGEE_RADIUS - radius of apogee (m)

PERIGEE_RADIUS - radius of perigee (m)

APØGEE_HEIGHT - apogee altitude above equatorial radius (nmi)

PERIGEE_HEIGHT - perigee altitude above equatorial radius (nmi)

11.4 Post-Processor Output

This subsection describes the output provided by two post processor modules optional available to the user.

11.4.1 Output Table Post-Processor

A nonzero input value of the variable NTABLE indicates that output tables generated by this post-processor module are desired. Upon termination of the trajectory simulation, the element 'ELTTBL', which contains

the Namelist array 'INPUT2', is automatically placed into the run stream. Therefore, the user should update this element before executing the table option (see Section 10 for input requirements).

The output table post-processor module, which is called B1PTBL, has been programmed to allow full flexibility to the user. By modification of a few lines of code, the user can select the number and the types of output variables required in the tables. The number of tables is controlled by the PARAMETER variable MT (which currently is set to 14) and the types of variables are specified trhough the array NNAME (10,MT). Figure 11-19 represents the currently programmed values of this array. The absolute value of quantities represents location of the variables in the trajectory block printout array (see Section 11.2). Figure 11-20 provides a location cross reference of these variables. The sign of the quantity signifies the types of units preferred such that the positive (+) sign signifies metric units and a minus (-) sign signifies english units.

```
CARD NO.

39.        DATA NNAME/
40.      *  1,   7,  13,   8,   9,  10,  11,  17,  12,  34,
41.      *  1,   2,   3,   4,  14,  15,  16, 111, 112,  35,
42.      *  1,  19,  20,  21,  22,  23,  24, 124, 125, 126,
43.      *  1,  25,  26,  27,  31,  32,  33, 119, 120,   0,
44.      *  1, -37,  85, -38, -39, -40, -41,  -5,   0,   0,
45.      *  1, -37, -43,  49, -55, -61, -67,  73,  79,  85,
46.      *  1, -38, -44,  50, -56, -62, -68,  74,  80,  86,
47.      *  1, -39, -45,  51, -57, -63, -69,  75,  81,   0,
48.      *  1, -40, -46,  52, -58, -64, -70,  76,  82,   0,
49.      *  1, -41, -47,  53, -59, -65, -71,  77,  83,   0,
50.      *  1,  97,  98,  99,-103,-104,-105, 121, 122, 123,
51.      *  1,  90,  96,  91,  92,  93,  94,  95, -87, -88,
52.      *  1,-100,-106,-109,-102,-108,-110,-101,-107,-115,
53.      *  1,  28,  29,  30, 113, 114, 117,   0,   0,   0/
```

Figure 11-19. NNAME Array

	0	1	2	3	4	5	6	7	8	9
0	------	TIME	VSUBE	GAME	AZE	THRST	XPLB	R	VSUBI	GAMI
1	AZI	GCLAT	LONG	ALT	VSUBR	GAMR	AZR	GDLAT	NODE	Z8-X
2	X8-Y	Y8-Z	Z08-W	X08-U	Y08-V	CHIP	CHIY	CHIR	TIMP	LTIMP
3	LNGMP	THERI	PSIRI	PHIRI	INCL	RANGE	RNGAN	THWPS	THSRM	THLBM
4	THOMS	THRCS	TH	TVMPS	TVSRM	TVLBM	TVOMS	TVRCS	TVAC	ISPMP
5	ISPSR	ISPLB	ISPOM	ISPRC	ISP	WOMPS	WOSRM	WOLBM	WOOMS	WORCS
6	WDOT	MPS F	SRM F	LBM F	OMS F	RCS F	FUEL	MPS D	SRM D	LBM D
7	OMS D	RCS D	SPENT	OPMPS	OPSRM	OPLBM	OPOMS	OPRCS	DELP	DYMPS
8	DYSRM	DYLBM	DYOMS	DYRCS	DELY	TAU	SRMAE	PDOME	SRMET	DRBET
9	CH VL	TN L	GV L	OR L	BKP L	GIM L	TD VL	MACH	ALPHA	BETA
10	FAA	FAS	FAN	Q	QALPH	QBETA	FAB	FNB	FNB	DRAG
11	LIFT	VWIND	AZW	HEAT	HRATE	PRESS	RHO	TEMP	REYNO	DELPC
12	DELYC	EMACH	INBO	OUTBO	AXACC	NRACC	LTACC	XCG	ZCG	

Figure 11-20. Variable Location Cross Reference

Using the first line of the NNAME array as an example, the module output provides publishable tables as shown in Figure 11-21. Upon completion of the desired tables, a summary table is provided which automatically defines the variable names, the table location, the desired units, and the definition of the variables are specified through the NNAME array (see Figure 11-22).

MSFC-EL24 16 MAR 1981 LRM TRAJECTORY - TEST CASE 1

TABLE NO. 1

	TIME	R	ALT	VSUBI	GAMI	AZI	GCLAT	GDLAT	LONG	INCL
IGNITION	234.40	6466050.	93968.	2527.40	7.083	-178.95	32.15	32.325	-121.165	90.893
	238.90	6467422.	95306.	2552.00	6.804	-178.87	32.05	32.224	-121.186	90.957
	238.90	6467422.	95306.	2552.00	6.804	-178.87	32.05	32.224	-121.186	90.957
	245.55	6469832.	97266.	2527.19	6.740	-178.74	31.90	32.075	-121.217	91.073
	245.55	6469832.	97266.	2527.19	6.740	-178.74	31.90	32.075	-121.217	91.073
BEGIN MINM	245.55	6469832.	97266.	2527.19	6.740	-178.74	31.90	32.075	-121.217	91.073
	453.46	6461719.	88629.	573.71	-28.448	135.55	29.08	29.239	-121.809	52.263
	453.46	6461719.	88629.	573.71	-28.448	135.55	29.08	29.239	-121.809	52.263
	480.98	6454121.	81019.	507.30	-32.530	84.17	29.04	29.200	-121.810	29.566
	480.98	6454121.	81019.	507.30	-32.530	84.17	29.04	29.200	-121.810	29.566
	534.40	6442197.	69168.	1137.11	-6.941	31.29	29.27	29.432	-121.760	63.056
	534.40	6442197.	69168.	1137.11	-6.941	31.29	29.27	29.432	-121.760	63.056
G-LIMIT	569.72	6441076.	68180.	1841.42	3.150	22.72	29.68	29.848	-121.674	70.396
	573.64	6441573.	68697.	1935.58	4.143	22.08	29.75	29.912	-121.661	70.955
INTERMEDIATE	573.64	6441573.	68697.	1935.58	4.143	22.08	29.75	29.912	-121.661	70.955
	582.64	6442826.	70000.	2180.81	3.655	20.57	29.90	30.066	-121.630	72.269
	582.64	6442826.	70000.	2180.81	3.655	20.57	29.90	30.066	-121.630	72.269
	588.64	6443606.	70816.	2287.77	3.040	20.02	30.01	30.178	-121.607	72.757
INJECTION	588.64	6443606.	70816.	2287.77	3.040	20.02	30.01	30.178	-121.607	72.757

Figure 11-21. Table Module Output Example

DEFINITION OF TABLE OUTPUT

PARAMETER	TABLE	UNITS	DEFINITION
TIME	1	SECONDS	INSTANTANEOUS TIME FROM SRB IGNITION
R	1	METERS	RADIUS FROM EARTH CENTER
ALT	1	METERS	ALTITUDE ABOVE REFERENCE ELLIPSOID
VSUBI	1	METERS/SEC	INERTIAL VELOCITY
GAMI	1	DEGREES	INERTIAL FLIGHT PATH ANGLE
AZI	1	DEGREES	INERTIAL AZIMUTH
GCLAT	1	DEGREES	GEOCENTRIC LATITUDE
GDLAT	1	DEGREES	GEODETIC LATITUDE
LONG	1	DEGREES	LONGITUDE
INCL	1	DEGREES	ORBITAL INCLINATION ANGLE
TIME	2	SECONDS	INSTANTANEOUS TIME FROM SRB IGNITION
VSUBE	2	METERS/SEC	EARTH-FIXED VELOCITY
GAME	2	DEGREES	EARTH-FIXED FLIGHT PATH ANGLE
AZE	2	DEGREES	EARTH-FIXED AZIMUTH
VSUBR	2	METERS/SEC	WIND RELATIVE VELOCITY
GAMR	2	DEGREES	WIND RELATIVE FLIGHT PATH ANGLE
AZR	2	DEGREES	WIND RELATIVE AZIMUTH
VWIND	2	METERS/SEC	WIND VELOCITY
AZW	2	DEGREES	WIND AZIMUTH
RANGE	2	KILOMETERS	RELATIVE SURFACE RANGE
TIME	3	SECONDS	INSTANTANEOUS TIME FROM SRB IGNITION
ZB-X	3	METERS	DOWNRANGE POSITION COMPONENT(INERTIAL)
XB-Y	3	METERS	VERTICAL POSITION COMPONENT(INERTIAL)
YB-Z	3	METERS	CROSSRANGE POSITION COMPONENT(INERTIAL)
ZDB-W	3	METERS/SEC	DOWNRANGE VELOCITY COMPONENT(INERTIAL)
XDB-U	3	METERS/SEC	VERTICAL VELOCITY COMPONENT(INERTIAL)
YDB-V	3	METERS/SEC	CROSSRANGE VELOCITY COMPONENT(INERTIAL)
AXACC	3	G'S	TOTAL AXIAL ACCELERATION
NRACC	3	G'S	TOTAL NORMAL ACCELERATION
LTACC	3	G'S	TOTAL LATERAL ACCELERATION
TIME	4	SECONDS	INSTANTANEOUS TIME FROM SRB IGNITION
CHIP	4	DEGREES	PITCH ATTITUDE (INERTIAL)
CHIY	4	DEGREES	YAW ATTITUDE (INERTIAL)
CHIR	4	DEGREES	ROLL ATTITUDE (INERTIAL)
THERI	4	DEGREES	PITCH ANGLE IN BOOSTER REF SYSTEM
PSIRI	4	DEGREES	YAW(HEADING) ANGLE IN BOOSTER REF SYSTEM
PHIRI	4	DEGREES	ROLL ANGLE IN BOOSTER REF SYSTEM
DELPC	4	DEGREES	COMPOSITE PITCH GIMBAL ANGLE
DELYC	4	DEGREES	COMPOSITE YAW GIMBAL ANGLE
TIME	5	SECONDS	INSTANTANEOUS TIME FROM SRB IGNITION

Figure 11-22. Summary Output Table Example

11.4.2 Plotting Post-Processor (Demand Terminal Only)

Upon termination of the trajectory simulation, the user, in a demand mode, has the option of creating a 'list-directed' file for use in the MIPS plotting package. Use of this package allows the plotting of all variables (as defined in Figure 11-23) in the manner selected by the user. For additional information on the MIPS plotting package, the user should consult Reference 12. Examples of generated plots are shown in Figure 11-24.

```
 DESCRIPTOR FORMAT - FILENAME.XNAME.YNAME.DRIVER
 DRIVER WILL DEFAULT TO XNAME
 ENTER FILENAME.HELP FOR TUTORING
 SPECIFY DESCRIPTOR NO 1 CURRENTLY  OFF
MIPS>MASTREDATA.TIME.XXX
 LDF    MASTREDATA   LAST WRITE 03/25/81  12:21:46    24% OF USABLE SPACE
   NAME XXX       IS NOT IN THE DICTIONARY FOR   MASTREDATA
  DO YOU WANT TO LOOK AT THE DICTIONARY?
MIPS>YES
TIME      VSUBE     GAME      AZE       THRST     XMLB      R         VSUBI     GAMI
AZI       GCLAT     LONG      ALT       VSUBR     GAMR      AZR       GDLAT     MODE
ZS-X      XS-Y      YS-Z      ZDS-W     XDS-U     YDS-V     CHIP      CHIY      CHIR
TIMP      LTIMP     LNGMP     THERI     PSIRI     PHIRI     INCL      RANGE     RNGAN
THMPS     THSRM     THLBM     THOMS     THRCS     TH        TVMPS     TVSRM     TVLBM
TVOMS     TVRCS     TVAC      ISPMP     ISPSR     ISPLB     ISPOM     ISPRC     ISP
WDMPS     WDSRM     WDLBM     WDOMS     WDRCS     WDOT      MPSF      SRMF      LBMF
OMSF      RCSF      FUEL      MPSD      SRMD      LBMD      OMSD      RCSD      SPENT
DPMPS     DPSRM     DPLBM     DPOMS     DPRCS     DELP      DYMPS     DYSRM     DYLBM
DYOMS     DYRCS     DELY      TAU       SRMAE     PDOME     SRMET     ORBET     CHVL
TNL       GVL       DRL       BKPL      GIML      IDVL      MACH      ALPHA     BETA
FAA       FAS       FAN       Q         QALPH     QBETA     FAB       FMB       FNB
DRAG      LIFT      VWIND     AZW       HEAT      HRATE     PRESS     RHO       TEMP
REYNO     DELPC     DELYC     EMACH     INBD      OUTBD     AXACC     NRACC     LTACC
XCG       ZCG       BLANK     BLANK
 SPECIFY DESCRIPTOR NO 1 CURRENTLY  OFF
MIPS>
```

Figure 11-23. Output Variable Dictionary

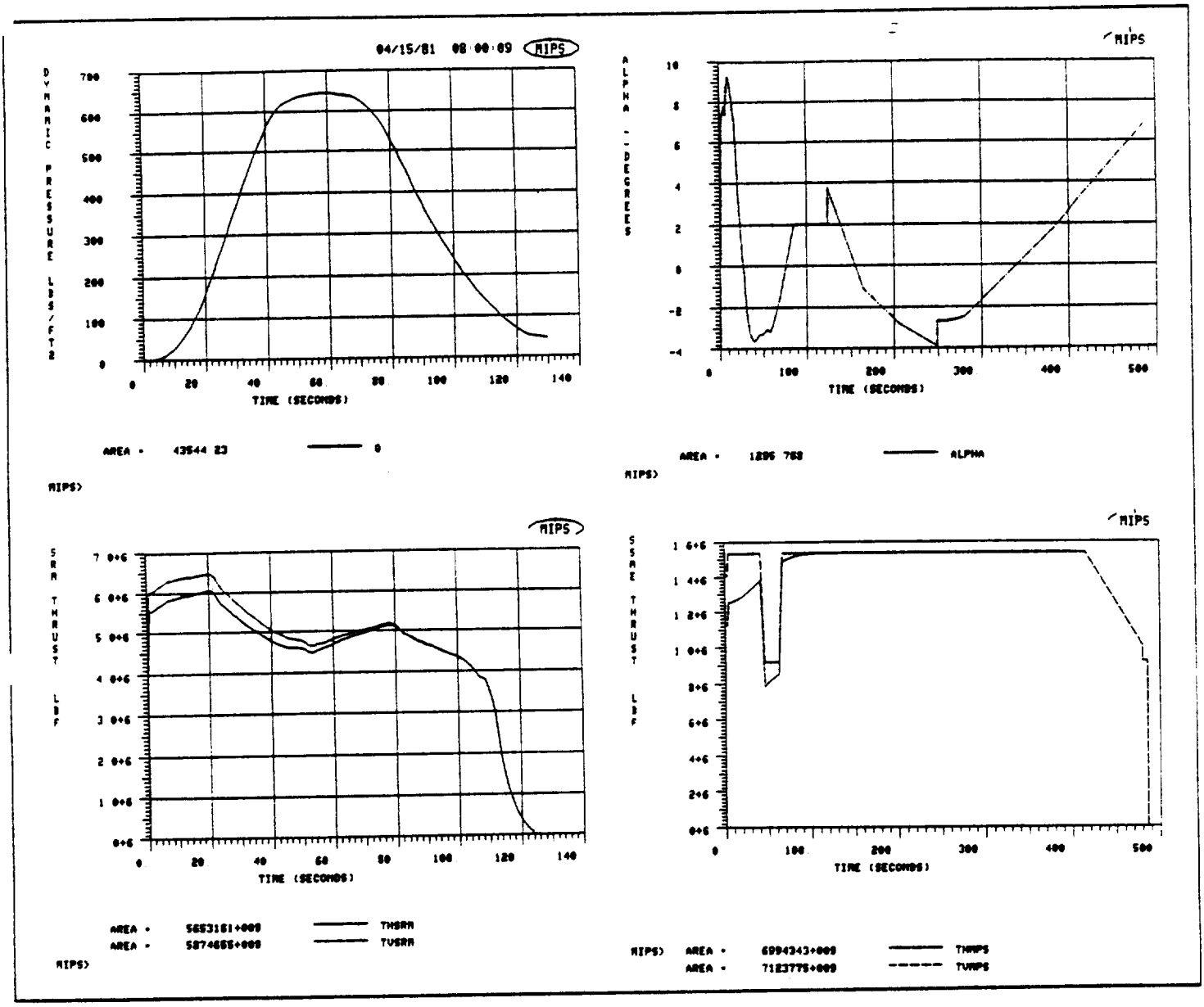


Figure 11-24. Examples of Plotting Post-Processor Output

Section 12

EFFICIENT USE OF MASTRE

The following subsections contain several comments and suggestions which will assist in using MASTRE more efficiently.

12.1 BODY AXIS COORDINATE SYSTEM

In order to agree with the present Shuttle Standard coordinate system, all vehicle input data uses the coordinate system shown in Figure 12-1, with the longitudinal axis as the x-axis, the lateral axis as the y-axis (out the right wing), and the vertical axis as the z-axis (down). At lift-off, the vehicle will be oriented with the z-axis pointed in the opposite direction of the input azimuth (FAZ), thus directing the orbiter tail in the direction of FAZ.

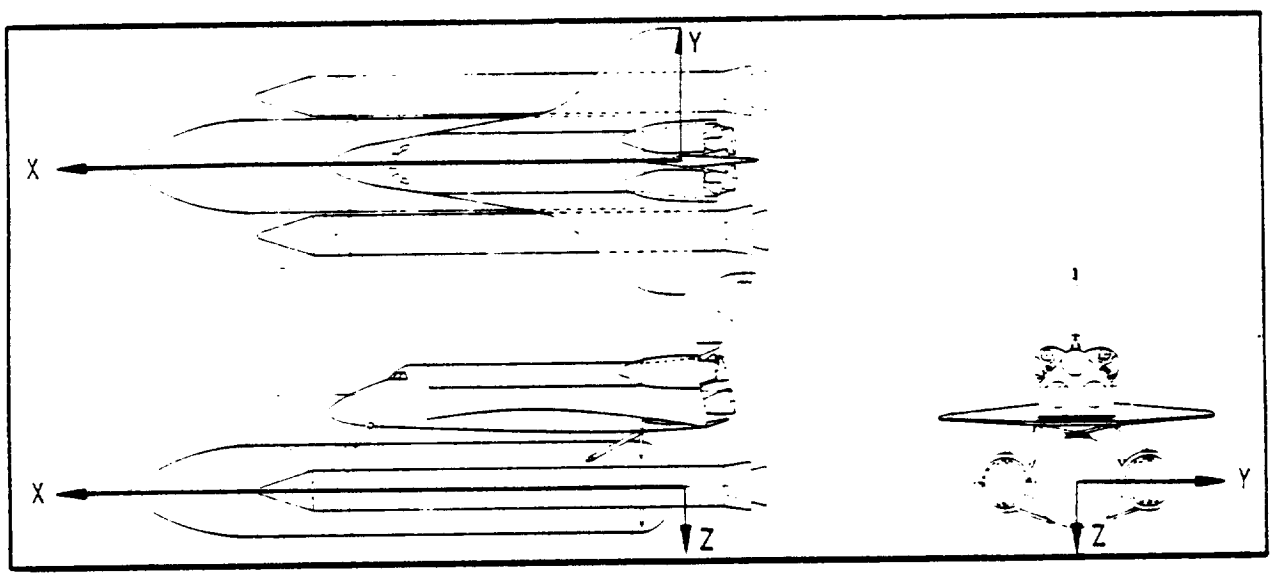


Figure 12-1. Input Body Axis System

12.2 ATTITUDE CONTROL

Vehicle attitude control polynomials are stage-dependent and therefore are carried over thrust events of a given stage. The order of the polynomials used to describe the attitude histories should be chosen with the following points in mind:

- The higher the order of the polynomials, the more optimum the trajectory can be.
- The higher the order of the polynomials, the more parameters are required and therefore the longer the run will take to converge. Therefore, the lowest order polynomial which yields an acceptable solution should be used when rapid solutions are required.
- An additional polynomial is useful in the event of a discontinuity in acceleration history.

In addition to the above points concerning selection of the order of the attitude polynomials, proper spacing of the points in the time tables can increase the efficiency of the program. The input tables TTBL contain the time points corresponding to the attitude angles in the input tables CPTBL and CYTBL. Since the program must fit the polynomials with the input times held constant, an even distribution of time points over the duration of the stage will provide the most stable condition for the program. (Consider, for example, the difficulty of fitting a nonlinear function to points grouped very closely in time, and the sensitivity of the function to very slight variations in the input time points when they are so grouped.)

The user should always keep in mind when using the MASTRE program that the booster attitude solutions are not the absolute optimum as would be determined from a calculus of variation solution. Therefore, it is advisable to examine several orders of polynomials before deciding on a trajectory.

12.3 JUMP-START

Whenever using the jump-start option the user must remember the following items:

- The initial state including losses and propellant flowrates must be input into the array VIV unless the ISTART flag has been used in the preceding trajectory. If the ISTART flags have been used, the variables VIV, TZERØ, WØ1, and JUMP are preloaded. TZERØ, WØ1, and JUMP must be input if ISTART has not been used.
- All intermediate constraints which occur prior to the jump-start should be turned off.
- The values of KDB which relate to time duration of thrust events prior to the jump-start and booster attitude are set equal to zero by the program and therefore do not require additional input.

Section 13

PROGRAMMER'S MANUAL

This manual furnishes information required by a programmer for modification or conversion of the Minimum Hamiltonion Ascent Shuttle Trajectory Evaluation (MASTRE) program. Included in this manual is a general program flow diagram, a subroutine cross reference, and a program listing. The code has been programmed in the ASCII FORTRAN programming language and therefore should be compatible with other computer systems.

Before a programmer begins work on this program he should acquire an understanding of the general flow of the program placing special emphasis on the integration package subroutines DESOLV and DPIR. Whenever these subroutines are called they control the flow of the integration process until termination (which is caused by calling the subroutine RTMRK). Since both forward and backward integrations are required by the steepest ascent algorithm, these subroutines are the focal points of the program.

Section 14

GENERAL PROGRAM FLOW

The program structure as shown in Figure 14.1 consists of seven primary segments:

1. Initiation
2. Demand terminal communication (demand mode only)
3. Forward trajectory integration and elevation
4. Backward trajectory integration
5. Partial derivative calculation
6. Parameter update and convergence tests
7. Table output

These seven segments are described briefly in the following paragraphs.

14.1 INITIATION

Initial flow begins by asking whether the run is a batch job or on the demand terminal. If the job is on the demand terminal the flow is transferred to the second segment of the program which provides communication between the user and the computer. If the job is a batch run the flow is transferred to the AINIT subroutine to preset variables and constants and to read input data from either namelist arrays or the aerodynamic data base.

14.2 DEMAND TERMINAL COMMUNICATION

The demand terminal user can directly interface with the computer via the MIPS system subroutines. As shown in Figure 14-1, the user is allowed many optimal paths depending upon his choice of numbers selected from the menus.

14.3 FORWARD TRAJECTORY INTEGRATION AND EVALUATION

The forward trajectory integration and evaluation is performed by calling AFORUN and AFORND. Upon return from these two subroutines, the

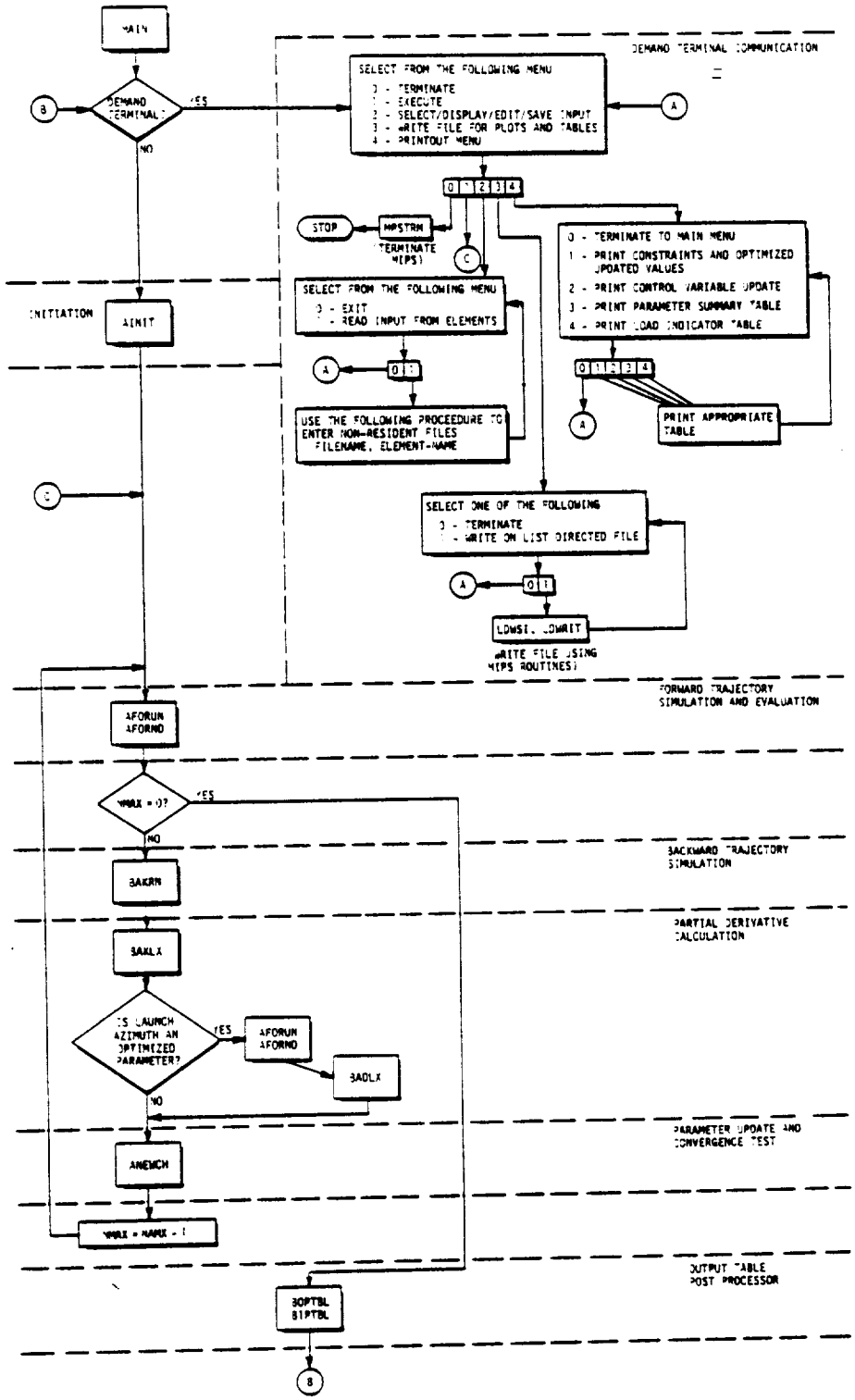


Figure 14-1. 'MASTRE' Flow Diagram

values of the constraint errors and the payoff are known for the input parameter set and the initial values for the backward trajectory are determined. The calling of the forward trajectory is the beginning of the iteration cycle.

14.4 BACKWARD TRAJECTORY INTEGRATION

A backward trajectory integration is performed from the terminal point of the forward trajectory to the min-H initialization point by calling subroutine BAKRN. During this time, the adjoint differential equations are integrated based upon interpolated values of the state variables stored during the forward trajectory. Impulse response functions are also evaluated to determine the effects of the pitch and yaw attitude angles on the trajectory.

14.5 PARTIAL DERIVATIVE CALCULATION

The partial derivatives of the payoff, terminal, and intermediate constraints with respect to the control parameters are determined by calling subroutine BADLX. These partial derivatives are determined by a combination of the results obtained by integration of the adjoint equations and forward numerical differences determined prior to min-H initiation. If launch azimuth is a control parameter (denoted by KDB (17) = 1), however, the associated partial derivatives are determined only by forward differences which encompass the total flight from lift-off to orbiter termination.

14.6 PARAMETER UPDATE AND CONVERGENCE TEST

The final part of the iteration cycle is the call to the subroutine ANEWCH which controls the following items:

- Determination of optimization and restoration step size requirements
- Control parameter update requirements
- Update requirements for the control variables during the min-H phase

- Investigation of the tests for convergence
- Prints outs (if coverged), the parameter summary table, orbital element table, the load indicator table, and the weight summary table.

If the convergence tests are met, the variable NMAX is set to 1 (and decremented to zero) to signal the termination of the iteration cycle after the forward trajectory has been determined. If the convergence tests are not met, the variable NMAX is decreased and the program flow is returned to the forward trajectory integration to continue the iteration cycle.

14.7 OUTPUT TABLES

Over a converged run is obtained, or the maximum number of iterations has been reached, a non zero value of NTABL specifies that special output tables are desired. If the tables are desired, the subroutines BOPTBL and B1PTBL are called.

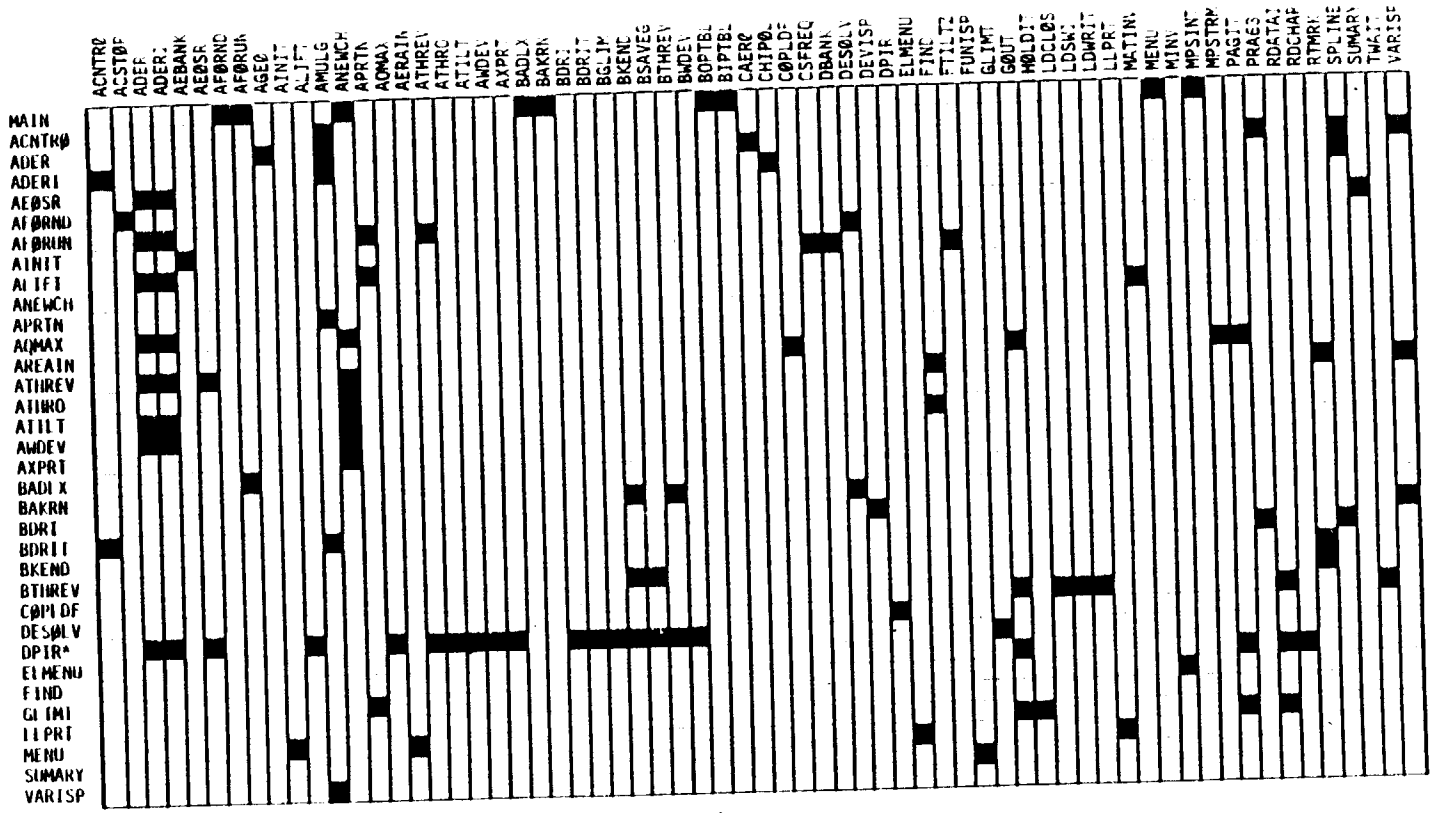
Section 15

SUBROUTINE DESCRIPTION

The following subsections gives an alphabetical listing of all subroutines and a brief statement of the function of each. Subsection 15.1 describes the nonsystem subroutines, Subsection 15.2 describes the non-system elements, and Subsection 15.3 gives the required FORTRAN system subroutines. A cross-index of the calling requirements of the non-system routines is presented in Figure 15-1. The current overlay structure is presented in Figure 15-2.

15.1 NON-SYSTEM SUBROUTINE NAMES

Subroutine Name	Function
ACNTRO	Determines the values of pitch and yaw attitude based on interpolation of the proper table. (Only used during the min-H phase.)
ACSTOP	Evaluates the payoff and determines the value of terminal and intermediate equality constraints. Also evaluates the partial derivatives of the payoff and constraints with respect to the state.
ADER	Evaluates the state derivatives.
ADERI	Evaluates the time dependent variables.
AEBANK	Reads first stage aerodynamic coefficient tables from aerodynamic data base.
AEOSR	Stores the state during the forward trajectory.
AFORND	Controls the call to ASCTOP to determine the payoff function and terminal and intermediate constraints, computes errors from desired values, sets up initial backward integration, and prints terminal summary.
AFORUN	Controls the setup logic for the forward trajectory integration and calls the integration package.
AEGO	Calculates the gravitational parameters for forward and backward integration.
AINIT	Reads input data and performs initial setup.



*Subroutines called by DPIR are brought in through calling arguments from AFORUN and BAKRN by way of the subroutine DESOLV.

Figure 15-1. Subroutine Cross Index

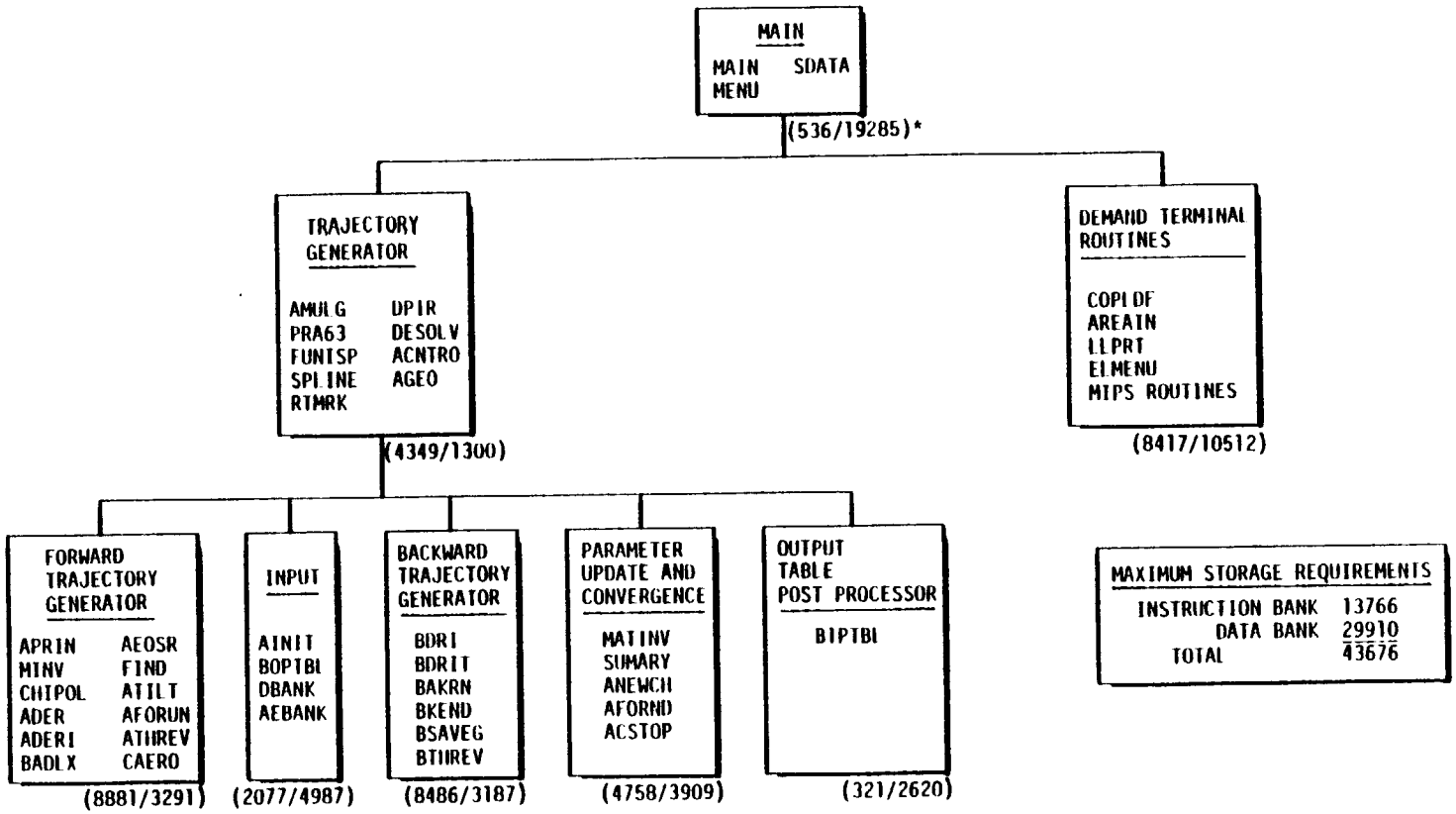


Figure 15-2. Overlay Structure

Subroutine Name	Function
ALIFT	Prints 'LIFTOFF' trajectory block when longitudinal acceleration reaches thrust-to-weight of one.
AMULG	Linear interpolation scheme.
ANEWCH	Evaluates the optimization and restoration equations to determine the required changes in the controlling parameters. Tests for convergence are made in this subroutine.
APRTN	Computes output parameters, prints data, and writes output tape for output tables.
AQMAX	Provides maximum dynamic pressure printout.
AREAIN	Provides main menu in the demand terminal mode.
ATHREV	Thrust event control routine; sets up vehicle geometry, aerodynamics, number of engines, thrust limits, and flow rates; and trigger flags for thrust event cutoff.
ATHRO	Provides discrete printout at MPS throttle level changes corresponding to the TIMMPS and FTBL input arrays.
ATILT	Controls time related events directly after lift-off.
AWDEV	Controls weight drop event.
AXPRT	Controls normal printout.
BADLX	Determines the influence coefficients (partial derivatives) of the payload terminal and intermediate constraints with respect to the parameters.
BAKRN	Controls the setup logic for the backward trajectory integration and calls the integration derivatives.
BDRI	Computes total adjoint derivatives.
BDII	Computes time dependent portion of adjoint derivatives.
BGLIM	Turns off acceleration limit switch in backward integration.
BKEND	Terminates adjoint run and adjusts influence coefficients.

Subroutine Name	Function
BSAVEG	Saves impulse response functions and performs Simpson integration on products of impulse response functions.
BTHREV	Backward thrust event control routine.
BWDEV	Backward weight drop event control routine.
BOPTBL	Reads input namelist for output table.
B1PTBL	Output table routine.
CAERO	Determines the first stage aerodynamic coefficients based on current values of Mach number, angle of attack, sideslip angle, and inboard/outboard elevon angles.
CHIPOL	Evaluates pitch and yaw attitude polynomials.
COPLDF	Writes list directed file for plotting.
DBANK	Reads other data from the aerodynamic data base.
DESOLV DPIR	Integration package.
DEVISP	Provides partial derivative of MPS specific impulse w.r.t. mass
ELMENU	Provides a mean for allowing demand terminal user to add data elements into the input flow.
FIND	Determines attitude polynomials based on attitude time history information.
FUNISP	Calculates the current value of MPS specific impulse based on throttle level and countout reference value.
GLIMT	Provides acceleration limit printout.
LLPRT	Provides speical output in demand mode.
MATINV	Performs matrix inversion.
MENU	Provides main logic flow for demand mode.
MINV	Same as MATINV (used for overlay).
PRA63	Evaluates the Patrick Air Force Base atmosphere model for 1963 and determines the partial derivatives of the parameters with respect to altitude by using a third order spline interpolation method.

Subroutine Name	Function
RTMRK	Part of integration package which flags termination of the integration.
SDATA	Block data routine (summary table data).
SPLINE	Third order spline interpolation routine.
SUMARY	Generates weight summary output tables.
VARISP	Calculates current value of MPS specific impulse based on throttle level and time varying reference (100%) specific impulse.

15.2 NON-SYSTEM ELEMENTS

File Name	Description
AERO79	Namelist for additional aerodynamic data.
ELTTBL	Namelist data for output table post processor.
LBMW/RI	Provides updates to programmed code and maps the program for the LBM trajectory simulation from Vandenburg.
LBMW-1	Namelist data for LBM pseudo targeting mission constraints.
LBMW-2	Namelist data for AOA mission.
LBMW-3	Namelist data for nominal mission.
LBMW-4	Namelist data for RTLS mission.
PAC63	Evaluates cold Patrick atmospheric model.
PHA63	Evaluates hot Patrick atmospheric model.
SYM	Program map.
VAC71	Evaluates cold Vandenburg atmospheric model
VHA71	Evaluates hot Vandenburg atmospheric model.
VRA71	Evaluates reference Vandenburg atmospheric model.

15.3 SYSTEM SUBROUTINE NAMES

Subroutine (or Function) Name	Description
ABS	Absolute value of the function.
ACOS	Determines arc cosine.

Subroutines (or Function) Name	Description
AINT	Truncates real variables to integer form.
ALOG	Determines the natural logarithm of a variable.
AMAX1	Selects a maximum value from argument list.
ASIN	Determines arc sine.
ATAN	Determines arc tangent based on one argument $(-\frac{\pi}{2} \leq 0 \leq \frac{\pi}{2})$.
ATAN2 (A,B))	Determines arc tangent based on two arguments $(-\pi \leq 0 \leq \pi)$.
COS	Determines the cosine of the argument.
DABS	Double precision ABS.
DCOS	Double precision COS.
DSIN	Double precision SIN.
DSQRT	Double prevision SQRT.
ENCODE	Converts variables to hollerith field data.
EXP	Exponential function.
EXTERNAL	FORTRAN statement stating that the list of subroutines names following the word EXTERNAL are defined outside of this subroutine. This allows subroutine names to be used as an argument in a subroutine or function call.
IABS	Absolute value of an integer variable.
NAMELIST	Variable format input routine that reads and prints data.
SIGN (A,B)	Transfers the sign of the second argument to the absolute value of the first argument.
SIN	Determines the sine of the argument.
SQRT	Determines the square root of the argument.

The following subroutines are in the MIPS library, for detailed information about these subroutines the reader should consult the MSFC MIPS documentation (Reference 12).

Subroutine (or Function) Name	Description
CSFREQ	Adds data from the terminal to the run stream.
FTITL2	Prints title on microfiche output.
GOUT	Outputs predetermined code to the terminal.
HOLDIT	Maintain current image on terminal screen.
LDCLOS	Closes list directed file.
LDSWI	Initializes and open list directed file.
LDWRIT	Writes list directed file.
MPSINT	Initiates MIPS systems.
MPSTRM	Terminates MIPS system.
PAGIT	Erases present screen.
RDATAI	Interrupts data from terminal.
RDCHAR	Interrupts character strings from terminal.
TWAIT	Program pause command.

Section 16

PROGRAM LISTING

Appendix G presents a listing of the computer program. The main routine is presented first followed by an alphabetical listing of all non-system subroutines and elements.

Appendix A

INTERPOLATION METHODS

Appendix A

INTERPOLATION METHODS

Rapid, consistent, and accurate interpolation of tabulated data is essential to the operation of MASTRE. To this end, two very efficient interpolation schemes are used in the program. The linear interpolation subroutine, AMULG, is used for the determination of the values of the pitch and yaw attitude during the min-H phase as a function of the control tables. A cubic spline interpolation routine, SPLINE, which ensures continuous first derivatives of the data across the data points, is used to find the aerodynamic coefficients as functions of Mach number, the base pressure force as a function of altitude, the atmospheric properties as functions of altitude, wind direction and wind speed as a function of altitude, and center of gravity location as a function of weight. Both interpolation routines keep track of the lower data point from the interval used in the previous call to the routine for each particular table. In this manner the routines are not required to repeatedly start from the beginning of the tables to find the current interval containing the independent variable.

The linear interpolation routine, AMULG, searches for the interval containing the current value of the independent variable. If, of course, $x = x_i$ (a data point), then $y = y_i$. If the independent data point x_{i+1} is less than x_i, it is assumed that the last valid data point has been passed and that extrapolation beyond that point will now be required. (Note that this requirement allows the number of data points to be less than the maximum number of points available in each table.) If $x < x_1$ then m is chosen as $m = 2$, (where m denotes the number of the base point of the interval) and backward extrapolation is performed. Once the base point, x_m, is located, the general linear interpolation is

$$y = y_{m-1} + \frac{x_m - x_{m-1}}{x_m - x_{m-1}} \left(y_m - y_{m-1}\right) .$$

The cubic spline interpolation routine, SPLINE, first locates the independent variable within the central span of a four point group, that is, $x_m \leq x \leq x_{m+1}$. (If $x < x_1$, $m = 1$.) If all of the available points in the independent variable table are filled, extrapolation will be performed when $x > x_m$ (where m is the number of locations in the table). If $x_{i+2} < x_{i+1}$, implying (as in AMULG) that the last valid data point has been passed, the routine sets y_{i+2} values in all the dependent variable tables equal to the y_{i+1} values. Thus, if fewer than the total number of available points are used in the tables, the scheme assumes that the dependent variables remain unchanged after the last valid input point. Having established the interval containing x ($x_m < x < x_{m+1}$), the dependent variables are found as

$$y_i = a_i x^3 + b_i x^2 + c_i x + y_m . \tag{A-1}$$

The coefficients a_1, b_1, and c_1 are derived from the following equations:

$$s \equiv \frac{y_{m+1} - y_m}{x_{m+1} - x_m}$$

$$s_m \equiv \frac{y_m - t_{m-1}}{x_m - x_{m-1}}$$

$$s_p \equiv \frac{y_{m+2} - y_{m+1}}{x_{m+2} - x_{m+1}} .$$

If $m = 1$, s_m is <u>chosen</u> so that

$$s = \frac{s_m + s_p}{2}$$

or

$$s_m = 2s - s_p .$$

If $m = n-1$, s_p is <u>chosen</u> so that

$$s = \frac{s_m + s_p}{2}$$

or

$$s_p = 2s - s_m \ .$$

Equation (A-1) can be rewritten as

$$\Delta y_1 = y_{m+1} - y_m = a\Delta x_1^3 + b\Delta x_1^2 + c\Delta x_1$$

where

$$\Delta x_1 = x_{m+1} - x_m \ .$$

Dividing by x_1, yields

$$s = a\ x_1^2 + b\ x_1 + c.$$

The spline assumption is

$$\left.\frac{dv}{dx}\right|_{x_m} = c = \frac{s + s_m}{2}$$

and

$$\left.\frac{dv}{dx}\right|_{x_{m+1}} = 3a\Delta x_1^2 + 2b\Delta x_1 + c = \frac{s + s_p}{2} \ .$$

Note that this equation leaves the first derivatives continuous as m changes. Thus, a, b, and c are

$$a = \frac{1}{\Delta x_1^2}\left(\frac{s_p - s_m}{2} - s + s_m\right)$$

$$b = \frac{1}{\Delta x_1} \left(\frac{s - s_p}{2} + s - s_m \right)$$

$$c = \frac{s + s_m}{2} .$$

With $R \equiv \frac{\Delta x}{\Delta x_1} = \frac{x - x_m}{x_{m+1} - x_m}$

we have $y = y_m + ((aR + b)\ R + c)\ \Delta x$

and

$$\frac{dy}{dx} = (3aR + 2b)R + c.$$

Since a, b, and c are functions of the data points only (and are therefore constant for a given interval of the independent variable data set), their values are stored and reused without recalculation if interpolation within the same interval is required on subsequent calls.

Appendix B

SOLUTION OF MOMENT BALANCE EQUATIONS

Appendix B

SOLUTION OF MOMENT BALANCE EQUATIONS

From Subsection 2.4.3 we have the moment equations in the center-of-gravity and gimbal point coordinate system

$$\left(T_{y_1} \tan \delta_{y_1} + T_{y_2} \tan \delta_{y_2}\right) dy + \left(T_{y_1} - T_{y_2}\right) dz = -M_{A_x} \tag{B-1}$$

$$\left(T_{y_1} \tan \delta_{p_1} - T_{y_2} \tan \delta_{p_2}\right) dz = -M_{A_y} \tag{B-2}$$

$$\left(T_{y_1} \tan \delta_{p_1} + T_{y_2} \tan \delta_{p_2}\right) dy = -M_{A_z} \tag{B-3}$$

$$\delta_{y_1} = \delta_{y_2} = \delta_y \ . \tag{B-4}$$

Since the total thrust of each engine is fixed ($T_1 = T_2 = T_T/2$) we can write

$$T_{x_1}^2 + T_{y_1}^2 + T_{z_1}^2 = \left(\frac{T_T}{2}\right)^2 \tag{B-5}$$

and

$$T_{x_2} + T_{y_2} + T_{z_2} = \left(\frac{T_T}{2}\right)^2 \ . \tag{B-6}$$

But, from the definitions of δ_p and δ_y

$$T_{x_1} = -T_{y_1} \tan \delta_{p_1}$$

$$T_{x_2} = -T_{y_2} \tan \delta_{p_2}$$

$$T_{z_1} = T_{y_1} \tan \delta_{y_1} = T_{y_1} \tan \delta_y$$

$$T_{z_2} = T_{y_2} \tan \delta_{y_2} = t_{y_2} \tan \delta_y$$

so that

$$T_{y_1} = \frac{T_{T/2}}{\sqrt{1 + \tan^2 \delta_{p_1} + \tan^2 \delta_y}} \tag{B-7}$$

and

$$T_{y_2} = \frac{T_{T/2}}{\sqrt{1 + \tan^2 \delta_{p_2} + \tan^2 \delta_y}} \quad . \tag{B-8}$$

We now assume that δ_{p_1}, δ_{p_2}, and δ_y are small enough so that

$$\tan^2 \delta_{p_1} << 1 \tag{B-9}$$

$$\tan^2 \delta_{p_2} << 1 \tag{B-10}$$

and

$$\tan^2 \delta_y << 1 \ . \tag{B-11}$$

This approximation yields, from Equations (B-7) and (B-8)

$$T_{y_1} \approx \frac{T_T}{2} \tag{B-12}$$

and

$$T_{y_2} \approx \frac{T_T}{2} \quad . \tag{B-13}$$

Using (B-4), (B-12), and (B-13) in (B-1), (B-2), and (B-3) yields

$$\left(T_T \tan \delta_y\right) dy = -M_{A_x} \tag{B-14}$$

$$\frac{T_T}{2} \left(\tan \delta_{p_1} - \tan \delta_{p_2}\right) dz = -M_{A_y} \tag{B-15}$$

$$\frac{T_T}{2}\left(\tan\delta_{p_1} + \tan\delta_{p_2}\right)dy = -M_{A_z} \tag{B-16}$$

which can be rewritten as

$$\tan\delta_y = \frac{M_{A_x}}{T_T dy} \tag{B-17}$$

$$\tan\delta_{p_1} - \tan\delta_{p_2} = -\frac{2M_{A_y}}{T_T dz} \tag{B-18}$$

and

$$\tan\delta_{p_1} + \tan\delta_{p_2} = -\frac{2M_{A_z}}{T_T dy}\,. \tag{B-19}$$

Adding (B-18) and (B-19) yields

$$\tan\delta_{p_1} = -\frac{1}{T_T}\left(\frac{M_{A_y}}{dz} + \frac{M_{A_z}}{dy}\right). \tag{B-20}$$

Subtracting (B-18) from (B-19) yields

$$\tan\delta_{p_2} = -\frac{1}{T_T}\left(\frac{M_{A_z}}{dy} - \frac{M_{A_y}}{dz}\right). \tag{B-21}$$

Since the total thrust is inherently enforced in the solution, any aiming error amounts to an error in the moments produced by the engine to balance the aerodynamic moments. (It is possible to obtain an approximate solution that produces no net unbalanced moment but changes the total thrust slightly.) The philosophy in MASTRE assumes a moment balance scheme to determine the performance of a vehicle based on realistic thrust vectoring; this goal is satisfied acceptably when the moment on the vehicle has been reduced to a very small level without changing the overall thrust of the vehicle.

As expected, for no aerodynamic moment $\delta_{p_1} = \delta_{p_2} = \delta_y = 0$. The thrust vector of each engine is parallel to the center of gravity and gimbal point y axis when aerodynamic moments are zero.

Note that the moment and distance components in the above solution are in the xyz (center-of-gravity and gimbal point) coordinate system. The moments must be transformed into this system from the body axis XYZ system in which they are computed, and the resulting thrust components must be transformed from the xyz system back to the XYZ system in order to determine the total forces acting on the vehicle. The xyz system is obtained from the XYZ system by rotating about Z the angle $(-\rho)$, where

$$\rho = \tan^{-1}\left(\frac{X_{GP} - X_{CG}}{Y_{GP} - Y_{CG}}\right) . \tag{B-22}$$

Thus, the transformation matrix from XYZ to xyz is

$$\begin{Bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{Bmatrix} = \begin{bmatrix} \cos(-\rho) & \sin(-\rho) & 0 \\ -\sin(-\rho) & \cos(-\rho) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{i}' \\ \hat{j}' \\ \hat{k}' \end{Bmatrix} \quad \text{or} \quad \begin{bmatrix} \cos\rho & -\sin\rho & 0 \\ \sin\rho & \cos\rho & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{i}' \\ \hat{j}' \\ \hat{k}' \end{Bmatrix}$$

and the reverse transformation matrix is then the transpose of the above, or

$$\begin{Bmatrix} \hat{i}' \\ \hat{j} \\ \hat{k} \end{Bmatrix} = \begin{bmatrix} \cos\rho & \sin\rho & 0 \\ -\sin\rho & \cos\rho & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} \hat{i} \\ \hat{j} \\ \hat{k} \end{Bmatrix} .$$

The aerodynamic moments in the xyz system are then

$$\begin{aligned} M_{A_x} &= M_{A_X} \cos\rho - M_{A_Y} \sin\rho \\ M_{A_y} &= M_{A_X} \sin\rho + M_{A_Y} \cos\rho \\ M_{A_z} &= M_{A_Z} . \end{aligned} \tag{B-23}$$

Now, the distances are defined

$$\Delta X = X_{GP_1} - X_{CG} = X_{GP_2} - X_{CG}$$

$$\Delta Y = Y_{GP_1} - Y_{CG} = Y_{GP_2} - Y_{CG}$$

$$\Delta Z = Z_{GP_2} - Z_{CG} = -\left(Z_{GP_1} - Z_{CG}\right)$$

so that

$$\rho = \tan^{-1}\left(\frac{\Delta X}{\Delta Y}\right)$$

$$\cos \rho = \frac{\Delta Y}{\sqrt{\Delta X^2 + \Delta Y^2}} \tag{B-24}$$

$$\sin \rho = \frac{\Delta X}{\sqrt{\Delta X^2 + \Delta Y^2}}$$

and

$$\Delta x = \Delta X \cos \rho - \Delta Y \sin \rho = 0$$

$$\Delta y = \Delta X \sin \rho + \Delta Y \cos \rho = \sqrt{X^2 + Y^2} \tag{B-25}$$

$$\Delta z = \Delta Z \ .$$

Using (B-21), (B-22), and (B-23) in (B-14), (B-17), and (B-18) we obtain

$$\delta_y = -\tan^{-1}\left[\frac{M_{A_Y}\dfrac{\Delta X}{\Delta Y} - M_{A_X}}{T_T\,\Delta Y\sqrt{1 + \dfrac{\Delta X^2}{\Delta Y^2}}}\right]$$

$$\delta_{p_1} = -\tan^{-1}\left[\frac{\dfrac{M_{A_Z}}{\Delta Y} + \dfrac{M_{A_Y}}{\Delta Z} - \dfrac{M_{A_X}\,\Delta X}{\Delta Y \Delta Z}}{T_T\sqrt{1 + \dfrac{\Delta X^2}{\Delta Y^2}}}\right]$$

$$\delta_{p_2} = -\tan^{-1}\left[\frac{\frac{M_{A_Z}}{\Delta Y} - \frac{M_{A_Y}}{\Delta Z} + \frac{M_{A_Y}\ \Delta X}{\Delta Y \Delta Z}}{T_T\sqrt{1+\frac{\Delta X^2}{\Delta Y^2}}}\right]$$

Note that the gimbal angles in the XYZ body axis coordinate system are given by

$$\delta_{p_1}' = \delta_{p_1} - \tan^{-1}\left(\frac{\Delta X}{\Delta Y}\right) = \delta_{p_1} - \rho$$

$$\delta_{p_2}' = \delta_{p_2} - \tan^{-1}\left(\frac{\Delta X}{\Delta Y}\right) = \delta_{p_2} - \rho$$

and

$$\delta_y' = \delta_y \ .$$

The total thrust components in the XYZ body axis coordinate system are then found by noting that

$$T_{X_1} = T_{Y_1} \tan \delta_{p_1}'$$

$$T_{X_2} = T_{Y_2} \tan \delta_{p_2}'$$

$$T_{Z_1} = T_{Y_1} \tan \delta_y$$

$$T_{Z_2} = T_{Y_2} \tan \delta_y$$

$$T_{Y_1} = \frac{T_T}{2\sqrt{1 + \tan^2 \delta_{p_1}' + \tan^2 \delta_y}}$$

and

$$T_{Y_2} = \frac{T_T}{2\sqrt{1 + \tan^2 \delta_{p_2}' + \tan^2 \delta_y}}$$

so that

$$T_X = -\frac{T_T}{2}\left[\frac{\tan \delta_{p_1}'}{\sqrt{1 + \tan^2 \delta_{p_1}' + \tan^2 \delta_y}} + \frac{\tan \delta_{p_2}'}{\sqrt{1 + \tan^2 \delta_{p_2}' + \tan^2 \delta_y}}\right]$$

$$T_Y = \frac{T_T}{2}\left[\frac{1}{\sqrt{1 + \tan^2 \delta_{p_1}' + \tan^2 \delta_y}} + \frac{1}{\sqrt{1 + \tan^2 \delta_{p_2}' + \tan^2 \delta_y}}\right]$$

$$T_Z = T_Y \tan \delta_y$$

where T_X, T_Y, and T_Z are the total thrust components in the body axis XYZ coordinate system.

Appendix C

VELOCITY LOSS EQUATIONS

Appendix C

VELOCITY LOSS EQUATIONS

The velocity loss equations result from the calculation of the change in the inertial velocity, V_I, during the time interval $t_a - t_b$.

It is noted that

$$V_I(t_b) - V_I(t_a) = \int_{t_a}^{t_b} \dot{V}_I dt \ . \tag{C-1}$$

Since,

$$V_I^2 = \overline{V}_I \cdot \overline{V}_I \tag{C-2}$$

it follows that

$$\dot{V}_I = \frac{\overline{V}_I \cdot \dot{\overline{V}}_I}{V_I} \ . \tag{C-3}$$

Therefore, Equation (C-1) can be rewritten as

$$\Delta V_I \equiv V_I(t_b) - V_I(t_a) = \int_{t_a}^{t_b} \frac{\overline{V}_I \cdot \dot{\overline{V}}_I}{V_I} dt \ . \tag{C-4}$$

Three different thrust levels will now be defined. The first, T_v, is the total resultant thrust calculated as the square root of the sum of the squares of the thrust components. The second, T_s, is the scalar sum of the thrusts of all engines. The third, T_{vac_s}, is the scalar sum of the vacuum thrust of all engines. Adding and subtracting each of these thrust levels divided by mass to the right-hand side of (C-4) leads to

$$\Delta V_I = \int_{t_a}^{t_b} \left[\frac{T_v}{M} - \frac{T_v}{M} + \frac{T_s}{M} - \frac{T_s}{M} + \frac{T_{vac_s}}{M} - \frac{T_{vac_s}}{M} + \frac{\overline{V}_I \cdot \dot{\overline{V}}_I}{V_I} \right] dt \ . \tag{C-5}$$

Since T_{vac_s} is the largest of the three, we choose it to define ideal velocity so that everything else will represent losses. This gives

$$\Delta V_I = \int_{t_a}^{t_b} \left[\frac{T_{vac_s}}{M} + \frac{T_s - T_{vac_s}}{M} + \frac{T_v - T_s}{M} + \frac{\overline{V}_I \cdot \dot{\overline{V}}_I}{V_I} - \frac{T_v}{M} \right] dt. \quad \text{(C-6)}$$

In MASTRE

$$\dot{\overline{V}}_I = \frac{1}{M} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} + \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} \quad \text{(C-7)}$$

where, assuming $\chi_R = 0$,

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \begin{bmatrix} \cos \chi_P & \cos \chi_Y \sin \chi_P & -\sin \chi_Y \sin \chi_P \\ \sin \chi_P & \cos \chi_Y \cos \chi_P & -\sin \chi_Y \cos \chi_P \\ 0 & \sin \chi_Y & \cos \chi_Y \end{bmatrix} \begin{bmatrix} T_x + A_x \\ T_y + A_y \\ T_z + A_z \end{bmatrix} \quad \text{(C-8)}$$

Taking the dot product of (C-7) and $\overline{V}_I/V_I$ leads to

$$\frac{\overline{V}_I \cdot \dot{\overline{V}}_I}{V_I} = \frac{\overline{V}_I}{V_I} \cdot \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} + \frac{1}{M} \left(T_x \cos \alpha_1 + T_y \cos \alpha_2 + T_z \cos \alpha_3 \right)$$

$$+ \frac{1}{M} \left(A_x \cos \alpha_1 + A_Y \cos \alpha_2 + A_Z \cos \alpha_3 \right) \quad \text{(C-9)}$$

where

$$\cos \alpha_1 \equiv \frac{\overline{V}_I}{V_I} \cdot \begin{bmatrix} \cos \chi_P \\ -\sin \chi_P \\ 0 \end{bmatrix}$$

$$\cos \alpha_2 \equiv \frac{\overline{V}_I}{V_I} \cdot \begin{bmatrix} \cos \chi_Y \sin \chi_P \\ \cos \chi_Y \cos \chi_P \\ \sin \chi_Y \end{bmatrix} \qquad \text{(C-10)}$$

$$\cos \alpha_3 \equiv \frac{\overline{V}_I}{V_I} \cdot \begin{bmatrix} -\sin \chi_Y \sin \chi_P \\ -\sin \chi_Y \cos \chi_P \\ \cos \chi_Y \end{bmatrix}$$

If there is a roll angle, χ_R, to be considered also, the variables in (C-10) are used to redefine $\cos \alpha_1$, $\cos \alpha_2$, and $\cos \alpha_3$ as

$$\begin{bmatrix} \cos \alpha_1 & \cos \alpha_2 & \cos \alpha_3 \end{bmatrix} \equiv \begin{bmatrix} \cos \alpha_1 & \cos \alpha_2 & \cos \alpha_3 \end{bmatrix} \begin{bmatrix} \cos \chi_R & 0 & \sin \chi_R \\ 0 & 1 & 0 \\ -\sin \chi_R & 0 & \cos \chi_R \end{bmatrix} \qquad \text{(C-11)}$$

Rewriting (C-6) and using either (C-10) or (C-11) as desired, yields

$$\Delta V_I = \int_{t_a}^{t_b} \left[\frac{T_{vac_s}}{M} + \frac{T_s - T_{vac_s}}{M} + \frac{T_v - T_s}{M} + \frac{\overline{V}_I}{V_I} \cdot \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} \right.$$

$$+ \frac{1}{M} \left\{ T_x \cos \alpha_1 + T_y \cos \alpha_2 + T_z \cos \alpha_3 - T_v \right\}$$

$$\left. + \frac{1}{M} \left\{ A_X \cos \alpha_1 + A_Y \cos \alpha_2 + A_Z \cos \alpha_3 \right\} \right] dt. \qquad \text{(D-12)}$$

With the exception of T_{vac_s}/M, each of the other terms in (C-12) is negative and represents a loss. Now, defining the following:

$$\text{ideal vel} \equiv \int_{t_a}^{t_b} \left(\frac{T_{vac_s}}{M} \right) dt \qquad \text{(C-13)}$$

$$\text{back pressure loss} \equiv \int_{t_a}^{t_b} - \left(\frac{T_s - T_{vac_s}}{M}\right) dt \tag{C-14}$$

$$\text{gimbal loss} \equiv \int_{t_a}^{t_b} - \left(\frac{T_v - T_s}{M}\right) dt \tag{C-15}$$

$$\text{gravity loss} \equiv \int_{t_a}^{t_b} - \frac{\overline{V}_I}{V_I} \cdot \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} dt \tag{C-16}$$

$$\text{turning loss} \equiv \int_{t_a}^{t_b} - \frac{1}{M}\left[T_x \cos \alpha_1 + T_y \cos \alpha_2 + T_z \cos \alpha_3 - T_v\right] dt \tag{C-17}$$

$$\text{aero loss} \equiv \int_{t_a}^{t_b} - \frac{1}{M}\left[A_x \cos \alpha_1 + A_y \cos \alpha_2 + A_z \cos \alpha_3\right] dt \tag{C-18}$$

characteristic velocity = ideal velocity - back pressure loss - gimbal loss.

These losses are such that

V_I = ideal velocity - pressure loss - gimbal loss - gravity loss - turning loss - aero loss.

The definition of ideal velocity used here has the added advantage of being an analytical function of the atmosphere. However, if other definitions of characteristic velocity are desired, they can be calculated easily from (C-13) through (C-15), that is

$$\int_{t_a}^{t_b} \frac{T_s}{M}\, dt = \text{ideal velocity - pressure loss}$$

$$\int_{t_a}^{t_b} \frac{T_v}{M}\, dt = \text{ideal velocity - pressure loss - gimbal loss.}$$

Appendix D

PARTIAL DERIVATIVES OF THE PAYOFF AND CONSTRAINTS WITH RESPECT TO THE FINAL STATE

Appendix D

PARTIAL DERIVATIVES OF THE PAYOFF AND CONSTRAINTS WITH RESPECT TO THE FINAL STATE*

The partial derivatives of the various payoff and terminal constraint quantities with respect to the final state (position XYZ, velocity WUV, mass M) are required to determine the boundary conditions of Section IV. These derivative calculations are presented in this appendix.

Mass M_f (final cutoff weight)

$$\frac{\partial M_f}{\partial X} = \frac{\partial M_f}{\partial Y} = \frac{\partial M_f}{\partial Z} = \frac{\partial M_f}{\partial W} = \frac{\partial M_f}{\partial U} = \frac{\partial M_f}{\partial V} = 0$$

$$\frac{\partial M_f}{\partial M} = 1.$$

Inertial Velocity V_I

$$V_I = \left(W^2 + U^2 + V^2\right)^{1/2}$$

$$\frac{\partial V_I}{\partial X} = \frac{\partial V_I}{\partial Y} = \frac{\partial V_I}{\partial Z} = \frac{\partial V_I}{\partial M} = 0$$

$$\frac{\partial V_I}{\partial W} = \frac{W}{V_I}$$

$$\frac{\partial V_I}{\partial U} = \frac{U}{V_I}$$

$$\frac{\partial V_I}{\partial V} = \frac{V}{V_I} \quad .$$

*Extracted in large part from Reference 3 (see Section VII).

Inertial Flight Path Angle

$$\gamma = \sin^{-1}\left((XW + YU + ZV)/RV_I\right)$$

$$\frac{\partial\gamma}{\partial X} = \left(N_{21} - \frac{W \sin \gamma}{V_I}\right) \Big/ \left(V_I \cos \gamma\right)$$

$$\frac{\partial\gamma}{\partial Y} = \left(N_{22} - \frac{U \sin \gamma}{V_I}\right) \Big/ \left(V_I \cos \gamma\right)$$

$$\frac{\partial\gamma}{\partial Z} = \left(N_{23} - \frac{V \sin \gamma}{V_I}\right) \Big/ \left(V_I \cos \gamma\right)$$

$$\frac{\partial\gamma}{\partial W} = \frac{N_{24}}{V_I \cos \gamma}$$

$$\frac{\partial\gamma}{\partial U} = \frac{N_{25}}{V_I \cos \gamma}$$

$$\frac{\partial\gamma}{\partial V} = \frac{N_{26}}{V_I \cos \gamma}$$

$$\frac{\partial\gamma}{\partial M} = 0 \ .$$

The equation for N is defined beginning on Page D-13.

Radius R

$$R = \left(X^2 + Y^2 + Z\right)^{1/2}$$

$$\frac{\partial R}{\partial X} = \frac{X}{R}$$

$$\frac{\partial R}{\partial Y} = \frac{Y}{R}$$

$$\frac{\partial R}{\partial Z} = \frac{Z}{R}$$

$$\frac{\partial R}{\partial W} = \frac{\partial R}{\partial U} = \frac{\partial R}{\partial V} = \frac{\partial R}{\partial M} = 0$$

Earth Fixed Velocity V_E

$$V_E = \left(\underline{W}^2 + \underline{U}^2 + \underline{V}^2\right)^{1/2}$$

$$\frac{\partial V_E}{\partial X} = \omega\left(-\underline{U}\ A_{32} + \underline{V}\ A_{22}\right)/V_R$$

$$\frac{\partial V_E}{\partial Y} = \omega\left(\underline{W}\ A_{32} - \underline{V}\ A_{12}\right)/V_R$$

$$\frac{\partial V_E}{\partial Z} = \omega\left(-\underline{W}\ A_{22} + \underline{U}\ A_{12}\right)/V_R$$

$$\frac{\partial V_E}{\partial W} = \underline{W}/V_R$$

$$\frac{\partial V_E}{\partial U} = \underline{U}/V_R$$

$$\frac{\partial V_E}{\partial V} = \underline{V}/V_R$$

where

$$\begin{bmatrix} \underline{W} \\ \underline{U} \\ \underline{V} \end{bmatrix} = \overline{V}_E \ .$$

Earth Fixed Azimuth (Heading Angle) A_{Z_E}

$$A_{Z_E} = \tan^{-1}\left(\frac{W_E}{-V_E}\right)$$

$$\begin{bmatrix} \frac{\partial A_{Z_E}}{\partial X} \\ \frac{\partial A_{Z_E}}{\partial Y} \\ \frac{\partial A_{Z_E}}{\partial Z} \\ \frac{\partial A_{Z_E}}{\partial W} \\ \frac{\partial A_{Z_E}}{\partial U} \\ \frac{\partial A_{Z_E}}{\partial V} \end{bmatrix} = - \frac{V_E}{V_E^2 + W_E^2} \begin{bmatrix} P_{11} \\ P_{12} \\ P_{13} \\ P_{14} \\ P_{15} \\ P_{16} \end{bmatrix} + \frac{W_E}{V_E^2 + W_E^2} \begin{bmatrix} P_{31} \\ P_{32} \\ P_{33} \\ P_{34} \\ P_{35} \\ P_{36} \end{bmatrix}$$

$$\frac{\partial A_{Z_E}}{\partial M} = 0.$$

The P vector and W_E, V_E are defined beginning on page D-13.

<u>Inertial Heading Angle A_Z</u>

$$A_Z = \tan^{-1} \left(\frac{W_S}{-V_S} \right)$$

$$\begin{bmatrix} \frac{\partial A_Z}{\partial X} \\ \frac{\partial A_Z}{\partial Y} \\ \frac{\partial A_Z}{\partial Z} \\ \frac{\partial A_Z}{\partial W} \\ \frac{\partial A_Z}{\partial U} \\ \frac{\partial A_Z}{\partial V} \end{bmatrix} = - \frac{V_S}{V_S^2 + V_S^2} \begin{bmatrix} N_{11} \\ N_{12} \\ N_{13} \\ N_{14} \\ N_{15} \\ N_{16} \end{bmatrix} + \frac{W_S}{V_S^2 + W_S^2} \begin{bmatrix} N_{31} \\ N_{32} \\ N_{33} \\ N_{34} \\ N_{35} \\ N_{36} \end{bmatrix}$$

$$\frac{\partial A_Z}{\partial M} = 0.$$

The N vector and W_S, V_S are defined beginning on page D-13.

<u>Earth Fixed Flight Path Angle γ_R</u>

$$\gamma_R = \sin^{-1} \left(X\underline{W} + Y\underline{U} + Z\underline{V}/RV_R \right)$$

$$\frac{\partial \gamma_R}{\partial X} = \frac{1}{\cos \gamma_R} \left(\frac{W}{RV_R} - \sin \gamma_R \left(\frac{D_{12}}{R} + \frac{\left(A_{22}\underline{V} - A_{32}\underline{U}\right)}{V_R^2} \right) \right)$$

$$\frac{\partial \gamma_R}{\partial Y} = \frac{1}{\cos \gamma_R} \left(\frac{U}{RV_R} - \sin \gamma_R \left(\frac{D_{22}}{R} + \frac{\left(A_{32}\underline{W} - A_{12}\underline{U}\right)}{V_R^2} \right) \right)$$

$$\frac{\partial \gamma_R}{\partial Z} = \frac{1}{\cos \gamma_R} \left(\frac{V}{RV_R} - \sin \gamma_R \left(\frac{D_{32}}{R} + \frac{\left(A_{12}\underline{U} - A_{22}\underline{W}\right)}{V_R^2} \right) \right)$$

$$\frac{\partial \gamma_R}{\partial W} = D_{12} \left(\frac{W \sin \gamma_R}{V_R}\right) \Big/ V_R \cos \gamma_R$$

$$\frac{\partial \gamma_R}{\partial U} = D_{22} - \left(\frac{U \sin \gamma_R}{V_R}\right) \Big/ V_R \cos \gamma_R$$

$$\frac{\partial \gamma_R}{\partial V} = D_{32} - \left(\frac{V \sin \gamma_R}{V_R}\right) \Big/ V_R \cos \gamma R$$

$$\frac{\partial \gamma_R}{\partial M} = 0.$$

<u>Inclination i</u>

$$i = \cos^{-1} (\sin \theta \cos A_Z)$$

$$\frac{\partial i}{\partial X} = \frac{\sin A_Z \cos \theta}{\sin i} N_{64} - \frac{\cos A_Z \sin \theta}{\sin i} \frac{\partial A_Z}{\partial X}$$

$$\frac{\partial i}{\partial Y} = \frac{\sin A_Z \cos \theta}{\sin i} N_{65} - \frac{\cos A_Z \sin \theta}{\sin i} \frac{\partial A_Z}{\partial Y}$$

$$\frac{\partial i}{\partial Z} = \frac{\sin A_Z \cos \theta}{\sin i} N_{66} - \frac{\cos A_Z \sin \theta}{\sin i} \frac{\partial A_Z}{\partial Z}$$

$$\frac{\partial i}{\partial W} = - \frac{\sin A_Z \cos \theta}{\sin i} N_{61} - \frac{\cos A_Z \sin \theta}{\sin i} \frac{\partial A_Z}{\partial W}$$

$$\frac{\partial i}{\partial U} = - \frac{\sin A_Z \cos \theta}{\sin i} N_{62} - \frac{\cos A_Z \sin \theta}{\sin i} \frac{\partial A_Z}{\partial U}$$

$$\frac{\partial i}{\partial V} = -\frac{\sin A_Z \cos \theta}{\sin i} N_{63} - \frac{\sin A_Z \cos \theta}{\sin i} \frac{\partial A_Z}{\partial V}$$

$$\frac{\partial i}{\partial M} = 0 \ .$$

<u>Line of Nodes Ω</u>

$$\Omega = \tan^{-1}\left(\frac{A_{11} D_{12} + A_{21} D_{22} + A_{31} D_{32}}{A_{13} D_{12} + A_{23} D_{22} + A_{33} D_{32}}\right) + \tan^{-1}\left(\frac{\cos \theta \, W_S}{V_S}\right)$$

Defining $A = \dfrac{W_S V_S \sin \theta}{V_S^2 + W_S^2 \cos^2 \theta}$

and $B = \dfrac{(V_S^2 + W_S^2) \cos \theta}{V_S^2 + W_S^2 \cos^2 \theta}$

we have

$$\frac{\partial \Omega}{\partial X} = \left(N_{44} - A N_{64}\right) + B \frac{\partial A_Z}{\partial X}$$

$$\frac{\partial \Omega}{\partial Y} = \left(N_{45} - A N_{65}\right) + B \frac{\partial A_Z}{\partial Y}$$

$$\frac{\partial \Omega}{\partial Z} = \left(N_{46} - A N_{66}\right) + B \frac{\partial A_Z}{\partial Z}$$

$$\frac{\partial \Omega}{\partial W} = B \frac{\partial A_Z}{\partial W}$$

$$\frac{\partial \Omega}{\partial U} = B \frac{\partial A_Z}{\partial U}$$

$$\frac{\partial \Omega}{\partial V} = B \frac{\partial A_Z}{\partial V}$$

$$\frac{\partial \Omega}{\partial M} = 0 \ .$$

Orbital Velocity Correction ΔV (Reference 7)

$$\Delta V = V_C \left\{ 1 + \frac{J}{3}\left(\frac{R_e}{R_I}\right)^2 \left[1 + \rho + \rho^2 - 2\rho^2 \sin^2 i - \cos^2 \theta \left(3 + 3\rho - \rho^2\right) \right] \right\}^{1/2} - V_I$$

where

$$V_C = \left(\frac{2\mu_e R_M}{R_I (R_I + R_M)} \right)^{1/2}$$

$$\rho = \frac{R_I}{R_M}$$

R_I = radius of initial point (apogee or perigee)

R_M = radius of point one half of an orbit later.

Let

$$A = \frac{V_C}{\left\{ 1 + K_1 \left[K_3 + K_4 \sin^2 i - K_2 \cos^2 \theta \right] \right\}^{1/2}}$$

and

$$B = K_1 K_4 \sin i \cos i.$$

The partial derivatives are defined as the following

$$\frac{\partial \Delta V}{\partial W} = AB \frac{\partial i}{\partial W} - \frac{W}{V_I}$$

$$\frac{\partial \Delta V}{\partial U} = AB \frac{\partial i}{\partial U} - \frac{U}{V_I}$$

$$\frac{\partial \Delta V}{\partial V} = AB \frac{\partial i}{\partial V} - \frac{V}{V_I}$$

$$\frac{\partial \Delta V}{\partial X} = A\left[B \frac{\partial i}{\partial X} + K_1 K_2 \cos \theta \sin \theta N_{64}\right]$$

$$\frac{\partial \Delta V}{\partial Y} = A\left[B \frac{\partial i}{\partial Y} + K_1 K_2 \cos \theta \sin \theta N_{65}\right]$$

$$\frac{\partial \Delta V}{\partial Z} = A\left[B \frac{\partial i}{\partial Z} + K_1 K_2 \cos \theta \sin \theta N_{66}\right]$$

where

$$K_1 = \frac{J}{3}\left(\frac{R_e}{R_I}\right)^{1/2}$$

$$K_2 = 3 + 3\rho - \rho^2$$

$$K_3 = 1 + \rho + \rho^2$$

$$K_4 = -2\rho^2 .$$

<u>Maximum Dynamic Pressure Q_{max}</u>

$$\frac{\partial\left(Q - Q_{MAX}\right)}{\partial \bar{p}} = 0 \quad \text{(not dependent on final state)} .$$

<u>Range from Launch Site R_{NG}</u>

$$R_{NG} = R_e \Phi_{NG}$$

where

$$\Phi_{NG} = \cos^{-1}(X_L D_{12} + Y_L D_{22} + Z_L D_{32})$$

$$X_L = \cos \theta_L A_{12} + \sin \theta_L\left(A_{11} \sin (\omega_e t) + A_{13} \cos (\omega_e t)\right)$$

$$Y_L = \cos \theta_L A_{22} + \sin \theta_L A_{23} \cos (\omega_e t)$$

$$Z_L = \cos \theta_L A_{32} + \sin \theta_L \left(A_{31} \sin (\omega_e t) + A_{33} \cos (\omega_e t)\right)$$

θ_L = geocentric colatitude of the launch site

ω_e = earth rotation rate

$$\frac{\partial R_{NG}}{\partial X} = - \frac{R_e}{R \sin \Phi_{NG}} \left[X_L - D_{12} \cos \Phi_{NG}\right]$$

$$\frac{\partial R_{NG}}{\partial Y} = - \frac{R_e}{R \sin \Phi_{NG}} \left[Y_L - D_{22} \cos \Phi_{NG}\right]$$

$$\frac{\partial R_{NG}}{\partial Z} = - \frac{R_e}{R \sin \Phi_{NG}} \left[Z_L - D_{32} \cos \Phi_{NG}\right]$$

$$\frac{\partial R_{NG}}{\partial W} = \frac{\partial R_{NG}}{\partial U} = \frac{\partial R_{NG}}{\partial V} = 0$$

$$\frac{\partial R_{NG}}{\partial M} = 0 \ .$$

In the case of range, the time derivative is required

$$\frac{\partial R_{NG}}{\partial t} = - \frac{R_e \sin \theta_L \, \omega_e}{R \sin \Phi_{NG}} \left[D_{12}\left(A_{11} \cos (\omega_e t) - A_{13} \sin (\omega_e t)\right) - D_{22}\left(A_{23} \sin (\omega_e t)\right) + D_{32}\left(A_{31} \cos (\omega_e t) - A_{33} \sin (\omega_e t)\right)\right].$$

The U_S, V_S, and W_S used in the above derivatives are the velocity components in the spherical inertial coordinate system.

RTLS Terminal Targeting Constraint (ΔV_{RTLS})

This constraint is based upon a linear relationship between the relative velocity and the range from the launch site. Additional input required for this constraint is the slope of the relative velocity with respect to the range (e) and the intercept point on the relative velocity scale (b). Thus, the constraint is formulated as

$$V_{RTLS} = b + e\ R_{NG} - V_R\ .$$

The partial derivatives for this constraint are:

$$\frac{\partial \Delta V_{RTLS}}{\partial X} = e\ \frac{\partial R_{NG}}{\partial X} - \frac{\partial V_R}{\partial X}$$

$$\frac{\partial \Delta V_{RTLS}}{\partial Y} = e\ \frac{\partial R_{NG}}{\partial Y} \quad \frac{\partial V_R}{\partial Y}$$

$$\frac{\partial \Delta V_{RTLS}}{\partial Z} = e\ \frac{\partial R_{NG}}{\partial Z} \quad \frac{\partial V_R}{\partial Z}$$

$$\frac{\partial \Delta V_{RTLS}}{\partial W} = -\ \frac{\partial V_R}{\partial W}$$

$$\frac{\partial \Delta V_{RTLS}}{\partial U} = -\ \frac{\partial V_R}{\partial U}$$

$$\frac{\partial \Delta V_{RTLS}}{\partial V} = -\ \frac{\partial V_R}{\partial V}$$

$$\frac{\partial \Delta V_{RTLS}}{\partial M} = 0.$$

Since the range is a function of time, the time derivative is also required

$$\frac{\partial \Delta V_{RTLS}}{\partial t} = e\ \frac{\partial R_{NG}}{\partial t}\ .$$

Definition of N Equation

To define N_{ii} used above, N is a 6 x 6 matrix which can be partitioned into four 3 x 3 submatrices

$$N = \left[\begin{array}{c|c} N_I & N_{II} \\ \hline N_{III} & N_{IV} \end{array}\right] .$$

Defining

$$\theta_1 = \frac{\pi}{2} - \theta_0$$

where

θ_0 launch site geodetic latitude

and

$$\theta = \cos^{-1}\left(\frac{X \cos A_z \sin \theta_1 + Y \cos \theta_1 - Z \sin A_2 \sin \theta_1}{R}\right)$$

where

A_{Z_L} = launch azimuth

portions of the N matrix are defined in the following manner

$$N_I = \begin{bmatrix} \frac{Z \cos \theta_1 + Y \sin A_{Z_L} \sin \theta_1}{R \sin \theta} & \frac{-X \sin A_{Z_L} \sin \theta_1 - Z \cos A_{Z_L} \sin \theta_1}{R \sin \theta} & \frac{Y \cos A_{Z_L} \sin \theta_1 - X \cos \theta_1}{R \sin \theta} \\ \frac{X}{R} & \frac{Y}{R} & \frac{Z}{R} \\ \frac{\frac{X}{R} \cos \theta - \cos A_{Z_L} \sin \theta_1}{\sin \theta} & \frac{\frac{Y}{R} \cos \theta - \cos \theta_1}{\sin \theta} & \frac{\frac{Z}{R} \cos \theta + \sin A_{Z_L} \sin \theta_1}{\sin \theta} \end{bmatrix} = D^T$$

$$N_{II} = \begin{bmatrix} \frac{\partial W_S}{\partial X} & \frac{\partial W_S}{\partial Y} & \frac{\partial W_S}{\partial Z} \\ \frac{\partial U_S}{\partial X} & \frac{\partial U_S}{\partial Y} & \frac{\partial U_S}{\partial Z} \\ \frac{\partial V_S}{\partial X} & \frac{\partial V_S}{\partial Y} & \frac{\partial V_S}{\partial Z} \end{bmatrix}$$

where

$$\frac{\partial W_S}{\partial X} = \frac{A_{32}U - A_{22}V - W_S\left(N_{31} \cos \theta + N_{21} \sin \theta\right)}{R \sin \theta}$$

$$\frac{\partial W_S}{\partial Y} = \frac{A_{12}V - A_{32}W - W_S\left(N_{32} \cos \theta + N_{22} \sin \theta\right)}{R \sin \theta}$$

$$\frac{\partial W_S}{\partial Z} = \frac{A_{22}W - A_{12}U - W_S\left(N_{33} \cos \theta + N_{23} \sin \theta\right)}{R \sin \theta}$$

$$\frac{\partial U_S}{\partial X} = \left(W - N_{21}\ U_S\right)\Big/R$$

$$\frac{\partial U_S}{\partial Y} = \left(U - N_{22}\ U_S\right)\Big/R$$

$$\frac{\partial U_S}{\partial Z} = \left(V - N_{23}\ U_S\right)\Big/R$$

$$\frac{\partial U_S}{\partial X} = \left(N_{11}\ W_S \operatorname{ctn} \theta - N_{31}\ U_S\right)\Big/R$$

$$\frac{\partial V_S}{\partial Y} = \left(N_{12}\ W_S \operatorname{ctn} \theta - N_{32}\ U_S\right)\Big/R$$

$$\frac{\partial V_S}{\partial Z} = \left(N_{13}\ W_S\ \text{ctn}\ \theta - N_{33}\ U_S\right)\Big/R$$

and

$$A = \begin{bmatrix} \sin A_{Z_L} & \cos A_{Z_L} \sin \theta_1 & -\cos A_{Z_L} \cos \theta_1 \\ 0 & \cos \theta_1 & \sin \theta_1 \\ \cos A_{Z_L} & -\sin A_{Z_L} \sin \theta_1 & \sin A_{Z_L} \cos \theta_1 \end{bmatrix}$$

$$N_{III} = 0$$

$$N_{IV} = \begin{bmatrix} \frac{N_{11}}{R \sin \theta} & \frac{N_{12}}{R \sin \theta} & \frac{N_{13}}{R \sin \theta} \\ N_{21} & N_{22} & N_{23} \\ \frac{N_{31}}{R} & \frac{N_{32}}{R} & \frac{N_{33}}{R} \end{bmatrix} .$$

Thus, N is the matrix of first partial derivatives

$$N = \frac{\partial \hat{W}_S}{\partial \hat{W}}$$

where $\hat{W}_S$ is the 6 x 1 vector of spherical coordinate system state components

$$\hat{W}_S = \begin{Bmatrix} W_S \\ R_S \\ V_S \\ \phi \\ R \\ \theta \end{Bmatrix}$$

where:

$$W_s = \overline{V} \cdot \hat{E}$$

$$U_s = \overline{V} \cdot \hat{R}$$

$$V_s = \overline{V} \cdot \hat{S}$$

$$\phi = \overline{R} \cdot \hat{E}$$

$$r = \overline{R} \cdot \hat{R}$$

$$\theta = \overline{R} \cdot \hat{S}$$

and

$$\hat{R} = \frac{\overline{X}}{R}$$

$$\hat{E} = \frac{\hat{\omega} \times \hat{R}}{|\hat{\omega} \times \hat{R}|}$$

$$\hat{S} = \hat{E} \times \hat{R}$$

and $\hat{W}$ is the 6 x 1 vector of plumbline components

$$\hat{W} = \begin{Bmatrix} w \\ u \\ v \\ x \\ y \\ z \end{Bmatrix} .$$

$$W_E = \overline{V}_E \cdot E$$

$$U_E = \overline{U}_E \cdot R$$

$$V_E = \overline{V}_E \cdot S$$

where $\overline{V}_E = \overline{V} - (\overline{W} \times \overline{R})$

$$P = \begin{bmatrix} \frac{\partial W_E}{\partial \overline{W}} & \frac{\partial U_E}{\partial \overline{W}} & \frac{\partial V_E}{\partial \overline{W}} \\ \frac{\partial W_E}{\partial \overline{X}} & \frac{\partial U_E}{\partial \overline{X}} & \frac{\partial V_E}{\partial \overline{X}} \end{bmatrix} .$$

Appendix E

FIRST PARTIAL DERIVATIVES OF GRAVITATIONAL ACCELERATION WITH RESPECT TO PLUMBLINE POSITION COORDINATES

Appendix E

FIRST PARTIAL DERIVATIVES OF GRAVITATIONAL ACCELERATION WITH RESPECT TO PLUMBLINE POSITION COORDINATES

The matrix K is defined as the matrix of the first partial derivatives of the gravitational acceleration vector in the plumbline system with respect to the plumbline position coordinates. This matrix is used in the gravity related terms of the adjoint (Euler-Langrange) equations.

$$K = \begin{bmatrix} \frac{\partial G_X}{\partial X} & \frac{\partial G_X}{\partial Y} & \frac{\partial G_X}{\partial Z} \\ \frac{\partial G_Y}{\partial X} & \frac{\partial G_Y}{\partial Y} & \frac{\partial G_Y}{\partial Z} \\ \frac{\partial G_Z}{\partial X} & \frac{\partial G_Z}{\partial Y} & \frac{\partial G_Z}{\partial Z} \end{bmatrix} = \begin{bmatrix} G_{XX} & G_{XY} & G_{XZ} \\ G_{YX} & G_{YY} & G_{YZ} \\ G_{ZX} & G_{ZY} & G_{ZZ} \end{bmatrix}$$

$$K = G_{11} I + \begin{bmatrix} X & A_{12} \\ Y & A_{22} \\ Z & A_{32} \end{bmatrix} \begin{bmatrix} G_{22} & G_{23} \\ G_{32} & G_{33} \end{bmatrix} \begin{bmatrix} X & Y & Z \\ A_{12} & A_{22} & A_{32} \end{bmatrix}$$

G_{11} is defined in subsection 3.1. I is a 3 x 3 identity matrix, the A_{ij} are elements of the A matrix, and

$$G_{22} = \frac{1}{R}\left(\frac{\partial G_{11}}{\partial R} + \frac{\cos\theta}{R}\frac{\partial G_{11}}{\partial \theta}\right)$$

$$= -\frac{3}{R^2}\left[G_{11} - \frac{u_e}{R^3}\left(J\left(\frac{R_e}{R}\right)^2\left(\frac{2}{3} - \frac{20}{3}\cos^2\theta\right) + H\left(\frac{R_e}{R}\right)^3\left(4 - 14\cos^2\theta\right)\cos\theta\right.\right.$$

$$\left.\left. + D\left(\frac{R_e}{R}\right)^4\left(\frac{4}{7} - \left(12 - 24\cos^2\theta\right)\cos^2\theta\right)\right)\right]$$

$$G_{23} = G_{32} = -\frac{1}{R\sin\theta}\frac{\partial G_{TO}}{\partial\theta} = -\frac{1}{R}\left(\frac{\partial G_{TO}}{\partial R} - \frac{\cot\theta}{R}\frac{\partial G_{TO}}{\partial\theta}\right)$$

$$= \frac{\mu_e}{R^4}\left[10\,J\left(\frac{R_e}{R}\right)^2\cos\theta - H\left(\frac{R_e}{R}\right)^3\left(3 - 21\cos^2\theta\right) + D\left(\frac{R_e}{R}\right)^4\left(12 - 36\cos^2\theta\right)\cos\theta\right]$$

$$G_{33} = \frac{1}{R\sin\theta}\frac{\partial G_{TO}}{\partial\theta}$$

$$= -\frac{\mu_e}{R^3}\left[2\,J\left(\frac{R_e}{R}\right)^2 + 6H\left(\frac{R_e}{R}\right)^3\cos\theta + \frac{12}{7}\,J\left(\frac{R_e}{R}\right)^4\left(1 - 7\cos^2\theta\right)\right]$$

The fact that K is symmetric can be anticipated, since K is also the matrix of second partial derivatives of the gravitational potential function $U(R, \theta)$ with respect to the plumbline position coordinates.

In the event that a spherical earth is being simulated, K reduces to

$$K = G_{11}I + \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} G_{22}\,[X\ Y\ Z]$$

with

$$G_{22} = -\frac{3}{R^2}\,G_{11}$$

since

$$G_{23} = G_{32} = G_{33} = 0\ .$$

Appendix F

PARTIAL DERIVATIVES OF THE EQUATIONS OF MOTION WITH RESPECT TO THE STATE AND THE CONTROL

GENERAL EQUATIONS OF MOTION

$$\ddot{\bar{x}} = \frac{1}{m}\begin{bmatrix} F_x & F_y & F_z \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} + \bar{G}$$

$\frac{\partial}{\partial \bar{x}}$	$\frac{1}{m}\begin{bmatrix} \frac{\partial F_x}{\partial \bar{x}} & \frac{\partial F_y}{\partial \bar{x}} & \frac{\partial F_z}{\partial \bar{x}} \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} + \frac{\partial}{\partial \bar{x}}\bar{G}^*$
$\frac{\partial}{\partial \dot{\bar{x}}}$	$\frac{1}{m}\begin{bmatrix} \frac{\partial F_x}{\partial \dot{\bar{x}}} & \frac{\partial F_y}{\partial \dot{\bar{x}}} & \frac{\partial F_z}{\partial \dot{\bar{x}}} \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix}$
$\frac{\partial}{\partial m}$	$-\frac{1}{m^2}\begin{bmatrix} F_x & F_y & F_z \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix} + \frac{1}{m}\begin{bmatrix} \frac{\partial F_x}{\partial m} & \frac{\partial F_y}{\partial m} & \frac{\partial F_z}{\partial m} \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix}$
$\frac{\partial}{\partial \chi_p}$	$\frac{1}{m}\left\{\begin{bmatrix} F_x & F_y & F_z \end{bmatrix}\begin{bmatrix} \frac{\partial \hat{X}}{\partial \chi_p} \\ \frac{\partial \hat{Y}}{\partial \chi_p} \\ \frac{\partial \hat{Z}}{\partial \chi_p} \end{bmatrix} + \begin{bmatrix} \frac{\partial F_x}{\partial \chi_p} & \frac{\partial F_y}{\partial \chi_p} & \frac{\partial F_z}{\partial \chi_p} \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix}\right\}$
$\frac{\partial}{\partial \chi_y}$	$\frac{1}{m}\left\{\begin{bmatrix} F_x & F_y & F_z \end{bmatrix}\begin{bmatrix} 0 \\ \frac{\partial \hat{Y}}{\partial \chi_y} \\ \frac{\partial \hat{Z}}{\partial \chi_y} \end{bmatrix} + \begin{bmatrix} \frac{\partial F_x}{\partial \chi_y} & \frac{\partial F_y}{\partial \chi_y} & \frac{\partial F_z}{\partial \chi_y} \end{bmatrix}\begin{bmatrix} \hat{X} \\ \hat{Y} \\ \hat{Z} \end{bmatrix}\right\}$

*See Appendix E

$$\frac{\partial^2}{\partial x_p^2} \quad \frac{1}{m}\left\{\left[\frac{\partial^2 F_x}{\partial x_p^2} - F_x \;\vdots\; \frac{\partial^2 F_y}{\partial x_p^2} - F_y \;\vdots\; \frac{\partial^2 F_z}{\partial x_p^2} - F_z\right]\begin{bmatrix}\hat{X}\\ \hat{Y}\\ \hat{Z}\end{bmatrix} + 2\left[\frac{\partial F_x}{\partial x_p}\;\frac{\partial F_y}{\partial x_p}\;\frac{\partial F_z}{\partial x_p}\right]\begin{bmatrix}\frac{\partial \hat{X}}{\partial x_p}\\ \frac{\partial \hat{Y}}{\partial x_p}\\ \frac{\partial \hat{Z}}{\partial x_p}\end{bmatrix}\right\}$$

$$\frac{\partial^2}{\partial x_y^2} \quad \frac{1}{m}\left\{\left[\frac{\partial^2 F_x}{\partial x_y^2} \;\vdots\; \frac{\partial^2 F_y}{\partial x_y^2} - F_y \;\vdots\; \frac{\partial^2 F_z}{\partial x_y^2} - F_z\right]\begin{bmatrix}\hat{X}\\ \hat{Y}\\ \hat{Z}\end{bmatrix} + 2\left[\frac{\partial F_x}{\partial x_y}\;\frac{\partial F_y}{\partial x_y}\;\frac{\partial F_z}{\partial x_y}\right]\begin{bmatrix}0\\ \frac{\partial \hat{Y}}{\partial x_y}\\ \frac{\partial \hat{Z}}{\partial x_y}\end{bmatrix}\right\}$$

$$\frac{\partial^2}{\partial x_p \partial x_y} \quad \frac{1}{m}\left\{\left[\frac{\partial^2 F_x}{\partial x_p \partial x_y}\;\frac{\partial^2 F_y}{\partial x_p \partial x_y}\;\frac{\partial^2 F_z}{\partial x_p \partial x_y}\right]\begin{bmatrix}\hat{X}\\ \hat{Y}\\ \hat{Z}\end{bmatrix} + \left[\frac{\partial F_x}{\partial x_p}\;\frac{\partial F_y}{\partial x_p}\;\frac{\partial F_z}{\partial x_p}\right]\begin{bmatrix}0\\ \frac{\partial \hat{Y}}{\partial x_y}\\ \frac{\partial \hat{Z}}{\partial x_y}\end{bmatrix} + \left[\frac{\partial F_x}{\partial x_y}\;\frac{\partial F_y}{\partial x_y}\;\frac{\partial F_z}{\partial x_y}\right]\begin{bmatrix}\frac{\partial \hat{X}}{\partial x_p}\\ \frac{\partial \hat{Y}}{\partial x_p}\\ \frac{\partial \hat{Z}}{\partial x_p}\end{bmatrix} + [F_x\;\; F_y\;\; F_z]\begin{bmatrix}0\\ \frac{\partial^2 \hat{Y}}{\partial x_p \partial x_y}\\ \frac{\partial^2 \hat{Z}}{\partial x_p \partial x_y}\end{bmatrix}\right\}$$

GENERAL FORCE EQUATIONS

$$\bar{F} = \bar{T} + q\,S\,\bar{A}$$

$\frac{\partial}{\partial \bar{x}}$	$\frac{\partial \bar{T}}{\partial \bar{x}} + S\left(\frac{\partial q}{\partial \bar{x}}\bar{A} + q\frac{\partial \bar{A}}{\partial \bar{x}}\right)$
$\frac{\partial}{\partial \dot{\bar{x}}}$	$\frac{\partial \bar{T}}{\partial \dot{\bar{x}}} + S\left(\frac{\partial q}{\partial \dot{\bar{x}}}\bar{A} + q\frac{\partial \bar{A}}{\partial \dot{\bar{x}}}\right)$
$\frac{\partial}{\partial m}$	$\frac{\partial \bar{T}}{\partial m}$
$\frac{\partial}{\partial x_p}$	$\frac{\partial \bar{T}}{\partial x_p} + q\,S\,\frac{\partial \bar{A}}{\partial x_p}$
$\frac{\partial}{\partial x_y}$	$\frac{\partial \bar{T}}{\partial x_y} + q\,S\,\frac{\partial \bar{A}}{\partial x_y}$
$\frac{\partial^2}{\partial x_p^2}$	$\frac{\partial^2 \bar{T}}{\partial x_p^2} + q\,S\,\frac{\partial^2 \bar{A}}{\partial x_p^2}$
$\frac{\partial^2}{\partial x_y^2}$	$\frac{\partial^2 \bar{T}}{\partial x_y^2} + q\,S\,\frac{\partial^2 \bar{A}}{\partial x_y^2}$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\frac{\partial^2 \bar{T}}{\partial x_p \partial x_y} + q\,S\,\frac{\partial^2 \bar{A}}{\partial x_p \partial x_y}$

TRAJECTORY PARAMETERS: Relative Velocity Vector ($\bar{V}_R$), Relative Velocity Magnitude (V_R), Radius Unit Vector ($\hat{R}$), and Radius Magnitude (R)

	$\bar{V}_R = \dot{\bar{x}} - (\bar{\omega} \times \bar{x}) + \bar{W}$	$V_R = (\bar{V}_R \cdot \bar{V}_R)^{1/2}$	$\hat{R} = \frac{\bar{x}}{R}$	$R = (\bar{x} \cdot \bar{x})^{1/2}$
$\frac{\partial}{\partial \bar{x}}$	$-\left(\bar{\omega} \times \frac{\partial \bar{x}}{\partial \bar{x}}\right) + \frac{\partial \bar{W}}{\partial \bar{x}}$	$\frac{\bar{V}_R}{V_R} \cdot \frac{\partial \bar{V}_R}{\partial \bar{x}}$	$\frac{1}{R}\left(\frac{\partial \bar{x}}{\partial \bar{x}} - \hat{R}\,\frac{\partial R}{\partial \bar{x}}\right)$	$\frac{\bar{x}}{R} \cdot \frac{\partial \bar{x}}{\partial \bar{x}}$
$\frac{\partial \dot{\bar{x}}}{\partial \dot{\bar{x}}}$	$\frac{\partial \dot{\bar{x}}}{\partial \dot{\bar{x}}}$	$\frac{\bar{V}_R}{V_R} \cdot \frac{\partial \bar{V}_R}{\partial \dot{\bar{x}}}$	0	0
$\frac{\partial}{\partial m}$	0	0	0	0
$\frac{\partial}{\partial x_p}$	0	0	0	0
$\frac{\partial}{\partial x_y}$	0	0	0	0
$\frac{\partial^2}{\partial x_p^2}$	0	0	0	0
$\frac{\partial^2}{\partial x_y^2}$	0	0	0	0
$\frac{\partial^2}{\partial x_p \partial x_y}$	0	0	0	0

TRAJECTORY PARAMETERS: Wind Vector ($\bar{W}$), East Direction Unit Vector ($\hat{E}$), and North Direction Unit Vector ($\hat{N}$)

	$\bar{W} = W_H[\sin A_{Z_W}\hat{E} + \cos A_{Z_W}\hat{N}]$	$\hat{E} = \frac{1}{\sin\theta}(\bar{\omega}\times\hat{R})$	$\hat{N} = \hat{R}\times\hat{E}$
$\frac{\partial}{\partial \bar{x}}$	$\frac{\partial W_H}{\partial h}\frac{\partial h}{\partial \bar{x}}[\sin A_{Z_W}\hat{E} + \cos A_{Z_W}\hat{N}] + W_H[\sin A_{Z_W}(\frac{\partial \hat{E}}{\partial \bar{x}} - \hat{N}\frac{\partial A_{Z_W}}{\partial h}\frac{\partial h}{\partial \bar{x}}) + \cos A_{Z_W}(\frac{\partial \hat{N}}{\partial \bar{x}} + \hat{E}\frac{\partial A_{Z_W}}{\partial h}\frac{\partial h}{\partial \bar{x}})]$	$\frac{\cos\theta}{\sin^3\theta}(\bar{\omega}\cdot\frac{\partial \hat{R}}{\partial \bar{x}})(\bar{\omega}\times\hat{R}) + \frac{1}{\sin\theta}(\bar{\omega}\times\frac{\partial \hat{R}}{\partial \bar{x}})$	$\hat{R}\times\frac{\partial \hat{E}}{\partial \bar{x}} + \frac{\partial \hat{R}}{\partial \bar{x}}\times\hat{E}$
$\frac{\partial}{\partial \dot{x}}$	0	0	0
$\frac{\partial}{\partial m}$	0	0	0
$\frac{\partial}{\partial x_p}$	0	0	0
$\frac{\partial}{\partial x_y}$	0	0	0
$\frac{\partial^2}{\partial x_p^2}$	0	0	0
$\frac{\partial^2}{\partial x_y^2}$	0	0	0
$\frac{\partial^2}{\partial x_p \partial x_y}$	0	0	0

TRAJECTORY PARAMETERS: Altitude (h), Dynamic Pressure (Q), Mach Number (Mn), and Reynolds Number (Rn)

$$h = R - \frac{(1-\epsilon)R_e}{\sqrt{(1-\epsilon)^2 \sin^2\theta + \cos^2\theta}} \qquad q = \frac{1}{2}\rho V_R^2 \qquad Mn = \frac{V_R}{V_S} \qquad Rn = \frac{\rho V_R}{\mu}\,\ell$$

$\frac{\partial}{\partial \bar{x}}$	$\frac{\bar{x}}{R} - \frac{R^3(\theta)\epsilon(2-\epsilon)\cos\theta}{R_e^3(1-\epsilon)^2}\left(\frac{\bar{x}}{R^2}\cos\theta - \frac{\bar{z}}{R}\right)$	$\rho V_R \frac{\partial V_R}{\partial \bar{x}} + \frac{1}{2} V_R^2 \frac{\partial \rho^*}{\partial h}\frac{\partial h}{\partial \bar{x}}$	$\frac{1}{V_S}\left(\frac{\partial V_R}{\partial \bar{x}} - Mn \frac{\partial V_S^*}{\partial h}\frac{\partial h}{\partial \bar{x}}\right)$	$\ell\left[\frac{1}{\mu}\left(\rho \frac{\partial V_R}{\partial \bar{x}} + V_R \frac{\partial \rho^*}{\partial h}\frac{\partial h}{\partial \bar{x}}\right) - \frac{\rho V_R}{\mu^2}\frac{\partial \mu^*}{\partial h}\frac{\partial h}{\partial \bar{x}}\right]$
$\frac{\partial}{\partial \bar{z}}$	0	$\rho V_R \frac{\partial V_R}{\partial \bar{z}}$	$\frac{1}{V_S}\frac{\partial V_R}{\partial \bar{z}}$	$\ell\left[\frac{\rho}{\mu}\frac{\partial V_R}{\partial \bar{z}}\right]$
$\frac{\partial}{\partial m}$	0	0	0	0
$\frac{\partial}{\partial x_p}$	0	0	0	0
$\frac{\partial}{\partial x_y}$	0	0	0	0
$\frac{\partial^2}{\partial x_p^2}$	0	0	0	0
$\frac{\partial^2}{\partial x_y^2}$	0	0	0	0
$\frac{\partial^2}{\partial x_p \partial x_y}$	0	0	0	0

*Determine by atmosphere subroutine using spline interpolation method

AERODYNAMIC FORCE AND MOMENT EQUATIONS: Angle of Attack (α) and Sideslip Angle (β)

	$\alpha = \tan^{-1}\left(\frac{\dot{X} - \dot{V}_M}{\dot{Z} - \dot{V}_M}\right)$	$\beta = \sin^{-1}\left(\frac{\dot{Y} - \dot{V}_M}{V_M}\right)$
$\frac{\partial}{\partial \dot{X}}$	$\frac{\cos^2\alpha}{(\dot{Z} - \dot{V}_M)}\left[\left(\dot{X} - \frac{\partial \dot{V}_M}{\partial \dot{X}}\right) - \tan\alpha\left(\dot{Z} - \frac{\partial \dot{V}_M}{\partial \dot{X}}\right)\right]$	$\frac{1}{V_M\cos\beta}\left[\left(\dot{Y} - \frac{\partial \dot{V}_M}{\partial \dot{X}}\right) - \sin\beta\,\frac{\partial V_M}{\partial \dot{X}}\right]$
$\frac{\partial}{\partial \dot{Z}}$	$\frac{\cos^2\alpha}{(\dot{Z} - \dot{V}_M)}\left[\left(\dot{X} - \frac{\partial \dot{V}_M}{\partial \dot{Z}}\right) - \tan\alpha\left(\dot{Z} - \frac{\partial \dot{V}_M}{\partial \dot{Z}}\right)\right]$	$\frac{1}{V_M\cos\beta}\left[\left(\dot{Y} - \frac{\partial \dot{V}_M}{\partial \dot{Z}}\right) - \sin\beta\,\frac{\partial V_M}{\partial \dot{Z}}\right]$
$\frac{\partial}{\partial \dot{V}_M}$	0	0
$\frac{\partial}{\partial x_p}$	$\frac{\cos^2\alpha}{(\dot{Z} - \dot{V}_M)}\left[\left(\frac{\partial \dot{X}}{\partial x_p} - \dot{V}_M\right) - \tan\alpha\left(\frac{\partial \dot{Z}}{\partial x_p} - \dot{V}_M\right)\right]$	$V_M\frac{1}{\cos\beta}\left(\frac{\partial \dot{Z}}{\partial x_p} - \dot{V}_M\right)$
$\frac{\partial}{\partial x_y}$	$\frac{\cos^2\alpha}{(\dot{Z} - \dot{V}_M)}\left[\left(\frac{\partial \dot{X}}{\partial x_y} - \dot{V}_M\right) - \tan\alpha\left(\frac{\partial \dot{Z}}{\partial x_y} - \dot{V}_M\right)\right]$	$V_M\frac{1}{\cos\beta}\left(\frac{\partial \dot{Z}}{\partial x_y} - \dot{V}_M\right)$
$\frac{\partial^2}{\partial x_p^2}$	$2\frac{\partial\alpha}{\partial x_p}\left[\tan\alpha\,\frac{\partial\alpha}{\partial x_p} + \frac{\left(\frac{\partial \dot{Z}}{\partial x_p} - \dot{V}_M\right)}{(\dot{Z} - \dot{V}_M)}\right] + \frac{\cos^2\alpha}{(\dot{Z} - \dot{V}_M)}\left[\left(\frac{\partial^2 \dot{X}}{\partial x_p^2} - \dot{V}_M\right) - \tan\alpha\left(\frac{\partial^2 \dot{Z}}{\partial x_p^2} - \dot{V}_M\right)\right]$	$\tan\beta\left(\frac{\partial\beta}{\partial x_p}\right)^2 + V_M\frac{1}{\cos\beta}\left(\frac{\partial^2 \dot{Z}}{\partial x_p^2} - \dot{V}_M\right)$
$\frac{\partial^2}{\partial x_y^2}$	$-2\frac{\partial\alpha}{\partial x_y}\left[\tan\alpha\,\frac{\partial\alpha}{\partial x_y} + \frac{\left(\frac{\partial \dot{Z}}{\partial x_y} - \dot{V}_M\right)}{(\dot{Z} - \dot{V}_M)}\right] - \frac{\cos^2\alpha}{(\dot{Z} - \dot{V}_M)}\tan\alpha\left(\frac{\partial^2 \dot{X}}{\partial x_y^2} - \dot{V}_M\right)$	$\tan\beta\left(\frac{\partial\beta}{\partial x_y}\right)^2 + V_M\frac{1}{\cos\beta}\left(\frac{\partial^2 \dot{Z}}{\partial x_y^2} - \dot{V}_M\right)$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$-2\tan\alpha\,\frac{\partial\alpha}{\partial x_p}\frac{\partial\alpha}{\partial x_y}\frac{\left(\frac{\partial \dot{Z}}{\partial x_p} - \dot{V}_M\right)}{(\dot{Z} - \dot{V}_M)}\frac{\partial\alpha}{\partial x_y}\frac{\left(\frac{\partial \dot{Z}}{\partial x_y} - \dot{V}_M\right)}{(\dot{Z} - \dot{V}_M)}\frac{\partial\alpha}{\partial x_p} - \frac{\cos^2\alpha}{(\dot{Z} - \dot{V}_M)}\tan\alpha\left(\frac{\partial^2 \dot{X}}{\partial x_p \partial x_y} - \dot{V}_M\right)$	$\tan\beta\left(\frac{\partial\beta}{\partial x_p}\right)\left(\frac{\partial\beta}{\partial x_y}\right) + V_M\frac{1}{\cos\beta}\left(\frac{\partial^2 \dot{Z}}{\partial x_p \partial x_y} - \dot{V}_M\right)$

AERODYNAMIC FORCE AND MOMENT EQUATIONS: Aerodynamic Force Coefficients: Axial Force Coefficient (C_A), Normal Force Coefficient (C_N), Side Force Coefficient (C_S), Zeronormal Force Angle of Attack (α_o)

	$C_A = C_{A_i} + \Delta C_A + F_{A_B/QS}$	$C_N = C_{N_i} + \Delta C_N + F_{N_B/QS}$	$C_S = C_{S_o} + C_{y_\beta}\beta$
$\frac{\partial}{\partial \bar{x}}$	$\frac{\partial C_{A_i}}{\partial \bar{x}} + \frac{\partial \Delta C_A}{\partial \bar{x}} + \left(\frac{\partial F_{A_B}}{\partial h}\frac{\partial h}{\partial \bar{x}} - \frac{F_{A_B}}{q}\frac{\partial q}{\partial \bar{x}}\right)/q\alpha$	$\frac{\partial C_{N_i}}{\partial \bar{x}} + \frac{\partial \Delta C_N}{\partial \bar{x}} + \left(\frac{\partial F_{N_B}}{\partial h}\frac{\partial h}{\partial \bar{x}} - \frac{F_{N_B}}{q}\frac{\partial q}{\partial \bar{x}}\right)/q\alpha$	$\left(\frac{\partial C_{S_\beta}}{\partial M} + \frac{\partial C_{y_\beta}}{\partial M}\right)\frac{\partial M}{\partial \bar{x}}\beta + \frac{C_S}{\beta}\frac{\partial \beta}{\partial \bar{x}}$
$\frac{\partial}{\partial \bar{z}}$	$\frac{\partial C_{A_i}}{\partial \bar{z}} + \frac{\partial \Delta C_A}{\partial \bar{z}} - F_{A_B}\frac{\partial q}{\partial \bar{z}}/q^2\alpha$	$\frac{\partial C_{N_i}}{\partial \bar{z}} + \frac{\partial \Delta C_N}{\partial \bar{z}} - F_{N_B}\frac{\partial q}{\partial \bar{z}}/q^2\alpha$	$\left(\frac{\partial C_{S_\beta}}{\partial M} + \frac{\partial C_{y_\beta}}{\partial M}\right)\frac{\partial M}{\partial \bar{z}}\beta + \frac{C_S}{\beta}\frac{\partial \beta}{\partial \bar{z}}$
$\frac{\partial}{\partial m}$	0	0	0
$\frac{\partial}{\partial x_p}$	$\frac{\partial C_{A_i}}{\partial x_p} + \frac{\partial \Delta C_A}{\partial x_p}$	$\frac{\partial C_{N_i}}{\partial x_p} + \frac{\partial \Delta C_N}{\partial x_p}$	$C_{y_\beta}\frac{\partial \beta}{\partial x_p}$
$\frac{\partial}{\partial x_y}$	$\frac{\partial C_{A_i}}{\partial x_y} + \frac{\partial \Delta C_A}{\partial x_y}$	$\frac{\partial C_{N_i}}{\partial x_y} + \frac{\partial \Delta C_N}{\partial x_y}$	$C_{y_\beta}\frac{\partial \beta}{\partial x_y}$
$\frac{\partial^2}{\partial x_p^2}$	$\frac{\partial^2 C_{A_i}}{\partial x_p^2} + \frac{\partial^2 \Delta C_A}{\partial x_p^2}$	$\frac{\partial^2 C_{N_i}}{\partial x_p^2} + \frac{\partial^2 \Delta C_N}{\partial x_p^2}$	$C_{y_\beta}\frac{\partial^2 \beta}{\partial x_p^2}$
$\frac{\partial^2}{\partial x_y^2}$	$\frac{\partial^2 C_{A_i}}{\partial x_y^2} + \frac{\partial^2 \Delta C_A}{\partial x_y^2}$	$\frac{\partial^2 C_{N_i}}{\partial x_y^2} + \frac{\partial^2 \Delta C_N}{\partial x_y^2}$	$C_{y_\beta}\frac{\partial^2 \beta}{\partial x_y^2}$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\frac{\partial^2 C_{A_i}}{\partial x_p \partial x_y} + \frac{\partial^2 \Delta C_A}{\partial x_p \partial x_y}$	$\frac{\partial^2 C_{N_i}}{\partial x_p \partial x_y} + \frac{\partial^2 \Delta C_N}{\partial x_p \partial x_y}$	$C_{y_\beta}\frac{\partial^2 \beta}{\partial x_p \partial x_y}$

AERODYNAMIC FORCE AND MOMENT EQUATIONS: Additional Aerodynamic Parameters

	$C_{N_\delta} = C_{N_\omega} + C_{N_{\dot\omega}}$	$C_{A_\delta} = C_{A_\omega} + C_{A_{\dot\omega}}$
$\frac{\partial}{\partial \dot{x}}$	$\left(\frac{\partial C^{*}_{N_{\dot\omega}}}{\partial M_{\dot\omega}} + \frac{\partial C^{*}_{N_{\dot\omega}}}{\partial M_{\dot\omega}}\omega\right)\frac{\partial M_{\dot\omega}}{\partial \dot{x}} + C_{N_{\dot\omega}}\frac{\partial \dot\omega}{\partial \dot{x}}$	$\left(\frac{\partial C^{*}_{A_{\dot\omega}}}{\partial M_{\dot\omega}} + \frac{\partial C^{*}_{A_{\dot\omega}}}{\partial M_{\dot\omega}}\omega\right)\frac{\partial M_{\dot\omega}}{\partial \dot{x}} + C_{A_{\dot\omega}}\frac{\partial \dot\omega}{\partial \dot{x}}$
$\frac{\partial}{\partial x}$	$\left(\frac{\partial C^{*}_{N_{\ddot\omega}}}{\partial M_{\ddot\omega}} + \frac{\partial C^{*}_{N_{\ddot\omega}}}{\partial M_{\ddot\omega}}\omega\right)\frac{\partial M_{\ddot\omega}}{\partial \ddot{x}} + C_{N_{\ddot\omega}}\frac{\partial \ddot\omega}{\partial \ddot{x}}$	$\left(\frac{\partial C^{*}_{A_{\ddot\omega}}}{\partial M_{\ddot\omega}} + \frac{\partial C^{*}_{A_{\ddot\omega}}}{\partial M_{\ddot\omega}}\omega\right)\frac{\partial M_{\ddot\omega}}{\partial \ddot{x}} + C_{A_{\ddot\omega}}\frac{\partial \ddot\omega}{\partial \ddot{x}}$
$\frac{\partial}{\partial m}$	0	0
$\frac{\partial}{\partial x_p}$	$C_{N_\omega}\frac{\partial \omega}{\partial x_p}$	$C_{A_\omega}\frac{\partial \omega}{\partial x_p}$
$\frac{\partial}{\partial x_y}$	$C_{N_\omega}\frac{\partial \omega}{\partial x_y}$	$C_{A_\omega}\frac{\partial \omega}{\partial x_y}$
$\frac{\partial^2}{\partial x_p^2}$	$C_{N_\omega}\frac{\partial^2 \omega}{\partial x_p^2}$	$C_{A_\omega}\frac{\partial^2 \omega}{\partial x_p^2}$
$\frac{\partial^2}{\partial x_y^2}$	$C_{N_\omega}\frac{\partial^2 \omega}{\partial x_y^2}$	$C_{A_\omega}\frac{\partial^2 \omega}{\partial x_y^2}$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$C_{N_\omega}\frac{\partial^2 \omega}{\partial x_p \partial x_y}$	$C_{A_\omega}\frac{\partial^2 \omega}{\partial x_p \partial x_y}$

*Determined by spline interpolation method

AERODYNAMIC FORCE AND MOMENT EQUATIONS: Viscous Interaction Terms for Axial and Normal Forces and Pitch Moment

$$\Delta C_i = (C_{i_{MF}} - C_{i_{CF}}) \sin(9.009(K_n - 0.01)) + C_{i_{CF}}$$

		FOR $K_n < 0.01$	FOR $K_n > 10$
$\frac{\partial}{\partial \bar{x}}$	$\left(\frac{\partial C_{i_{MF}}}{\partial \bar{x}} - \frac{\partial C_{i_{CF}}}{\partial \bar{x}}\right) \sin(9.009(K_n - 0.01)) + (C_{i_{MF}} - C_{i_{CF}}) \cos(9.009(K_n - 0.01))\, 9.009 \frac{\partial K_n}{\partial \bar{x}} + \frac{\partial C_{i_{CF}}}{\partial \bar{x}}$	$\frac{\partial C_{i_{CF}}}{\partial \bar{x}}$	$\frac{\partial C_{i_{MF}}}{\partial \bar{x}}$
$\frac{\partial}{\partial \dot{x}}$	$\left(\frac{\partial C_{i_{MF}}}{\partial \dot{x}} - \frac{\partial C_{i_{CF}}}{\partial \dot{x}}\right) \sin(9.009(K_n - 0.01)) + (C_{i_{MF}} - C_{i_{CF}}) \cos(9.009(K_n - 0.01))\, 9.009 \frac{\partial K_n}{\partial \dot{x}} + \frac{\partial C_{i_{CF}}}{\partial \dot{x}}$	$\frac{\partial C_{i_{CF}}}{\partial \dot{x}}$	$\frac{\partial C_{i_{MF}}}{\partial \dot{x}}$
$\frac{\partial}{\partial m}$	0	0	0
$\frac{\partial}{\partial x_p}$	$\left(\frac{\partial C_{i_{MF}}}{\partial x_p} - \frac{\partial C_{i_{CF}}}{\partial x_p}\right) \sin(9.009(K_n - 0.01)) + \frac{\partial C_{i_{CF}}}{\partial x_p}$	$\frac{\partial C_{i_{CF}}}{\partial x_p}$	$\frac{\partial C_{i_{MF}}}{\partial x_p}$
$\frac{\partial}{\partial x_y}$	$\left(\frac{\partial C_{i_{MF}}}{\partial x_y} - \frac{\partial C_{i_{CF}}}{\partial x_y}\right) \sin(9.009(K_n - 0.01)) + \frac{\partial C_{i_{CF}}}{\partial x_y}$	$\frac{\partial C_{i_{CF}}}{\partial x_y}$	$\frac{\partial C_{i_{MF}}}{\partial x_y}$
$\frac{\partial^2}{\partial x_p^2}$	$\left(\frac{\partial^2 C_{i_{MF}}}{\partial x_p^2} - \frac{\partial^2 C_{i_{CF}}}{\partial x_p^2}\right) \sin(9.009(K_n - 0.01)) + \frac{\partial^2 C_{i_{CF}}}{\partial x_p^2}$	$\frac{\partial^2 C_{i_{CF}}}{\partial x_p^2}$	$\frac{\partial^2 C_{i_{MF}}}{\partial x_p^2}$
$\frac{\partial^2}{\partial x_y^2}$	$\left(\frac{\partial^2 C_{i_{MF}}}{\partial x_y^2} - \frac{\partial^2 C_{i_{CF}}}{\partial x_y^2}\right) \sin(9.009(K_n - 0.01)) + \frac{\partial^2 C_{i_{CF}}}{\partial x_y^2}$	$\frac{\partial^2 C_{i_{CF}}}{\partial x_y^2}$	$\frac{\partial^2 C_{i_{MF}}}{\partial x_y^2}$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\left(\frac{\partial^2 C_{i_{MF}}}{\partial x_p \partial x_y} - \frac{\partial^2 C_{i_{CF}}}{\partial x_p \partial x_y}\right) \sin(9.009(K_n - 0.01)) + \frac{\partial^2 C_{i_{CF}}}{\partial x_p \partial x_y}$	$\frac{\partial^2 C_{i_{CF}}}{\partial x_p \partial x_y}$	$\frac{\partial^2 C_{i_{MF}}}{\partial x_p \partial x_y}$

AERODYNAMIC FORCE AND MOMENT EQUATION: Viscous Interaction Terms, Knudsen Number (K) and mean molecular flow and continuous flow coefficient general forms ($C_{i_{MF}}$, $C_{i_{CF}}$) axial, and normal forces and pitch moment

	$K_n = 0.1407 \frac{\lambda}{V_R} \frac{P}{\rho}^{1/2}$	$(C_{i_{MF}}, C_{i_{CF}}) = a_i + b_i\alpha + c_i\alpha^2$
$\frac{\partial}{\partial \bar{x}}$	$K_n\left[\frac{1}{2}\left(\frac{1}{P}\frac{\partial P}{\partial h} - \frac{1}{\rho}\frac{\partial \rho}{\partial h} + \frac{2K_n}{\lambda}\frac{\partial T}{\partial h}\right)\frac{\partial h}{\partial x} - \frac{1}{V_R}\frac{\partial V_R}{\partial \bar{x}}\right]$	$(b_i + 2c_i\alpha)\frac{\partial \alpha}{\partial \bar{x}}$
$\frac{\partial}{\partial \dot{\bar{x}}}$	$-K_n \frac{1}{V_R}\frac{\partial V_R}{\partial \dot{\bar{x}}}$	$(b_i + 2c_i\alpha)\frac{\partial \alpha}{\partial \dot{\bar{x}}}$
$\frac{\partial}{\partial m}$	0	0
$\frac{\partial}{\partial \chi_p}$	0	$(b_i + 2c_i\alpha)\frac{\partial \alpha}{\partial \chi_p}$
$\frac{\partial}{\partial \chi_y}$	0	$(b_i + 2c_i\alpha)\frac{\partial \alpha}{\partial \chi_y}$
$\frac{\partial^2}{\partial \chi_p^2}$	0	$(b_i + 2c_i\alpha)\frac{\partial^2 \alpha}{\partial \chi_p^2} + 2c_i\left(\frac{\partial \alpha}{\partial \chi_p}\right)^2$
$\frac{\partial^2}{\partial \chi_y^2}$	0	$(b_i + 2c_i\alpha)\frac{\partial^2 \alpha}{\partial \chi_y^2} + 2c_i\left(\frac{\partial \alpha}{\partial \chi_y}\right)^2$
$\frac{\partial^2}{\partial \chi_p \partial \chi_y}$	0	$(b_i + 2c_i\alpha)\frac{\partial^2 \alpha}{\partial \chi_p \partial \chi_y} + 2c_i\left(\frac{\partial \alpha}{\partial \chi_p}\right)\left(\frac{\partial \alpha}{\partial \chi_y}\right)$

AERODYNAMIC FORCE AND MOMENT EQUATIONS: Aerodynamic Forces (A_x, A_y, and A_z)

	$A_x = -qsC_N$	$A_y = -qsC_A$	$A_z = qsC_S$
$\frac{\partial}{\partial \bar{x}}$	$-qs\frac{\partial C_N}{\partial \bar{x}} - sC_N\frac{\partial q}{\partial \bar{x}}$	$-qs\frac{\partial C_A}{\partial \bar{x}} - sC_A\frac{\partial q}{\partial \bar{x}}$	$qs\frac{\partial C_S}{\partial \bar{x}} + sC_S\frac{\partial q}{\partial \bar{x}}$
$\frac{\partial}{\partial \dot{\bar{x}}}$	$-qs\frac{\partial C_N}{\partial \dot{\bar{x}}} - sC_N\frac{\partial q}{\partial \dot{\bar{x}}}$	$-qs\frac{\partial C_A}{\partial \dot{\bar{x}}} - sC_A\frac{\partial q}{\partial \dot{\bar{x}}}$	$qs\frac{\partial C_S}{\partial \dot{\bar{x}}} + sC_S\frac{\partial q}{\partial \dot{\bar{x}}}$
$\frac{\partial}{\partial M}$	0	0	0
$\frac{\partial}{\partial \chi_p}$	$-qs\frac{\partial C_N}{\partial \chi_p}$	$-qs\frac{\partial C_A}{\partial \chi_p}$	$-qs\frac{\partial C_S}{\partial \chi_p}$
$\frac{\partial}{\partial \chi_y}$	$-qs\frac{\partial C_N}{\partial \chi_y}$	$-qs\frac{\partial C_A}{\partial \chi_y}$	$-qs\frac{\partial C_S}{\partial \chi_y}$
$\frac{\partial^2}{\partial \chi_p^2}$	$-qs\frac{\partial^2 C_N}{\partial \chi_p^2}$	$-qs\frac{\partial^2 C_A}{\partial \chi_y^2}$	$-qs\frac{\partial^2 C_S}{\partial \chi_p^2}$
$\frac{\partial^2}{\partial \chi_y^2}$	$-qs\frac{\partial^2 C_N}{\partial \chi_y^2}$	$-qs\frac{\partial^2 C_A}{\partial \chi_y^2}$	$-qs\frac{\partial^2 C_S}{\partial \chi_y^2}$
$\frac{\partial^2}{\partial \chi_p \partial \chi_y}$	$-qs\frac{\partial^2 C_N}{\partial \chi_p \partial \chi_y}$	$-qs\frac{\partial^2 C_A}{\partial \chi_p \partial \chi_y}$	$-qs\frac{\partial^2 C_S}{\partial \chi_p \partial \chi_y}$

AERODYNAMIC FORCE AND MOMENT EQUATIONS: Aerodynamic Moment Equations: Yaw (A_{M_x}) and Roll (A_{M_y})

	$A_{M_x} = qSt\left[C_{n_\beta}\beta - \frac{C_S \Delta Y'}{\ell}\right]$ $\Delta Y' = Y_{CG} - Y_{REF}$	$A_{M_y} = qSt\left[C_{\ell_\beta}\beta + C_S \frac{\Delta X'}{\ell}\right]$ $\Delta X' = X_{CG} - X_{REF}$
$\frac{\partial}{\partial x}$	$St\left\{\frac{\partial q}{\partial x}\left[C_{n_\beta}\beta - \frac{C_S \Delta Y'}{\ell}\right] + q\left[\beta\frac{\partial C_{n_\beta}}{\partial Mn}\frac{\partial Mn}{\partial x} - \frac{\partial C_S}{\partial x}\frac{\Delta Y'}{\ell} + C_{n_\beta}\frac{\partial \beta}{\partial x}\right]\right\}$	$St\left\{\frac{\partial q}{\partial x}\left[C_{\ell_\beta}\beta + C_S \frac{\Delta X'}{\ell}\right] + q\frac{\partial C_{\ell_\beta}}{\partial Mn}\frac{\partial Mn}{\partial x}\beta + C_{\ell_\beta}\frac{\partial \beta}{\partial x} + \frac{\partial C_S}{\partial x}\frac{\Delta X'}{\ell}\right]\right\}$
$\frac{\partial^2}{\partial x^2}$	$St\left\{\frac{\partial q}{\partial x}\left[C_{n_\beta}\beta - \frac{C_S \Delta Y'}{\ell}\right] + q\left[\beta\frac{\partial C_{n_\beta}}{\partial Mn}\frac{\partial Mn}{\partial x} + C_{n_\beta}\frac{\partial \beta}{\partial x} - \frac{\partial C_S}{\partial x}\frac{\Delta Y'}{\ell}\right]\right\}$	$St\left\{\frac{\partial q}{\partial x}\left[C_{\ell_\beta}\beta + C_S \frac{\Delta X'}{\ell}\right] + q\frac{\partial C_{\ell_\beta}}{\partial Mn}\frac{\partial Mn}{\partial x}\beta + C_{\ell_\beta}\frac{\partial \beta}{\partial x} + \frac{\partial C_S}{\partial x}\frac{\Delta X'}{\ell}\right]\right\}$
$\frac{\partial}{\partial m}$	0	0
$\frac{\partial}{\partial x_p}$	$qSt\left[C_{n_\beta}\frac{\partial \beta}{\partial x_p} - \frac{\partial C_S}{\partial x_p}\frac{\Delta Y'}{\ell}\right]$	$qSt\left[C_{\ell_\beta}\frac{\partial \beta}{\partial x_p} + \frac{\partial C_S}{\partial x_p}\frac{\Delta X'}{\ell}\right]$
$\frac{\partial}{\partial x_y}$	$qSt\left[C_{n_\beta}\frac{\partial \beta}{\partial x_y} - \frac{\partial C_S}{\partial x_y}\frac{\Delta Y'}{\ell}\right]$	$qSt\left[C_{\ell_\beta}\frac{\partial \beta}{\partial x_y} + \frac{\partial C_S}{\partial x_y}\frac{\Delta X'}{\ell}\right]$
$\frac{\partial^2}{\partial x_p^2}$	$qSt\left[C_{n_\beta}\frac{\partial^2 \beta}{\partial x_p^2} - \frac{\partial^2 C_S}{\partial x_p^2}\frac{\Delta Y'}{\ell}\right]$	$qSt\left[C_{\ell_\beta}\frac{\partial^2 \beta}{\partial x_p^2} + \frac{\partial^2 C_S}{\partial x_p^2}\frac{\Delta X'}{\ell}\right]$
$\frac{\partial^2}{\partial x_y^2}$	$qSt\left[C_{n_\beta}\frac{\partial^2 \beta}{\partial x_y^2} - \frac{\partial^2 C_S}{\partial x_y^2}\frac{\Delta Y'}{\ell}\right]$	$qSt\left[C_{\ell_\beta}\frac{\partial^2 \beta}{\partial x_y^2} + \frac{\partial^2 C_S}{\partial x_y^2}\frac{\Delta X'}{\ell}\right]$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$qSt\left[C_{n_\beta}\frac{\partial^2 \beta}{\partial x_p \partial x_y} - \frac{\partial^2 C_S}{\partial x_p \partial x_y}\frac{\Delta Y'}{\ell}\right]$	$qSt\left[C_{\ell_\beta}\frac{\partial^2 \beta}{\partial x_p \partial x_y} + \frac{\partial^2 C_S}{\partial x_p \partial x_y}\frac{\Delta X'}{\ell}\right]$

AERODYNAMIC FORCE AND MOMENT EQUATIONS: Aerodynamic Moment Equations
Pitch Moment Coefficient (C_m) and Pitch (A_{M_z})

$$C_m = C_{m_0} + C_{m_\alpha}\alpha + \frac{M_{m_B}}{q S \ell} + \Delta C_m$$

$$A_{M_z} = qS\ell\left(C_m + C_A\frac{\Delta X'}{\ell} - C_N\frac{\Delta Y'}{\ell}\right) - T_{Y_F}\Delta X + T_{X_F}\Delta Y$$

$$\Delta X = X_{CG} - X_{CP_F}$$

$$\Delta Y = Y_{CG} - Y_{CP_F}$$

$\frac{\partial}{\partial \bar{x}}$	$\left(\frac{\partial C_{m_0}}{\partial Mn} + \frac{\partial C_{m_\alpha}}{\partial Mn}\right)\frac{\partial Mn}{\partial \bar{x}} + C_{m_\alpha}\frac{\partial \alpha}{\partial \bar{x}} + \left(\frac{\partial M_{m_B}}{\partial h}\frac{\partial h}{\partial \bar{x}} - \frac{M_{m_B}}{q}\frac{\partial q}{\partial \bar{x}}\right)\frac{1}{qS\ell} + \frac{\partial \Delta C_m}{\partial \bar{x}}$	$S\ell\left(\frac{\partial q}{\partial \bar{x}}\left[C_m + C_A\frac{\Delta X'}{\ell} - C_N\frac{\Delta Y'}{\ell}\right] + q\left[\frac{\partial C_m}{\partial \bar{x}} + \frac{\partial C_A}{\partial \bar{x}}\frac{\Delta X'}{\ell} - \frac{\partial C_N}{\partial \bar{x}}\frac{\Delta Y'}{\ell}\right]\right) - \frac{\partial T_{Y_F}}{\partial \bar{x}}\Delta X + \frac{\partial T_{X_F}}{\partial \bar{x}}\Delta Y$
$\frac{\partial}{\partial \dot{\bar{x}}}$	$\left(\frac{\partial C_{m_0}}{\partial Mn} + \frac{\partial C_{m_\alpha}}{\partial Mn}\alpha\right)\frac{\partial Mn}{\partial \dot{\bar{x}}} + C_{m_\alpha}\frac{\partial \alpha}{\partial \dot{\bar{x}}} - \frac{M_{m_B}}{q^2 S\ell}\frac{\partial q}{\partial \dot{\bar{x}}} + \frac{\partial \Delta C_m}{\partial \dot{\bar{x}}}$	$S\ell\left(\frac{\partial q}{\partial \dot{\bar{x}}}\left[C_m + C_A\frac{\Delta X'}{\ell} - C_N\frac{\Delta Y'}{\ell}\right] + q\left[\frac{\partial C_m}{\partial \dot{\bar{x}}} + \frac{\partial C_A}{\partial \dot{\bar{x}}}\frac{\Delta X'}{\ell} - \frac{\partial C_N}{\partial \dot{\bar{x}}}\frac{\Delta Y'}{\ell}\right]\right)$
$\frac{\partial}{\partial m}$	0	$qS\left[C_A\frac{\partial X_{CG}}{\partial m} - C_N\frac{\partial Y_{CG}}{\partial m}\right] - T_{Y_F}\frac{\partial X_{CG}}{\partial m} + T_{X_F}\frac{\partial Y_{CG}}{\partial m}$
$\frac{\partial}{\partial x_p}$	$C_{m_\alpha}\frac{\partial \alpha}{\partial x_p} + \frac{\partial \Delta C_m}{\partial x_p}$	$qS\ell\left[\frac{\partial C_m}{\partial x_p} + \frac{\partial C_A}{\partial x_p}\frac{\Delta X'}{\ell} - \frac{\partial C_N}{\partial x_p}\frac{\Delta Y'}{\ell}\right]$
$\frac{\partial}{\partial x_y}$	$C_{m_\alpha}\frac{\partial \alpha}{\partial x_y} + \frac{\partial \Delta C_m}{\partial x_y}$	$qS\ell\left[\frac{\partial C_m}{\partial x_y} + \frac{\partial C_A}{\partial x_y}\frac{\Delta X'}{\ell} - \frac{\partial C_N}{x_Y}\frac{\Delta Y'}{\ell}\right]$
$\frac{\partial^2}{\partial x_p^2}$	$C_{m_\alpha}\frac{\partial^2 \alpha}{\partial x_p^2} + \frac{\partial^2 \Delta C_m}{\partial x_p^2}$	$qS\ell\left[\frac{\partial^2 C_m}{\partial x_p^2} + \frac{\partial^2 C_A}{\partial x_p^2}\frac{\Delta X'}{\ell} - \frac{\partial^2 C_N}{\partial x_p^2}\frac{\Delta Y'}{\ell}\right]$
$\frac{\partial^2}{\partial x_y^2}$	$C_{m_\alpha}\frac{\partial^2 \alpha}{\partial x_y^2} + \frac{\partial^2 \Delta C_m}{\partial x_y^2}$	$qS\ell\left[\frac{\partial^2 C_m}{\partial x_y^2} + \frac{\partial^2 C_A}{\partial x_y^2}\frac{X'}{\ell} - \frac{\partial^2 C_N}{\partial x_y^2}\frac{Y'}{\ell}\right]$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$C_{m_\alpha}\frac{\partial^2 \alpha}{\partial x_p \partial x_y} + \frac{\partial^2 \Delta C_m}{\partial x_p \partial x_y}$	$qS\ell\left[\frac{\partial^2 C_m}{\partial x_p \partial x_y} + \frac{\partial^2 C_A}{\partial x_p \partial x_y}\frac{\Delta X'}{\ell} - \frac{\partial^2 C_N}{\partial x_p \partial x_y}\frac{\Delta Y'}{\ell}\right]$

THRUST FORCE EQUATION: Thrust Magnitude (T) and Fixed Thrust Components in Body System

	$T = T_{VAC_{ORB}} - A_{E_{ORB}} P$	$T = T_{VAC_i} - A_{E_i} P$	$T_{X_F} = T_{Y_F} \tan \delta_{P_F}$	$T_{Y_F} = \frac{T}{(1 + \tan^2 \delta_{P_F} + \tan^2 \delta_{Y_F})^{1/2}}$
$\frac{\partial}{\partial \bar{x}}$	$-A_{E_{ORB}} \frac{\partial P^*}{\partial h} \frac{\partial h}{\partial \bar{x}}$	$A_{E_i} \frac{\partial P^*}{\partial h} \frac{\partial h}{\partial \bar{x}}$	$\frac{\partial T_{Y_F}}{\partial \bar{x}} \tan \delta_{P_F}$	$\frac{1}{(1 + \tan^2 \delta_{P_F} + \tan^2 \delta_{Y_F})^{1/2}} \frac{\partial T}{\partial \bar{x}}$
$\frac{\partial}{\partial \bar{x}}$	0	0	0	0
$\frac{\partial}{\partial m}$	0, when not throttling ag_o, when throttling to acceleration a	0	0	0
$\frac{\partial}{\partial x_p}$	0	0	0	0
$\frac{\partial}{\partial x_y}$	0	0	0	0
$\frac{\partial^2}{\partial x_p^2}$	0	0	0	0
$\frac{\partial^2}{\partial x_y^2}$	0	0	0	0
$\frac{\partial^2}{\partial x_p \partial x_y}$	0	0	0	0

*Determined by spline interpolation method in atmosphere routine

i Corresponds to LBM, OMS, or RCS propulsion systems

THRUST FORCE EQUATIONS: Sine and Cosine of Transformation Angles to Convert From Moment Balance Related Coordinated System to Body System ($\cos\rho$ and $\sin\rho$)

$$\cos\rho = \frac{1}{\sqrt{1. + \Delta x / \Delta y}}$$

$$\Delta x = X_{CG} - X_{GP}$$

$$\Delta y = Y_{CG} - Y_{GP}$$

$$\sin\rho = \frac{\Delta x/\Delta y}{\sqrt{1 + \left(\frac{\Delta x}{\Delta y}\right)^2}}$$

$\frac{\partial}{\partial \bar{x}}$	0	0
$\frac{\partial}{\partial \dot{\bar{x}}}$	0	0
$\frac{\partial}{\partial m}$	$-\sin\rho \frac{1}{\Delta y\left[1 + \left(\frac{\Delta x}{\Delta y}\right)^2\right]}\left(-\frac{\partial X_{CG}^{*}}{\partial m} + \frac{\Delta x}{\Delta y}\frac{\partial Y_{CG}^{*}}{\partial m}\right)$	$\frac{\Delta x}{\Delta y}\frac{\partial \cos\rho}{\partial m} + \cos\rho \frac{1}{\Delta y}\left(-\frac{\partial X_{CG}^{*}}{\partial m} + \frac{\Delta x}{\Delta y}\frac{\partial Y_{CG}^{*}}{\partial m}\right)$
$\frac{\partial}{\partial x_p}$	0	0
$\frac{\partial}{\partial x_y}$	0	0
$\frac{\partial^2}{\partial x_p^2}$	0	0
$\frac{\partial^2}{\partial x_y^2}$	0	0
$\frac{\partial^2}{\partial x_p \partial x_y}$	0	0

*Determined by spline interpolation method

THRUST FORCE EQUATIONS: Total Thurst Forces in Body Coordinates

	$T_x = T_x' \cos\phi + T_y' \sin\phi + T_{x_f}$	$T_y = -T_x' \sin\phi + T_y' \cos\phi + T_{y_f}$	$T_z = T_y' \tan\delta_y$
$\frac{\partial}{\partial \bar{x}}$	$\frac{\partial T_x'}{\partial \bar{x}} \cos\phi + \frac{\partial T_y'}{\partial \bar{x}} \sin\phi + \frac{\partial T_{x_f}}{\partial \bar{x}}$	$-\frac{\partial T_x'}{\partial \bar{x}} \sin\phi + \frac{\partial T_y'}{\partial \bar{x}} \cos\phi + \frac{\partial T_{y_f}}{\partial \bar{x}}$	$T_y' \frac{\partial(\tan\delta_y)}{\partial \bar{x}} + \tan\delta_y \frac{\partial T_y'}{\partial \bar{x}}$
$\frac{\partial}{\partial \dot{\bar{x}}}$	$\frac{\partial T_x'}{\partial \dot{\bar{x}}} \cos\phi + \frac{\partial T_y'}{\partial \dot{\bar{x}}} \sin\phi$	$-\frac{\partial T_x'}{\partial \dot{\bar{x}}} \sin\phi + \frac{\partial T_y'}{\partial \dot{\bar{x}}} \cos\phi$	$T_y' \frac{\partial(\tan\delta_y)}{\partial \dot{\bar{x}}} + \tan\delta_y \frac{\partial T_y'}{\partial \dot{\bar{x}}}$
$\frac{\partial}{\partial m}$	$T_x' \frac{\partial(\cos\phi)}{\partial m} + T_y' \frac{\partial(\sin\phi)}{\partial m} + \cos\phi \frac{\partial T_x'}{\partial m} + \sin\phi \frac{\partial T_y'}{\partial m}$	$-T_x' \frac{\partial(\sin\phi)}{\partial m} + T_y' \frac{\partial(\cos\phi)}{\partial m} - \sin\phi \frac{\partial T_x'}{\partial m} + \cos\phi \frac{\partial T_y'}{\partial m}$	$T_y' \frac{\partial(\tan\delta_y)}{\partial m} + \tan\delta_y \frac{\partial T_y'}{\partial m}$
$\frac{\partial}{\partial x_p}$	$\frac{\partial T_x'}{\partial x_p} \cos\phi + \frac{\partial T_y'}{\partial x_p} \sin\phi$	$-\frac{\partial T_x'}{\partial x_p} \cos\phi + \frac{\partial T_y'}{\partial x_p} \sin\phi$	$T_y' \frac{\partial(\tan\delta_y)}{\partial x_p} + \tan\delta_y \frac{\partial T_y'}{\partial x_p}$
$\frac{\partial}{\partial x_y}$	$\frac{\partial T_x'}{\partial x_y} \cos\phi + \frac{\partial T_y'}{\partial x_y} \sin\phi$	$-\frac{\partial T_x'}{\partial x_y} \cos\phi + \frac{\partial T_y'}{\partial x_y} \sin\phi$	$T_y' \frac{\partial(\tan\delta_y)}{\partial x_y} + \tan\delta_y \frac{\partial T_y'}{\partial x_y}$
$\frac{\partial^2}{\partial x_p^2}$	$\frac{\partial^2 T_x'}{\partial x_p^2} \cos\phi + \frac{\partial^2 T_y'}{\partial x_p^2} \sin\phi$	$-\frac{\partial^2 T_x'}{\partial x_p^2} \cos\phi + \frac{\partial^2 T_y'}{\partial x_p^2} \sin\phi$	$T_y' \frac{\partial^2(\tan\delta_y)}{\partial x_p^2} + \tan\delta_y \frac{\partial^2 T_y'}{\partial x_p^2}$
$\frac{\partial^2}{\partial x_y^2}$	$\frac{\partial^2 T_x'}{\partial x_y^2} \cos\phi + \frac{\partial^2 T_y'}{\partial x_y^2} \sin\phi$	$-\frac{\partial^2 T_x'}{\partial x_y^2} \cos\phi + \frac{\partial^2 T_y'}{\partial x_y^2} \sin\phi$	$T_y' \frac{\partial^2(\tan\delta_y)}{\partial x_y^2} + \tan\delta_y \frac{\partial^2 T_y'}{\partial x_y^2}$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\frac{\partial^2 T_x'}{\partial x_p \partial x_y} \cos\phi + \frac{\partial^2 T_y'}{\partial x_p \partial x_y} \sin\phi$	$-\frac{\partial^2 T_x'}{\partial x_p \partial x_y} \cos\phi + \frac{\partial^2 T_y'}{\partial x_p \partial x_y} \sin\phi$	$T_y' \frac{\partial^2(\tan\delta_y)}{\partial x_p \partial x_y} + \tan\delta_y \frac{\partial^2 T_y'}{\partial x_p \partial x_y}$

THRUST FORCE EQUATIONS: Total Thrust Forces in Moment Balance
Related Coordinate System

	$T'_x = -\tan\delta_{p_1} T_{y_1} - \tan\delta_{p_2} T_{y_2}$	$T'_y = T_{y_1} + T_{y_2}$
$\frac{\partial}{\partial \bar{x}}$	$\left[\frac{\partial(\tan\delta_{p_1})}{\partial \bar{x}} T_{y_1} + \frac{\partial(\tan\delta_{p_2})}{\partial \bar{x}} T_{y_2} + \tan\delta_{p_1} \frac{\partial T_{y_1}}{\partial \bar{x}} + \tan\delta_{p_2} \frac{\partial T_{y_2}}{\partial \bar{x}}\right]$	$\frac{\partial T_{y_1}}{\partial \bar{x}} + \frac{\partial T_{y_2}}{\partial \bar{x}}$
$\frac{\partial}{\partial \dot{\bar{x}}}$	$\left[\frac{\partial(\tan\delta_{p_1})}{\partial \dot{\bar{x}}} T_{y_1} + \frac{\partial(\tan\delta_{p_2})}{\partial \dot{\bar{x}}} T_{y_2} + \tan\delta_{p_1} \frac{\partial T_{y_1}}{\partial \dot{\bar{x}}} + \tan\delta_{p_2} \frac{\partial T_{y_2}}{\partial \dot{\bar{x}}}\right]$	$\frac{\partial T_{y_1}}{\partial \dot{\bar{x}}} + \frac{\partial T_{y_2}}{\partial \dot{\bar{x}}}$
$\frac{\partial}{\partial m}$	$\left[\frac{\partial(\tan\delta_{p_1})}{\partial m} T_{y_1} + \frac{\partial(\tan\delta_{p_2})}{\partial m} T_{y_2} + \tan\delta_{p_1} \frac{\partial T_{y_1}}{\partial m} + \tan\delta_{p_2} \frac{\partial T_{y_2}}{\partial m}\right]$	$\frac{\partial T_{y_1}}{\partial m} + \frac{\partial T_{y_2}}{\partial m}$
$\frac{\partial}{\partial x_p}$	$\left[\frac{\partial(\tan\delta_{p_1})}{\partial x_p} T_{y_1} + \frac{\partial(\tan\delta_{p_2})}{\partial x_p} T_{y_2} + \tan\delta_{p_1} \frac{\partial T_{y_1}}{\partial x_p} + \tan\delta_{p_2} \frac{\partial T_{y_2}}{\partial x_p}\right]$	$\frac{\partial T_{y_1}}{\partial x_p} + \frac{\partial T_{y_2}}{\partial x_p}$
$\frac{\partial}{\partial x_y}$	$\left[\frac{\partial(\tan\delta_{p_1})}{\partial x_y} T_{y_1} + \frac{\partial(\tan\delta_{p_2})}{\partial x_y} T_{y_2} + \tan\delta_{p_1} \frac{\partial T_{y_1}}{\partial x_y} + \tan\delta_{p_2} \frac{\partial T_{y_2}}{\partial x_y}\right]$	$\frac{\partial T_{y_1}}{\partial x_y} + \frac{\partial T_{y_2}}{\partial x_y}$
$\frac{\partial^2}{\partial x_p^2}$	$\left\{\frac{\partial^2(\tan\delta_{p_1})}{\partial x_p^2} T_{y_1} + \frac{\partial^2(\tan\delta_{p_2})}{\partial x_p^2} T_{y_2} + 2\left[\frac{\partial(\tan\delta_{p_1})}{\partial x_p} \frac{\partial T_{y_1}}{\partial x_p} + \frac{\partial(\tan\delta_{p_2})}{\partial x_p} \frac{\partial T_{y_2}}{\partial x_p}\right] + \tan\delta_{p_1} \frac{\partial^2 T_{y_1}}{\partial x_p^2} + \tan\delta_{p_2} \frac{\partial^2 T_{y_2}}{\partial x_p^2}\right\}$	$\frac{\partial^2 T_{y_1}}{\partial x_p^2} + \frac{\partial^2 T_{y_2}}{\partial x_p^2}$
$\frac{\partial^2}{\partial x_y^2}$	$\left\{\frac{\partial^2(\tan\delta_{p_1})}{\partial x_y^2} T_{y_1} + \frac{\partial^2(\tan\delta_{p_2})}{\partial x_y^2} T_{y_2} + 2\left[\frac{\partial(\tan\delta_{p_1})}{\partial x_y} \frac{\partial T_{y_1}}{\partial x_y} + \frac{\partial(\tan\delta_{p_2})}{\partial x_y} \frac{\partial T_{y_2}}{\partial x_y}\right] + \tan\delta_{p_1} \frac{\partial^2 T_{y_1}}{\partial x_y^2} + \tan\delta_{p_2} \frac{\partial^2 T_{y_2}}{\partial x_y^2}\right\}$	$\frac{\partial^2 T_{y_1}}{\partial x_y^2} + \frac{\partial^2 T_{y_2}}{\partial x_y^2}$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\left\{\frac{\partial^2(\tan\delta_{p_1})}{\partial x_p \partial x_y} T_{y_1} + \frac{\partial^2(\tan\delta_{p_2})}{\partial x_p \partial x_y} T_{y_2} + \frac{\partial(\tan\delta_{p_1})}{\partial x_p} \frac{\partial T_{y_1}}{\partial x_y} + \frac{\partial(\tan\delta_{p_2})}{\partial x_p} \frac{\partial T_{y_2}}{\partial x_y} + \frac{\partial(\tan\delta_{p_1})}{\partial x_y} \frac{\partial T_{y_1}}{\partial x_p} + \frac{\partial(\tan\delta_{p_2})}{\partial x_y} \frac{\partial T_{y_2}}{\partial x_p} + \tan\delta_{p_1} \frac{\partial^2 T_{y_1}}{\partial x_p \partial x_y} + \tan\delta_{p_2} \frac{\partial^2 T_{y_2}}{\partial x_p \partial x_y}\right\}$	$\frac{\partial^2 T_{y_1}}{\partial x_p \partial x_y} + \frac{\partial^2 T_{y_2}}{\partial x_p \partial x_y}$

THRUST FORCE EQUATIONS: Breakdown of Thrust Forces into Equivalent Engines in Moment Balance Realted Y Coordinate

$$T_{y_i} = \frac{T/\ell}{\sqrt{1 + \tan^2\delta_{p_i} + \tan^2\delta_y}} \qquad (i = 1,2)$$

$$\frac{\partial}{\partial \bar{x}}: \quad T_{y_i}\left\{\frac{1}{T}\frac{\partial T}{\partial \bar{x}} - \frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial \bar{x}} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial \bar{x}}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

$$\frac{\partial}{\partial \bar{y}}: \quad -T_{y_i}\left\{\frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial \bar{y}} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial \bar{y}}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

$$\frac{\partial}{\partial m}: \quad T_{y_i}\left\{\frac{1}{T}\frac{\partial T}{\partial m} - \frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial m} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial m}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

$$\frac{\partial}{\partial x_p}: \quad -T_{y_i}\left\{\frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial x_p} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial x_p}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

$$\frac{\partial}{\partial x_y}: \quad T_{y_i}\left\{\frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial x_y} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial x_y}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

$$\frac{\partial^2}{\partial x_p^2}: \quad T_{y_i}\left\{\frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial x_p} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial x_p}\right]^2}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)^2} - \frac{\left[\left(\frac{\partial(\tan\delta_{p_i})}{\partial x_p}\right)^2 + \tan\delta_{p_i}\frac{\partial^2\tan\delta_{p_i}}{\partial x_p^2} + \left(\frac{\partial(\tan\delta_y)}{\partial x_p}\right)^2 + \tan\delta_y\frac{\partial^2(\tan\delta_y)}{\partial x_p^2}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

$$\frac{\partial^2}{\partial x_y^2}: \quad T_{y_i}\left\{\frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial x_y} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial x_y}\right]^2}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)^2} - \frac{\left[\left(\frac{\partial(\tan\delta_{p_i})}{\partial x_y}\right)^2 + \tan\delta_{p_i}\frac{\partial^2\tan\delta_{p_i}}{\partial x_y} + \left(\frac{\partial(\tan\delta_y)}{\partial x_y}\right)^2 + \tan\delta_y\frac{\partial^2(\tan\delta_y)}{\partial x_y^2}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

$$\frac{\partial^2}{\partial x_p \partial x_y}: \quad T_{y_i}\left\{\frac{\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial x_p} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial x_p}\right]\left[\tan\delta_{p_i}\frac{\partial(\tan\delta_{p_i})}{\partial x_y} + \tan\delta_y\frac{\partial(\tan\delta_y)}{\partial x_y}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)^2} - \frac{\left[\frac{\partial(\tan\delta_{p_i})}{\partial x_p}\frac{\partial(\tan\delta_{p_i})}{\partial x_y} + \tan\delta_{p_i}\frac{\partial^2(\tan\delta_{p_i})}{\partial x_p \partial x_y} + \frac{\partial(\tan\delta_y)}{\partial x_p}\frac{\partial(\tan\delta_y)}{\partial x_y} + \tan\delta_y\frac{\partial^2(\tan\delta_y)}{\partial x_p \partial x_y}\right]}{(1 + \tan^2\delta_{p_i} + \tan^2\delta_y)}\right\}$$

THRUST FORCE EQUATIONS: Engine Deflection Angles: Pitch (δ_{P1})

$$\tan\delta_{P_1} = \frac{1}{T}\left(C_{xx}A_{H_x} + C_{xy}A_{H_y} + C_{xz}A_{H_z}\right)$$

$\frac{\partial}{\partial \bar{x}}$	$-\frac{1}{T}\left[\tan\delta_{P_1}\frac{\partial T}{\partial \bar{x}} + C_{xx}\frac{\partial A_{H_x}}{\partial \bar{x}} + C_{xy}\frac{\partial A_{H_y}}{\partial \bar{x}} + C_{xz}\frac{\partial A_{H_z}}{\partial \bar{x}}\right]$
$\frac{\partial}{\partial \dot{x}}$	$-\frac{1}{T}\left[C_{xx}\frac{\partial A_{H_x}}{\partial \dot{x}} + C_{xy}\frac{\partial A_{H_y}}{\partial \dot{x}} + C_{xz}\frac{\partial A_{H_z}}{\partial \dot{x}}\right]$
$\frac{\partial}{\partial m}$	$\frac{1}{T}\left\{C_{xx}\left[\frac{A_{H_x}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_x}}{\partial m}\right] + C_{xy}\left[\frac{A_{H_y}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_y}}{\partial m}\right] + C_{xz}\left[\frac{A_{H_z}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_z}}{\partial m}\right] - \left[\frac{\partial C_{xx}}{\partial m}A_{H_x} + \frac{\partial C_{xy}}{\partial m}A_{H_y} + \frac{\partial C_{xz}}{\partial m}A_{H_z}\right]\right\}$
$\frac{\partial}{\partial x_p}$	$-\frac{1}{T}\left[C_{xx}\frac{\partial A_{H_x}}{\partial x_p} + C_{xy}\frac{\partial A_{H_y}}{\partial x_p} + C_{xz}\frac{\partial A_{H_z}}{\partial x_p}\right]$
$\frac{\partial}{\partial x_y}$	$-\frac{1}{T}\left[C_{xx}\frac{\partial A_{H_z}}{\partial x_y} + C_{xy}\frac{\partial A_{H_y}}{\partial x_y} + C_{xz}\frac{\partial A_{H_z}}{\partial x_y}\right]$
$\frac{\partial^2}{\partial x_p^2}$	$\frac{1}{T}\left[C_{xx}\frac{\partial^2 A_{H_x}}{\partial x_p^2} + C_{xy}\frac{\partial^2 A_{H_y}}{\partial x_p^2} + C_{xz}\frac{\partial^2 A_{H_z}}{\partial x_p^2}\right]$
$\frac{\partial^2}{\partial x_y^2}$	$\frac{1}{T}\left[C_{xx}\frac{\partial^2 A_{H_x}}{\partial x_y^2} + C_{xy}\frac{\partial^2 A_{H_y}}{\partial x_y^2} + C_{xz}\frac{\partial^2 A_{H_z}}{\partial x_y^2}\right]$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\frac{1}{T}\left[C_{xx}\frac{\partial^2 A_{H_x}}{\partial x_p \partial x_y} + C_{xy}\frac{\partial^2 A_{H_y}}{\partial x_p \partial x_y} + C_{xz}\frac{\partial^2 A_{H_z}}{\partial x_p \partial x_y}\right]$

THRUST FORCE EQUATIONS: ENGINE DEFLECTION ANGLES: Pitch (δ_{P2})

$$\sin\delta_{P_2} = -\frac{1}{T}\left(C_{yx}A_{H_x} + C_{yy}A_{H_y} + C_{yz}A_{H_z}\right)$$

$\frac{\partial}{\partial \dot{x}}$	$-\frac{1}{T}\left[\sin\delta_{P_2}\frac{\partial T}{\partial \dot{x}} + C_{yx}\frac{\partial A_{H_x}}{\partial \dot{x}} + C_{yy}\frac{\partial A_{H_y}}{\partial \dot{x}} + C_{yz}\frac{\partial A_{H_z}}{\partial \dot{x}}\right]$
$\frac{\partial}{\partial \ddot{x}}$	$-\frac{1}{T}\left[C_{yx}\frac{\partial A_{H_x}}{\partial \ddot{x}} + C_{yy}\frac{\partial A_{H_y}}{\partial \ddot{x}} + C_{yz}\frac{\partial A_{H_z}}{\partial \ddot{x}}\right]$
$\frac{\partial}{\partial m}$	$\frac{1}{T}\left\{C_{yx}\left[\frac{A_{H_x}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_x}}{\partial m}\right] + C_{yy}\left[\frac{A_{H_y}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_y}}{\partial m}\right] + C_{yz}\left[\frac{A_{H_z}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_z}}{\partial m}\right] - \left[\frac{\partial C_{yx}}{\partial m}A_{H_x} + \frac{\partial C_{yy}}{\partial m}A_{H_y} + \frac{\partial C_{yz}}{\partial m}A_{H_z}\right]\right\}$
$\frac{\partial}{\partial x_p}$	$-\frac{1}{T}\left[C_{yx}\frac{\partial A_{H_x}}{\partial x_p} + C_{yy}\frac{\partial A_{H_y}}{\partial x_p} + C_{yz}\frac{\partial A_{H_z}}{\partial x_p}\right]$
$\frac{\partial}{\partial x_y}$	$-\frac{1}{T}\left[C_{yx}\frac{\partial A_{H_x}}{\partial x_y} + C_{yy}\frac{\partial A_{H_y}}{\partial x_y} + C_{yz}\frac{\partial A_{H_z}}{\partial x_y}\right]$
$\frac{\partial^2}{\partial x_p^2}$	$\frac{1}{T}\left[C_{yx}\frac{\partial^2 A_{H_x}}{\partial x_p^2} + C_{yy}\frac{\partial^2 A_{H_y}}{\partial x_p^2} + C_{yz}\frac{\partial^2 A_{H_z}}{\partial x_p^2}\right]$
$\frac{\partial^2}{\partial x_y^2}$	$\frac{1}{T}\left[C_{yx}\frac{\partial^2 A_{H_x}}{\partial x_y^2} + C_{yy}\frac{\partial^2 A_{H_y}}{\partial x_y^2} + C_{yz}\frac{\partial^2 A_{H_z}}{\partial x_y^2}\right]$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\frac{1}{T}\left[C_{yx}\frac{\partial^2 A_{H_x}}{\partial x_p \partial x_y} + C_{yy}\frac{\partial^2 A_{H_y}}{\partial x_p \partial x_y} + C_{yz}\frac{\partial^2 A_{H_z}}{\partial x_p \partial x_y}\right]$

THRUST FORCE EQUATIONS: Engine Deflection Angles: Yaw (δ_Y)

$$\tan\delta_y = -\frac{1}{T}\left[C_{zx}A_{H_x} + C_{zy}A_{H_y}\right]$$

$\frac{\partial}{\partial \dot{x}}$	$-\frac{1}{T}\left[\tan\delta_y \frac{\partial T}{\partial \dot{x}} + C_{zx}\frac{\partial A_{H_x}}{\partial \dot{x}} + C_{zy}\frac{\partial A_{H_y}}{\partial \dot{x}}\right]$
$\frac{\partial}{\partial \ddot{x}}$	$-\frac{1}{T}\left[C_{zx}\frac{\partial A_{H_x}}{\partial \ddot{x}} + C_{zy}\frac{\partial A_{H_y}}{\partial \ddot{x}}\right]$
$\frac{\partial}{\partial m}$	$\frac{1}{T}\left\{C_{zx}\left[\frac{A_{H_x}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_x}}{\partial m}\right] + C_{zy}\left[\frac{A_{H_y}}{T}\frac{\partial T}{\partial m} - \frac{\partial A_{H_y}}{\partial m}\right] - \left[\frac{\partial C_{zx}}{\partial m}A_{H_x} + \frac{\partial C_{zy}}{\partial m}A_{H_y}\right]\right\}$
$\frac{\partial}{\partial x_p}$	$-\frac{1}{T}\left[C_{zx}\frac{\partial A_{H_x}}{\partial x_p} + C_{zy}\frac{\partial A_{H_y}}{\partial x_p}\right]$
$\frac{\partial}{\partial x_y}$	$-\frac{1}{T}\left[C_{zx}\frac{\partial A_{H_x}}{\partial x_y} + C_{zy}\frac{\partial A_{H_y}}{\partial x_y}\right]$
$\frac{\partial^2}{\partial x_p^2}$	$-\frac{1}{T}\left[C_{zx}\frac{\partial^2 A_{H_x}}{\partial x_p^2} + C_{zy}\frac{\partial^2 A_{H_y}}{\partial x_p^2}\right]$
$\frac{\partial^2}{\partial x_y^2}$	$-\frac{1}{T}\left[C_{zx}\frac{\partial^2 A_{H_x}}{\partial x_y^2} + C_{zy}\frac{\partial^2 A_{H_y}}{\partial x_y^2}\right]$
$\frac{\partial^2}{\partial x_p \partial x_y}$	$\frac{1}{T}\left[C_{zx}\frac{\partial^2 A_{H_x}}{\partial x_p \partial x_y} + C_{zy}\frac{\partial^2 A_{H_y}}{\partial x_p \partial x_y}\right]$

THRUST FORCE EQUATIONS: Moment Related Parameters

	$C_{xx} = \frac{\sin\phi}{\Delta z}$	$C_{xy} = \frac{\cos\phi}{\Delta z}$	$C_{xz} = \frac{\cos\phi}{\Delta y} = C_{yz}$	$C_{yx} = -C_{xx}$	$C_{yy} = -C_{xy}$	$C_{zx} = \frac{1}{\Delta y\left[1 + \left(\frac{\Delta x}{\Delta y}\right)^2\right]}$	$C_{zy} = -\frac{\Delta x}{\Delta y} C_{zx}$
$\frac{\partial}{\partial \dot{x}}$	0	0	0	0	0	0	0
$\frac{\partial}{\partial \dot{x}}$	0	0	0	0	0	0	0
$\frac{\partial}{\partial m}$	$\frac{1}{\Delta z}\frac{\partial(\sin\phi)}{\partial m}$	$\frac{1}{\Delta z}\frac{\partial(\cos\phi)}{\partial m}$	$\frac{1}{\Delta y}\left(-\frac{\cos\phi}{\Delta y}\frac{\partial Y_{\hat{CG}}}{\partial m} + \frac{\partial(\cos\phi)}{\partial m}\right)$	$-\frac{\partial C_{xx}}{\partial m}$	$-\frac{\partial C_{xy}}{\partial m}$	$-C_{zx}\left[\frac{1}{\Delta y}\frac{\partial Y_{\hat{CG}}}{\partial m} + 2C_{zx}\left(-\frac{\partial X_{\hat{CG}}}{\partial m} + \frac{\Delta x}{\Delta y}\frac{\partial Y_{\hat{CG}}}{\partial m}\right)\right]$	$\left(\frac{\partial X_{\hat{CG}}}{\partial m} - \frac{\Delta x}{\Delta y}\frac{\partial Y_{\hat{CG}}}{\partial m}\right)\frac{C_{zx}}{\Delta y} - \frac{\Delta x}{\Delta y}\frac{\partial C_{zx}}{\partial m}$
$\frac{\partial}{\partial x_p}$	0	0	0	0	0	0	0
$\frac{\partial}{\partial x_y}$	0	0	0	0	0	0	0
$\frac{\partial^2}{\partial x_p^2}$	0	0	0	0	0	0	0
$\frac{\partial^2}{\partial x_y^2}$	0	0	0	0	0	0	0
$\frac{\partial^2}{\partial x_p \partial x_y}$	0	0	0	0	0	0	0

*Determined by spline interpolation method

DIRECTION COSINES

	$X_1 = \cos x_r \cos x_p$	$X_2 = -\cos x_r \sin x_p$	$Y_1 = \cos x_y \sin x_p$	$Y_2 = \cos x_y \cos x_p$	$Y_3 = \sin x_y$	$Z_1 = -\cos x_r \sin x_y \sin x_p$	$Z_2 = -\cos x_r \sin x_y \cos x_p$	$Z_2 = \cos x_r \cos x_y$
$\frac{\partial}{\partial x}$	0	0	0	0	0	0	0	0
$\frac{\partial}{\partial x}$	0	0	0	0	0	0	0	0
$\frac{\partial}{\partial x}$	0	0	0	0	0	0	0	0
$\frac{\partial}{\partial x_p}$	$-\cos x_r \sin x_p$ (X_2)	$-\cos x_r \cos x_p$ $(-X_1)$	$\cos x_y \cos x_p$ (Y_2)	$-\cos x_y \sin x_p$ $(-Y_1)$	0	$-\cos x_r \sin x_y \cos x_p$ (Z_2)	$\cos x_r \sin x_y \sin x_p$ $(-Z_1)$	0
$\frac{\partial}{\partial x_y}$	0	0	$-\sin x_y \sin x_p$	$-\sin x_y \cos x_p$	$\cos x_y$	$-\cos x_r \cos x_y \sin x_p$	$-\cos x_r \cos x_y \cos x_p$	$-\cos x_r \sin x_y$
$\frac{\partial^2}{\partial x_p^2}$	$-\cos x_r \cos x_p$ $(-X_1)$	$\cos x_r \sin x_p$ $(-X_2)$	$-\cos x_y \sin x_p$ $(-Y_1)$	$-\cos x_y \cos x_p$ $(-Y_2)$	0	$\cos x_r \sin x_y \sin x_p$ $(-Z_1)$	$\cos x_r \sin x_y \cos x_p$ $(-Z_2)$	0
$\frac{\partial^2}{\partial x_y^2}$	0	0	$-\cos x_y \sin x_p$ $(-Y_1)$	$-\cos x_y \cos x_p$ $(-Y_2)$	$-\sin x_y$ $(-Y_3)$	$\cos x_r \sin x_y \sin x_p$ $(-Z_1)$	$\cos x_r \sin x_y \cos x_p$ $(-Z_2)$	$-\cos x_r \cos x_y$ $(-Z_3)$
$\frac{\partial^2}{\partial x_p \partial x_y}$	0	0	$-\sin x_y \cos x_p$	$\sin x_y \sin x_p$	0	$-\cos x_r \cos x_y \cos x_p$	$\cos x_r \cos x_y \sin x_p$	0

Appendix G

PROGRAM LISTING

```
C     MAIN PROGRAM
      CHARACTER*60 HEAD
      CHARACTER*6 MISION,HEADER,ATMOSN,NAME,SRID
      CHARACTER*12 OFFICE,DATE
      CHARACTER*48 TITLE
      LOGICAL DEMAND,LEMP
      COMMON/CCC/GRAPH,JC,DEMAND
      COMMON/AERO/NTALPH(2),NTBETA(2),NTMACH,TALPH(11,2),TALTH(11,2),
     *TXMAC(28),CAB(15,11,6),CNA(15,11,6),CMA(15,11,6),CYA(15,11,6),
     *CXA(15,11,6),CLA(15,11,6)
      COMMON/AERODI/AEROC(30,16),AEROB(50,5,3),TCMACH(20),DELCA(20),
     *FNPGNE(15),BMFAET(50,2),BMFTHR(50,2),DELCN(20),DELCM(20)
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *DTZ,TG,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *PEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FA2,BG(3,2),CBAXIL,TCHIR,VRCUT,FFREAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIP,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVARS(12),CELXOW(15,3),DELXOR(7,4),CELXC(15,7),WZERO,ALT,
     *XMACH,GDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *BYL,BEL,BMMA(15),BMMN,BSTEP(15),HNMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITHR,IWO,IXR,JUMP,KAT,KCYTAB,KFAGE,KS1,KWIA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINOB,KRDERB,MISION,
     *IRTFLE,NROLL,NCCAS1,IFACT,NOBASE,LSTGE(15),MSWCH(15)
      COMMON/AGEN4/QALPHA,QBETA,QCN,STH1,CTH1,XJX(3),XJV(3)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TJX,TYY,TZZ,APX,APY,AMZ,
     *IM2,THFA,SICE,KPAG,AZW,FARC,FLATC,FOM,VISC
      COMMON/AFPGN/VI,CAP,C1,C3,XINC,XNOD,XMLB,THETG,THETA,PHI,RNGO,RNGA
     *A,AZI,AZRE
      COMMON/ALAT/ALAT,ALONGO,ALTLS,NTABLE
      COMMON/ARDC/FAT(14),ATMOSN
      COMMON/ASTGR/ASTGR(4,67)
      COMMON/AUTO/KOB(40),SAVCP(40),XLAMB(40,10),CP2
      COMMON/AVGGP/XGPA,YGPA,ZGPA
      COMMON/BAKE/BISS(10,10),YLF(7,10)
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NOP1,NF(4)
      COMMON/BLAMB/YLET(15,10),YLBW(15,10),GAPHI(10),FROC(2,10)
      COMMON/BODY/TBODY(15,2),TXCG(15,2),TYCG(15,2),XLEN(2),XREF(2),
     *YREF(2),XP(2),XY(2),AP(10,2),AY(10,2)
      COMMON/BOPCOP/NCASE,OFFICE,TITLE,DATE,SRID(60),NOTAB,NERR
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,GM,CMUE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,FTN,FTKG,FSFTM,PFN,PFKG,PSFFP
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/CONVK/CONVK(15,3)
      COMMON/COVL/COVL(10)
      COMMON/DDELX/DDELX(10)
      COMMON/DELFAE/DELALT(25),DELFAB(25),M3C,M3I,ITCL(15)
      COMMON/DELPAR/SFRACB,SYRACB
      COMMON/DELVP/DELVP,DELVG
      COMMON/DELX/DELX(7)
      COMMON/DELXH/DELXH(4)
      COMMON/DRHC/DRHC(9),E1,WIPPB,WGNU
      COMMON/DRNGET/DRNGET(2)
      COMMON/ELEVON/EPACH(25,2),ELEVON(25,4),M10,M11
      COMMON/ENPUT/PSIFST(6,4),HEADER(20),NVRST(4),KCDRES(6,4),LAST
      COMMON/FMXX/FMXX,MSW,WG,SPOA
      COMMON/FORINT/HBANN(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,HPX,HMN,HAP
      COMMON/GNU/GNU(10)
      COMMON/HEAT/MAXHAT
      COMMON/IMP/GAPTB(90,10),GSPTB(90,10),ANTB(50),SITB(90),TS(3)
     *,DELTS(3),KJTB
```

```
      COMMON/IPR/IFR,LFMR,ILAST,KDT(15),FRM4,PRTCT
      COMMON/IRTAOA/IRTACA
      COMMON/ISTART/JSTART,ISTART(2),STARTV(30,2),SVXINC,SVXNOC,SVAZRE
      COMMON/IUSE/IUSE
      COMMON/JJUMP/JJLPP
      COMMON/JUMPST/JLPPST,TSAVET,WZERST
      COMMON/LPRINT/LFRINT,INMAX,PREDP,GY,TGL
      COMMON/LOADS/QLCAD(16),QTIME(16),QVALUE(8),SAVE(8,8)
      COMMON/LONOFF/LBPM
      COMMON/LBM/TIMLBM(30),THRLBM(30),WCTLBM(30),PLBM(2),MLBM1,MLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TALALT(5),TALTBL
     *(5),MPS1,MPS2,TGLIM,GLIMIT
      COMMON/NAME/NAME(128),WORD(129)
      COMMON/NALTW/NALTW
      COMMON/NTIME/NTIME
      COMMON/OLOW/CLOW,WINT
      COMMON/OMSRCS/FCPS(15),FRCS(15),WCOMPS(15),WCRCS(15),WOCUPP(15)
      COMMON/OUTPUT/OLITM(6),OUTTV(6),OUTISP(6),CLTMC(6)
      COMMON/OMAX/OMAX,OFLG,OPEN,MAXO
      COMMON/REST/JSTR,NCMRS(4),NTCN,TR(4),LSB,NFB,NOPAR,WIBT(15)
      COMMON/RESTX/RESTX(10,4)
      COMMON/RICGN1/TTABLE(20),SIGMA,SIGTAB(20),P15,P16
      COMMON/RTLS/VRINT(2),VSLOPE(2)
      COMMON/SRM/SRMTTB(250),SRMFTB(250),SRMWDT(250),SRMAE(2),MSRB
      COMMON/STAUT/STALT(15)
      COMMON/TABISP/MTAU,TIMISP(20),TRFISP(20)
      COMMON/TABS/TBL(101),WBL(101),UBL(101),VBL(101),XBL(101),YBL(101),
     *ZBL(101),XMBL(101)
      COMMON/TAULIM/TALLIM,NTAU,NBETA,TIMTAU
      COMMON/TGL/TGL(6),TVL(6),TLL(6),RATIO
      COMMON/VAERO/VAERO(10,19),M18,M19,MBS18,MES19
      COMMON/VCHSTG/VCHSTG
      COMMON/VIC/VIC(2),XK1(2),XK2(2),XK3(2),XK4(2),LK
      COMMON/WING/ALTTBL(75),WTBL(75,2),NWINC
      COMMON/XTRA/XMD(15),CFR(15),F(15),CISP(15)
      COMMON/YLI/YLINT(7,10),PHITE(10),PSIREC(10),XCDPHI(10),FSTR(10)
      DATA LBPM/.FALSE./
C
C     **********************
C     THIS PROGRAM WAS FORMULATED AND PROGRAMMED
C     DURING 1973 AND 1974 BY J.T.LYONS OF
C     NORTHROP SERVICES, INC., HUNTSVILLE, ALABAMA.
C
C     FURTHER MODIFICATIONS WERE IMPLEMENTED BY
C     J.T.LYONS OF BYNET TECS,INC HUNTSVILLE,ALA
C     DURING THE FALL OF 1980
C*************MASTRE************
C
C     P    MINIMUM HAMILTONIAN (MIN-H)
C     A    ASCENT
C     S    SHUTTLE
C     TR   TRAJECTORY
C     E    EVALUATION
C          PROGRAM
C
C     **********************
C
      CALL PFSINT
      NTIME=1
      NCASE=1
      ISTART=0
C
C     READ THE INPUT
C     CONTINUE
C-----------------------------OBTAIN INPUT AND INITIALIZATION VALUES-----
      CALL MENU
C     INTEGRATE THE TRAJECTORY
```

```
      2 IF(IPR.EQ.0.OR.NCM1.LE.5) GO TO 5
        CELVP=FPRFAC*CHVEL
        FRK4=1.-EXP(-CELVP*ENGDAT(7,1)/ENGDAT(6,1))
      5 CALL AFORUN
        CALL AFORNO
        IF(NMAX.LE.0) GO TO 4
     15 CALL BAKRN
        WRITE(1,30) T,(COVL(I),I=1,7)
     30 FORMAT(/1X,11HCC( LAMBDAS/,3X,8E14.6)
        CALL BADLX
        IF(NOB(17).EQ.0) GO TO 35
        NOSS=NOSS-NTCN
        CALL AFORUN
        CALL AFORNC
        NOSS=NOSS+NTCN
        CALL BADLX
     35 CALL ANEWCH
        NMAX=NMAX-1
        GO TO 2
C
      4 NMAX=1
        CALL BCPTBL
        CALL BIPTBL
        IF(DEPAND .OR. LAST.EQ.1) GO TO 2
        STOP
        END
```

```
      SUBROUTINE ACNTRC
C
C      DETERMINES THE VALUES OF PITCH AND YAW ATTITUDE ANGLES FROM CONTR . OL
C      TABLES
C
      CHARACTER*60 HEAD
      CHARACTER*6 MISION
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CT2,TG,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FA2,BCH3,2),CBAXIL,TCHIR,VRCUT,FPGFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIP,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVARS(12),CELXDW(15,3),DELXOR(7,4),CELXD(15,7),WZERO,ALT,
     *XMACH,GDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *BYL,BEL,BHMAX(15),BMMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITER,IVD,IXR,JUMP,KAT,KCYTAB,XFAGE,KS1,XMIA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PIAH,PPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINDP,KRDERB,MISION,
     *IRTFLG,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/BGEN3/NOSS,JTB,NENT,NACT,NOM1,NF(4)
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,C..,CMLE,OMEGA,ALT1,ALT2,PSL,
     *GZERO,PTN,PTNG,PSFTM,PFN,PFNG,PSFFP
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYOTBL(91)
      COMMON/FORINT/HBANK(2),T,TT,W,U,V,X,Y,Z,XM1,ZAF(18),CVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,HPX,HPN,HAN
      COMMON/TABLK/MCON1,MCON2,MCON3,MCON4
C     GET CHIP AND CHIY AS FUNCTIONS OF TIME
      I=KWTA(KCYTAB)-1
      NN=NP(KCYTAB)
      IF(NN.EQ.0) RETURN
      GO TO (1,2,3,4),KCYTAB
    1 CALL AMULG(I,MCON1,NN,T,TOBL(  1),CHIP,CPOTBL(  1),CHIY,CYCTBL(  1
     *))
      GO TO(5,6),I
    2 CALL AMULG(I,MCON2,NN,T,TOBL( 31),CHIP,CPOTBL( 31),CHIY,CYCTBL( 31
     *))
      GO TO(5,6),I
    3 CALL AMULG(I,MCON2,NN,T,TOBL( 61),CHIP,CPOTBL( 61),CHIY,CYCTBL( 61
     *))
      GO TO(5,6),I
    4 CONTINUE
C     COME HERE IF CHIY=0
    5 CHIY=0
C     COME HERE IF CHIY.NE.0
    6 RETURN
      END
```

ACSTOP

```
      SUBROUTINE ACST(P,VALE,YL,I)
C
C     EVALUATES THE PAYOFF AND CONSTRAINTS AND THE INITIAL ADJOINT VARI - ABLES
C           P        : CONSTRAINT INDEX
C           VALE     : VALUE OF CONSTRAINT
C           YL       : INITIAL ADJOINT VARIABLES
C           I        : INDEX FOR ARRAY LOCATION
      CHARACTER*60 HEAD
      CHARACTER*6 PISICN
      COMMON/CCC/GRAPH,JC,DEMAND
      COMMON/AGEN1/TIPE(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CTZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *HEAD,FRINT(15),STEP(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XR,SW,SU,SV,Q,
     *VI(25),DVARS(13),CELXDW(15,3),DELXDR(7,4),CELXD(19,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYE,BEL,BHMAX(15),EHMN,BSTEP(15),HMMB,HMXB,TPOLY,XMIAO
      COMMON/AGEN3/ITER,IWO,IXR,JUMP,KAT,KCYTAB,NPAGE,KS1,KWTA(4),LINES,
     *NBGCT(4),NENCET(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINOB,KRDERB,PISION,
     *IRTFLG,NROLL,NCGAST,IFACT,NOBASE,LSTGE(16),PSWC(15)
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,C,CMLE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,PTN,PTNG,FSFTM,PFN,PFNG,PSFFP
      COMMON/CSTOP/W,U,V,X,Y,Z,XMXM,T,HEAT,ZZZ
      COMMON/DRNGDT/DRNGDT(2)
      COMMON/HEAT/MAXHAT
      COMMON/IPR/IFR,LPMX,ILAST,KOT(15),FRKN,PRTCT
      COMMON/QMAX/QMAX,QFLG,OPEN,MAXQ
      COMMON/RTLS/VRINT(2),VSLOPE(2)
      COMMON/REST/LSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NCFAR,WIBT(15)
      COMMON/VGHSTE/VGHSTG
      COMMON/VIC/VVIC(2),VXK1(2),VXK2(2),VXK3(2),VXK4(2),LK
      DIMENSION Q(2,3),STATE(3)
      EQUIVALENCE (STATE,W)
      DIMENSION YL(7,10)
      REAL A(6,6),AD(3)
      DIMENSION P(3,6)
      DOUBLE PRECISION VALE,DPM,SS,DPW,DPU,DPV,DPX,DPY,DPZ
     *,DV,DF,E,B,C3,DWR,DUR,DVR,VVR
      CPMASS=XMXM
      CPW=W
      CPU=U
      CPV=V
      CPX=X
      CPY=Y
      CPZ=Z
      IF(I.EQ.1) MAXQ=0
      IF(M.EQ.0)GO TO 1000
      GO TO (1,2,3,4,5,6,7,8,9,8,8,12,12,8,15,16,16,1,19,20,21,22),M
C
C     PAYLOAD
    1 VALE=XMXM
      YL(7,I)=1.
      RETURN
C
C     INERTIAL VELOCITY
    2 VALE=DSQRT(DPW*DPW+DPU*DPU+DPV*DPV)
      YL    (1,I)=W/VALE
      YL    (2,I)=U/VALE
      YL    (3,I)=V/VALE
      RETURN
C
C     INERTIAL FLIGHT PATH ANGLE
    3 SING=LS/VI
```

```
      TEMP=RAD/(VI*SQRT(AMAX1(0.,1.-SING**2)))
      VALE=ASIN(SING)*RAD
      IF(VALE.GT.180.)VALE=VALE-360.
      YL(1,I)=(N(2,1)-SING*W/VI)*TEMP
      YL(2,I)=(N(2,2)-SING*U/VI)*TEMP
      YL(3,I)=(N(2,3)-SING*V/VI)*TEMP
      DO 103 J=4,6
  103 YL(J,I)=N(2,J)*TEMP
      RETURN
C
C     RADIUS
    4 VALE=DSQRT(DPX*DPX+DPY*DPY+DPZ*DPZ)
      YL    (4,I)=X/VALE
      YL    (5,I)=Y/VALE
      YL    (6,I)=Z/VALE
      RETURN
    5 RETURN
    6 RETURN
C
C     RELATIVE VEL(CI)
    7 VALE=VVR
      YL(1,I)=DWR/VVR
      YL(2,I)=DUR/VVR
      YL(3,I)=DVR/VVR
      YL(4,I)=(-DUR*A32W+DVR*A22W)/VVR
      YL(5,I)=(DWR*A32W-DVR*A12W)/VVR
      YL(6,I)=(-DWR*A22W+DUR*A12W)/VVR
      IF(M.EQ.7) RETURN
      YL4=YL(4,I)
      YL5=YL(5,I)
      YL6=YL(6,I)
      GO TO 20
C
    8 CONTINUE
      IF(M.EQ.10.OR.M.EQ.11) GO TO 30
C
C     EARTH RELATIVE AZIMUTH(HEADING ANGLE)
      VALE=ATAN2(WRE,-VRE)
      TEMP=VRE/(VRE*VRE+WRE*WRE)
      TEMP1=WRE/(VRE*VRE+WRE*WRE)
      DO 80 J=1,6
   80 YL(J,I)=-(TEMP*P(1,J)-TEMP1*P(3,J))*RAD
      VALE=VALE*RAD
      RETURN
C
C     INERTIAL HEADING ANGLE
   30 VALE=ATAN2(WS,-VS)
      TEMP1=WS/(VS*VS+WS*WS)
      TEMP=VS/(VS*VS+WS*WS)
   32 DO 108 J=1,6
  108 YL(J,I)=-(TEMP*N(1,J)-TEMP1*N(3,J))*RAD
      IF(M.EQ.10.OR.M.EQ.14) GO TO 10
      IF(M.EQ.11)GO TO 11
      VALE=VALE*RAD
      RETURN
C
C     RELATIVE GAMMA
    9 SING=LRE/VVR
      VALE=ASIN(SING)*RAD
      TEMP=RAD/(VVR*SQRT(AMAX1(0.,1.-SINE*SING)))
      YL(1,I)=TEMP*(D(1,2)-DWR*SING/VVR)
      YL(2,I)=TEMP*(D(2,2)-DUR*SING/VVR)
      YL(3,I)=TEMP*(D(3,2)-DVR*SING/VVR)
      YL(4,I)=TEMP*(W/R-SING*(D(1,2)*VVR/R+(A22W*DVR-A32W*DUR)/VVR))
      YL(5,I)=TEMP*(V/R-SING*(D(2,2)*VVR/R+(A32W*DWR-A12W*DVR)/VVR))
      YL(6,I)=TEMP*(V/R-SING*(D(3,2)*VVR/R+(A12W*DUR-A22W*DWR)/VVR))
      RETURN
C
```

```
C       INCLINATION
   10   SINB=SIN(VALE)
        COSB=COS(VALE)
        CINC=STH*SINE
        VALE=ACOS(CINC)*RAD
        TEMP1=SINB*CTH/SIN(VALE/RAD)*RAD
        TEMP2=COSB*STH/SIN(VALE/RAD)
        DO 110 J=1,6
  110   YL(J,I)=-(TEMP1*N(6,J)+TEMP2*YL(J,I))
        IF(P.EQ.14) GO TO 14
        RETURN
C
C       LINE OF NODES
   11   TEMP1=A(1,1)*D(1,2)+A(2,1)*D(2,2)+A(3,1)*D(3,2)
        TEMP2=A(1,3)*D(1,2)+A(2,3)*D(2,2)+A(3,3)*D(3,2)
        PHI=ATAN2(TEMP1,TEMP2)
        VALE=PHI+ATAN2(CTH*WS,VS)
        VALE=VALE*RAD
        TA=WS*VS*STH/(VS**2+(WS*CTH)**2)
        TB=(VS**2+WS**2)*CTH/(VS**2+(WS*CTH)**2)
        DO 111 J=1,6
  111   YL(J,I)=(N(4,J)-TB*YL(J,I)/RAD-TA*N(6,J))*RAD
        RETURN
   12   RETURN
C
C       ORBITAL VELOCITY CORRECTION
   14   SENC=SIN(VALE/RAD)
        CENC=COS(VALE/RAD)
        VIC=VVIC(LK)
        XK1=VXK1(LK)
        XK2=VXK2(LK)
        XK3=VXK3(LK)
        XK4=VXK4(LK)
        TEMP=1.+XK1*(XK3+XK4*SENC*SENC-CTH*CTH*XK2)
        TEMP1=XK1*XK2*CTH*STH
        VALE=VIC*SQRT(TEMP)-VI
        TEMP2=XK1*XK4*SENC*CENC
        DO 140 J=1,3
  140   YL(J,I)=VIC*(TEMP2*YL(J,I)/RAD)/SQRT(TEMP)-STATE(J)/VI
        DO 141 J=4,6
  141   YL(J,I)=VIC*(TEMP2*YL(J,I)/RAD+TEMP1*N(6,J))/SQRT(TEMP)
        RETURN
C
C       DYNAMIC PRESSURE PENALTY FUNCTION
C
   15   VALE=CPEN
        MAXC=I
        RETURN
   16   VALE=HEAT
        MAXHAT=I
        RETURN
C
C       RTLS VELOCITY REQUIREMENT
   23   E=VRIAT(LK)
        E=VSL(FE(LK)
        VALE=B+E*VALE-VVF
        YL(1,I)=-YL(1,I)
        YL(2,I)=-YL(2,I)
        YL(3,I)=-YL(3,I)
        YL(4,I)=E*YL(4,I)
        YL(5,I)=E*YL(5,I)
        YL(6,I)=E*YL(6,I)
        DRNGDT(LK)=E*DRNGDT(LK)
        RETURN
C
C
   19   CONTINUE
C
```

```
C       RANGE FROM LAUNCH SITE
   20   OMT=OMEGA*T
        SOMT=SIN(OMT)
        COMT=COS(OMT)
        STHC=STHL
        CTHC=CTHL
        XLR=CTHC*A(1,2)+STHC*(A(1,1)*SOMT+A(1,3)*COMT)
        YLR=CTHC*A(2,2)+STHC*A(2,3)*COMT
        ZLR=CTHC*A(3,2)+STHC*(A(3,1)*SOMT+A(3,3)*COMT)
        CRANGE=XLR*D(1,2)+YLR*D(2,2)+ZLR*D(3,2)
        RANGLE=ACOS(CRANGE)
        CNMM=1./1852.
        VALE=RE*RANGLE*CNMM
        TEMP=-RE*CNMM/(R*SIN(RANGLE))
        YL(4,I)=TEMP*(XLR-C(1,2)*CRANGE)+YL4
        YL(5,I)=TEMP*(YLR-C(2,2)*CRANGE)+YL5
        YL(6,I)=TEMP*(ZLR-C(3,2)*CRANGE)+YL6
        DRNGDT(LK)=-RE*STHC*OMEGA*CNMM/SIN(RANGLE)*(D(1,2)*(A(1,1)*
       *(COMT-A(1,3)*SOMT)-C(2,2)*A(2,3)*SOMT+C(3,2)*(A(3,1)*COMT-A(3,3)*SO
       *MT))
        IF(M.EQ.18) GO TO 23
        RETURN
   21   RETURN
   22   RETURN
C***    INITIALIZE PARTIAL MATRICES ***
C       CALCULATE D MATRIX
 100    R2=X*X+Y*Y+Z*Z
        R=SQRT(R2)
        VI2=W*W+U*U+V*V
        VI=SQRT(VI2)
        ER=0.5*(DPX*DPX+DPY*DPY+DPZ*DPZ)
        EV=DPW*DPW+DPU*DPU+DPV*DPV
        C3=EV-2.*CMUE/DR
        CWR=DPW-A22W*DPZ+A32W*DPY
        CUR=DPU-A32W*DPX+A12W*DPZ
        CVR=DPV-A12W*DPY+A22W*DPX
        VVR=0.5*SQRT(DWR*DWR+DUR*DUR+DVR*DVR)
        C(1,2)=X/R
        C(2,2)=Y/R
        C(3,2)=Z/R
        CTH=A(1,2)*D(1,2)+A(2,2)*D(2,2)+A(3,2)*D(3,2)
        STH=SQRT(AMAX1(0.,1.-CTH*CTH))
        RSTH=R*STH
        C(1,1)=(-Y*A(3,2)+Z*A(2,2))/RSTH
        C(2,1)=(X*A(3,2)-Z*A(1,2))/RSTH
        C(3,1)=(-X*A(2,2)+Y*A(1,2))/RSTH
        C(1,3)=(D(1,2)*CTH-A(1,2))/STH
        C(2,3)=(D(2,2)*CTH-A(2,2))/STH
        C(3,3)=(D(3,2)*CTH-A(3,2))/STH
        WS=D(1,1)*W+C(2,1)*U+D(3,1)*V
        US=D(1,2)*W+C(2,2)*U+D(3,2)*V
        VS=D(1,3)*W+C(2,3)*U+D(3,3)*V
C       CALCULATE N MATRIX
        DO 102 I=1,3
        ND(I)=D(I,3)*CTH+D(I,2)*STH
        N(4,I+3)=D(I,1)/RSTH
        N(5,I+3)=D(I,2)
        N(6,I+3)=D(I,3)/R
        DO 102 J=1,3
        N(I,J)=D(J,I)
 102    N(I+3,J)=0.
        TEMP=WS*CTH/STH
        N(1,4)=(A(3,2)*U-A(2,2)*V-WS*ND(1))/RSTH
        N(1,5)=(A(1,2)*V-A(3,2)*W-WS*ND(2))/RSTH
        N(1,6)=(A(2,2)*W-A(1,2)*U-WS*ND(3))/RSTH
        N(2,4)=(W-D(1,2)*US)/R
        N(2,5)=(U-D(2,2)*US)/R
        N(2,6)=(V-D(3,2)*US)/R
```

```
      A(3,4)=(D(1,1)*TEMF-D(1,3)*US)/R
      A(3,5)=(D(2,1)*TEMF-D(2,3)*US)/R
      A(3,6)=(D(3,1)*TEMF-D(3,3)*US)/R
      WRE=D(1,1)*DWR+C(2,1)*DUR+D(3,1)*DVR
      WRE=D(1,2)*DWR+C(2,2)*DUR+D(3,2)*DVR
      WRE=D(1,3)*DWR+C(2,3)*DUR+D(3,3)*DVR
      F(1,1)=D(1,1)
      F(1,2)=D(2,1)
      F(1,3)=D(3,1)
      F(1,4)=-D(2,1)*A22W+D(3,1)*A22W+(A(3,2)*DUR-A(2,2)*DVR-WRE*
     *AD(1))/RSTH
      F(1,5)=+D(1,1)*A22W-D(3,1)*A12W+(A(1,2)*DVR-A(3,2)*DWR-WRE*
     *AD(2))/RSTH
      F(1,6)=-D(1,1)*A22W+D(2,1)*A12W+(A(2,2)*DWR-A(1,2)*DUR-WRE*
     *AD(3))/RSTH
      TEMP=WRE*CTH/STH
      F(3,1)=D(1,3)
      F(3,2)=D(2,3)
      F(3,3)=D(3,3)
      F(3,4)=-D(2,3)*A22W+D(3,3)*A22W+(D(1,1)*TEMF-D(1,3)*URE)/R
      F(3,5)=+D(1,3)*A22W-D(3,3)*A12W+(C(2,1)*TEMF-D(2,3)*URE)/R
      F(3,6)=-D(1,3)*A22W+D(2,3)*A12W+(D(3,1)*TEMF-C(3,3)*URE)/R
      RETURN
      END
```

```
      SUBROUTINE ACER
C
C     EVALUATES THE STATE DERIVATIVES
C
      CHARACTER*60 HEADE
      CHARACTER*6 MISICN,ATMOSN
      REAL LIFT
      LOGICAL GLIMIT,CFLG,BOOST,ORBITR
      COMMON/AERODI/AERODI(30,16),AEROB(50,5,3),TCMACH(20),DELCA(20),
     *FNMONE(15),BKFABT(50,2),BKFTHR(50,2),CELCN(20),CELCM(20)
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTIL1,TMINH,
     *CT2,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WD(15),WJET(15),
     *HEAD,FRINT(15),STEP(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AZ,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XP,SW,SL,SV,Q,
     *IV(25),DVAR(12),EELXDW(15,3),DELXDR(7,4),EELXD(15,7),W2ERO,ALT,
     *MACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMAX(15),EHMN,BSTEP(15),HAMB,HMXB,IPOLY,XMIAD
      COMMON/AGEN3/ITER,IWD,IXR,JUMP,KAT,KCYTAB,MPAGE,KS1,KWIA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PINH,PESFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINDB,KRDERB,MISION,
     *IRTFLE,NROLL,NGCAST,IFAGT,NBBASE,LSTGE(15),KSWC(15),
      COMMON/AGEN4/QALFHA,QBETA,QCN,STH1,CTH1,XJX(3),XJY(3)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,APX,APY,AMZ,
     *IMZ,THPFA,SIDE,HPAE,AZW,FANC,FLATC,FOM,VISC
      COMMON/ALAT/ALAT,ALONG,ALTLS,NTABLE
      COMMON/APRINT/ELEIN,ELEOUT,DINBD,DOUTBD,FABT(3),CAX,CNP,CM2,CS,
     *CMX,CPT,SEMACH,DRAG,LIFT,XLAMDA,XKAUD
      COMMON/ARDC/FAT(14),ATMOSN
      COMMON/AVGGP/XGPA,YGPA,ZGPA
      COMMON/BODYF/TBCAT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,M1,M2,MS,
     *M6,M7,M8,M9,MSO,MS1,MSRW,AP(10),AY(10)
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,D,CMUE,OMEGA,ALT1,ALT2,PSL,
     *GZERO,PTN,PTKG,PSFIN,PFN,PFKG,PSFFN
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/DELFAB/DELALT(25),DELFAB(25),M30,M31,ITCL(15)
      COMMON/ELEVON/EPACH(25,2),ELEVON(25,4),M10,M11
      COMMON/FORINT/HBANK(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,NINC,KRDER,EU,EL,AYL,HPX,HPN,HAP
      COMMON/GARBGE/DX,DYI,DZI,DXDY,UDXDYI,UCXDY2,XCG,YCG,TX1,TX2,TY1,TY
     *2,THAT(3),TXS,TYS,TAU,SINRHO
      COMMON/GRAV/GTO,G11,G22,G23,G33
      COMMON/IMACH/IMACH
      COMMON/LBM/TIMLEP(30),TLBM(30,2)            ,PLBM(2),MLBM1,MLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MLAST/MLAST,MMMM,KKKK
      COMMON/M20/M17,M20,M21,M22,M23,M24
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MPS1,MPS2,TGLIP,GLIMIT
      COMMON/NTILT/NTILT
      COMMON/OMSRCS/FCMS(15),FRCS(15),WDCMS(15),WCRCS(15),WDDUMP(15)
      COMMON/OUTPUT/OUTTH(6),OUTTV(6),OUTTSP(6),OUTWE(6)
      COMMON/OMAX/CMAX,QFLG,QPEN,MAXQ
      COMMON/LOADS/QLOAD(16),QTIME(16),QVALUE(8),SAVE(8,8)
      COMMON/REST/USTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NOPAR,MIBT(15)
      COMMON/REYNO/RETNO,VE,SWE,SUE,SVE
      COMMON/RICCN1/TTABLE(20),SIGMA,SIETAB(20),M15,M16
      COMMON/SRM/SRMTTB(250),SRMFTB(250),SRMWDT(250),SRMAE(2),MSER
      COMMON/SRMEXT/SRMEXT
      COMMON/TCL/TCL(6),TVL(6),TLL(6),RATIO
      COMMON/VAERO/VAEROD(10,19),M18,M19,MBS18,MBS19
```

```
      COMMON/WIND/ALTTBL(75),WTBL(75,2),AWINC
      COMMON/XV/XV(4)
      DIMENSION WARG(2),WARGD(2),AW(2),BW(2),CW(2)
      EQUIVALENCE (WMAG,WARG(1)),(AZW,WARG(2))
      DIMENSION AERO(12),WXR(3),WIND(3),RXWXR(3),UR(3),UWXR(3),
     *AERODD(15),DDDX(6),AA(12),BB(12),CC(12)          ,AFAB(3),BFAB(3),
     *CFAB(3)
      EQUIVALENCE (FAB,FABT(1)),(FNB,FABT(2)),(FPB,FABT(3))
      EQUIVALENCE (CALP,AERO(1)),(CAO,AERO(2)),(CLBETA,AERO(3)),
     *(CMALP,AERO(4)),(CMO,AERO(5)),(CNALP,AERO(6)),(CNBETA,AERO(7)),
     *(CNO,AERO(8)),(CYBETA,AERO(9)),(CLC,AERO(10)),(CYO,AERO(11)),
     *(CNBO,AERO(12)),(PHMOH,WARGD(1)),(PAZWDH,WARGD(2))
      DIMENSION VAERO(18),VAERDD(18),VA(18),VB(18),VC(18)
      EQUIVALENCE(ACACF,VAERO(1)),(BCACF,VAERO(2)),(CCACF,VAERO(3)),
     *(ACNCF,VAERO(4)),(BCNCF,VAERO(5)),(CCNCF,VAERO(6)),
     *(ACMCF,VAERO(7)),(BCMCF,VAERO(8)),(CCMCF,VAERO(9)),
     *(ACAMF,VAERO(10)),(BCAMF,VAERO(11)),(CCAMF,VAERO(12)),
     *(ACNMF,VAERO(13)),(BCNMF,VAERO(14)),(CCNMF,VAERO(15)),
     *(ACMMF,VAERO(16)),(BCMMF,VAERO(17)),(CCMMF,VAERO(18))
      DIMENSION BAERO(6)
      EQUIVALENCE (CAX,BAERO(1)),(CNP,BAERO(2)),(CMZ,BAERO(3)),
     *(CS,BAERO(4)),(CPX,BAERO(5)),(CMY,BAERO(6))
      DIMENSION DFAB(1),CDFAB(1),AF(1),BF(1),CF(1)
      DATA XK1,XK2,XK3,XK4,XK5,XK6,XK7/17700.,.0019*03,.3048,3048.,
     *116.4,.432,5E063./
      BOOST=ITHR.LE.NCEVNT(1)
      CRBITR=.NOT.BOOST
      NUMC=1
      R2=X*X+Y*Y+Z*Z
      R=SQRT(R2)
      SCHIP=XV(1)
      CCHIP=XV(2)
      SCHIY=XV(3)
      CCHIY=XV(4)
      CTH=(A(1,2)*X+A(2,2)*Y+A(3,2)*Z)/R
      CT2=CTH*CTH
      CALL AGEO(1)
      EX=X*E11-A(1,2)*ETC
      EY=Y*E11-A(2,2)*ETC
      EZ=Z*E11-A(3,2)*ETC
      RTHE=LMF*RE/(UMF*UMF*(1.-CT2)+CT2)**.5
      ALT=R-RTHE
    2 DO 19 I=1,3
   19 WIND(I)=0.
      WMAG=0.
      AZW=0.
      WXR(1)=A22W*Z-A32W*Y
      WXR(2)=A32W*X-A12W*Z
      WXR(3)=A12W*Y-A22W*X
      WXRM=SQRT(WXR(1)*WXR(1)+WXR(2)*WXR(2)+WXR(3)*WXR(3))
      DO 20 I=1,3
   20 WXR(I)=WXR(I)/WXRM
      UR(1)=X/R
      UR(2)=Y/R
      UR(3)=Z/R
      RXWXR(1)=UR(2)*WXR(3)-UR(3)*WXR(2)
      RXWXR(2)=UR(3)*WXR(1)-UR(1)*WXR(3)
      RXWXR(3)=UR(1)*WXR(2)-UR(2)*WXR(1)
      IF(NWIND.EQ.0) GO TO 22
      IF(ALT.LT.ALTTBL(1).OR.ALT.GT.ALTTBL(NWIND)) GO TO 22
      CALL SPLINE(2,M7,75,ALT,ALTTBL,WARG,WTBL,WARGD,2,MSRW,AW,BW,CW)
      SAZW=-SIN(AZW)
      CAZW=-COS(AZW)
      DO 21 I=1,3
   21 WIND(I)=WMAG*(SAZW*UWXR(I)+CAZW*RXWXR(I))
   22 SW=W-WXR(1)-WIND(1)
      SU=U-WXR(2)-WIND(2)
      SV=V-WXR(3)-WIND(3)
```

```
      VR2=SW*SW+SU*SU+SV*SV
      VR=SQRT(VR2)
C
C*****
      FAT(1)=ALT
      CALL FRA63(PAT,ATMCSN,IER)
      FAM=PAT(2)*1000.
      RHO=PAT(6)
      XMU=PAT(10)
      REYNO=RHO*VR*XLEN/XMU
      XMACH=VR/PAT(9)
C
C     CALCULATE ALTITUDE FOR BASE FORCE TABLE LOOKUP
C
      DALTB=AEROB(1,1,1)-ALTLS
      HGT=ALT+DALTB
      IF(BOOST) GO TO 300
C
C     SPLINE INTERPOLATION FOR SECOND STAGE AERODYNAMIC DATA(MONOVARIATE
C
      SMACH=XMACH
      IF(XMACH.GT.10.) XMACH=10.
      CALL SPLINE(12,M1,30,XMACH,AEROD(1,14),AERC,AERCD,AEROCD,1,M2,AA,
     *EB,CC)
      XMACH=SMACH
  300 CONTINUE
      IF(ITHR.GE.NCBASE) GO TO 31
C
C     SPLINE INTERPOLATION FOR POWER-ON BASE FORCE EFFECTS
C
      L=LSTEE(ITHR)
      IF(L.EQ.2.AND.TAE(1,ITHR).LT.2.9) L=3
      CALL SPLINE(3,ME,50,HGT,AEROB(1,1,L),FABT,AERCE(1,2,L),AERCOD(13),
     *1,M9,AFAB,BFAB,CFAB)
      GO TO 32
   31 FAB=CEAXIL
      FNB=0.
      FMB=0.
   32 CONTINUE
      XM=XMIAD+XMALG
C     EARTH RELATIVE VELOCITY
      SWE=W-WXR(1)
      SUE=U-WXR(2)
      SVE=V-WXR(3)
      VE=SQRT(SWE*SWE+SUE*SUE+SVE*SVE)
      Q=.5*RHO*VR2
      QPEN=AMAX1(QPEN,Q*FSFFM)
      SQ=S(ITHR)*Q
      NCHI=0
   25 IF(IFACT.NE.3.OR.ITHR.GE.MINH) GO TO 27
      CALL AMULG(2,M16,30,VE,TTBL,THETA,CPTBL,PSI,CYTBL)
C
C     COMPUTE ATTITUDE ANGLES IN BOOST REFERENCE COORD SYSTEM FOR PRINT
C     BOOST COORD SYSTEM : X-NORTH
C                          Y-EAST
C                          Z-DOWN LOCAL VERTICAL
C
      SPSI=SIN(PSI/RAD)
      CPSI=COS(PSI/RAD)
      SINT=SIN(THETA/RAD)
      COST=COS(THETA/RAD)
      SINS=SIN(SIGMA/RAD)
      COSS=COS(SIGMA/RAD)
      XX=COST*(CPSI*CA+SPSI*SA)
      YY=SINT
      XZ=COST*(SPSI*CA-CPSI*SA)
      YZ=SA*COST*(-SINT*(CPSI*COSS-SPSI*SINS)+CA*(SINT*SPSI*COSS-CPSI*
     *SINS)
```

```
      ZZ=SA*COST*(-SINT*(PSI*SINS-SPSI*COSS)+CA*(SINT*SPSI*SINS+CPSI*
     *(OSS)
      GO TO 260
   27 IF(TENOR.LE..001.OR.T.GT.TENOR) GO TO 26
      XX=-CCHIY*SCHIP*CCHIR-SCHIY*SCHIR
      YY=CCHIY*CCHIP
      YZ=CCHIY*SCHIP*SCHIR-SCHIY*CCHIR
      XZ=-CCHIP*SCHIR
      ZZ=SCHIY*SCHIP*SCHIR+CCHIY*CCHIR
  260 CONTINUE
      CCHIY=SQRT(YX*YX+YY*YY)
      SCHIP=YX/CCHIY
      CCHIP=YY/CCHIY
      SCHIY=YZ
      SCHIR=-XZ/CCHIY
      CCHIR=ZZ/CCHIY
   26 CONTINUE
   80 VWB=SCHIP*SW+CCHIP*SU
      VZB=SCHIY*VWB-CCHIY*SV
      VYB=CCHIP*SW-SCHIP*SU
      VXB=CCHIR*VYB+SCHIR*VZB
      VZB=SCHIR*VYB-CCHIR*VZB
      VYB=CCHIY*VWB+SCHIY*SV
      ALF=0.
      ALFY=0.
      IF(T.LT..3) GO TO 50
      ALF=ATAN2(VXB,VYB)
      ALFY=ASIN(VZB/VB)
   50 CONTINUE
      CLDA=ALF
      CLDB=ALFY
      CALPH=ABS(Q*ALF)
      CBETA=ABS(Q*ALFY)
      IF(ITHR.GE.MINM.OR.IFACT.NE.2) GO TO 504
      IF(NTILT.NE.1.OR.ITHR.GE.IPOLY) GO TO 504
      CALL AMULG(2,M20,3G,XMACH,AEROD(1,16),ALF,AEROD(1,13),ALFY,
     *AEROD(1,15))
      COLDA=COS(CLDA)
      SOLDA=SIN(CLDA)
      COLDB=COS(CLDB)
      SOLDB=SIN(CLDB)
      SALP=SIN(ALF)
      CALP=COS(ALF)
      SBETA=SIN(ALFY)
      CBETA=COS(ALFY)
      SINDB=SOLDB*CBETA-COLDB*SBETA
      COSDB=COLDB*CBETA+SOLDB*SBETA
      XX=CALP*COLDA+SALP*SOLDA*COSDB
      XY=-CALP*SOLDA+SALP*COLDA*COSDB
      XZ=SALP*SINDB
      YX=-SALP*COLDA+CALP*SOLDA*COSDB
      YY=SALP*SOLDA+CALP*COLDA*COSDB
      YZ=CALP*SINDB
      ZX=-SOLDA*SINDB
      ZY=-COLDA*SINDB
      ZZ=COSDB
      XN=XX*(CCHIR*CCHIR+SCHIR*SCHIY*SCHIP)+YY*(CCHIY*SCHIP)+YZ*(SCHIR*
     *CCHIP-CCHIR*SCHIY*SCHIP)
      YN=XX*(-CCHIR*SCHIP+SCHIR*SCHIY*CCHIP)+YY*(CCHIY*CCHIP)+YZ*(
     *-SCHIR*SCHIP-CCHIR*SCHIY*CCHIP)
      ZN=XX*(-SCHIR*CCHIY)+YY*SCHIY+YZ*CCHIR*CCHIY
      XN=XX*(-SCHIR*CCHIY)+YY*SCHIY+XZ*CCHIR*CCHIY
      ZN=ZX*(-SCHIR*CCHIY)+ZY*SCHIY+ZZ*CCHIR*CCHIY
      CCHIY=SQRT(YXN*YXN+YYN*YYN)
      SCHIP=YXN/CCHIY
      CCHIP=YYN/CCHIY
      SCHIY=YZN
      SCHIR=-XZN/CCHIY
      CCHIR=ZZN/CCHIY
```

```
  504 CONTINUE
C
      ALP=ALF*RAD
      BETA=ALFY*RAD
      SMACH=VE/PA(9)
      SREYNC=SQRT(REYNC)
C
C     DEFINITION OF AERODYNAMIC COEFFICIENTS
C
C     CAX= AXIAL FORCE COEFFICIENT.
C     CS=  SIDE FORCE COEFFICIENT.
C     CNP=NORMAL FORCE COEFFICIENT.
C     CMZ=PITCHING MOMENT COEFFICIENT
C     CMY=ROLLING MOMENT COEFFICIENT
C     CMX=YAWING MOMENT COEFFICIENT
C
      IF(ORBITR) GO TO 302
      IF(ITHR.GE.NGBASE) GO TO 301
C
C     LINEAR INTERPOLATION FOR SSME THROTTLING,Q AND ALPHA-BETA EFFECTS
C     ON POWER-ON BASE FORCE--BOOSTER ONLY
C
      CALL AMULG(2,M24,50,HGT,AEROB(1,1,1),FK1THR,BKFTHR(1,1),
     *FK2THR,BKFTHR(1,2))
C
      CALL AMULG(2,M23,50,HGT,AEROB(1,1,1),FABORB,AEROB(1,5,1),
     *CREF,AEROB(1,5,2))
C
C     SSME THROTTLING EFFECT-BASE FORCE
C
  320 IF(PPSFLG(ITHR).EQ.1) TAU=THROT(ITHR)
      IF(PPSFLG(ITHR).EQ.2) CALL AMULG(1,M51,15,1,THRPS,TAU,FTBL,C,C)
      IF(PPSFLG(ITHR).EQ.3.AND..NOT.QFLG) TAU=THRCT(ITHR)
      DTHRTL=1.0-TAU
      BASTHR=(FK2THR*DTHRTL+FK1THR)*DTHRTL+1.0
C
C     ORBAST=ORBITER BASE FORCE-AXIAL,DUE TO SSME THROTTLING
C     FABORB=ORBITER BASE FORCE-AXIAL,FOR SSME=100% EFF.
C
      ORBAST=FABORB*BASTHR
C     TOTAL AXIAL BASE FORCE-SSME THROTTLING EFFECT
      FAB=FAB-FABORB+ORBAST
C     TOTAL NORMAL BASE FORCE-SSME THROTTLING EFFECT
      FNB=FNB*BASTHR
C     TOTAL PIT MOMENT BASE FORCE-SSME THROTTLING EFFECT
      FMB=FMB*BASTHR
C
C     Q EFFECT-BASE FORCE
C
      IF(QREF.LE.ME-) GO TO 330
      QRATIO=Q/QREF
C     AXIAL BASE FORCE-Q EFFECT
      FAB=FAB*QRATIO
C     NORMAL BASE FORCE-Q EFFECT
      FNB=FNB*QRATIO
C     PIT MOMENT BASE FORCE-Q EFFECT
      FMB=FMB*QRATIO
C
C
C     LINEAR INTERPOLATION FOR ALPHA-BETA FACTOR FOR POWER-ON BASE
C     FORCE EFFECT
C
  330 IF(T.LT.2.0) GO TO 301
      CALL AMULG(2,M17,50,HGT,AEROB(1,1,1),FKALF,BKFABT(1,1),
     *FKBETA,BKFABT(1,2))
      ABFACT=1.0+FKALF*ALP+FKBETA*BETA**2
      FAB=FAB*ABFACT
      FNB=FNB*ABFACT
      FMB=FMB*ABFACT
C
```

```
C       FIRST STAGE AERODYNAMIC COEFFICIENTS
C
C
C       DETERMINE ELEVON DEFLECTION ANGLE
C
  301 CONTINUE
      CALL AMULG(2,M11,25,VE/,3048,EMACH(1,1),ELEIN,ELEVON(1,1),ELEOUT,
     *ELEVON(1,2))
      CALL AMULG(2,M11,25,SEMACH,EMACH(1,2),DINBD,ELEVON(1,3),DOUTBD,
     *ELEVON(1,4))
C
C     DELTA(1)=INBOARD ELEVON DEFLECTION  (SCHEDULE 6)
C     DELTA(2)=OUTBOARD ELEVON DEFLECTION (SCHEDULE 6)
C     DELTA(3)=INBOARD ELEVON DEFLECTION  (USC/RI AERO DATA BASE)
C     DELTA(4)=OUTBOARD ELEVON DEFLECTION (USC/RI AERO DATA BASE)
C
      DINBD=ELEIN-DINBD
      DOUTBD=ELEOUT-DOUTBD
C
      CALL CAERO(BAERO,XMACH,ALP,BETA,DINBD,DOUTBD)
      GO TO 61
C
C     SECOND STAGE AERODYNAMIC COEFFICIENTS
C
  302 CONTINUE
C
      CAX=CAO+CAALF*ALF
      CNP=CNO+CNALF*ALF
      CMZ=CMO+CMALF*ALF
      CS=CYO+CYBETA*ALFY
      CNX=CNBO+CNBETA*ALFY
      CMY=CLO+CLBETA*ALFY
C
      IF(XMACH.LE.10.0) GO TO 61
C
C     HYPERSONIC VISCOUS INTERACTION EFFECTS (OREITER STAGE ONLY)
C
C                REYNO  - REYNOLDS NUMBER
C                XLAMDA - MEAN FREE PATH (PAGE 13,U.S. STD ATMOS, 1962)
C                XKNUD  - KNUDSEN NUMBER (AERO DATA BOOK 4.1.5)
C                CF     - CONTINUUM FLOW
C                MF     - FREE MOLECULAR FLOW
C
      XLAMDA=2.332353E-05*PAT(12)/PAM
      XKNUD=.1407*XLAMDA/VR*(PAM/RHO)**.5
      CALL SPLINE(10,P18,10,XMACH,VAERO(1,19),VAERO,VAEROD,VAEROD,1,
     *P19,VA,VB,VC)
      CACF=ACACF+(BCACF+CCACF*ALP)*ALP
      CNCF=ACNCF+(BCNCF+CCNCF*ALP)*ALP
      CMCF=ACMCF+(BCMCF+CCMCF*ALP)*ALP
      CAMF=ACAMF+(BCAMF+CCAMF*ALP)*ALP
      CNMF=ACNMF+(BCNMF+CCNMF*ALP)*ALP
      CMMF=ACMMF+(BCMMF+CCMMF*ALP)*ALP
      IF(XKNUD.LT..01) XKNUD=.01
      IF(XKNUD.GT.10.) XKNUD=10.
      TEMP=SIN(9.0(9*(XKNUD-.01)/RAD)
      CAX=(CAMF-CACF)*TEMP+CACF
      CNP=(CNMF-CNCF)*TEMP+CNCF
      CMZ=(CMMF-CMCF)*TEMP+CMCF
      CCN=Q*CNP
C
   61 CONTINUE
C
C     DOC AERO COEFF TOLERANCES=F(MACH NBR) IF ITCL NOT EQUAL TO ZERO
C
      IF(ITCL(ITHR).EQ.0) GO TO 550
      CALL AMULG(2,M21,20,XMACH,TOMACH,DCAX,DELCA,DCNP,DELCN)
      CALL AMULG(1,M21,20,XMACH,TOMACH,DCMZ,DELCM,0,0)
      CAX=CAX+DCAX
```

```
      CNF=CAF+DCNP
      CM2=CM2+DCM2
      CALL SPLINE(1,M3E,25,MGT,DELALT,DFAB,DELFAB,DCFAB,1,M31,AF,BF,CF)
      FAB=FAB+DFAB(1)
  55E CONTINUE
C
C MOMENT COEFFICIENTS ABOUT C.G. NOW FORMED
      TEMP=1./XLEN
      CXPR=(XCG-XREF)*TEMP
      CYPR=(YCG-YREF)*TEMP
      CMCG=CM2-CNP*CYPR+CAX*DXPR
      CNCG=CMX-CS*CYPR
      CLCG=CMY+CS*CXPR
C
      FAA=CAX*SQ+FAB
      FAN=CNP*SQ+FNB
      SIDE=CS*SQ
      AMX=CNCG*SQ*XLEN
      AMY=CLCG*SQ*XLEN
      AM2=CMCG*SQ*XLEN+FMB-FNB*DYPR*XLEN+FAB*DXPR*XLEN
C END OF NEW AERODYNAMICS.
C
C     THRUST AND MOMENT CALCULATIONS FOR MULTIPLE PROPULSION SYSTEMS
C
      TXX=0.
      TYY=0.
      TZZ=0.
      TXS=0.
      TYS=0.
      TZS=0.
      TMOT=0.
      TM2=0.
      DO 50C NSY=1,7
      N=NSY
      IF(ORBITR.AND.N.EQ.1) GO TO 500
      IF(ORBITR.AND.N.EQ.7) N=1
      IF(BOOST.AND.ITHR.LT.ICONSW.AND.N.EQ.2) GO TO 500
      IF(BOOST.AND.ITHR.LT.ICONSW.AND.N.EQ.7) N=2
      IF(BOOST.AND.ITHR.GE.ICONSW.AND.N.EQ.1) GO TO 500
      IF(BOOST.AND.ITHR.GE.ICONSW.AND.N.EQ.7) N=1
      CVAR(N+7)=0.
      CUTISP(N)=0.
      CUTTH(N)=0.
      CUTTV(N)=0.
      CUTWD(N)=0.
      TCL(N)=0.
      TVL(N)=0.
      TLL(N)=0.
      IF(TNE(N,ITHR).LT..001) GO TO 500
      MMIN=0
      DO 400 I=1,15
      IF(NSYST(I).NE.N) GO TO 400
      IF(MMIN.EQ.0) MMIN=I
      MMAX=I
  400 CONTINUE
      GO TO (410,420,430,440,450,460),N
C
C     MPS SYSTEM
C
C
  410 MPS=MFSELG(ITHR)
      TAUMIN=.65
      IF(ORBITR) GO TO 411
      CALL AMULG(1,MSC,5,ALT,TAUALT,TAUMIN,TAUTBL,0,0)
  411 GO TO (412,413,415,416),MPS
C
C     NORMAL OPERATIONS
C
  412 TAU=THROT(ITHR)
```

```
      IF(QFLG) QFLG=.FALSE.
  412 THRVAC=TAU*ENGDAT(6,MMIN)*PTN
      GO TO 417
C
C     TABLE OF THROTTLE VS TIME
C
  413 CALL APULG(1,MS1,15,T,TIMMPS,TAU,FTBL,C,D)
      IF(TAL.LT.TALMIN) TAU=TAUMIN
      GO TO 413
C
C     GMAX THROTTLE
C
  415 IF(.NOT.QFLG) GO TO 412
      IF(TAL.LT.TALMIN) TAU=TAUMIN
      IF(TAL.GE.THROT(ITHR)) GO TO 412
      GO TO 413
C
C     G LIMIT THROTTLE
C
  416 IF(.NOT.GLIMIT) GO TO 412
      GM=XM*GLIMIT(ITHR)*GZERO
      TAU=GP/(ENGDAT(6,MMIN)*TNE(1,ITHR)*PTN)
      IF(TAL.LT.TALMIN) TAU=TAUMIN
      GO TO 413
  417 XISP=VARISP(TAU,T)
  418 WDOT=THRVAC/(GZERO*XISP)*TNE(N,ITHR)
  419 CUTISP(N)=XISP
      CUTWD(N)=WDOT
      THRUST=THRVAC-ENGDAT(8,MMIN)*PAM
      IF(THRUST.LT.0.) THRUST=0.
      CUTTH(N)=THRUST*TNE(N,ITHR)
      CUTTV(N)=THRVAC*TNE(N,ITHR)
      XMDOT=XMDOT+WDOT
      CVAR(N+7)=-WDOT
      IF(OREITR.AND.N.EQ.1) GO TO 480
      IF(BOOST.AND.N.EQ.2.AND.ITHR.LT.ICONSW) GO TO 460
      IF(BOOST.AND.N.EQ.1.AND.ITHR.GE.ICONSW) GO TO 480
      GO TO 470
C
C     SRB SYSTEM
C
  420 CALL APULG(2,MSRB,250,T,SRMTTB,THRVAC,SRMFTB,WDOT,SRPWDT)
      XISP=0.
      IF(WDOT.GT.0.) XISP=THRVAC/WDOT
      THRVAC=THRVAC*PTN
      SRMEXT=SRMAE(1)+SRPAE(2)*T
      ENGDAT(8,MMIN)=SRMEXT
      WDOT=WDOT*TNE(2,ITHR)*PTKG
      GO TO 419
C
C     LBM SYSTEM
C
  430 CALL APULG(2,MLBM1,30,T,TIMLBM,THRVAC,TLBM(1,1),WDOT,TLBM(1,2))
      XISP=0.
      IF(WDOT.GT.0.) XISP=THRVAC/WDOT
      THRVAC=THRVAC*PTN
      WDOT=WDOT*TNE(3,ITHR)*PTKG
      GO TO 419
C
C     OMS SYSTEM
C
  440 THRVAC=FOMS(ITHR)*PTN
      XISP=THRVAC/(WDOPS(ITHR)*GZERO*PTKG)
      WDOT=(WDOMS(ITHR)*TNE(4,ITHR)+WDOUPP(ITHR))*PTKG
      GO TO 419
C
C     RCS SYSTEM
C
  450 THRVAC=FRCS(ITHR)*PTN
```

```
      ISP=0.
      IF(WDRCS(ITHR).LT..001) XISP=THRVAC/(WDRCS(ITHR)*GZERO*PTKG)
      WDOT=WDRCS(ITHR)*TNE(5,ITHR)*PTKG
      GO TO 419
C
C     FUTURE SYSTEM
C
      GO TO 500
C
C     NON-GIMBALLED THRUST AND PITCH MOMENT COMPONENTS
C
470   NENG=MMAX-MMIN+1
      IF(INT(TNE(N,ITHR)+.1).NE.NENG) MMAX=INT(TNE(N,ITHR)+.1)+MMIN-1
      DO 475 I=MMIN,MMAX
      TANP=TAN(ENGDAT(4,I)/RAD)
      TANY=TAN(ENGDAT(5,I)/RAD)
      TEMP=THRUST/SQRT(1.+TANP*TANP+TANY*TANY)
      TX=-TEMP*TANP
      TY=TEMP
      TZ=TEMP*TANY
      TCL(N)=TCL(N)+TY
      TVL(N)=TVL(N)+TX
      TLL(N)=TLL(N)+TZ
      TMZ=-TY*(XCG-ENGDAT(3,I))+TX*(YCG-ENGDAT(1,I))+TMZ
      IF(.NOT.QFLG.OR.A.NE.1) GO TO 471
      TXS=TXS+TX
      TYS=TYS+TY
      TZS=TZS+TZ
471   CONTINUE
      TXX=TXX+TX
      TYY=TYY+TY
      TZZ=TZZ+TZ
475   CONTINUE
      GO TO 500
C
C     MOMENT BALANCE EQUATIONS
C
480   IF(OUTTH(N).LT.1E0E0..OR.NTRG7.GT.0) GO TO 470
      THRUST=OUTTH(N)
      DX=XGPA-XCG
      DY=YGPA-YCG
      DZ=ABS(ZGPA)
      DYI=1./DY
      DZI=1./DZ
      DXDY=DX*DYI
      UDXDY2=1.+DX(Y*DXDY
      COSRHO=1./SQRT(UDXDY2)
      SINRHO=DXDY*COSRHO
      TMZ=AMZ+TMZ
      TANDY=-(AMX-AMY*DX(Y)*DYI/(THRUST*UDXDY2)
      TEMP=(AMX*DX(Y+AMY)*DZI/THRUST
      TEMP1=-TMZ*DYI/THRUST
      TANDP1=(TEMP1-TEMP1)*COSRHO
      TANDP2=(TEMP1+TEMP1)*COSRHO
      TO2=THRUST/2.
      TEMP2=TANDY*TANDY
      TY1=TO2/SQRT(1.+TANDP1*TANDP1+TEMP2)
      TY2=TO2/SQRT(1.+TANDP2*TANDP2+TEMP2)
      TX2=-TY2*TANDP2
      TX1=TX2-TY1*TANDP1
      TY1=TY1+TY2
      TXX=TX1*COSRHO+TY1*SINRHO+TXX
      TYY=-TX1*SINRHO+TY1*COSRHO+TYY
      TZZ=TY1*TANDY+TZZ
      TCL(N)=-TX1*SINRHO+TY1*COSRHO
      TVL(N)=TX1*COSRHO+TY1*SINRHO
      TLL(N)=TY1*TANDY
500   CONTINUE
      THMFA=TYY-FAA
```

```
      FANC=TXX-FAN
      FLATC=TZZ+SIDE
      FOM=SQRT(THMFA*THMFA+FANC*FANC+FLATC*FLATC)/XM/GZERO
      TXXX=TXX
      TYYY=TYY
      CVAR(7)=-XMOCT
C
C     COMPUTE LIFT AND DRAG
C
      CALPHA=COS(ALF)
      SALPHA=SIN(ALF)
      CBETA=COS(ALFY)
      SBETA=SIN(ALFY)
      CRAG=FAA*CALPHA+FAA*SALPHA
      LIFT=-FAA*SALPHA+FAA*CALPHA
C
      FZ=TZZ+SIDE
      FW=SCHIR*FANC-CCHIR*FZ
      FY=SCHIY*FW+CCHIY*THMFA
      FU=CCHIR*FANC+SCHIR*FZ
      FX=CCHIP*FU+SCHIF*FY
      FY=-SCHIP*FU+CCHIP*FY
      FZ=-CCHIY*FW+SCHIY*THMFA
   13 CVAR(1)=FX/XM+GX
      CVAR(2)=FY/XM+GY
      CVAR(3)=FZ/XM+GZ
      IF(INTEG7.LT.0) GO TO 603
      CVAR(1)=0.
      CVAR(2)=0.
      CVAR(3)=0.
  603 CONTINUE
      CVAR(4)=W
      CVAR(5)=U
      CVAR(6)=V
    6 IF(NFB.LE.13) GO TO 16
      VMAG=SQRT(W*W+U*U+V*V)
      VOGTG=(W*GX+U*GY+V*GZ)/VMAG
      CALP1=(W*CCHIP-U*SCHIP)*CCHIR/VMAG
      CALP3=W*SCHIF+U*CCHIP
      CALP2=(CALP3*CCHIY+V*SCHIY)/VMAG
      CALP3=(V*CCHIY-CALP3*SCHIY)*CCHIR/VMAG
      TS=OUTTH(1)+OUTTH(2)+OUTTH(3)+OUTTH(4)+OUTTH(5)+OUTTH(6)
      TV=SQRT(TXXX*TXX)+TYYY*TYYY+TZZ*TZZ)
      DVAR(14)=XK1*(XK2*GMO)**.5*(VR/XK4)**3.07*(1.-XM5/(XK6*PAT(12)+((
     *VR/XK3)**2)/XK7))
      CMASS=1./XM
      CVAR(16)=DMASS*(CUTTV(1)+OUTTV(2)+CUTTV(3)+CUTTV(4)+OUTTV(5)+
     *CUTTV(6))
      CVAR(15)=DVAR(15)-TS*DMASS
      CVAR(17)=(TS-TV)*DMASS
      IF(NFB.LE.17) GO TO 16
      CVAR(18)=-VOCTG
      CVAR(19)=(-TXXX*CALP1-TYYY*CALP2-TZZ*CALP3+TV)*DMASS
      CVAR(20)=(FAN*CALP1+FAA*CALP2-SIDE*CALP3)*CMASS
      CVAR(21)=0.
      IF(Q.GT.QMAX*PSFTM) DVAR(21)=Q*PSFFM-QMAX
      CVAR(22)=MODLMP(ITHR)
   16 CONTINUE
      IF(NMAX.NE.0.OR.CUMP.NE.1) GO TO 17
      CVALUE(1)=Q*ALF*PSFFM*RAD
      CVALUE(2)=Q*ALFY*PSFFM*RAD
      CVALUE(3)=Q*PSFFM
      CVALUE(4)=THMFA/XM/GZERO
      CVALUE(5)=(TZZ+SIDE)/XM/GZERO
      CVALUE(6)=FANC/XM/GZERO
      CVALUE(7)=SQRT(CVALUE(4)**2+CVALUE(5)**2+CVALUE(6)**2)
      CVALUE(8)=ZAF(14)
```

```
      17 CONTINUE
         IF(OREITR) RETURN
C
C        CALCULATE Q DOT
C        CALCULATE P(RHO)WRT ALT
         DRDR=FAT(N)
C        CALCULATE DDEX
         LMFC=FLAT*(2.-FLAT)/(RE*RE*UMF*UMF)
         DRTHE=STHE**3*UMFE*CTH/R2
         FHX=X/R-(X*CTH-R*A(1,2))*DRTHE
         FHY=Y/R-(Y*CTH-R*A(2,2))*DRTHE
         FHZ=Z/R-(Z*CTH-R*A(3,2))*DRTHE
C
         CMX=A12W
         CMY=A22W
         CMZ=A32W
         TEMPP=WXR(1)
         TEMP1=WXR(2)
         TEMPP2=WXR(3)
         TEMP3=TEMP*TEMP+TEMPP1*TEMP1+TEMPP2*TEMPP2
         F1=(TEMP1*OMZ-TEMPP2*OMY)/TEMP3
         F2=(TEMPP2*OMX-TEMPP *OMZ)/TEMP3
         F3=(TEMP *OMY-TEMPP1*OMX)/TEMP3
         TEMPP4=1./SQRT(TEMPP3)
C
         E1=TEMPP *TEMPP4
         E2=TEMPP1*TEMPP4
         E3=TEMPP2*TEMPP4
         PE1X=-E1*P1
         PE1Y=-E1*P2-CMZ*TEMPP4
         PE1Z=-E1*P3+CMY*TEMPP4
         PE2X=-E2*P1+CMZ*TEMPP4
         PE2Y=-E2*P2
         PE2Z=-E2*P3-CMX*TEMPP4
         PE3X=-E3*P1-CMY*TEMPP4
         PE3Y=-E3*P2+CMX*TEMPP4
         PE3Z=-E3*P3
C
         TEMPP5=1./R
         TEMPP6=X*TEMPP5
         TEMPP7=Y*TEMPP5
         TEMPP8=Z*TEMPP5
         PRXX=TEMPP5*(1.-TEMPP6*TEMPP6)
         PRXY=TEMPP5*(   -TEMPP6*TEMPP7)
         PRXZ=TEMPP5*(   -TEMPP6*TEMPP8)
         PRYY=TEMPP5*(1.-TEMPP7*TEMPP7)
         PRYZ=TEMPP5*(   -TEMPP7*TEMPP8)
         PRZZ=TEMPP5*(1.-TEMPP8*TEMPP8)
C
         TEMPP4=E3*TEMPP7-E2*TEMPP8
         TEMPP3=SAZW*E1+CAZW*TEMPP4
         FW1X=FWMDH*PHX*TEMPP3+WMAG*(SAZW*(PE1X-PAZWDH*PHX*TEMPP4)+CAZW*((PE1
        *X*Y-PE2X*Z)*TEMPP5+(E3*PRXY-E2*PRXZ)+PAZWDH*PHX*E1))
         FW1Y=FWMDH*PHY*TEMPP3+WMAG*(SAZW*(PE1Y-PAZWDH*PHY*TEMPP4)+CAZW*((PE1
        *Y*Y-PE2Y*Z)*TEMPP5+(E3*PRYY-E2*PRYZ)+PAZWDH*PHY*E1))
         FW1Z=FWMDH*PHZ*TEMPP3+WMAG*(SAZW*(PE1Z-PAZWDH*PHZ*TEMPP4)+CAZW*((PE1
        *Z*Y-PE2Z*Z)*TEMPP5+(E3*PRYZ-E2*PRZZ)+PAZWDH*PHZ*E1))
C
         TEMPP4=E1*TEMPP8-E3*TEMPP6
         TEMPP3=SAZW*E2+CAZW*TEMPP4
         FW2X=FWMDH*PHX*TEMPP3+WMAG*(SAZW*(PE2X-PAZWDH*PHX*TEMPP4)+CAZW*((PE1
        *X*Z-PE3X*X)*TEMPP5+(E1*PRXZ-E3*PRXX)+PAZWDH*PHX*E2))
         FW2Y=FWMDH*PHY*TEMPP3+WMAG*(SAZW*(PE2Y-PAZWDH*PHY*TEMPP4)+CAZW*((PE1
        *Y*Z-PE3Y*X)*TEMPP5+(E1*PRYZ-E3*PRXY)+PAZWDH*PHY*E2))
         FW2Z=FWMDH*PHZ*TEMPP3+WMAG*(SAZW*(PE2Z-PAZWDH*PHZ*TEMPP4)+CAZW*((PE1
        *Z*Z-PE3Z*X)*TEMPP5+(E1*PRZZ-E3*PRXZ)+PAZWDH*PHZ*E2))
C
         TEMPP4=E2*TEMPP6-E1*TEMPP7
         TEMPP3=SAZW*E3+CAZW*TEMPP4
```

```
      FW3X=FWMOH*PHX*TEMF3+WMAG*(SAZW*(PE3X-PAZWCH*PHX*TEMP4)+CAZW*((PE2
     *)*X-PE1X*Y)*TEMF5+(E2*PRXX-E1*PRXY)+PAZWOH*FHX*E3))
      FW3Y=FWMOH*PHY*TEMF3+WMAG*(SAZW*(PE3Y-PAZWCH*PHY*TEMP4)+CAZW*((PE2
     *Y*X-PE1Y*Y)*TEMF5+(E2*PRXY-E1*PRYY)+PAZWOH*FHY*E3))
      FW3Z=FWMOH*PHZ*TEMF3+WMAG*(SAZW*(PE3Z-PAZWCH*PHZ*TEMP4)+CAZW*((PE2
     *Z*X-PE1Z*Y)*TEMF5+(E2*PRXZ-E1*PRYZ)+PAZWOH*FHZ*E3))
C
      TEMP=1./VR
      FVRX=(SV*OMY-SU*OMZ)*TEMP
      FVRY=(SW*OMZ-SV*OMX)*TEMP
      FVRZ=(SU*OMX-SW*OMY)*TEMP
      IF(NWIND.EQ.0) GO TO 201
      FVRX=FVRX+(-SW*PW1X-SU*PW2X-SV*PW3X)*TEMP
      FVRY=FVRY+(-SW*PW1Y-SU*PW2Y-SV*PW3Y)*TEMP
      FVRZ=FVRZ+(-SW*PW1Z-SU*PW2Z-SV*PW3Z)*TEMP
  201 FVRW=SW*TEMP
      FVRU=SU*TEMP
      FVRV=SV*TEMP
C
      TEMP1=RHO*VR
      TEMP2=.5*VR*VR*CFOR
      CQCX(1)=TEMP1*PVRW
      CQCX(2)=TEMP1*PVRU
      CQCX(3)=TEMP1*PVRV
      CQCX(4)=TEMP1*PVRX+TEMP2*PHX
      CQCX(5)=TEMP1*PVRY+TEMP2*PHY
      CQCX(6)=TEMP1*PVRZ+TEMP2*PHZ
   10 CDOT=0.
      DO 11 I=1,6
   11 CDOT=CDOT+QQCX(I)*CVAR(I)
      RETURN
      END
```

```
      SUBROUTINE ACER1
C
C     EVALUATES THE TIME DEPENDENT VARIABLES
C
      CHARACTER*60 HEAD
      CHARACTER*6 MISION
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINM,
     *CTZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WD(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPFFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVAR(13),DELXDW(15,3),DELXDR(7,4),DELXD(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CTZ,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMA(15),BHMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITER,IWC,IXR,JUMP,KAT,KCYTAB,KFAGE,KS1,KWTA(4),LINES,
     *NBGET(4),NENEET(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PINM,PPSFLG(15),NSYST(15),ICONSW,IFTLS,IPOLY,KINDB,KRDERB,MISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),MSWCH(15)
      COMMON/AGEN4/QALPHA,QBETA,QCN,STH1,CTH1,XJX(3),XJV(3)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,APX,APY,APZ,
     *IM2,TPMFA,SICE,BPAE,AZW,FANC,FLATC,FOM,VISC
      COMMON/AVGGP/XGPA,YGPA,ZGPA
      COMMON/BODYF/TBCHT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,M1,M2,M5,
     *M6,M7,M8,M9,PSO,PS1,MSRW,AP(10),AY(10)
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,DL,CMLE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,FTN,FTNG,FSFTM,PFN,PFNG,PSFFM
      COMMON/FORINT/HBANM(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINE,KFOER,EU,EL,AYL,HMX,HMN,HAP
      COMMON/GARBGE/DX,DYI,DZI,DXDY,UDXDYI,UDXDY2,XCG,YCG,TX1,TX2,TY1,TY
     *2,ZHAT(3),TXS,TYS,TAU,SINRHO
      COMMON/LBM/TIMLBP(30),THRLBM(30),WCTLBP(30),PLBP(2),HLBM1,HLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/KP/KP,KY
      COMMON/MPS/TIMMFS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MFS1,MFS2,TGLIP,GLIMIT
      COMMON/OLOW/CLOW,WINT
      COMMON/OMSRCS/FCMS(15),FRCS(15),WCCMS(15),WCRCS(15),WDDUMP(15)
      COMMON/RICCNT/TTABLE(28),SIGMA,SIGTAB(20),P15,P16
      COMMON/XV/XV(4)
      DIMENSION CG(2),CGBL(15,2),PCG(2),ACG(2),BCG(2),CCG(2)
      EQUIVALENCE (XCG,CG(1)),(YCG,CG(2)),(TXCG(1),CGBL(1,1)),
     *(TYCG(1),CGBL(1,2))
      CHIP=0.
      CHIY=0.
      CHIR=0.
      SCHIY=0.
      CCHIY=1.
      IF(IHEAD.EQ.1) CHIP=PI
      IF(T.LE.TBEGR) CHIP=CHIRO
      IF(NROLL.EQ.1) CHIP=CHIRO+CHRDOT*(T-TBEGR)
      IF(JUMP.NE.1) CHIR=CHIRO+CHRDOT*(TENDR-TBEGR)
      SCHIR=SIN(CHIR)
      XMIAD=XMI
      CCHIR=COS(CHIR)
      IF(KS1.GE.2)GO TO 59
C**   GET HERE DURING INITIAL LIFT OFF
      PHIL=DPHIZ+OMEGA*(T-TZERO)
      STSP=STH1*SIN(PHIL)
      CPST=STH1*COS(PHIL)
      XHAT=A(1,1)*STSP+A(1,2)*CTH1+A(1,3)*CPST
      YHAT=              A(2,2)*CTH1+A(2,3)*CPST
      SCHIY=A(3,1)*STSP+A(3,2)*CTH1+A(3,3)*CPST
      YX=XHAT*(CCHIP-SCHIP)*SCHIR
      SCHIY=-XHAT*SCHIR-SCHIY*CCHIR
```

```
      CHIP=ATAN2(-YX,Y+A1)
      CHIY=ASIN(SCHIY)
      CHPBAS=CHIP
      CHYBAS=CHIY
      CHISAV=CHIP
      IF(IFACT.EQ.3) GO TO 38
      IF(SCHIY.LE.1.AND.SCHIY.GE.-1.)GO TO 111
      IF(SCHIY.LE.1.)GO TO 112
      CHIY=PI/2.
      GO TO 6
   99 IF(ITER.LT.MINH) GO TO 38
      CALL ACNTRO
      GO TO 6
   38 IF(IFACT.NE.3) GO TO 40
      CALL ANULG(1,MIS,20,T,TTABLE,SIGMA,SIGTAB,C,D)
      GO TO 100
   40 IF(ITER.GE.IPOLY) GO TO 39
      IF(IFACT.GT.1.AND.T.GT.TTILT) GO TO 101
      CALL CHIPOL(NP,XY,T-TLIFT,CHIP,AP,CHIY,AY)
      GO TO 6
   39 CALL CHIPOL(NP,XY,T-TPOLY,CHIP,AP,CHIY,AY)
      GO TO 6
  112 CHIY=1.5*PI
      GO TO 6
  111 CHIY=ASIN(SCHIY)
    6 SCHIY=SIN(CHIY)
      CCHIY=COS(CHIY)
    8 SCHIP=SIN(CHIP)
      CCHIP=COS(CHIP)
      CHISAV=CHIP
  101 XV(1)=SCHIP
      XV(2)=CCHIP
      XV(3)=SCHIY
      XV(4)=CCHIY
  100 XM=XMIAD+XMALG
      CELWT=WINT-XP
      IF(ITER.GT.NCEVR1(1)) DELWT=OLOW-XP
      IF(DELWT.LT..01)DELWT=.01
   33 CALL SPLINE(2,MS,IS,DELWT,TBOWT,CG,CGBL,PCG,1,MG,ACG,BCG,CCG)
      RETURN
      END
```

```
      SUBROUTINE AEBANK(A,NF,IM,I1,I2,NTMACH)
      COMMON/CCC/GRAPH,JC,DEMAND
      COMMON/SBLCCK/BLCCK(1500)
      DIMENSION A(1)
C
      DEFINE FILE E(65,1500,U,NOUM)
C
      NR=NF
      IN=0
      DO 1 K=1,6
      READ(E'NR,ERR=2) BLOCK
      NR=NR+1
      I=I2
      L=IM*I1*(I-1)
      DO 1 J=1,I1
      P=L+IM*(J-1)
      DO 1 K=1,NTMACH
      NK=K+P
      IN=IN+1
    1 A(IN)=BLOCK(NK)
      RETURN
C
    2 WRITE(6,3) NR
    3 FORMAT(1X,'ERROR IN READING RECORD NO.',I6)
      STOP
      END
```

```
      SUBROUTINE AEOSR
C
C     STORES THE STATE DURING THE FORWARD TRAJECTORY
C
      CHARACTER*60 HEAE
      CHARACTER*6 MISION
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),T2ERG,TLIFT,TTILT,TMINH,
     *CT2,TC,TL,XMAUG,INE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FA2,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,GHIF,SGHIP,CGHIP,GHIY,SGHIY,CGHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *WIV(25),DVARS(13),CELXOW(15,3),DELXOR(7,4),CELXC(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMAX(15),EHMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAO
      COMMON/AGEN3/ITER,IWO,IXR,JUMP,KAT,KCYTAB,KPAGE,KS1,KWTA(4),LINES,
     *BGET(4),NENEET(4),NMAX,NOEVNT(5),NBWD(15),NVNT,NWVNT,IHEAC,
     *PINH,PPSFLG(15),ASYST(15),ICONSW,IRTLS,IPOLY,KINOB,KRDERB,MISION,
     *IRTFLG,NROLL,NGEAST,IFACT,NOBASE,LSTGE(15),PSWGH(15)
      COMMON/FORINT/HBANH(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,HING,KRDER,EU,EL,AYL,HMX,HMN,HAP
      COMMON/LOADS/QLOAD(16),QTIME(16),QVALUE(8),SAVE(8,8)
      COMMON/TABS/TBL(101),WBL(101),UBL(101),VBL(101),XBL(101),YBL(101),
     *ZBL(101),XMBL(101)
C
C     SAVES STATE VARIABLE PTS. FROM FORWARD INTEGRATION
      IF(NMAX.NE.0.OR.NOPR.NE.1) GO TO 2
      CALL ADER1
      CALL ADER
      DO 11 I=1,8
      IF(QVALUE(I).GT.QLOAD(I)) GO TO 10
      QLOAD(I)=QVALUE(I)
      QTIME(I)=T
   10 K=I+8
      IF(QVALUE(I).LT.QLOAD(K)) GO TO 11
      QLOAD(K)=QVALUE(I)
      QTIME(K)=T
   11 CONTINUE
    2 CONTINUE
      IF(ITER.LT.MINH) RETURN
      IF(TL.GT.T)RETURN
      IXR=IXR+1
C**   SET 4 STATES AND TIME IN TABLES
      TBL(IXR)=T
      WBL(IXR)=W
      UBL(IXR)=U
      VBL(IXR)=V
      XBL(IXR)=X
      YBL(IXR)=Y
      ZBL(IXR)=Z
      XMBL(IXR)=XMIAO
    1 TL=T+STEP(ITER)-.2
      IF(IXR.LT.100) RETURN
C     WRITE OUT SAVED VALUES (4 STATES)
      WRITE(3)(TBL(I),WBL(I),UBL(I),VBL(I),XBL(I),YBL(I),ZBL(I),XMBL(I),
     *I=1,100)
      IXR=0
      RETURN
      END
```

```
$AEROIN
ALTBAS(31,2)=250000.,275000.,300000.,325000.,350000.,375000.,400000.,425000.,
             450000.,475000.,500000.,525000.,550000.,575000.,600000.,
BAXIAL(31,2)=15*-900.,
BNORM(31,2)=15*-492.,
BPIT(31,2)=15*53081.,
ALTBAS(31,3)=250000.,275000.,300000.,325000.,350000.,375000.,400000.,425000.,
             450000.,475000.,500000.,525000.,550000.,575000.,600000.,
BAXIAL(31,3)=15*-300.,
BNORM(31,3)=15*-164.,
BPIT(31,3)=15*17700.,
$END
```

```
      SUBROUTINE AFGRAC
C     COMPUTES CONSTRAINT ERRORS;SETS UP BACKWARD TRAJECTORY INITIALIZA
G     AND PRINTS TERMINAL SUMMARY
      CHARACTER*6  NAME(16),IUNIT(16),HEADER
      CHARACTER*6 MISIGN
      CHARACTER*60 HEAD
      DOUBLE PRECISION VALE,DSIREQ
      LOGICAL DEMAND
      COMMON/CGC/GRAPH,JG,DEMAND
      COMMON/AERODI/AEROD(30,16),AERO8(50,5,3),TCPACH(20),DELCA(20),
     *FNMONE(15),BKFAB1(50,2),BKFTHR(50,2),DELCN(20),CELCM(20)
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CTZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WO(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BG(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIP,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,SCHI,CCHI,STHL,CTHL,DPHI2,RITE,R,VR,XN,SH,SL,SV,Q,
     *XIV(25),DVARS(13),CELXOW(15,3),DELXOR(7,4),CELXC(15,7),WZERO,ALT,
     *XMACH,GDOT,FAA,FAN,A(3,3),CASE,CT2,UHF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMAX(15),BHMN,BSTEP(15),HMMB,HMXB,TPOLY,XMIAO
      COMMON/AGEN3/ITER,IWO,IXR,JUMP,KAT,KCYTAB,KPAGE,KS1,KWTA(4),LINES,
     *NBGCT(4),NENECT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *IANH,PFSFLG(15),ASYST(15),ICONSW,IRTLS,IPOLY,KIADB,KRDERB,MISICN,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/AFPRN/VI,GAP,C1,C3,XINC,XNOC,XMLB,THETG,THETA,PHI,RNGO,RNGA
     *A,AZI,AZRE
      COMMON/AUTO/NOB(40),SAVCP(40),XLAMB(40,10),CP2
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NOP1,NF(4)
      COMMON/BODY/IBDLK(15,2),IXCG(15,2),IYCG(15,2),XLEN(2),XREF(2),
     *YREF(2),KP(2),KY(2),AP(10,2),AY(10,2)
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,GM,CMUE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,FTN,PTNG,FSFTM,PFN,PFKG,PSFFM
      COMMON/CONTRC/TTEL(30),CPTBL(30),CYTBL(30),TORL(91),CPCTBL(91),
     *CYOTBL(91)
      COMMON/CSTOR/CSTVAR(10)
      COMMON/DRMO/DRHO(9),E1,WIPPB,WGNU
      COMMON/ENPUT/PSIRST(6,4),HEADER(20),NVRST(4),KCCRES(6,4),LAST
      COMMON/FORINT/HEANH(2),T,TT,W,U,V,X,Y,Z,XM1,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,HMX,HMN,HAN
      COMMON/LOADS/QLCAD(16),QTIME(16),QVALUE(8),SAVE(8,8)
      COMMON/LPRINT/LPRINT,INMAX,PREOP,CY,TOL
      COMMON/MPS/TIMMPS(15),FTBL(15),THROT(15),GLIM(15),TAUALT(15),TAUTBL
     *(5),MPS1,MPS2,TELIM,GLIMIT
      COMMON/OMAX/OMAX,QFLG,OPEN,MAXQ
      COMMON/REST/LSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NOPAR,WIBT(15)
      COMMON/RESTX/RESTX(10,4)
      COMMON/TAULIM/TALLIM,NTAU,NBETA,TIMTAU
      COMMON/VIC/VIC(2),XK1(2),XK2(2),XK3(2),XK4(2),LX
      COMMON/YLI/YLINT(7,10),PHITE(10),PSIREQ(10),KCCPHI(10),PSTR(10)
      DIMENSION GOT(10),ASK(10),PERC(10)
      DATA NAME/6H G ALP,6HHA    ,6H Q BET,6HA     ,6H Q     ,6H      ,6H
     * LONG ,6HACC    ,6H LAT A,6HCC     ,6H NORM ,6HACC    ,6H TOTAL,6H AC
     *C  ,6H STAG ,6HPEAT  /
      DATA IUNIT/6HLB-SEC,6H/FT**2,6HLB-SEC,6H/FT**2,6HLB/F,6HT**2  ,
     *6H      G,6H'S     ,6H      G,6H'S     ,6H      E,6H'S     ,6H      G,
     *6H'S    ,6H  BTU/,6HFT**2 /
C**   SET LAST STATE POINT IN TABLES
C     EVALUATE TERMINAL CONDITIONS AND INITIALIZE LAMBDAS
C*****NOSS AT THIS POINT IS EQUAL TO REGULAR SETS-SET BY ANEWCH
      IF(I.NVNT(I).EQ.1
      CSTVAR(1)=W
      CSTVAR(2)=U
      CSTVAR(3)=V
      CSTVAR(4)=X
      CSTVAR(5)=Y
```

```
      CSTVAR(6)=Z
      CSTVAR(7)=XM
      CSTVAR(8)=T
      CSTVAR(9)=ZAF(7)
      CSTVAR(10)=0.
      CALL ACSTOP(C,VALE,YLINT,I)
      LK=C
      DO 20 I=1,7
      DO 20 J=1,10
   20 YLINT(I,J)=0.
      DO 4 I=1,NOSS
      KL=KCDPHI(I)
      IF(KL.EQ.14.OR.KL.GE.18.AND.KL.LE.20) LK=1
      CALL ACSTOP(KL,VALE,YLINT,I)
      IF(I.EQ.1)GO TO 3
      CSIREC=PSIREC(I-1)
      FHITE(I)=VALE-CSIREC
      GO TO 4
    3 FHITE(I)=VALE
    4 CONTINUE
      JA=NOSS
      KR=R
      DO 50 L=1,4
      K=5-L
      IF(NVRST(K).EQ.C) GO TO 50
C**** IF NVRST.NE.C EVALUATE RESTART TERMINAL CONC AND INIT LAMS
      KR=R
      DO 16 I=1,10
   16 CSTVAR(I)=RESTX(I,K)
      CALL ACSTOP(C,VALE,YLINT,I)
      NCN=NCNRS(K)
      DO 18 I=1,NCN
      JA=JA+1
      KL=KCDRES(I,K)
      IF(KL.EQ.14.OR.KL.GE.18.AND.KL.LE.20) LK=2
      CALL ACSTOP(KL,VALE,YLINT,JA)
      CSIREC=PSIRST(I,K)
   18 FHITE(JA)=VALE-CSIREC
   50 CONTINUE
      R=RR
C**   WRITE OUT TEST MATRIX
   15 IF(LSB.EQ.1)RETURN
      N=2*(NOSS+NTCN)
      IF(LSB.NE.C)GO TO 1
      WRITE(6,56) CASE,HEAD
   56 FORMAT(1H1,12HMASTRE CASE=,F11.4,10X,A60//)
      GO TO 2
    1 WRITE(6,17) CASE,HEAD
   17 FORMAT(//1X,12HMASTRE CASE=,F11.4,10X,A60//)
    2 CONTINUE
      WRITE(6,5)(HEADER(I),I=1,K)
    5 FORMAT(//,1X,6H      ,20A6)
      IF(DEBAND.AND.NCP1.EQ.0) WRITE(6,52)(HEADER(I),I=1,K)
      ASK(1)=E1*XJEXT*(WIPPB-WGNU)
      JA=NOSS+NTCN-1
      ASK(1)=ASK(1)-PFEDF
      DO 6 I=1,JA
    6 ASK(I+1)=BRHC(I)
      JA=JA+1
      DO 7 I=1,JA
      GOT(I)=PHITE(I)-FSTR(I)
    7 PERC(I)=GOT(I)/ASK(I)*100.
      WRITE(J0,8)(FSTR(I),I=1,JA)
      WRITE(J0,9)(PHITE(I),I=1,JA)
      WRITE(J0,10)(ASK(I),I=1,JA)
      WRITE(J0,11)(GOT(I),I=1,JA)
      WRITE(J0,12)(PERC(I),I=1,JA)
    8 FORMAT(1X,6HOLD    ,10E12.5)
    9 FORMAT(1X,6HNEW    ,10E12.5)
```

```
   10 FORMAT(1X,6HASK    ,10E12.5)
   11 FORMAT(1X,6HGOT    ,10E12.5)
   12 FORMAT(1X,6HFCNT   ,10E12.5//////)
      IF(.NOT.DEMAND)GO TO 1234
      NCASE=CASE
      ICASE=INT((CASE-NCASE)*10000.)
      WRITE(6,51) ICASE,(GOT(I),I=1,JA)
   51 FORMAT(2X,I3,2X,10E12.5)
   52 FORMAT(//,1X,4HCASE,2X,20A6)
C-----------------------------------------------------------------------
      WRITE(10)JA,K,CASE,HEADER,PHITE
C-----------------------------------------------------------------------
 1234 CONTINUE
      DO 13 I=1,JA
   13 FSTR(I)=PHITE(I)
      IF(LSE.NE.0)RETURN
      WRITE(60,31) AP,AY
   31 FORMAT(//24H POLYNOMIAL COEFFICIENTS/1X,2HAP,5E15.8,3(/3X,5E15.8)
     */1X,2HAY,5E15.8,3(/3X,5E15.8))
      IF(LPRINT.NE.2.AND.NMAX.NE.0) RETURN
      DO 14 I=1,15
   14 SAVCP(I)=TAUT(I)
      SAVCP(16)=WZERO/.45359237
      SAVCP(17)=AA*RAD
      DO 60 I=1,5
      SAVCP(I+17)=CPTBL(I)
   60 SAVCP(I+25)=CYTBL(I)
      SAVCP(23)=CPTBL(6)
      SAVCP(24)=CPTBL(7)
      SAVCP(25)=CPTBL(8)
      SAVCP(31)=TIMPS(NTAU)
      SAVCP(32)=TIMTAU
      SAVCP(33)=FTBL(NTAU+1)
      SAVCP(34)=AEROD(NBETA,15)
      SAVCP(35)=AEROD(NBETA+1,15)
      SAVCP(36)=AEROD(NBETA+2,15)
      WRITE(60,30) (SAVCP(I),I=1,36)
   30 FORMAT(/1X,1HPARAMETERS/4X,4HTAUT/8X,8E15.8/8X,7E15.8/5X,3HWO1/
     *6X,E15.8/6X,2HA2/8X,E15.8/3X,5HCPTBL/8X,8E15.8/3X,5HCYTBL/8X,5E15.
     *8/3X,19HTHROTTLE PARAMETERS/8X,3E15.8/3X,16HSIDESLIP CONTROL/8X,
     *3E15.8)
      SAVCP(16)=SAVCP(16)*.45359237
      RNGAN=RNGAN*RAD
      HIS=R*VI*COS(GAM)
      GAM=GAM*RAD
      XMLB=XM*PFKG
      XINC=XINC*RAD
      AZI=AZI*RAD
      P=HIS**2/CMUE
      CJD=CMUE/ABS(C3)
      E=SQRT(ABS(1.-(P/CJD)))
      IF(E.GT..00005)GO TO 34
      E=0.
   34 RA=CJD*(1.+E)
      RP=CJD*(1.-E)
      HA=(RA-RE)/1852.
      HP=(RP-RE)/1852.
      WRITE(60,37)
   37 FORMAT(1H1)
      WRITE(1) HEAD,T,RNGO,RNGAN,VI,R,GAM,XINC,XNGD,SAVCP(17),THETG,
     *THETA,PHI,AZI,C1,XMLB,C3,E,RA,RP,HA,HP
      WRITE(60,38) HEAD
   38 FORMAT(29X,A60//)
      WRITE(60,36)T,RNGO,RNGAN,VI,R,GAM,XINC,XNGD,SAVCP(17),THETG,THETA,
     *PHI,AZI,C1,XMLB,C3
      SAVCP(17)=AA
   36 FORMAT(5X,4HTIME,E15.8,1X,3HSEC,12X,5HRANGE,E15.8,1X,1HM,10X,
```

```
     *11HRANGE ANGLE,E15.8,1X,3HDEG///1X,8HINERTIAL,E15.8,1X,5HM/SEC,9X,
     *6HRADIUS,E15.8,1X,1HM,11X,6HFLIGHT,4X,E15.8,1X,3HDEG/1X,8HVELOCITY
     *,62X,10HPATH ANGLE///5X,4HINCL,E15.8,1X,3HDEG,13X,4HDES.,E15.8,1X,
     *3HDEG,12X,6HFLIGHT,1X,E15.8,1X,3HDEG/40X,4HNODE,30X,7HAZIMUTH///
     *4X,5HGCLAT,E15.8,1X,3HDEG,12X,5HGCLAT,E15.8,1X,3HDEG,15X,4HLONG,
     *E15.8,1X,3HDEG///1X,8HINERTIAL,E15.8,1X,3HDEG,14X,2HC1,1X,E15.8,
     *1X,8HM**2/SEC/2X,7HAZIMUTH///3X,6HWEIGHT,E15.8,1X,3HLBS,14X,2HC3,
     *1X,E15.8,1X,11HM**2/SEC**2///)
      WRITE(JO,40)E,RA,RP,HA,HP
   40 FORMAT(1X,28HKEPLERIAN ORBITAL PARAMETERS//1X,8HECNTRICY,E15.8,15X
     *,6HAPOGEE,E15.8,1X,1HM,14X,7HPERIGEE,E15.8,1X,1HM/38X,6HRADIUS,31X
     *,6HRADIUS///3X,6HAPOGEE,E15.8,1X,2HNM,11X,7HPERIGEE,E15.8,1X,2HNM/
     *3X,6HHEIGHT,29X,6HHEIGHT)
      CALL SUMARY
      IF(JUMP.NE.1) RETURN
      WRITE(JO,37)
      WRITE(JO,38) HEAD
      WRITE(JO,41)
      DO 39 I=1,8
      N=2*I-1
   39 WRITE(JO,42)NAME(N),NAME(N+1),(SAVE(K,I),K=1,8),IUNIT(N)
     *,IUNIT(N+1)
      WRITE(2) HEAD,NAME,SAVE,IUNIT
   41 FORMAT(///42X,11HFIRST STAGE29X,12HSECOND STAGE/34X,7HMINIMUM11X,
     *7HMAXIMUM16X,7HMINIMUM11X,7HMAXIMUM/9X,14HLOAD INDICATOR6X,2(1X,
     *4HTIME5X,5HVALUE3X),5X,2(1X,4HTIME5X,5HVALUE3X),10X,5HUNITS//)
   42 FORMAT(10X,2A6,7X,2(F6.2,2X,F8.2,2X),5X,2(F6.2,2X,F8.2,2X),6X,2A6/
     *)
      RETURN
      END
```

```
      SUBROUTINE AFORGN
C
C     CONTROLS THE SETUP LOGIC FOR THE FORWARD TRAJECTORY AND CALLS THE
C     INTEGRATION PACKAGE
      CHARACTER*60 HEAG
      CHARACTER*6 PISICN
      LOGICAL GLIMIT,GFLG
      COMMON/CCC/GRAPH,JC,CEMAND
      COMMON/AGEN1/TIPE(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CT2,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WG(15),WJET(15),
     *EAB,PRINT(15),STEP(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHROOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCPIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFT,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTPE,R,VR,XM,SW,SL,SV,Q,
     *XIV(25),DVARS(13),DELXOW(15,3),DELXOR(7,4),EELXO(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *CYL,BEL,BHMA(15),BHMN,BSTEP(15),HNMB,HMXB,TPOLY,XMIAB
      COMMON/AGEN3/ITPF,IWO,IXR,JUMP,KAT,KCYTAB,KPAGE,KS1,KWTA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWO(15),NVNT,NWVNT,IHEAG,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINDB,KRDERB,PISION,
     *IRTFLE,NROLL,NGCAST,IFAGT,NOBASE,LSTGE(15),PSWGH(15)
      COMMON/AGEN4/CALFHA,QBETA,QCN,STH1,CTH1,XJX(3),XJV(3)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,THX,TYY,TZZ,AMX,AMY,AMZ,
     *IMZ,TPPFA,SICE,BPAE,AZW,FANC,FLATC,FOM,VISC
      COMMON/ASTOR/ASTCR(4,67)
      COMMON/AUTO/KOB(40),SAVCP(40),XLAMB(40,10),CP2
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NOM1,NP(4)
      COMMON/BODY/ATBCLT(15,2),ATXCG(15,2),ATYCG(15,2),AXLEN(2),AXREF(2)
     *,AYREF(2),KPI(2),KYI(2),API(10,2),AYI(10,2)
      COMMON/BODYF/TBCLT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,M1,M2,M5,
     *M6,M7,M8,M9,M50,M51,MSRW,AP(10),AY(10)
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,DL,CMUE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,PTN,PTKG,PSFTM,PFN,PFKG,PSFFM
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/FORINT/HEANK(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,HPX,HPN,HAP
      COMMON/JJUMP/JJLPP
      COMMON/HEAT/PAXHAT
      COMMON/IMACH/IMACH
      COMMON/IPR/IPR,WFMX,ILAST,KOT(15),FRK4,PRTCT
      COMMON/ITHRO/ITHRO
      COMMON/JUMPST/JLPPST,TSAVET,WZERST
      COMMON/LOADS/QLOAD(16),QTIME(16),QVALUE(8),SAVE(8,8)
      COMMON/LBM/TIMLBM(30),THRLBM(30),WCTLBM(30),PLBM(2),MLBM1,MLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MPS1,MPS2,TCLIP,GLIMIT
      COMMON/MLAST/MLAST,MMMM,KKKK
      COMMON/NALTW/NALTW
      COMMON/NTILT/NTILT
      COMMON/OLOW/CLOW,WINT
      COMMON/OMAX/CMAX,QFLG,QPEN,MAXQ
      COMMON/REST/STF,NCNRS(14),NTGN,TR(4),LSB,NFC,NOPAR,WIGT(15)
      COMMON/RICONT/TTABLE(20),SIGMA,SIGTAB(20),M15,M16
      COMMON/SRM/SRMTTB(250),SRMFTB(250),SRMWOT(250),SRMAE(2),MSRB
      COMMON/STAUT/STALT(15)
      COMMON/TAULIP/TALLIM,NTAU,NBETA,TIMTAU
      EXTERNAL  AXFRT,ATILT,AQMAX,GLIMT,AWDEV,AECSR,ALIFT,ATHRO
      MAXHAT=0
      CPEN=0.
      IWO=0
      KCYTAB=1
      IMACH=0
      CHIBAS=0.
      CHPBAS=0.
```

```
      CHYBAS=0.
      ATILT=0
      M15=1
      M16=1
C     CALCULATE A MATRIX
      SA=SIN(AA)
      CA=COS(AA)
      A(1,1)=SA
      A(2,1)=0.
      A(3,1)=CA
      A(1,2)=CA*STH1
      A(2,2)=CTH1
      A(3,2)=-SA*STH1
      A(1,3)=-CA*CTH1
      A(2,3)=STH1
      A(3,3)=SA*CTH1
      DO 14 I=1,16
      CLOAD(I)=0.
   14 CTIME(I)=0.
      NJUM=0
      IF(JJLPP.EQ.0.OR.NMAX.EQ.0.OR.JUMPST.EQ.0.OR.LSP.EQ.1) GO TO 60
      JUMPS=JUMP
      JUMP=JUMPST
      KS2=KS1
      KS1=2
      WZEROS=WZERO
      TZEROS=TZERO
      TZERO=TSAVET
      TLIFTS=TLIFT
      TLIFT=TLIFT+TZERO
      WZERO=WZERST
      CNOM=CMAX
      ELOWS=ELOW
      CLOWS=CLOW
      CHIP=CFOTBL(1)
      CHIY=CYOTBL(1)
      NJUM=1
   60 CONTINUE
C
      A12W=A(1,2)*CMEGA
      A22W=A(2,2)*CMEGA
      A32W=A(3,2)*CMEGA
      IF(JUMP.NE.1 .AND. VIV(7).LT.1.)GO TO 41
      XZ=A(1,1)*XJX(1)+A(1,2)*XJX(2)+A(1,3)*XJX(3)
      YZ=A(2,1)*XJX(1)+A(2,2)*XJX(2)+A(2,3)*XJX(3)
      ZZ=A(3,1)*XJX(1)+A(3,2)*XJX(2)+A(3,3)*XJX(3)
      WZ=A(1,1)*XJV(1)+A(1,2)*XJV(2)+A(1,3)*XJV(3)
      LZ=A(2,1)*XJV(1)+A(2,2)*XJV(2)+A(2,3)*XJV(3)
      NZ=A(3,1)*XJV(1)+A(3,2)*XJV(2)+A(3,3)*XJV(3)
      GO TO 99
   41 W=VIV(1)
      L=VIV(2)
      N=VIV(3)
      X=VIV(4)
      Y=VIV(5)
      Z=VIV(6)
   99 CONTINUE
      DO 42 I=1,18
   42 ZAP(I)=VIV(I+7)
      LLOSOL=5
      IF(KIND.NE.2)LLOSOL=3*(KORDER+2)+4
C **  KORDER CAN NOT BE GREATER THAN 5
C     INITIALIZE TABLE LOOK-UP INDICES
      MLAST=0
      MMMM=1
      MKKK=2
      MS1=1
```

```
      IF(JUMP.GT.1)MS1=2
      LINES=6
      IL=0.
      CDOT=.1
      IWC=0
      NFB=7
      IF(JUMP.NE.1)TO=TZERO
      IF(JUMP.NE.1)GO TO 102
      W=WZ
      U=UZ
      V=VZ
      X=XZ
      Y=YZ
      Z=ZZ
      DO 101 I=1,6
      ZAP(I)=PROP(I)*PTKE
      ZAP(I+6)=0.
  101 ZAP(I+12)=0.
      ZAP(1)=ZAP(1)-(ASTOR(1,64)+ASTOR(1,65))*PTKE
  102 CONTINUE
      WINT=WZERO
C
C     TAUT IS DURATION TIME OF THRUST EVENTS
C
      XMAUG=WJET(JUMP)
      TIME(1,JUMP)=TZERO
      DO 1 I=JUMP,NVNT
      TIME(1,I+1)=TIME(1,I)+TAUT(I)
    1 XMAUG=XMAUG+WJET(I)
      IWOJ=0
      J=1
    2 J=NOWE(I)
      IF(J.GE.JUMP) GO TO 3
      IF(I.GE.NWVNT) GO TO 5
      I=I+1
      GO TO 2
    3 IWOJ=I
      DO 4 I=IWOJ,NWVNT
      J=NOWE(I)
      TIME(2,I)=TIME(1,J)+TAUW(I)
    4 XMAUG=XMAUG+WB(I)
    5 XM=WZERO
      XMI=XP-XMAUG
      XMIAD=XMI
      T=TZERO
      NROLL=0
      IF(ABS(CHRDOT).LT.1.E-04) NROLL=-1
      IF(NROLL.EQ.-1.AND.IHEAD.EQ.0) FAZ=AA+PI
      IF(NROLL.EQ.-1.AND.IHEAD.NE.0) FAZ=AA
      IF(FAZ.GT.2.*PI) FAZ=FAZ-2.*PI
      CHIRO=PI-FAZ+AA
      IF(CHIRO.GE.2.*PI) CHIRO=CHIRO-2.*PI
      IF(NROLL.NE.0) GO TO 13
      CHIRF=0.
      IF(IHEAD.NE.0) CHIRF=PI
      N=0
      TBEGR=TCHIR
      TENDR=0.
      TENDO=TENDR-TBEGR
   19 IF(CHIRO.GT.CHIRF) GO TO 20
      IF(CHRDOT.LT.0.) TENDR=(-2.*PI+CHIRF-CHIRO)/CHRDOT
      IF(CHRDOT.GT.0.) TENDR=(CHIRF-CHIRO)/CHRDOT
      GO TO 21
   20 IF(CHRDOT.LT.0.) TENDR=(CHIRF-CHIRO)/CHRDOT
      IF(CHRDOT.GT.0.) TENDR=(2.*PI+CHIRF-CHIRO)/CHRDOT
   21 IF(TENDO.LE.0..OR.TENDR.LT.TENDO+PI/(2.*ABS(CHRDOT))) GO TO 29
      N=N+1
      CHRDOT=-CHRDOT
```

```
      IF(N.EQ.2) STOP
      GO TO 19
   29 CONTINUE
      TENDR=TENDR+TBEGR
   13 CONTINUE
      CHIR=CHIRO
      DO 7 I=1,10
      AP(I)=0.
      AY(I)=0.
      DO 7 J=1,2
      API(I,J)=0.
    7 AYI(I,J)=0.
C
C     TRIGGERS 1-PRINT,2-THRUST EVENT,3-VR CUTOFF,4-TILT,5-ALT,6-Q MAX,
C     7-FUEL CUTOFF,9-G-LIMIT,12-WT DROP,13-MPS THROTTLE TABLE
      NOTRG=7
C
C *1* PRINT TRIGGER
      NTRG1=-1
      IF(NMAX.EQ.0)NTRG1=1
      TV1=AINT(T)+PRINT(JUMP)
C
C *4* TILT TRIGGER
    6 NTRG4=2
      IF(JUMP.GT.1)NTRG4=-2
      TV4=TLIFT
      IF(TBEGR.LE.TLIFT.AND.NROLL.EQ.0) TV4=TBEGR
      IF(IFACT.EQ.3) TV4=TLIFT
C
C *6* MAX Q TRIGGER
      NTRG6=-1
      TV6=0.
C
C*12* WEIGHT DROP EVENT TRIGGER(REQ'D IN MIN H)
      NTRG12=2
      IF(IWEQ.LE.0) NTRG12=-2
      TV12=TIME(2,IWDJ)
C
C
C*13* SSME THROTTLE TABLE TRIGGER
C
      NTRG13=-1
      IF(NTAU.EQ.0) GO TO 40
      FTBL(NTAU+2)=FTBL(NTAU+1)
      TIMMPS(NTAU+1)=TIMMPS(NTAU)+(FTBL(NTAU)-FTBL(NTAU+1))/TAULIM
      TIMMPS(NTAU+2)=TIMMPS(NTAU+1)+50.
      TIMMPS(NTAU+3)=TIMMPS(NTAU+2)+4.
   40 CONTINUE
C
      GLIMIT=.FALSE.
      CFLG=.FALSE.
      NF8=14
      IF(IPR.NE.0) NF8=17
      IF(NMAX.EQ.0) NF8=22
C
C     INITIALIZE THRUST,STEP,MDOT,ETC.
      ITHR=JUMP-1
      NTRG7=-2
      CALL ATHREV
      IF(ITHR.EQ.MINH.OR.JUMP.GT.NOEVNT(1)) GO TO 8
C
      CALL ADER1
      CALL ADER
      IF(FOM.GE.1.) GO TO 30
      TV7=1.
      NTRG7=2
      CALL APRTN(3)
      GO TO 31
   30 CALL APRTN(2)
   31 CONTINUE
```

```
      IF(JUPP.NE.1) NTRG7=-2
C
      IF(LSE.EQ.1) GO TO 110
C
      LOAD=6
      IF(NMAX.EQ.0.AND.JUMP.EQ.1) LOAD=2
      CALL DESOLV(25,NF8,ADER1,ADER,AEOSR,LOAD,KERR,HNM,HBANK,FSAVE,KIND
     *,LLDSOL,EU,EL,HMX,HMN,AYL,NOTRG,
     * NTRG1,AXPRT,T,TV1,     NTRG2,ATHREV,T,TV2,  NTRG4,ATILT,T,TV4,
     * NTRG6,AQMAX,QDOT,TV6, NTRG12,AWDEV,T,TV12, NTRG7,ALIFT,FOM,TV7,
     *NTRG13,ATHRO,T,TV13,  0,0,0,0,   0,0,0,0,   0,0,0,0,   0,0,0,0)
C
      GO TO 111
  110 CALL DESOLV(25,NF8,ADER1,ADER,0,6,KERR,HNM,HBANK,FSAVE,KIND,LLDSOL
     *,EU,EL,HMX,HMN,AYL,6,
     * NTRG2,ATHREV,T,TV2,  NTRG4,ATILT,T,TV4,  NTRG6,AQMAX,QDOT,TV6,
     *NTRG13,AWDEV,T,TV12, NTRG7,ALIFT,FOM,TV7, NTRG13,ATHRO,T,TV13,
     *C,C,0,C,   0,0,C,0,   0,0,0,0,   C,0,C,0,   0,C,0,C)
C
  111 IF(KERR.NE.0) GO TO 61
    8 IF(LSB.EQ.1.AND.AALTW.EQ.0) RETURN
      IF(JUPP.LE.NCEVNT(1)) GO TO 10
      DO 9 I=1,4
      IF(NP(I).EQ.0) GO TO 9
      IF(JUPP.GE.NEND(I)) KCYTAB=I+1
      IF(KCYTAB.GT.4) KCYTAB=4
    9 CONTINUE
   10 IF(LSE.EQ.1) GO TO 112
C
      IXR=0
      NOTRG=4
      CALL DESOLV(25,NF8,ADER1,ADER,AEOSR,2,KERR,HNM,HBANK,FSAVE,KIND,
     *LLDSOL,EU,EL,HMX,HMN,AYL,NOTRG,
     *NTRG1,AXPRT,T,TV1, NTRG2,ATHREV,T,TV2, NTRG9,GLIMT,T,TV9,
     *NTRG13,AWDEV,T,TV12, 0,0,0,0,   0,0,0,0,   0,0,0,0,   0,0,0,0,
     *C,C,0,C,   0,0,C,0,   0,0,0,0)
C
      IF(NMAX.NE.0.OR.ITER.NE.NCOAST) GO TO 114
      CALL ADER1
      CALL ADER
      CALL AERTN(6)
      AVNT=AVNT+1
      GO TO 10
  114 IF(KJUP.EQ.0) GO TO 113
      JUMP=JUMPS
      NS1=NS2
      LZERO=LZEROS
      TZERO=TZEROS
      TLIFT=TLIFTS
      ELOW=ELOWS
      CLOW=CLOWS
      GO TO 113
C
  112 CALL DESOLV(25,NF8,ADER1,ADER,0,6,KERR,HNM,HBANK,FSAVE,KIND,LLDSOL
     *,EU,EL,HMX,HMN,AYL,3,
     *NTRG2,ATHREV,T,TV2,NTRG9,GLIMT,T,TV9,NTRG12,AWDEV,T,TV12,
     *C,C,0,C,   0,0,C,0,   0,0,0,0,   C,0,0,0,   C,C,C,0,   C,0,C,C,
     *C,C,0,C,   0,0,C,0)
  113 IF(KERR.NE.0) GO TO 61
      RETURN
   61 CONTINUE
      WRITE(6,0,12)
   12 FORMAT(//1X,25HINTEGRATION ERROR. DUMPED)
      STOP
      END
```

```
      SUBROUTINE AGEO(N)
C
C     CALCULATES THE GRAVITATIONAL PARAMETERS FOR FORWARD AND BACKWARD
C     TRAJECTORIES
C           N          = 1,FORWARD TRAJECTORY;2,BACKWARD TRAJECTORY
C
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHI2,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VI(25),DVARS(13),DELXDW(15,3),DELXDR(7,4),DELXD(15,7),WZERO,ALT,
     *MACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,DEL,BHMA(15),BHMN,BSTER(15),HNMB,HMXB,TPOLY,XMIAD
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,D,CMLE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,PTN,PTMG,PSFTM,PFN,PFKG,PSFFM
      COMMON/GRAV/ETO,E11,G22,G23,G33
      NEEDS CTH,CT2,    RE,R,CMUE,J,H,D,R2
C
C     GET OBLATE EARTH GRAVITY TERMS
      R2=R*R
      RER=RE/R
      RER2=RER*RER
      CJR=CJ*RER2
      HR=H*RER2*RER
      DJR=D*RER2*RER2
      CMT=CMUE/(R2*R)
      E11=-CMT*(1.+CJR*(1.-5.*CT2)+HR*(3.-7.*CT2)*CTH
     1     +DJR*(.42857143-(6.-9.*CT2)*CT2))
      IF(N.NE.1)GO TO 1
      ETC=R*CMT*(2.*CJR*CTH-HR*(.6-3.*CT2)+DJR*(1.7142857-4.*CT2)*CTH)
      RETURN
    1 E22=-(3./R2)*(E11)-(MT*(CJR*(.66666667-6.66666667*CT2)
     1      +HR*(4.-14.*CT2)*CTH+DJR*(.57142857-(12.-24.*CT2)*CT2)))
      E23=(CMT/R)*(10.*CJR*CTH-HR*(3.-21.*CT2)+DJR*(12.-36.*CT2)*CTH)
      E33=-CMT*     (2.*CJR*6. *HR*CTH          +DJR*(1.7142857-12.*CT2))
      RETURN
      END
```

```
      SUBROUTINE AINIT
C
C     READS INPUT DATA;PERFORMS INITIALIZATION
      LOGICAL DEMAND,GRAPH
      CHARACTER*60 MGAC,CN2,BLANK
      CHARACTER*6 MISSCN,MISION,BCD,HEADER,ATMOSN
      CHARACTER*36 IAR
      COMMON/CCC/GRAPH,JC,DEMAND
      COMMON/AERO/NTALFH(2),NTBETA(2),NTMACH,TALFH(11,2),TALYH(11,2),
     *IXMAC(28),CAA(15,11,6),CNA(15,11,6),CMA(15,11,6),CYA(15,11,6),
     *CXA(15,11,6),CLA(15,11,6)
      COMMON/AERO01/AERO0(30,16),AERO8(50,5,3),TCPACH(20),DELCA(20),
     *FNPONE(15),NFAB1(50,2),BKFTHR(50,2),DELCN(20),CELCM(20)
      COMMON/AGEN1/TIME(12,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *ETZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WB(15),WJET(15),
     *HEAD,FRINT(15),STEP(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHRB,CHIRB,FA22,BC(3,2),CBAXIL,TCHIR,VRCUT,FPFFAG,GNVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),OVARS(13),CELXOW(15,3),DELXOR(7,4),CELXC(15,7),WZERO,ALT,
     *XMACH,GDOT,FAA,FAN,A(3,3),CASE,GT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMA(15),BHMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITHR,IWO,IXR,JUMP,KAT,KGYTAB,KPAGE,KSI,KUTA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PINH,PPSFLG(15),ASYST(15),ICONSW,IRTLS,IPOLY,XINOB,KRDERB,MISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/AGEN4/GALPHA,GBETA,QCN,STH1,CTH1,XJX(3),XJV(3)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,APX,APY,AMZ,
     *TM2,TIPFA,STGE,PPAG,A2W,FANC,FLATC,FOM,VISC
      COMMON/ALAT/ALAT,ALONG,ALTLS,NTABLE
      COMMON/ARDC/FAT(14),ATMOSN
      COMMON/ASTCR/ASTCR(4,67)
      COMMON/AUTO/KOB(40),SAVGP(40),XLAMB(40,10),CP2
      COMMON/BODY/TBDLT(15,2),TZCG(15,2),TXCG(15,2),XLEN(2),ZREF(2),
     *XREF(2),XP(2),XI(2),AP(10,2),AY(10,2)
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NCM1,NF(4)
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,G0,CMLE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,PTN,PTKG,PSFTM,PFN,PFKG,PSFFP
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/DELFAB/DALT(25),DFAB(25),M3C,M31,ITCL(15)
      COMMON/DELPAR/SFRAOB,SYRAOB
      COMMON/DELVP/DELVP,DELVG
      COMMON/ELEVON/EPACH(25,2),ELEVON(25,4),M10,P11
      COMMON/ENPUT/PSIRST(6,4),HEADER(20),NVRST(4),KCCRES(6,4),LAST
      COMMON/FORINT/HBANK(2),T,TT,W,U,V,X,Y,Z,XM1,ZAP(18),OVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,NINC,KRDER,EU,EL,AYL,HPX,HPN,HAP
      COMMON/IUSE/IUSE
      COMMON/IIELE/IIELE,DN2(2)
      COMMON/INITP/MISSON(6),WTJET(15),WCLBS(15),AZWTPL(75),BCD(44)
      COMMON/IPR/IPR,HFMX,ILAST,KOT(15),PRK4,PRTCT
      COMMON/IRTAOB/IRTACA
      COMMON/ISTART/JSTART,ISTART(2),STARTV(30,2),SVXINC,SVXNOD,SVAZRE
      COMMON/JJUMP/JJUMP
      COMMON/JUMPST/JUMPST,TSAVET,WZERST
      COMMON/LPRINT/LPRINT,INMAX,PREDP,GY,TCL
      COMMON/LBM/TIMLBM(30),THRLBM(30),WCTLBM(30),PLBM(2),MLBM1,MLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(15),MFS1,MPS2,TCLIM,GLIMIT
      COMMON/NTIME/NTIME
      COMMON/OMSRCS/FCPS(15),FRCS(15),WCCMS(15),WCRCS(15),WOCUMP(15)
      COMMON/OLOW/CLOW,WINT
      COMMON/OMAX/CMAX,QFLG,OPEN,MAXQ
      COMMON/REST/JSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NOPAR,WIBT(15)
      COMMON/RIGGNT/TTABLE(20),SIGMA,SIGTAB(20),P15,P16
```

```
      COMMON/RTLS/VRINT(2),VSLOPE(2)
      COMMON/SRM/SRMTTB(250),SRMFTB(250),SRMWDT(250),SRMAE(2),MSRR
      COMMON/STAUT/STALT(15)
      COMMON/TABISP/MTAU,TIMISP(20),TRFISP(20)
      COMMON/TAULIM/TALLIM,NTAU,NBETA,TIMTAU
      COMMON/TEL/TEL(6),TVL(6),TLL(6),RATIO
      COMMON/VAERO/VAEROC(10,19),M18,M19,MBS18,MES19
      COMMON/VIC/VIC(2),XK1(2),XK2(2),XK3(2),XK4(2),LK
      COMMON/WIND/ALTTBL(75),WTBL(75),AZTBL(75),NWIND
      COMMON/YLI/YLINT(7,10),PHITE(10),PSIREQ(10),KCDPHI(10),PSTR(10)
      DIMENSION CAC(30),CAALP(30),CNO(30),CNALP(30),CMO(30),CMALP(30),
     *CYO(30),CYBETA(30),CNBO(30),CNBETA(30),CLO(30),CLBETA(30),FNM(30),
     *TALFP(30),TBETA(30),TMACH(30),ALTBAS(50,3),BAXIAL(50,3),
     *ENGRM(50,3),BPIT(50,3),OREFT(50),ORBASE(50)
      DIMENSION DELALT(25),DELFAB(25)
C
      NAMELIST/INPUT/ CASE,HEAD,LAST,MISION,
C     GENERAL PROPULSION
     *NSYST,ENGDAT,TNE,PROP,
C     PPS DATA
     *PPSFLE,TIMMPS,FTBL,THROT,GLIM,TAUALT,TAUTBL,IPR,DELVG,TAULIM,NTAU,
     *TIPTAL,TIMISP,TRFISP,RATIO,
C     SRP AND LBP DATA
     *SRMTTB,SRMFTB,SRMWDT,SRMAE,      TIMLBM,TTRLBM,WDTLBM,
C     OMS AND RCS DATA
     *FOMS,WDOMS,WEDUPF,    FRCS,WDRCS,
C     THRUST EVENT DEPENDENT DATA
     *TAUT,TAUW,WT,ET,WDLBS,NOWD,PRINT,NCEVNT,
C     ATTITUDE CONTROL DATA
     *TTBL,CFTBL,CYTBL,KWTA,TOBL,CPOTBL,CYOTBL,IFACT,TMACH,TALFP,TBETA,
     *CHRDOT,KP,KY,TTABLE,SIGTAB,SPRADB,SYRADB,IFCLY,THEAD,NBETA,
C     EXTRA AERO DATA
     *S,XLEN,XREF,ZREF,TCMACH,DELCA,DELCN,DELCM,NCBASE,CBAXIL,ITCL,
     *DELALT,DELFAB,
C     CG AND WIND DATA
     *TBGWT,TXCG,TZCG,  NWIND,ALTTBL,AZWTBL,WTBL,
C     TIMES
     *DTZ,TZERO,TLIFT,TTILT,TCHIR,
C     CONSTANTS
     *CJ,DJ,H,GMUE,OMEEA,GZERO,RE,FLAT,PIN,PIKG,FSFTP,
C     INTEGRATION DATA
     *AYL,BYL,EU,BEU,PPN,BHMN,KRDER,KRDERB,KIND,KINDB,STEP,BSTEP,
C     LAUNCH AND INITIAL CONDITIONS
     *ALAT,ALONG,ALTLS,AZ,FAZ,JUMP,VIV,WC1,
C     OPTIONS
     *ISTART,JSTART,JETET,LPRINT,NCOAST,NMAX,NTAELE,KCDPHI,KCDRES,KDB,
     *MDT,PSIREQ,PSIRST,NVRST,VRCUT,VRINT,VSLOPE,ICONSW,FPRFAC,
     *IRTLS,PSWCH,CMAX,IRTACA,
C     OPTIMIZATION DATA
     *XJEXT,DP2,GY,TOL,MINH,NBGCT,NENDCT,NP,WIBT
      NAMELIST/AERCIN/
C     FIRST STAGE AERO
     *NTALPH,NTBETA,NTMACH,TALPH,TALYH,TXMAC,
C     SECOND STAGE AERO
     *FNM,CAC,CAALP,CNO,CNALP,CMO,CMALP,CYO,CYBETA,CNBO,CNBETA,CLO,
     *CLBETA,
C     BASE DRAG DATA
     *ALTBAS,ORBASE,BAXIAL,BNORM,BPIT,BKFABT,BKFTHR,OREFT,
C     OTHER DATA
     *EMACH,ELEVON,VAEROC
      DATA MISSON/6HNOP   ,6HAOA    ,6HRTLS  ,6HPOST N,6HPOST A,6HVAP IY/
      DATA MISION/6HACA   /
      DATA BLANK/6H
     *      /
      DATA BCD/6H  MA,6HSS    ,6H    VE,6HL     ,6H  GA,6HMI    ,
     *         6H   F,6H      ,6H      ,6H      ,6H      ,6H    ,
     *         6H  VE,6HLE    ,6H    AZ,6HE     ,6H  GA,6HME    ,
     *         6H  IN,6HCL    ,6H    NC,6HDE    ,6H      ,6H    ,
     *         6H     ,6H     ,6H    BE,6HL V   ,6H  C M,6HAX   ,
```

```
     *          6H  STG ,6HHEAT  ,6H      ,6H      ,6H   RTL,6HS CV  ,
     *          6H      ,6H      ,6H    RA,6HNGE   ,6H      ,6H      ,
     *          6H      ,6H      /
      DATA IAR/36H L N LGTT BIN 207 MASTRE TRAJECTORY /
      IF(NTIME.NE.1) GO TO 3
      CALL FTITL2(IAR)
C *   REWIND BOPTBL TAPE
      REWIND 9
      REWIND 3
      NOBASE=20
      OBAXIL=0.
      IHEAD=1
      IPR=0
      IFACT=1
      MIND=1
      MINDB=3
      MRDER=3
      MRDERB=3
      SPRADB=.1
      SYRADB=.1
      ALTLS=0.
      TLIFT=8.
      ISTART(1)=0
      ISTART(2)=0
      TOL=.005
      IPOLY=20
      ICONSW=20
      TBEGR=0.
      TENDR=0.
      NROLL=0
      LPRINT=0
      LAST=0
      NMAX=0
      NTABLE=0
      NWIND=0
      DO 1 I=1,15
1     ITGL(I)=0
1     WJET(I)=0.
1     MOB(I)=0
1     MOT(I)=0
1     MOB(I+15)=0
1     NOWO(I)=0.
1     S(I)=0.
1     TAUT(I)=0.
1     TAUW(I)=0.
1     MSWCH(I)=0
1     MPSFLG(I)=0
1     STEP(I)=4.
1     ESTEP(I)=16.
1     HMAX(I)=100000.
1     EHMAX(I)=260000.
1     PRINT(I)=8.
1     ELIM(I)=.1E20
1     WIBT(I)=1.
1     IF(I.GT.10) GO TO 1
1     MCDPHI(I)=0
1     PSIREC(I)=0.
1     IF(I.GT.5) GO TO 1
1     NOEVNT(I)=0
1     MCDRES(I,1)=0
1     MCDRES(I,2)=0
1     MCDRES(I,3)=0
1     MCDRES(I,4)=0
1     IF(I.GT.4) GO TO 1
1     MWTA(I)=2
1     NP(I)=0
1     NVRST(I)=0
```

```
1          MCDRES(6,I)=0
1          IF(I.GT.2) GO TO 1
1          MP(I)=0
1          MY(I)=0
1        1 CONTINUE
           STEP(1)=1.
           PI=3.14159265
           RAD=180./PI
           JUMP=1
           AA=90./RAD
           ALAT=28.608422
           ALONG=80.6041334
           DP2=.25
           XJEXT=1.
           CY=.8
           CASE=1.
           DELVG=0.
           EU=1.E-05
           EEL=2.E-05
           AYL=2.E-03
           EYL=4.E-05
           HMN=.25
           CMMN=.5
           CMUE=3.986012E14
           CMEGA=.729211595E-04
           CJ=1.62405E-03
           H=-6.40E-06
           CJ=6.5125E-06
           FLAT=1./298.25
           RE=6378160.
           GZERO=9.80665
         3 CONTINUE
           NCOAST=0
           FRK4=0.
           FRTOT=0.
           HEAD=BLANK
           NOM1=0
           LSB=0
           MPRT=1
           MPAGE=0
C          THE FOLLOWING TWO CARDS REINITIALIZE CASE BY TRUNCATING IT.
           ICASE=CASE
           CASE=ICASE
           NMAX=INMAX
           PTN=4.44822162
           PTKG=.45359237
           PSFTM=47.8802589
           PFN=1./PTN
           PFKG=1./PTKG
           PSFFM=1./PSFTM
           NTAU=0
           NBETA=0
           TAULIM=.1
           FTM=.3048
           DO 200 I=1,15
1          WDLBS(I)=WD(I)/PTKG
1          WTJET(I)=WJET(I)/PTKG
1      200 CONTINUE
           CHRDOT=CHRD*RAD
           AZ=AA*RAD
           WO1=WZERO/PTKG
           FAZ=FAZZ*RAD
           IF(JSTART.EQ.0) GO TO 205
           J=2
           IF(JSTART.EQ.ISTART(1)) J=1
           DO 204 I=1,25
1      204 VIV(I)=STARTV(I,J)
           TZERO=STARTV(26,J)
```

```
      GO1=STARTV(27,J)/PIKG
      CHIP=STARTV(28,J)
      CHIY=STARTV(29,J)
      CLCW=STARTV(30,J)
      CPCTBL(1)=CHIP
      CYCTBL(1)=CHIY
      JUMP=ISTART(J)
      N=NOEVNT(I)+1
      ISTART(J)=G
  205 CONTINUE
      DO 207 I=1,75
  207 AZWTBL(I)=AZTBL(I)*RAD
      DO 300 I=1,30
      CAALP (I)=AEROD(I, 1)/RAD
      CAO   (I)=AEROD(I, 2)
      CLBETA(I)=AEROD(I, 3)/RAD
      CMALP (I)=AEROD(I, 4)/RAD
      CMG   (I)=AEROD(I, 5)
      CNALP (I)=AEROD(I, 6)/RAD
      CNBETA(I)=AEROD(I, 7)/RAD
      CNG   (I)=AEROD(I, 8)
      CYBETA(I)=AEROD(I, 9)/RAD
      CLG   (I)=AEROD(I,10)
      CYG   (I)=AEROD(I,11)
      CNBO  (I)=AEROD(I,12)
      TALFP (I)=AEROD(I,13)*RAD
      FNM   (I)=AEROD(I,14)
      TBETA (I)=AEROD(I,15)*RAD
  300 TMACH (I)=AEROD(I,16)
      DO 402 I=1,25
      CELALT(I)=DALT(I)/FTM
  402 CELFAB(I)=DFAB(I)*FFN
      DO 305 I=1,50
      DO 305 J=1,3
      ALTBAS(I,J)=AEROB(I,1,J)/FTM
      BAXIAL(I,J)=AEROB(I,2,J)/PTN
      ENORM(I,J)=AEROB(I,3,J)/PTN
      BPIT(I,J)=AEROB(I,4,J)/PTN/FTM
      IF(J.EQ.1) ORBASE(I)=AEROB(I,5,J)/FTN
      IF(J.EQ.2) QREFT(I)=AEROB(I,5,J)/PSFTN
  305 CONTINUE
      DO 312 I=1,90
      CPCTBL(I)=CPCTBL(I)*RAD
  312 CYCTBL(I)=CYCTBL(I)*RAD
C
C-----------------------READ INPUT AS REQUESTED-------------
      IF(NTIME.NE.1) GO TO 324
      IF(DEMAND) WRITE(6,685) ON2(1)
  685 FORMAT(1X,A60)
      IF(DEMAND) CALL CSFREQ(ON2(1))
C     READ AERO DATA BASE
C
      CALL CBANK(EPACH(1,2),1,170,194)
      CALL CBANK(ELEVCN(1,3),1,-220,244)
      CALL CBANK(ELEVCN(1,4),1,-270,294)
      CALL CBANK(EPACH(1,1),1,-1070,1094)
      CALL CBANK(ELEVCN(1,1),1,-1100,1124)
      CALL CBANK(ELEVCN(1,2),1,-1130,1154)
      CALL CBANK(NTALFH,7,1,2)
      CALL CBANK(NTBETA,7,-101,102)
      CALL CBANK(NTMACH,7,-201,201)
      CALL CBANK(TALPH,7,-301,322)
      CALL CBANK(TALYH,7,-401,422)
      CALL CBANK(TMMAC,7,-501,528)
      CALL CBANK(ALTBAS(1,1),7,-701,750)
      CALL CBANK(BNORM(1,1),7,-801,850)
      CALL CBANK(BPIT(1,1),7,-901,950)
      CALL CBANK(BAXIAL(1,1),7,-1001,1050)
```

```
      CALL CBANK(BKFAB(1,1),7,-1101,1200)
      CALL CBANK(BKFTFR(1,1),7,-1201,1250)
      CALL CBANK(BKFTFR(1,2),7,-1301,1350)
      CALL CBANK(OFBASE,7,-1401,1450)
      CALL CBANK(OREFT,7,-1451,1500)
      CALL CBANK(BAXIAL(1,2),50,1,30)
      CALL CBANK(BAXIAL(1,3),50,-31,60)
      CALL CBANK(ALTBAS(1,2),50,-1350,1379)
      CALL CBANK(ALTBAS(1,3),50,-1350,1379)
      CALL CBANK(BNORM(1,2),51,1,30)
      CALL CBANK(BNORM(1,3),51,-31,60)
      CALL CBANK(BFIT(1,2),52,1,30)
      CALL CBANK(BFIT(1,3),52,-31,60)
      CALL CBANK(CNO,44,1400,1429)
      CALL CBANK(CNALF,44,-1451,1480)
      CALL CBANK(CNO,45,1400,1429)
      CALL CBANK(CNALP,45,-1451,1480)
      CALL CBANK(CMO,46,1400,1429)
      CALL CBANK(CMALF,46,-1451,1480)
      CALL CBANK(CYO,47,1400,1429)
      CALL CBANK(CYBETA,47,-1451,1480)
      CALL CBANK(CNBO,48,1400,1429)
      CALL CBANK(CNBETA,48,-1451,1480)
      CALL CBANK(CLO,49,1400,1429)
      CALL CBANK(CLBETA,49,-1451,1480)
      CALL CBANK(PAM,45,-1350,1379)
      CALL CBANK(VAERCC(1,19),53,1350,1359)
      CALL CBANK(VAERCC(1,1),53,-1401,1430)
      CALL CBANK(VAERCC(1,10),53,-1451,1480)
      CALL CBANK(VAERCC(1,4),54,1401,1430)
      CALL CBANK(VAERCC(1,13),54,-1451,1480)
      CALL CBANK(VAERCC(1,7),55,1401,1430)
      CALL CBANK(VAERCC(1,16),55,-1451,1480)
      READ(5,AERCIN)
      NTB=NTBETA(1)
      DO 321 I=1,NTB
      IF(ABS(TALPH(I,1)).LT.1.E-04) GO TO 322
  321 CONTINUE
  322 IB=I
      CALL AEBANK(CAA, 8,28,11,IB,NTMACH)
      CALL AEBANK(CNA,14,28,11,IB,NTMACH)
      CALL AEBANK(CMA,20,28,11,IB,NTMACH)
      NTA=NTALPH(2)
      DO 323 I=1,NTA
      IF(ABS(TALPH(I,2)).LT.1.E-04) GO TO 325
  323 CONTINUE
  325 IA=I
      CALL AEBANK(CYA,26,28,11,IA,NTMACH)
      CALL AEBANK(CXA,32,28,11,IA,NTMACH)
      CALL AEBANK(CLA,38,28,11,IA,NTMACH)
  324 CONTINUE
      IF(DEMAND) WRITE(6,685) DN2(IIELE)
      IF(DEMAND) CALL CSFREQ(DN2(IIELE))
      READ(5,INPUT)
C-----------------------------------------------------------------
      REWIND 1
      REWIND 2
      REWIND 4
      REWIND 10
      REWIND 11
      REWIND 12
      REWIND 13
C-----------------------------------------------------------------
      IF(JUMP.EQ.1) GO TO 350
      W01=WC1-WTJET(JUMP-1)
      IF(MISSION.NE.MISSON(3)) GO TO 351
C
C     SET UP AZRE CONSTRAINT FOR RTLS
C
```

```
      DO 352 I=1,6
      IF(KCDRES(I,1).NE.8.AND.KCDRES(I,1).NE.0) GO TO 352
      KCDRES(I,1)=8
      FSIRST(I,1)=SVAZRE+180.
      IF(SVAZRE.GE.0.) PSIRST(I,1)=SVAZRE-180.
      GO TO 350
  352 CONTINUE
  351 CONTINUE
      IF(MISSION.NE.MISSON(6)) GO TO 350
C
C     SET UP INCLINATION AND NODE CONSTRAINTS FOR AOA(VARIABLE IY)
C
      DO 320 I=1,10
      IF(KCDPHI(I).NE.10 .AND. KCDPHI(I).NE.0) GO TO 320
      KCDPHI(I)=10
      KCDPHI(I+1)=11
      FSIREC(I-1)=SVXINC*RAD
      FSIREC(I)=SVXNOD
      MISION=MISSON(2)
      GO TO 350
  320 CONTINUE
  350 CONTINUE
      IF(JETET.EQ.0) GO TO 1345
      M=1
      IF(MISION.EQ.MISSON(5)) M=2
      WTJET(JETET)=ASTOR(M,30)-ASTOR(M,22)+ASTOR(M,18)+ASTOR(M,19)+
     *ASTOR(M,20)+ASTOR(M,21)
 1345 CONTINUE
      IF(NTABLE.LE.1) REWIND 9
      IF(JSTART.EQ.0) JSTART=ISTART(1)
      DO 206 I=1,15
      IF(I.LT.JUMP.AND.KOB(I).NE.0) KOB(I)=0
      IF(I.GT.13) GO TO 206
      IF(JUMP.GT.NCEVAT(1).AND.KOB(I+15).NE.0) KOB(I+15)=0
  206 CONTINUE
      IF(IPR.NE.0.AND.KOB(IPR).NE.0) KOB(IPR)=0
      IF(ABS(CHROOT).LT.1.E-04) NROLL=-1
      FAZZ=FAZ/RAD
      IF(FAZZ.LT.0.) FAZZ=FAZZ+2.*PI
      CHRO=CHROOT/RAD
      AA=AZ/RAD
      INMAX=NMAX
      THET1=(90.-ALAT)/RAD
      WZERO=W01*PTWG
      WRITE(6,314) HEAD
  314 FORMAT(1H1,33X,A40,//)
C
      DO 8 I=1,15
      STAUT(I)=TAUT(I)
      W0(I)=W0LBS(I)*PTWG
    8 WJET(I)=WTJET(I)*PTWG
      EL=EU/2.**(KRDER+2)
      EEL=BEU/2.**(KRCERE+2)
      DO 302 I=1,4
      IF(NP(I).EQ.0) GO TO 302
      IF(NP(I).EQ.1) NP(I)=3
      IF(NP(I).GE.10) NP(I)=29
      J=(NP(I)/2)*2
      IF(J.EQ.NP(I)) NP(I)=NP(I)+1
  302 CONTINUE
C*****
C     CALCULATE NUMBER OF PARAMETERS
      NOPAR=0
      DO 12 I=1,40
      IF(KOB(I).EQ.0)GO TO 12
      NOPAR=NOPAR+1
   12 CONTINUE
C*****
```

```
C       CALCULATE NUMBER OF REGULAR SETS
        NCSS=0
        DO 13 I=1,7
        IF(KCDPHI(I).EQ.0)GO TO 13
        NCSS=NCSS+1
   13   CONTINUE
C*****
C       CALCULATE NUMBER OF RESTART CONSTRAINTS
        NTCN=0
        DO 15 K=1,4
        NCNRS(K)=0
        IR(K)=1.E19
        IF(NVRST(K).EQ.0) GO TO 15
        DO 14 I=1,6
        IF(KCDRES(I,K).EQ.0) GO TO 14
        NCNRS(K)=NCNRS(K)+1
   14   CONTINUE
        NTCN=NTCN+NCNRS(K)
   15   CONTINUE
        IF((NCSS+NTCN).LE.10) GO TO 17
        WRITE(6,16)
   16   FORMAT(1X,23H    TOO MANY CONSTRAINTS)
        WRITE(6,INPUT)
        STOP
   17   CONTINUE
C***    CALCULATE NO OF THRUST EVENTS
        NVNT=0
        DO 18 I=1,5
        IF(NOEVNT(I).EQ.0)GO TO 18
        NVNT=NVNT+NOEVNT(I)
        NA=I
   18   CONTINUE
        DO 180 I=1,15
        IF(KDB(I).EQ.0) GO TO 180
        IF(JUMP.LE.I.AND.NVNT.GE.I) GO TO 180
        KDB(I)=0
        NOPAR=NOPAR-1
  180   CONTINUE
        ILAST=NVNT-NCEVNT(NA)+1
C***    CALCULATE NO OF WEIGHT EVENTS
        NWVNT=0
        DO 303 I=1,15
        IF(NOWD(I).EQ.0) GO TO 303
        NWVNT=NWVNT+1
  303   CONTINUE
C***    CALCULATE PERFORMANCE RESERVE CONSTANTS
        IF(IPR.EQ.0) GO TO 304
        XISP=ENGDAT(6,1)/ENGDAT(7,1)
        FRK4=1.-EXP(-DELVG/(GZERO*XISP))
C***    CHECK KDT SETUP FOR IPR
        DO 191 J=ILAST,NVNT
        IF(KDB(J).EQ.0) GO TO 191
        KDT(J)=IPR-J
        IF(KDT(J).GT.0) GO TO 191
        KDT(J)=0
        KDB(J)=0
        NOPAR=NOPAR-1
  191   CONTINUE
        IF(KDB(16).EQ.1) GO TO 304
        KDB(16)=1
        NOPAR=NOPAR+1
  304   CONTINUE
        II=NOEVNT(1)
        TJETWT=0.
        DO 2 I=JUMP,NVNT
    2   TJETWT=TJETWT+WJET(I)*WD(I)
        DO 24 I=1,II
```

```
      24 LSTGE(I)=1
         II=II+1
         DO 25 I=II,NVNT
      25 LSTGE(I)=2
         IF(NVNT.NE.15) LSTGE(NVNT+1)=LSTGE(NVNT)
      20 CONTINUE
         DO 199 I=1,40
         DO 199 J=1,10
     199 XLAMB(I,J)=0.
C
         PAT(1)=0.
         CALL FRA63(PAT,ATMOSN,IER)
         PSL=PAT(2)*10000.
         STH1=SIN(THET1)
         CTH1=COS(THET1)
         STHL=STH1
         CTHL=CTH1
C        OBLATE EARTH
         UMF=1.-FLAT
         TEMP=PI/2.-ATAN2(UMF*UMF*CTH1,STH1)
         STH=SIN(TEMP)
         CTH=COS(TEMP)
         STHL=STH
         CTHL=CTH
         R=UMF*RE/SQRT((UMF*STH)**2+CTH*CTH)+ALTLS
      30 WRZ=R*STH *OMEGA
         DPHIZ=OMEGA*CTZ
         SDPHI=SIN(DPHIZ)
         CDPHI=COS(DPHIZ)
         IF(JUMP.NE.1)GO TO 40
         EO(1,1)=CDPHI
         EO(1,2)=SDPHI*STH
         EO(2,1)=0.
         EO(2,2)=CTH
         EO(3,1)=-SDPHI
         EO(3,2)=CDPHI*STH
         XJV(1)=WRZ*EO(1,1)
         XJV(2)=0.
         XJV(3)=WRZ*EO(3,1)
         XJX(1)=R*EO(1,2)
         XJX(2)=R*EO(2,2)
         XJX(3)=R*EO(3,2)
      40 ALT1=10000.0?0
         ALT2=14000.028
         ALT=0.
C*****SET UP HEADERS FOR CONSTRAINTS
         NV=1
         DO 51 I=1,10
         IF(KGEPHI(I).EQ.0)GO TO 51
         K=2*KCDPHI(I)-1
         HEADER(KV)=ECD(K)
         HEADER(KV+1)=ECC(J+1)
         KV=KV+2
      51 CONTINUE
C-----------PRINT NAMELIST INPUT HERE-----------
         IF(NTIME.EQ.1) WRITE(JO,AERDIN)
         WRITE(JO,INPUT)
         NTIME=NTIME+1
C
         DO 301 I=1,30
         AEROD(I, 1)=CALF (I)*RAD
         AEROD(I, 2)=CAO  (I)
         AEROD(I, 3)=CLBETA(I)*RAD
         AEROD(I, 4)=CMALF (I)*RAD
         AEROD(I, 5)=CMO  (I)
         AEROD(I, 6)=CNALF (I)*RAD
         AEROD(I, 7)=CNBETA(I)*RAD
         AEROD(I, 8)=CNO  (I)
```

```
      AEROD(I, 9)=CYBETA(I)*RAD
      AEROD(I,10)=CLO    (I)
      AEROD(I,11)=CYO    (I)
      AEROD(I,12)=CNBC   (I)
      AEROD(I,13)=TALFF  (I)/RAD
      AEROD(I,14)=FNM    (I)
      AEROD(I,15)=TBETA  (I)/RAD
  301 AEROD(I,16)=TMACH  (I)
      DO 403 I=1,25
      DALT(I)=DELALT(I)*FTM
  403 DFAB(I)=DELFAB(I)*FTM
      DO 308 I=1,50
      AEROB(I,1,J)=ALTBAST(I,J)*FTM
      AEROB(I,2,J)=BAXIAL(I,J)*PTN
      AEROB(I,3,J)=BNORM(I,J)*PTN
      AEROB(I,4,J)=BPIT(I,J)*PTN*FTM
      IF(J.EQ.1) AEROB(I,5,J)=QRBASE(I)*FTN
      IF(J.EQ.2) AEROB(I,5,J)=QREFT(I)*PSFTM
  308 CONTINUE
      DO 313 I=1,90
      CPOTBL(I)=CPCTBL(I)/RAD
  313 CYCTBL(I)=CYCTBL(I)/RAD
      DO 307 I=1,75
  307 AZTBL(I)=AZWTBL(I)/RAD
      JJUMP=0
      IF(JUMP.NE.1.OR.FINM.LE.NOEVNT(1)) GO TO 64
      J=1
      M=NOEVNT(1)
      L=1
      GO TO 62
   61 J=16
      M=28
      L=2
   62 DO 63 I=J,M
      IF(KOE(I).EQ.1) GO TO 64
   63 CONTINUE
      IF(L.EQ.1) GO TO 61
      JJUMP=1
   64 CONTINUE
      JUMPST=0
      DO 68 L=1,4
      M=5-L
      IF(NVEST(M).EQ.0) GO TO 68
      DO 52 I=1,6
      IF(MCDRES(I,M).EQ.0) GO TO 52
      J=2*MCDRES(I,M)-1
      HEADER(KV)=BCD(J)
      HEADER(KV+1)=BCD(J+1)
      KV=KV+2
   52 CONTINUE
   68 CONTINUE
      IK1=0
      IJ1=0
      LK=0
      M=0
      DO 69 I=2,10
      IF(MCDFHI(I).EQ.4) IJ=I-1
      IF(MCDFHI(I).EQ.14) IJ1=I-1
      IF(I.GT.7) GO TO 69
      DO 73 L=1,4
      IF(MCDRES(I-1,L).EQ.4) IK=I-1
      IF(MCDRES(I-1,L).EQ.14)IK1=I-1
      IF(IK.NE.0.AND.M.EQ.0) M=L
   73 CONTINUE
   69 CONTINUE
      DO 70 I=1,2
      IF(I.EQ.2) GO TO 71
      IF(IJ1.EQ.0) GO TO 70
      LK=LK+1
```

```
      RM=PSIREQ(IJ1)
      RI=PSIREQ(IJ)
      PSIREQ(IJ1)=0.
      GO TO 72
   71 IF(IK1.EQ.0) GO TO 70
      LK=LK+1
      RM=PSIRST(IK-1,K)
      RI=PSIRST(IK,K)
      PSIRST(IK1,K)=0.
   72 VIC(LK)=SQRT(2.*CHLE*RM/(RI*(RI+RM)))
      RHO=RI/RM
      XK1(LK)=CJ/3.*RE*RE/(RI*RI)
      XK2(LK)=3.+RHO*(1.-RHO)
      XK3(LK)=1.+RHO*(1.+RHO)
      XK4(LK)=-2.*RHO*RHO
   70 CONTINUE
      RETURN
      END
```

APULG

```
      SUBROUTINE APULG(L,M,N,X,XT,Y1,Y1T,Y2,Y2T)
C
C     LINEAR INTERPOLATION SCHEME
C          L        : NUMBER OF DEPENDENT VARIABLES(1 OR 2)
C          M        : PLACE IN TABLE(SET=0 INITIALLY;UPDATED BY ROUTINE)
C          N        : NUMBER OD POINTS IN INDEPENDENT TABLES
C          X        : VALUE OF INDEPENDENT VARIABLE
C          XT       : INDEPENDENT VARIABLE TABLE
C          Y1       : VALUE OF DEPENDENT VARIABLE 1
C          Y1T      : DEPENDENT VARIABLE TABLE FOR 1
C          Y2       : VALUE OF DEPENDENT VARIABLE 2
C          Y2T      : DEPENDENT VARIABLE TABLE FOR 2
C
      COMPILER(ARGCHK=OFF)
      DIMENSION XT(1),Y1T(1),Y2T(1)
      DIFF=X-XT(M)
      IF(DIFF)1,3,2
C     GET HERE IF X.LT.XT(M)
    1 IF(M.EQ.1)GO TO 6
      P=P-1
      IF(X-XT(M))1,3,4
C     GET HERE IF X.GT.XT(M)
    2 IF(M.EQ.N)GO TO 7
      MP1=M+1
      IF(XT(MP1).LT.XT(M)) GO TO 7
      P=MP1
      IF(X-XT(M))5,3,2
    3 Y1=Y1T(M)
      IF(L.NE.1)Y2=Y2T(M)
      RETURN
    4 P=P+1
    5 MM1=M-1
      DEL=(X-XT(MM1))/(XT(M)-XT(MM1))
      Y1=Y1T(MM1)+DEL*(Y1T(M)-Y1T(MM1))
      IF(L.NE.1)Y2=Y2T(MM1)+DEL*(Y2T(M)-Y2T(MM1))
      RETURN
    6 Y1=Y1T(1)
      IF(L.EQ.2) Y2=Y2T(1)
      RETURN
    7 Y1=Y1T(M)
      IF(L.EQ.2) Y2=Y2T(P)
      RETURN
      END
```

```
      SUBROUTINE ANEWCH
C
C     DETERMINES THE STEPSIZE AND EVALUATES THE REQUIRED CHANGES IN
C     THE PARAMETERS.CONVERGENCE TESTS ARE MADE FOR ITERATION TERMINATIO
      CHARACTER*60-HEAD
      CHARACTER*6 MISION
      LOGICAL DEMAND
      COMMON/CCC/GRAPH,JC,DEMAND
      COMMON/AERODI/AERO(30,16),AERO8(50,5,3),TCMACH(20),DELCA(20),
     *FNPONE(15),BHFAB1(50,2),BHFTHR(50,2),CELCN(20),CELCM(20)
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CTZ,TC,TL,XMAUG,INE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),FLOP(6),TBEGR,TENDR,
     *CHROOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SU,SV,Q,
     *VIV(25),BVAR5(12),EEL,DW(15,3),DEL,DR(7,4),GEL,C(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMAX(15),BHMN,BSTEP(15),WNMB,HMXB,TPOLY,XMIAO
      COMMON/AGEN3/ITER,IWC,IXR,JUMP,KAT,KCYTAB,KFAGE,KS1,KWTA(4),LINES,
     *NBECT(4),NENECT(4),NMAX,NOEVNT(5),NOWO(15),NVNT,NWVNT,IHEAC,
     *NIN,KFSFLG(15),NSYST(15),ICONSW,IFTLS,IPOLY,KINOB,KRDERB,MISION,
     *IRTFLG,NROLL,NGEAST,IFACT,NOBASE,LSTGE(15),KSWCH(15)
      COMMON/AUTG/KOB(40),SAVCP(40),XLAMB(40,10),CP2
      COMMON/BAKE/WISS(10,10),YLF(7,10)
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NCM1,NF(4)
      COMMON/BLAMB/YLE1(15,10),YLBW(15,10),GAPHI(10),FROD(2,10)
      COMMON/BODY/IBOLT(15,2),TXCG(15,2),TYCG(15,2),XLEN(2),XREF(2),
     *YREF(2),XP(2),XY(2),AP(10,2),AY(10,2)
      COMMON/CONST/RAC,PI,RE,FLAT,CJ,H,CJ,CMLE,OMEGA,ALT1,ALT2,PSL,
     *GZERO,FTN,PTNG,FSFTM,PFN,PFKG,PSFFM
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/CONVK/CONVK(15,3)
      COMMON/CRHC/CRHC(9),E1,WIPPB,WGNU
      COMMON/FMXX/FMXX,MSW,WG,SPOA
      COMMON/FORINT/HEANN(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),CVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,FMX,HPN,HAP
      COMMON/GNU/BAU(10)
      COMMON/IMP/GAPTE(90,10),GSPTB(90,10),APTB(90),SITB(90),TS(3)
     *,DELTS(3),KJTB
      COMMON/IPR/IFR,WPMX,ILAST,KOT(15),FRM4,PRTCT
      COMMON/LPRINT/LPRINT,INMAX,PREDP,CY,TOL
      COMMON/LBM/TIMLEP(30),THRLBM(30),WCTLBM(30),PLBM(2),MLBM1,MLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MPS/TIMMPS(15),FTBL(15),THRGT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MPS1,MPS2,TGLIM,GLIMIT
      COMMON/OMSRCS/FCPS(15),FRCS(15),WOCMS(15),WCRCS(15),WOCUMP(15)
      COMMON/REST/NSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NCFAR,WIBT(15)
      COMMON/TAULIM/TALLIM,NTAU,NBETA,TIMTAU
      COMMON/XTRA/XMO(15),CFR(15),F(15),CISP(15)
      COMMON/YLI/YLIN1(7,10),PHITE(10),PSIREQ(10),KCDPHI(10),PSTR(10)
      DIMENSION          SWIBT(17),OBETA(17),WAPS(9),WASS(10,10),WOS(9)
     *DIMENSION HUMAX(2),WIPSB(9),WISSB(9,9),XLBENU(9),XLBWOS(9)
      LOGICAL BETCON
C*****
C*****
C*****
C     BEGIN MAIN NEWCH
C*****
   88 KAY=1.
      LP=0
      IF(LPRINT.NE.1) GO TO 10
      LP=1
      LPRINT=2
```

```
      16 CONTINUE
         CONVP=1.-.5*E1
         CASE=CASE+.0001
         EETCOM=.TRUE.
         LSB=0
         NV=1
         DO 12 I=1,40
         IF(NOB(I).NE.1)GO TO 12
         DO 11 J=1,NOSS
      11 XLAMB(NV,J)=XLAMB(I,J)
         NV=NV+1
      12 CONTINUE
         WRITE(6,97)
      97 FORMAT(/,1X,22HINFLUENCE COEFFICIENTS)
         DO 81 I=1,NOFAR
      81 WRITE(6,9)(XLAMB(I,J),J=1,NOSS)
       9 FORMAT(1X,8E15.8)
C*****
C*****
C        CALCULATE I-SY-ST MATRIX
      71 NK=NOSS-1
         WIPP=WISS(1,1)
         DO 16 I=1,9
      16 WAPS(I)=WISS(I+1,1)
         DO 17 I=2,9
         DO 17 J=2,I
         IM=I-1
         JM=J-1
         WASS(IP,JM)=WISS(I,J)
      17 WASS(JP,IM)=WASS(IP,JM)
         IF(LPRINT.NE.2)GO TO 82
         WRITE(6,18)
      18 FORMAT(1X,12H    I PSI PSI)
         WRITE(6,21)WIPP
         DO 20 I=1,KK
      20 WRITE(6,21)WAPS(I),(WASS(I,J),J=1,KK)
      21 FORMAT(1X,8E15.8)
C*****GET WIBT
C*****
      82 DO 14 I=1,NOFAR
         SWIBT(I)=0.
         DO 13 J=2,NOSS
         TEMP=0.
         IF(ABS(WASS(J-1,J-1)).GT.0.) TEMP=XLAMB(I,J)*XLAMB(I,J)/
        *WASS(J-1,J-1)
      13 SWIBT(I)=AMAX1(SWIBT(I),TEMP)
      14 SWIBT(I)=WIBT(I)/SWIBT(I)
C*****
         IF(NOFAR.EQ.0)SWIBT(1)=0.
C******GOT WIBT
         IF(LPRINT.NE.2)GO TO 83
         WRITE(6,98)
      98 FORMAT(/,1X,5HPSLBI)
      15 WRITE(6,21)(SWIBT(I),I=1,NOFAR)
         WRITE(6,99)
      99 FORMAT(/,1X,4HWIBT)
         WRITE(6,21)(WIBT(I),I=1,NOFAR)
      83 WIPPB=WIPP
         DO 22 I=1,NOFAR
      22 WIPPB=WIPPB+XLAMB(I,1)*SWIBT(I)*XLAMB(I,1)
         DO 23 I=1,KK
         WIPSB(I)=WAPS(I)
         DO 23 J=1,NOFAR
      23 WIPSB(I)=WIPSB(I)+XLAMB(J,1)*SWIBT(J)*XLAMB(J,I+1)
         DO 24 I=1,KK
         DO 24 J=1,KK
         WISSB(I,J)=WASS(I,J)
         DO 24 K=1,NOFAR
```

```
      24 WISSB(I,J)=WISSB(I,J)+XLAMB(K,I+1)*SWIBT(K)*XLAMB(K,J+1)
C     PRINT TOTAL I MATRIX
      IF(LPRINT.NE.2)GO TO 84
      WRITE(6,25)
      25 FORMAT(1X,23H   TOTAL I SY SY MATRIX)
      WRITE(6,21)WIPSB
      DO 251 I=1,KK
      251 WRITE(6,21)WIPSB(I),(WISSB(I,J),J=1,KK)
      84 IF(KK.GT.1)GO TO 26
      WISSB(1,1)=1./WISSB(1,1)
      GO TO 271
      26 CALL MATINV(WISSB(1,1),KK)
      DO 2005 I=1,KK
      DO 2005 J=1,I
      WISSB(I,J)=(WISSB(I,J)+WISSB(J,I))/2.
      2005 WISSB(J,I)=WISSB(I,J)
C*****
C     CALCULATE MATRIX PRODUCTS
C*****
      271 DO 27 I=1,KK
      27 DRHO(I)=-PSTR(I+1)
      DO 28 I=1,KK
      WDS(I)=0.
      DO 28 J=1,KK
      28 WDS(I)=WDS(I)+WISSB(I,J)*DRHO(J)
      DO 30 I=1,KK
      GNU(I)=0.
      DO 30 J=1,KK
      30 GNU(I)=GNU(I)-WISSB(I,J)*WIPSB(J)
      WGNU=0.
      DO 31 I=1,KK
      31 WGNU=WGNU-GNU(I)*WIPSB(I)
      IF(LPRINT.NE.2)GO TO 85
      WRITE(6,1000)WGNU
      1000 FORMAT(1X,4HWGNU,E15.8)
C*****
C     BEGIN E1 LOGIC
C*****
      85 NOM2=NOM1+1
      IF(NOM2.GE.5) GO TO 325
      GO TO(321,322,323,324),NOM2
      321 E1=0.
      XKAY=AMAX1(DF2,.25)
      IF(XKAY.LT..4995) GO TO 34
      IF(XKAY.GT..5995) GO TO 32
      NOM1=1
      XKAY=.5
      GO TO 34
      32 NOM1=2
      GO TO 36
      322 E1=0.
      XKAY=.5
      GO TO 34
      323 E1=0.
      GO TO 36
      324 E1=QY/2.
      GO TO 36
      325 E1=QY
      GO TO 36
      34 DO 35 I=1,KK
      DRHO(I)=DRHO(I)*XKAY
      35 WDS(I)=XKAY*WDS(I)
C*****
C     CALCULATE DBETA
C*****
      36 IF(NOFAR.EQ.0)GO TO 50
      DO 37 I=1,NOFAR
      CONVK(I,3)=ABS(XLAMB(I,1))
```

```
      XLBGNU(I)=0.
      XLBWDS(I)=0.
      DO 37 J=1,KK
      TEMP=XLAMB(I,J+1)*GNU(J)
      XLBGNU(I)=XLBGNU(I)+TEMP
      XLBWDS(I)=XLBWDS(I)+XLAMB(I,J+1)*WDS(J)
      TEMP=TEMP+XJEXT*XLBWDS(I)
   37 CONVK(I,3)=AMAX1(CONVK(I,3),ABS(TEMP))
      IF(LPRINT.NE.2)GO TO 86
      WRITE(JO,1001)XKAY,E1
 1001 FORMAT(  1X,4HXKAY,E15.8,3H E1,E15.8)
      WRITE(JO,450)
  450 FORMAT(  1X,12HDRHO,WDS,GNU)
      DO 470 I=1,KK
  470 WRITE(JO,21)DRHO(I),WDS(I),GNU(I)
   86 DO 40 I=1,NOPAR
      TEMP=XLAMB(I,1)+XLBGNU(I)
      CONVK(I,2)=TEMP/CONVK(I,3)
   40 DBETA(I)=SWIET(I)*(     TEMP                    *E1*XJEXT+XLBWDS(I))
      WRITE(JO,96)
   96 FORMAT(/,7X,4HDPAR,9X,8HOLD PCON,7X,8HNEW PCON,8X,7HP SCALE)
      IK=0
      DO 1002 I=1,NOPAR
      IF(ABS(CONVK(I,1)).GT..005)BETCON=.FALSE.
      IF(BETCON) GO TO 1002
      IK=IK+1
      IF(IK.GT.1) GO TO 1002
      IF(ABS(CONVK(I,2)*CONVK(I,3)).GT.1.) GO TO 1002
      BETCON=.TRUE.
      IK=0
 1002 WRITE(JO,21)DBETA(I),(CONVK(I,J),J=1,3)
C*****
C     ADD DBETA TO BETA
C*****
      IF(LP.EQ.1) LPRINT=1
      NV=1
      DO 45 I=1,40
      IF(KDB(I).EQ.0)GO TO 45
      IF(I.GT.15)GO TO 43
      TTOL=1.*TAUT(I)
      IF(TTOL.GT.3.) TTOL=3.
      IF(ABS(DBETA(NV)).GT.TTOL) DBETA(NV)=SIGN(TTOL,DBETA(NV))
   41 TAUT(I )=TAUT(I )+DBETA(NV)
C
C     IF(I.LT.NVNT)TAUT(I+1)=TAUT(I+1)-DBETA(NV)
C
      IF(MSW.EQ.1) GO TO 42
      IF(KDT(I).EQ.0)GO TO 42
      J=I +KDT(I)
      TEMP=TNE(1,I)*CFR(I)/(TNE(1,J)*CFR(J))
      TAUT(J)=TAUT(J)-TEMP*DBETA(NV)
      IF(TAUT(J).LE..01)TAUT(J)=.01
   42 IF(ABS(DBETA(NV)/(TAUT(I)-DBETA(NV))).GT..0025) BETCON=.FALSE.
      IF(TAUT(I ).LE..01)TAUT(I )=.01
      WRITE(JO,242) I,TAUT(I)
      IF(.NOT.DEMAND)GO TO 44
      IJWC=1
      WRITE(11)IJWC,I,TAUT(I)
  242 FORMAT(1X,5HTAUT(,I2,2H)=,E15.8)
      GO TO 44
   43 I2=I-15
      GO TO (141,142,143,143,143,143,143,143,143,144,144,144,144,145,
     *145,145,146,146,146),I2
  141 WZERO=WZERO+DBETA(NV)
      IF(ABS(DBETA(NV)).GT.100.)BETCON=.FALSE.
      WO1=WZERO/.45359237
      WRITE(JO,243) WO1
  243 FORMAT(1X,4HWO1=,E15.8)
```

```
      IF(.NOT.DEMAND)GO TO 44
      IJWC=2
      WRITE(11)IJWC,IJWC,NO1
      GO TO 44
  142 AA=AA+DBETA(NV)/RAD
      AD=AA*RAD
      IF(ABS(DBETA(NV)/(AD-DBETA(NV))).GT..005) BETCON=.FALSE.
      WRITE(6,244) AD
  244 FORMAT(1X,3HAZ=,E15.8)
      IF(.NOT.DEMAND)GO TO 44
      IJWC=3
      WRITE(11)IJWC,IJWC,AD
      GO TO 44
  143 JJ=I2-1
      IF(ABS(DBETA(NV)).GT.2.) DBETA(NV)=SIGN(2.,DBETA(NV))
      CPTBL(JJ)=CPTBL(JJ)+DBETA(NV)
      IF(ABS(DBETA(NV)/(CPTBL(JJ)-DBETA(NV))).GT..01) BETCON=.FALSE.
      WRITE(6,245) JJ,CPTBL(JJ)
  245 FORMAT(1X,6HCPTBL(,I2,2H)=,E15.8)
      IF(.NOT.DEMAND)GO TO 44
      IJWC=4
      WRITE(11)IJWC,JJ,CPTBL(JJ)
      GO TO 44
  144 JJ=I2-8
      IF(ABS(DBETA(NV)).GT.2.) DBETA(NV)=SIGN(2.,DBETA(NV))
      CYTBL(JJ)=CYTBL(JJ)+DBETA(NV)
      IF(ABS(DBETA(NV)/(CYTBL(JJ)-DBETA(NV))).GT..01) BETCON=.FALSE.
      WRITE(6,246) JJ,CYTBL(JJ)
  246 FORMAT(1X,6HCYTBL(,I2,2H)=,E15.8)
      IF(.NOT.DEMAND)GO TO 44
      IJWC=5
      WRITE(11)IJWC,JJ,CYTBL(JJ)
      GO TO 44
  145 JJ=I2-13
      IF(ABS(DBETA(NV)).GT.5.) DBETA(NV)=SIGN(5.,DBETA(NV))
      GO TO (1450,1451,1452),JJ
 1450 PAR=TIMMPS(NTAU)
      TIMMPS(NTAU)=PAR+DBETA(NV)
      TEMP=TIMMPS(NTAU)
      GO TO 1453
 1451 PAR=TIMTAU
      TIMTAU=PAR+DBETA(NV)
      NTINV=0
      IF(TIMTAU.LT.0.) NTINV=NV
      IF(TIMTAU.LT.0.) TIMTAU=.01
      TEMP=TIMTAU
      GO TO 1453
 1452 PAR=FTBL(NTAU+1)*100.
      FTBL(NTAU+1)=(PAR+DBETA(NV))/100.
      NTNV=0
      IF(FTBL(NTAU+1).LT..65) NTNV=NV
      IF(FTBL(NTAU+1).LT..65) FTBL(NTAU+1)=.65
      TEMP=FTBL(NTAU+1)
 1453 IF(ABS(DBETA(NV)/PAR).GT..01) BETCON=.FALSE.
      WRITE(6,1454) TEMP
 1454 FORMAT(1X,13HTHROTTLE PAR(,I2,2H)=,E15.8)
      IF(.NOT.DEMAND) GO TO 44
      IJWC=6
      WRITE(11) IJWC,JJ,TEMP
      GO TO 44
  146 JJ=I2-16
      L=NBETA+JJ-1
      AEROD(L,15)=AEROD(L,15)+DBETA(NV)/RAD
      IF(ABS(DBETA(NV)/(AEROD(L,15)*RAD-DBETA(NV))).GT..01) BETCON=
     *.FALSE.
      PAR=AEROD(L,15)*RAD
      WRITE(6,1460) JJ,PAR
 1460 FORMAT(1X,13HSIDESLIP PAR(,I2,2H)=,E15.8)
      IF(.NOT.DEMAND) GO TO 44
```

```
      IJKC=1
      WRITE(11) IJKC,...,FAR
      GO TO 44
   44 NV=NV+1
   45 CONTINUE
      IJKC=6
      IF(GE.AND.)WRITE(11)IJKC,IJKC,IJKC
      IF(MSK.EQ.0) GO TO 203
      WY=0.
      K=IPR-1
      TEMP=WZERO
      DO 200 I=JLMF,K
      TEMP=TEMP-TNE(1,I)*XMD(I)*TAUT(I)-WJET(I)-WG(I)
      WY=WY+TNE(1,I)*CFR(I)*TAUT(I)
  200 CONTINUE
      TEMP=TEMP-SRPRP
      WY=WY+TNE(1,IPR)*CFR(IPR)*(TEMP-WG)/(TNE(1,IPR)*XMD(IPR))
      NV=1
      DO 202 I=1,15
      IF(MDB(I).EQ.0) GO TO 202
      IF(MDT(I).EQ.0) GO TO 201
      TEMP=TNE(1,I)*XPC(I)/(TNE(1,IPR)*XPD(IPR))-SPCA*(TNE(1,I)*CFR(I)-
     *CFR(IPR)/XMD(IPR)*TNE(1,I)*XMD(I))/(WPPX-WY-CFR(IPR)/XMD(IPR)*WG)
      TAUT(IPR)=TAUT(IPR)-TEMP*DBETA(NV)
  201 NV=NV+1
  202 CONTINUE
  203 CONTINUE
C*****
C     CALCULATE CHANGE IN CHIP AND CHIY
C*****
   50 HUMAX(1)=0.
      HUMAX(2)=0.
      NV=1
      WRITE(6,95)
   95 FORMAT(/,7X,4HTIME,9X,6HNEW CP,9X,6HOLD CP,8X,6HNEW CY,8X,
     1 6HOLD CY,10X,3HDCP,10X,3HDCY)
  202 DO 60 II=1,4
      DELCP=0.
      DELCY=0.
      IF(NP(KV).EQ.0)GO TO 59
      KJTB=(KV-1)*30
      I=TS(KV)
      ETB4=DELTS(KV)
      JTB=NF(KV)
      JA=NOSS
      IF(T+.1.GT.TR(1).AND.TR(1).GT.0.) JA=NOSS-NCNRS(1)
      IF(T+.1.GT.TR(2).AND.TR(2).GT.0.) JA=NOSS-NCNRS(1)-NCNRS(2)
      IF(T+.1.GT.TR(3).AND.TR(3).GT.0.) JA=NOSS-NCNRS(1)-NCNRS(2)-NCNRS(
     *3)
      IF(T+.1.GT.TR(4).AND.TR(4).GT.0.) JA=NOSS-NTCN
      SRPRP=0.
      DO 46 I=1,JTB
      JJ=KJTB+I
      TEMP=0.
      TEMP1=0.
      DO 47 J=2,JA
      TEMP=TEMP+GAPTB(JJ,J)*GNU(J-1)
   47 TEMP1=TEMP1+ESPTB(JJ,J)*WDS(J-1)
      DELCP=GAPTB(JJ,1)+TEMP1*XJEXT
      HUMAX(1)=AMAX1(ABS(DELCP+TEMP1),HUMAX(1))
      DELCP=DELCP*E1+TEMP1
C***  END OF CHIP
      IF(KWTA(KV).EQ.2) GO TO 57
      TEMP=0.
      TEMP1=0.
      DO 48 J=2,JA
      TEMP=TEMP+GSPTB(JJ,J)*GNU(J-1)
   48 TEMP1=TEMP1+ESPTB(JJ,J)*WDS(J-1)
```

```
      DELCY=(GSPTB(JJ,1)+TEMP)*XJEXT
      HUMAX(2)=AMAX1(ABS(DELCY+TEMP1),HUMAX(2))
      DELCY=DELCY*E1+TEMP1
C***  END OF CHIY
C     UPDATE TTBL,CPTBL,CYTBL
   57 TOBL(JJ)=T
      CPOTBL(JJ)=APTB(JJ)+DELCP
      CYOTBL(JJ)=SITB(JJ)+DELCY
      IF(CYOTBL(JJ).GT.1.5) CYOTBL(JJ)=(1.5+SITB(JJ))/2.
      IF(CYOTBL(JJ).LT.-1.5) CYOTBL(JJ)=-(1.5-SITB(JJ))/2.
      IF(LPRINT.EQ.2)GO TO 75
      IF(NMAX.EQ.INMAX)GO TO 75
      IF(I.EQ.1.OR.I.EQ.NTB) GO TO 75
      GO TO 76
   75 CP=CPOTBL(JJ)*RAD
      AM=AMTB(JJ)*RAD
      CY=CYOTBL(JJ)*RAD
      SI=SITB(JJ)*RAD
      DEP=DELCP*RAD
      DEY=DELCY*RAD
      IF(DEPAND)WRITE(12)T,CP,AM,CY,SI,DEP,DEY,JJ
      WRITE(JO,49) T,CP,AM,CY,SI,DEP,DEY,JJ
   49 FORMAT(1X,7E14.7,I3)
   76 CONTINUE
   46 T=T+DT84
      KV=KV+1
   60 CONTINUE
      IF(DEPAND)WRITE(13)HUMAX(1),HUMAX(2),BETCON,KAT
      WRITE(JO,61)HUMAX(1),HUMAX(2),BETCON,KAT
   61 FORMAT(1X,13HDEL CHIP MAX=,E11.4,16H   DEL CHIY MAX=,E11.4,
     1 1GH   BETCON=,L2,6H   KAT=,I3)
C*****RESTORE NOSS FOR FORWARD RUN
      NOSS=NOSS-NTCN
      NOP1=NCM1+1
      IF(NOP1.LT.5)BETCON=.FALSE.
      IF(IPR.NE.0.AND.NOP1.LE.6) BETCON=.FALSE.
      IF(HUMAX(1).LT..CCS.AND.HUMAX(2).LT..CCS.AND.BETCON)NMAX=1
C*****UPDATE CONVERGENCE SCALE FACTORS*****
      IF(NOP1.LE.5)GO TO 64
      DO 63 I=1,NOFAR
      IF(ABS(CONVK(I,2)).LT.TOL) GO TO 63
      IF(ABS(CONVK(I,2)).LT.CONVF*ABS(CONVK(I,1)))GO TO 63
      IF(ABS(CONVK(I,2)-CONVK(I,1)).LT.ABS(CONVK(I,2)))GO TO 62
      WIBT(I)=WIBT(I)/2.
      GO TO 63
   62 WIBT(I)=WIBT(I)*2.
   63 CONTINUE
      IF(NTINV.NE.0) WIBT(NTINV)=WIBT(NTINV)/6.
      IF(NTNV.NE.0) WIBT(NTNV)=WIBT(NTNV)/6.
   64 DO 65 I=1,NOFAR
   65 CONVK(I,1)=CONVK(I,2)
      FREDP=0.
      DO 66 I=1,NK
      FREDP=FREDP+GNU(I)*ORHO(I)
      IF(NOP1.GT.2)GNU(I)=GNU(I)+XJEXT*WES(I)
   66 CONTINUE
      LA=NOSS+NTCN
      RETURN
      END
```

```
      SUBROUTINE AFRTN(NN)
C
C      COMPUTES OUTPUT PARAMETERS,PRINTS DATA,AND WRITES OUTPUT TAPE FOR
C     OUTPUT TABLES
C          NN      = INDEX FOR PRINTOUT IDENTIFICATION
      CHARACTER*60 HEAD
      CHARACTER*6 PISION,RID(50),AERONM(2),NAME,GGQ,ATMOSN
      COMMON/CCC/GRAPH,JC,DEMAND
      REAL LIFT
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CT2,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WD(15),WJET(15),
     *HEAD,FRINT(15),STEF(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FRFFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XP,SW,SU,SV,Q,
     *VIV(25),DVARS(13),CELXDW(15,3),DELXOR(7,4),CELXC(15,7),WZERO,ALT,
     *MACH,GDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12,A22,A32,XJEXT,BEL,
     *EYL,BEL,BHPA(15),EMMN,BSTEP(15),HAMB,HMXB,IPOLY,XMIAD
      COMMON/AGEN3/ITHF,IWD,IXR,JUMP,KAT,KCYTAB,NPAGE,KS1,KWTA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PINH,PPSFLG(15),ASYST(15),ICONSW,IFTLS,IPOLY,KINDB,KRDERB,PISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/AGEN4/CALPHA,CBETA,QCN,STH1,CTH1,XJ1(3),XJV(3)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,AMX,AMY,AMZ,
     *TMZ,THMFA,SIDE,HPAE,AZW,FANC,FLATC,FOM,VISC
      COMMON/AFPRN/VI,GAP,C1,C3,XINC,XNOD,XMLB,THETG,THETA,PHI,RNGO,RNGA
     *N,AZI,AZRE
      COMMON/ALAT/ALAT,ALONGO,ALTLS,NTABLE
      COMMON/APRINT/ELEIN,ELEOUT,DINBD,DCUTBC,FABT(3),CAX,CNF,CM2,CS,
     *CMX,CPY,SEPACH,DRAG,LIFT,XLAMDA,XKAUD
      COMMON/ARDC/FAT(14),ATMOSN
      COMMON/ASTOR/ASTC(4,67)
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,G,CMLE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,FTN,PTKG,FSFTM,PFN,PFKG,PSFFP
      COMMON/FORINT/HBANK(2),T,TT,W,U,V,X,Y,Z,XM1,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,MINC,KRDER,EU,EL,AYL,HPX,HPN,HAP
      COMMON/GARBGE/DX,DYI,DZI,DXDY,UDXDYI,UDXDY2,XCG,YCG,TX1,TX2,TY1,TY
     *2,THAT(3),TXS,TYS,TAU,SINRHO
      COMMON/JT/JT
      COMMON/LPRINT/LFRINT,INMAX,PREDP,QY,TOL
      COMMON/REST/LSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NCFAR,NIBT(15)
      COMMON/MPS/TIMMFS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MFS1,MFS2,TELIF,GLIMIT
      COMMON/NAME/NAME(128),WORD(129)
      COMMON/OUTPUT/OUTTH(6),OUTTV(6),OUTISP(6),CLTME(6)
      COMMON/REYNO/REYNO,VE,SWE,SUE,SVE
      COMMON/SRMEXT/SRMEXT
      COMMON/TCL/TCL(6),TVL(6),TLL(6),RATIO
      COMMON/VISC/XMU
      DIMENSION D(3,3)
      DIMENSION XS(3),DS(3),XDDS(3),XEC(3),XDEC(3),XCDEC(3),AT(3,3),
     *WET(3,3),DWET(3,3),DDWET(3,3),CTEF1(9),CTEF2(9),WORDRS(12)
      DIMENSION WLCXT(5),HFADT(9)
      EQUIVALENCE (SRP,IR1,ASTO(1,47))
      DATA(RID(I),I=1,50)/6H      ,6H      ,6HLIFT-O,6HFF    ,
     *6HIGNITI,6HON    ,6H      ,6H      ,6HTHRUST,6H EVENT,
     *6HBEGIN ,6HCCAST ,6HBEGIN ,6HTILT  ,6HEND TI,6HLT    ,
     *6HSRB SE,6HPARATE,6HBEGIN ,6HMINH  ,6HEND CC,6HAST   ,
     *6HLBM SE,6HPARATE,6HLBM IG,6HNITION,6HMAX CY,6HN PRES,
     *6HINJECT,6HICN   ,6HBEGIN ,6HCHIFRI,6HLEIG1,6H DROP ,
     *6HBEGIN ,6HRCLL  ,6HEND RO,6HLL    ,6HINTERP,6HEDIATE,
     *6HRTLS/A,6HOA    ,6HG-LIMI,6HT     ,6H      ,6H      ,
```

```
     *6H       ,6H       ,6HMPS TH,6HROTTLE/
      DATA AERONM/6H  LN 7,6H9 BIV)/
      DATA (NAME(I),I=1,114)/
     *6H  TIME,6H VSUBE,6H  GAME,6H   AZE,6H THRST,6H   XMLB,
     *6H      R,6H VSUBI,6H  GAMI,6H   AZI,6H GCLAT,6H   LONG,
     *6H    ALT,6H VSUBR,6H  GAMR,6H   AZR,6H GDLAT,6H   NODE,
     *6H  ZB-X,6H  XB-Y,6H  YB-Z,6H ZDB-L,6H XDB-L,6H YDB-V,
     *6H  CHIP,6H  CHIY,6H  CHIR,6H  TIME,6H LTIME,6H LNGME,
     *6H THERI,6H PSIRI,6H PHIRI,6H  INCL,6H RANGE,6H RNGAN,
     *6H THPPS,6H THSRP,6H THLBM,6H THOMS,6H THRCS,6H     TH,
     *6H TVPPS,6H TVSRP,6H TVLBM,6H TVOMS,6H TVRCS,6H   TVAC,
     *6H ISPPS,6H ISPSR,6H ISPLB,6H ISPOM,6H ISPRC,6H    ISP,
     *6H WDPPS,6H WDSRP,6H WDLBM,6H WDOMS,6H WDRCS,6H   WDOT,
     *6H MPS F,6H SRM F,6H LBM F,6H OMS F,6H RCS F,6H   FUEL,
     *6H MPS D,6H SRM D,6H LBM D,6H OMS D,6H RCS D,6H SPENT,
     *6H DPPPS,6H DPSRP,6H DPLBM,6H DPOMS,6H DPRCS,6H   DELP,
     *6H DYPPS,6H DYSRP,6H DYLBM,6H DYOMS,6H DYRCS,6H   DELY,
     *6H    TAU,6H SRMAE,6H PDOME,6H SRMET,6H ORBET,6H CH VL,
     *6H  TA L,6H  GV L,6H  OR L,6H BKP L,6H GIM L,6H ID VL,
     *6H  MACH,6H ALPHA,6H  BETA,6H   FAA,6H   FAS,6H   FAN,
     *6H     Q,6H CALPH,6H DBETA,6H   FAB,6H   FPB,6H   FNB,
     *6H  DRAG,6H  LIFT,6H VWIND,6H   AZL,6H  HEAT,6H HRATE/
      DATA (NAME(I),I=115,128)/
     *6H PRESS,6H    RHO,6H   TEMP,6H REYNO,6H DELFC,6H DELYC,
     *6H EMACH,6H   INEC,6H OUTBD,6H AXACC,6H NRACC,6H LTACC,
     *6H    XCG,6H    ZCG/
      DATA COO/6H  OPEN/
C     INITIALIZATION DATA FOR LOAD INDICATOR CALCULATIONS
C     SFT=SRB DYNAMIC THRUST SCALE FACTOR(PER SRB)
C     SFD=SRB DRAG SCALE FACTOR(PER SRB)
C     SFDO=ORBITER DRAG SCALE FACTOR
C     DENLOX=LOX DENSITY(LBS/CU IN)
C     UPRESS=ULLAGE PRESSURE(LBS/IN2)
C     SUMPS=MPS THRUST BUILDUP PROPELLANT FROM SSME IGN TO T=0(LBS)
C     BLEED=LOX OVERBOARD BLEED(LBS)
C     OFOB=LOX OVERFILL AND DRAINBACK(LBS)
      DATA SFT/1./, SFD/.15/, DENLOX/.04C989/, UPRESS/25./
      DATA PLOAD/1/, SFDO/.38/
C     TABLE OF LOX HEAD(IN) VS LOX WEIGHT(MLBS)
      DATA HEADT/0.,270.,300.,332.,370.,410.,465.,507.,550./
      DATA WLOXT/0.,.8,.9,1.0,1.1,1.2,1.3,1.35,1.37/
      IF(LSE.EQ.1) RETURN
      IF(LPRINT.GT.1)GO TO 81
      IF(NMAX.NE.0.AND.NMAX.NE.INMAX) RETURN
   81 XM=XMIAD+XMALG
      R=SQRT(X*X+Y*Y+Z*Z)
      C(1,2)=X/R
      C(2,2)=Y/R
      C(3,2)=Z/R
      CTH=A(1,2)*O(1,2)+A(2,2)*O(2,2)+A(3,2)*O(3,2)
      SPHI=A(1,1)*O(1,2)+A(2,1)*O(2,2)+A(3,1)*O(3,2)
      CPHI=A(1,3)*O(1,2)+A(2,3)*O(2,2)+A(3,3)*O(3,2)
      PHI=ATAN2(SPHI,CPHI)
      IF(T.LT..1)PHI=CTZ*OMEGA
      PHIO=PHI
      PHI=(PHI-OMEGA*(T+CTZ))*RAD-ALONGO
      VI=(W*W+U*U+V*V)**.5
      GAM=0.
      IF(ABS(VI).GT..1E-05) GAM=ASIN((W*X+U*Y+V*Z)/(R*VI))
      C1=VI*R*COS(GAM)
      C3=VI*VI-2.*CMUG/R
      CHIP=ATAN2(SCHIP,CCHIP)
      CHIY=ATAN2(SCHIY,CCHIY)
      CHIR=ATAN2(SCHIR,CCHIR)
      IF(CHIY.GT.PI) CHIY=CHIY-2.*PI
C
C     CALCULATE LOAD INDICATORS
C
```

```
      MLOAD=1
      FLOX=RATIO*ZAP(1)*FFKG/1.E+06
      CALL AMULG(1,MLOAD,9,PLOX,WLOXT,HDLOX,HEADT,0,0)
      HITW=SFT*THMFA/XP
      FDOME=WPRESS+DENLOX*HITW*HDLOX/GZERO
      SRMET=0.
      WTSRM=0.
      IF(ITHR.GT.NCEVNT(1)) GO TO 82
      WTSRM=SRMIRT*PTKG+ZAP(2)
      SRMET=(SFT*TCL(2)-HITW*WTSRM)/TNE(2,ITHR)-SFD*FAA
      SRMET=SRMET*FFN
   82 CRBET=TCL(1)-HITW*(XM-WTSRM)-SFDO*FAA
      CRBET=CRBET*FFN
C
C     COMPUT YAW,PITCH,ROLL ATTITUDE IN BOOST REFERENCE COORD SYSTEM
C     FOR PRINT ONLY
C
      IF(T .GT. TLIFT) GO TO 108
      PSIRI=CHIY*RAD+AA*RAD
      THERI=90.
      PHIRI=CHIR*RAD
      GO TO 30
  108 CONTINUE
C     YAW
      PSIRI=ATAN2(SA*SCHIP*CCHIY+SCHIY*CA,CA*SCHIP*CCHIY-SCHIY*SA)*RAD
      IF(PSIRI.LT. 0.) PSIRI=PSIRI+360.
C     PITCH
      THERI=ASIN(CCHIP*CCHIY)*RAD
C     ROLL
      PHIRI=ATAN2(SCHIP*SCHIR+CCHIR*SCHIY*CCHIP,CCHIR*SCHIP-SCHIR*SCHIY*
     *CCHIP)*RAD
   30 ALTP=ALT
      STH=(1.-CTH*CTH)**.5
      RSTH=R*STH
      D(1,3)=(D(1,2)*CTH-A(1,2))/STH
      D(2,3)=(D(2,2)*CTH-A(2,2))/STH
      D(3,3)=(D(3,2)*CTH-A(3,2))/STH
      D(1,1)=(-Y*A(3,2)+Z*A(2,2))/RSTH
      D(2,1)=(X*A(3,2)-Z*A(1,2))/RSTH
      D(3,1)=(-X*A(2,2)+Y*A(1,2))/RSTH
      GAMR=0.
      IF(ABS(VR).GT..1E-05) GAMR=ASIN((SW*X+SU*Y+SV*Z)/(R*VR))*RAD
      GAMRE=0.
      IF(ABS(VE).GT..1E-05) GAMRE=ASIN((SWE*X+SUE*Y+SVE*Z)/(R*VE))*RAD
      VTH=D(1,3)*SW+D(2,3)*SU+D(3,3)*SV
      VPH=D(1,1)*SW+D(2,1)*SU+D(3,1)*SV
      AZR=ATAN2(VPH,-VTH)*RAD
      VTHE=D(1,3)*SWE+D(2,3)*SUE+D(3,3)*SVE
      VPHE=D(1,1)*SWE+D(2,1)*SUE+D(3,1)*SVE
      AZRE=ATAN2(VPHE,-VTHE)*RAD
      VTH=D(1,3)*W+D(2,3)*U+D(3,3)*V
      VPH=D(1,1)*W+D(2,1)*U+D(3,1)*V
      AZI=ATAN2(VPH,-VTH)
      THET=ACOS(CTH)
      THETA=90.-THET*RAD
      THETG=ATAN2(SIN(THETA/RAD),UMF*UMF*COS(THETA/RAD))*RAD
      CINC=SIN(AZI)*SIN(THET)
      XINC=ACOS(CINC)
      XNOD=(XNOD+ATAN2(CTH*VPH,VTH))*RAD
      A1=ABS(CMUE/C3)
      P=C1*C1/CMUE
      ECC=1.+P*C3/CMUE
      RESAVE=RE
      IF(ECC.LT.0.)GO TO 31
      ECC=SQRT(ECC)
      APOGEE=((P/(1.0-ECC))-RE)/1852.0
      PERIGE=((P/(1.0+ECC))-RE)/1852.0
      IF(RE-P/(1.+ECC).LT.RE/100.)GO TO 31
      CETANW=(P-R)/(R*ECC)
```

```
      ETANW=ABS(CETANW)
      IF(ETANW.LE.1.)GO TO 140
      ETANW=0.
      GO TO 142
  140 ETANW=ACOS(ETANW)
  142 IF(CETANW.LT.0.)ETANW=PI-ETANW
      IF(GAP.LT.0.)ETANW=2.*PI-ETANW
      CETATN=(P-RE)/(RE*ECC)
      ETATN=ABS(CETATN)
      IF(ETATN.LE.1.)GO TO 150
      ETATN=0.
      GO TO 151
  150 ETATN=ACOS(ETATN)
  151 IF(CETATN.LT.0.)ETATN=PI-ETATN
      CELETA=2.*PI-ETATN-ETANW
      CBGENW=(A1-R)/(A1*ECC)
      BGENOW=ABS(CBGENW)
      IF(BGENOW.LE.1.)GO TO 152
      BGENOW=0.
      GO TO 153
  152 BGENOW=ACOS(BGENOW)
  153 IF(CBGENW.LT.0.)BGENOW=PI-BGENOW
      IF(GAP.LT.0.)BGENOW=2.*PI-BGENOW
      CBGETN=(A1-RE)/(A1*ECC)
      BGETHN=ABS(CBGETN)
      IF(BGETHN.LE.1.)GO TO 154
      BGETHN=0.
      GO TO 155
  154 BGETHN=ACOS(BGETHN)
  155 IF(CBGETN.LT.0.)BGETHN=PI-BGETHN
      BGETHN=2.*PI-BGETHN
      BELBGE=BGETHN-BGENOW
      CTI=(CELBGE-ECC*(SIN(BGETHN)-SIN(BGENOW)))/SQRT(CMUE/(A1*A1*A1))
      THT1=ACOS(CTH*COS(CELETA)+STH*SIN(CELETA)*COS(AZI))
      DLPHI=ASIN(SIN(CELETA)*SIN(AZI)/SIN(THT1))
      PHIMP=PHI+(DLPHI-OMEGA*DTI)*RAD
      THIMP=90.-THT1*RAD
   31 CONTINUE
      RE=RESAVE
   90 PHIPP=ATAN2(CTHL,STHL)
      DLTL=DTZ
      DO 79 I=1,2
      RNGAN=ACOS(SIN(PHIPP)*SIN(THETA/RAD)+COS(PHIPP)*COS(THETA/RAD)*
     * COS(PHI+ALONG(I)/RAD+OMEGA*DLTL))
      IF(I.NE.1) GO TO 79
      IF(RNGAN.LT..0001) RNGAN=0.
      RNGO=RNGAN*RE
      DLTL=T+DTZ
   79 CONTINUE
      DTX=2.*TX2-TX1
      DTY=TY1-2.*TY2
      DTXP=DTX*UDXDYI+DTY*SINRHO
      DTYP=-DTX*SINRHO+DTY*UDXDYI
  155 THDX=AMX+DTYP/DZI+TZZ/DYI
      THDY=AMY+DTXP/DZI-TZZ*DX
      THDZ=AMZ-TXX/DYI+TYY*DX
      IF(ALTP.LT.0..AND.T.LT.10.) ALTP=0.
      N=2*NN-1
   50 IF(LINES)55,55,60
C     WRITE HEADER
   55 CONTINUE
      XPAGE=XPAGE+1
      WRITE(JO,56)CASE,HEAD,XPAGE,AERONM
   56 FORMAT(1H1,12HMASTRE CASE=,F11.4,10X,A60,10X,4HPAGE,I3,8H (AERO -
     *,2A6)
      WRITE(JO,57) ATMOSN
   57 FORMAT(111X,8H (ATMOS-,A6,1H))
   41 LINES=35
   60 CONTINUE
```

```
WORD(1)=T
WORD(2)=VE
WORD(3)=GAMRE
WORD(4)=AZRE
WORD(5)=(OUTTH(1)+OUTTH(2)+OUTTH(3)+OUTTH(4)+OUTTH(5)+OUTTH(6))*
*FFN
WORD(6)=XM*PFKG
WORD(7)=R
WORD(8)=VI
WORD(9)=GAM*RAD
WORD(10)=AZI*RAD
WORD(11)=THETA
WORD(12)=PHI
WORD(13)=ALTF
WORD(14)=VR
WORD(15)=GAMR
WORD(16)=AZR
WORD(17)=THETG
WORD(18)=XNOD
WORD(19)=X
WORD(20)=Y
WORD(21)=Z
WORD(22)=W
WORD(23)=U
WORD(24)=V
WORD(25)=CHIF*RAD
WORD(26)=CHIY*RAD
WORD(27)=CHIR*RAD
WORD(28)=DTI
WORD(29)=THIPP
WORD(30)=PHIPP
WORD(31)=THEFI
WORD(32)=PSIFI
WORD(33)=PHIFI
WORD(34)=XINC*RAD
WORD(35)=RNGD/1852.
WORD(36)=RNGAN*RAD
WORD(37)=OUTTH(1)*FFN
WORD(38)=OUTTH(2)*FFN
WORD(39)=OUTTH(3)*FFN
WORD(40)=OUTTH(4)*FFN
WORD(41)=OUTTH(5)*FFN
WORD(42)=OUTTH(6)*FFN
WORD(43)=OUTTV(1)*FFN
WORD(44)=OUTTV(2)*FFN
WORD(45)=OUTTV(3)*FFN
WORD(46)=OUTTV(4)*FFN
WORD(47)=OUTTV(5)*FFN
WORD(48)=OUTTV(6)*FFN
WORD(49)=OUTISP(1)
WORD(50)=OUTISP(2)
WORD(51)=OUTISP(3)
WORD(52)=OUTISP(4)
WORD(53)=OUTISP(5)
WORD(54)=OUTISP(6)
WORD(55)=OUTBD(1)*FFKG
WORD(56)=OUTBD(2)*FFKG
WORD(57)=OUTBD(3)*FFKG
WORD(58)=OUTBD(4)*FFKG
WORD(59)=OUTBD(5)*FFKG
WORD(60)=OUTBD(6)*FFKG
WORD(61)=ZAP(1)*FFKG
WORD(62)=ZAP(2)*FFKG
WORD(63)=ZAP(3)*FFKG
WORD(64)=ZAP(4)*FFKG
WORD(65)=ZAP(5)*FFKG
WORD(66)=ZAP(6)*FFKG
WORD(67)=PROF(1)-WCRD(61)
WORD(68)=PROF(2)-WCRD(62)
WORD(69)=PROF(3)-WCRD(63)
```

```
      WORD(70)=PROF(4)-WCRD(64)
      WORD(71)=PROF(5)-WCRD(65)
      WORD(72)=PROF(6)-WCRD(66)
      DO 70 I=1,6
      WORD(I+72)=0.
      WORD(I+78)=0.
      IF(TCL(I).LT.1.E-04) GO TO 70
      WORD(I+72)=ATAN2(-TVL(I),TCL(I))*RAD
      WORD(I+78)=ATAN2(TLL(I),TCL(I))*RAD
   70 CONTINUE
      WORD(85)=TAU
      WORD(86)=SRMEXT
      WORD(87)=PDOME
      WORD(88)=SRMET
      WORD(89)=ORBET
      WORD(90)=ZAP(8)-ZAP(9)-ZAP(10)
      WORD(91)=ZAP(12)
      WORD(92)=ZAP(11)
      WORD(93)=ZAP(13)
      WORD(94)=ZAP(9)
      WORD(95)=ZAP(10)
      WORD(96)=ZAP(8)
      WORD(97)=XMACH
      WORD(98)=ALF*RAD
      WORD(99)=ALFY*RAD
      WORD(100)=FAA*PFN
      WORD(101)=SIDE*PFN
      WORD(102)=FAN*PFN
      WORD(103)=Q*FSFFP
      WORD(104)=WORD(103)*WORD(98)
      WORD(105)=WORD(103)*WORD(99)
      WORD(106)=FABT(1)*PFN
      WORD(107)=FABT(3)*PFN/.3048
      WORD(108)=FABT(2)*PFN
      WORD(109)=DRAG*PFN
      WORD(110)=LIFT*PFN
      WORD(111)=WMAG
      WORD(112)=AZW*RAD
      WORD(113)=ZAP(7)
      WORD(114)=DVAR(14)
      WORD(115)=PAT(2)*208.85434
      WORD(116)=PAT(6)
      WORD(117)=PAT(12)
      WORD(118)=REYNO
      WORD(119)=0.
      WORD(120)=0.
      IF(TYY.LT.1.E-04) GO TO 71
      WORD(119)=ATAN2(-TXX,TYY)*RAD
      WORD(120)=ATAN2(TZZ,TYY)*RAD
   71 CONTINUE
      WORD(121)=SEMACH
      WORD(122)=ELEIN
      WORD(123)=ELEOUT
      WORD(124)=THMFA/XM/GZERO
      WORD(125)=FANC/XM/GZERO
      WORD(126)=FLATC/XM/GZERO
      WORD(127)=YCE*39.3701
      WORD(128)=XCE*39.3701
      WORD(129)=ZAP(15)
C
C****************** RANGE SAFETY CALCULATIONS***************************
C
C
C     COMPUTE X,XDOT,XDOT IN EARTH FIXED,EARTH CENTERED COORD
C     SYSTEM(PACSS11)
C
C     XEC=[A][WET][AT]XS
C     XDEC=[A][DWET][AT]XDS
C     XDDEC=[A][DDWET][AT]XDDS
C
```

```
      XS(1)=X
      XS(2)=Y
      XS(3)=Z
      XDS(1)=W
      XDS(2)=U
      XDS(3)=V
      XDDS(1)=DVAR(1)
      XDDS(2)=DVAR(2)
      XDDS(3)=DVAR(3)
      WT=OMEGA*T
C     DWET) MATRIX FORMED BY A POSITIVE OMEGA(WE) ROTATION ABOUT
C     THE Y-EC AXIS
      DO 8 I=1,3
      DO 8 J=1,3
      WET(I,J)=0.
      DWET(I,J)=0.
    8 DDWET(I,J)=0.
      WET(1,1)=COS(WT)
      WET(3,1)= SIN(WT)
      WET(2,2)=1.
      WET(1,3)=-SIN(WT)
      WET(3,3)=COS(WT)
C
      DWET(1,1)=-OMEGA*SIN(WT)
      DWET(3,1)= OMEGA*COS(WT)
      DWET(1,3)=-DWET(3,1)
      DWET(3,3)=DWET(1,1)
C
      DDWET(1,1)=-OMEGA**2*COS(WT)
      DDWET(3,1)=-OMEGA**2*SIN(WT)
      DDWET(1,3)=-DDWET(3,1)
      DDWET(3,3)=DDWET(1,1)
C
      DO 3 I=1,3
      DO 3 J=1,3
    3 AT(J,I)=A(I,J)
C
      CALL TRNSFM(CTER1,AT,XS)
      CALL TRNSFM(CTER1(4),AT,XDS)
      CALL TRNSFM(CTER1(7),AT,XDDS)
C     POSITION
      CALL TRNSFM(CTER2,WET,CTER1)
      CALL TRNSFM(XEC,A,CTER2)
C     VELOCITY
      CALL TRNSFM(CTER2,DWET,CTER1)
      CALL TRNSFM(CTER2(4),WET,CTER1(4))
      DO 9 I=1,3
    9 CTER2(I)=CTER2(I)+CTER2(I+3)
      CALL TRNSFM(XDEC,A,CTER2)
C     ACCELERATION
      CALL TRNSFM(CTER2,DDWET,CTER1)
      CALL TRNSFM(CTER2(4),DWET,CTER1(4))
      CALL TRNSFM(CTER2(7),WET,CTER1(7))
      DO 10 I=1,3
   10 CTER2(I)=CTER2(I)+2.*CTER2(I+3)+CTER2(I+6)
      CALL TRNSFM(XDDEC,A,CTER2)
C
      XORDRS(1)=XEC(2)
      XORDRS(2)=XEC(3)
      XORDRS(3)=XEC(1)
      XORDRS(4)=XDEC(2)
      XORDRS(5)=XDEC(3)
      XORDRS(6)=XDEC(1)
      XORDRS(7)=XDDEC(2)
      XORDRS(8)=XDDEC(3)
      XORDRS(9)=XDDEC(1)
      XORDRS(10)=0.
      XORDRS(11)=0.
      XORDRS(12)=0.
```

```
C
C
C ****************** END RANGE SAFETY CALCULATIONS ********************
C
      WRITE(6,61) RID(N),RID(N+1),(NAME(I),WORD(I),I=1,128)
     *,DCO,ZAP(14)
   61 FORMAT(1X,2A6/(4X,6(A6,E15.8)))
   62 LINES=LINES-23
C
      IF(NMAX.EQ.0) GO TO 101
      RETURN
C
C***  WRITE OUTPUT FOR REPORT
  101 WRITE(9) RID(N),RID(N+1),WORD,MISION
  100 RETURN
C
      SUBROUTINE TRNSFM(C,E,B)
      DIMENSION C(3),E(3,3),B(3)
      DO 10 I=1,3
      C(I)=0.
      DO 10 J=1,3
   10 C(I)=C(I)+E(I,J)*B(J)
      RETURN
      END
```

```
      SUBROUTINE AREAIN
      COMMON/IUSE/IUSE
      COMMON/LLFLAG/LLFLAG
      CALL FAGIT
      IUSE=0
1     CALL GOUT(' SELECT FROM THE FOLLOWING MENU:',22)
      CALL GOUT(' 0 - TERMINATE',15)
      CALL GOUT(' 1 - EXECUTE',13)
      CALL GOUT(' 2 - SELECT/DISPLAY/EDIT/SAVE INPUT.',38)
      CALL GOUT(' 3 - WRITE FILE FOR PLOTS AND TABLES',37)
      CALL GOUT(' 4 - PRINTOUT MENU',19)
      CALL RDATAI(1,ILSE,$1)
      IF(IUSE.LT.0 .OR. IUSE.GT.6)GO TO 1
      IUSE=ILSE+1
      GO TO(10,11,12,13,14),IUSE
C-----------------------------TERMINATE-------------------------
10    CALL PPSTRM
      STOP
C-----------------------------EXECUTE-----------------------------
11    CALL GOUT(' YOU ARE NOW EXECUTING.',24)
      LLFLAG=1
C-----------------------------DISPLAY/EDIT INPUT-------------------
12    IUSE=ILSE-1
      RETURN
C-----------------------------WRITE FILE FOR PLOTS/TABLES----------
13    CALL CCPLOF
      GO TO 1
C-----------------------------PRINTOUT MENU----------------------
14    CONTINUE
      GO TO 12
      END
```

```
      SUBROUTINE ATHREV
C
C     THRUST EVENT CONTROL ROUTINE;SETS UP VEHICLE GEOMETRY,AERODYNAMIC
C     NUMBER OF ENGINES,THRUST LEVELS,FLOWRATES,TRIGGER FLAGS
C
      CHARACTER*60 HEAD
      CHARACTER*6 PISION,HEADER
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINM,
     *CT2,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WO(15),WJET(15),
     *HEAD,PRINT(15),STEF(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHRDO1,CHIRO,FA2,BC13,21,CBAXIL,TCHIR,VRCUT,FPREAC,CHVEL
      COMMON/AGEN2/AZ,SA,CA,ALF,ALFY,CHIF,SCHIP,(CHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SM,SU,SV,Q,
     *VIV(25),DVARS(13),CELXDW(15,3),DELXOR(7,4),CELXD(15,7),WZERO,ALT,
     *MACH,QDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *BYL,BEL,BHMAX(15),BHMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAO
      COMMON/AGEN3/ITER,IWO,IXR,JUMP,KAT,KCYTAB,KPAGE,KS1,KWIA(4),LINES,
     *ABGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWO(15),NVAT,NWVNT,IHEAD,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IFTLS,IPOLY,KINOB,KRDERB,PISION,
     *IRTFLG,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,APX,APY,AMZ,
     *TMZ,TPPFA,SICE,BPAE,AZW,FANC,FLATC,FOM,VISC
      COMMON/AFPRN/VI,GAM,C1,C3,XINC,XNCC,XMLB,THETG,THETA,PHI,RAGO,RNGA
     *A,AZI,AZRE
      COMMON/AUTC/KOB(40),SAVCP(40),XLAME(40,10),CP2
      COMMON/AVGGP/XGPA,YGPA,ZGPA
      COMMON/BGEN3/NOSS,TB,NENT,NACT,NCP1,NF(4)
      COMMON/BODY/ATBCHT(15,2),ATXCG(15,2),ATYCG(15,2),AXLEN(2),AXREF(2)
     *,AYREF(2),KPI(2),KYI(2),API(10,2),AYI(10,2)
      COMMON/BODYF/TBCHT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,M1,M2,M5,
     *M6,M7,M8,M9,PSO,PS1,MSRW,AP(10),AY(10)
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,C,CPLE,OMEGA,ALTI,ALT2,PSL,
     *EZERO,FTN,FTNG,FSFTM,PFN,PFNG,PSFFP
      COMMON/CONTRC/TTEL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/DELFAB/DELALT(25),DELFAB(25),M3C,M3I,ITOL(15)
      COMMON/DELX/CELX(7)
      COMMON/DELXH/CELXH(4)
      COMMON/ELEVON/EPACH(25,2),ELEVON(25,4),M10,M11
      COMMON/ENPUT/PSIFST(6,4),HEADER(20),NVRST(4),KCCRES(6,4),LAST
      COMMON/FMXX/FMXX,MSW,WG,SPOA
      COMMON/FORINT/HBANK(2),T,TT,VAR(25)                ,DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRE10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,NINC,KRDER,EU,EL,AYL,HPX,HPN,HAP
      COMMON/IPR/IFR,WFMX,ILAST,KOT(15),FRK4,PRTCT
      COMMON/IRTAGA/IRTAGA
      COMMON/ISTART/JSTART,ISTART(2),STARTV(30,2),SVXINC,SVXNOD,SVAZRE
      COMMON/ITHRO/ITHRO
      COMMON/JT/JT
      COMMON/JUMPST/JLPPST,TSAVET,WZERST
      COMMON/KP/KP,KY
      COMMON/LOADS/CLCAD(16),QTIME(16),CVALUE(8),SAVE(8,8)
      COMMON/M20/M17,M20,M21,M22,M23,M24
      COMMON/LBM/TIMLEP(30),THRLBM(30),WCTLBM(30),PLBM(2),MLBM1,MLBM2,
     *PLBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MPS1,MPS2,TELIP,GLIMIT
      COMMON/OLOK/CLOK,WINT
      COMMON/OMSRCS/FCPS(15),FRCS(15),WOGMS(15),WCRCS(15),WOOUMP(15)
      COMMON/REST/WSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NOFAR,WIBT(15)
      COMMON/RESTX/RESTX(16,4)
      COMMON/RICONT/TTABLE(20),SIGMA,SIGTAB(20),M15,M16
      COMMON/SRM/SRMTTE(250),SRMFTB(250),SRMWOT(250),SRMAE(2),MSRB
      COMMON/TABISP/MTAU,TIMISP(20),TRFISP(20)
      COMMON/TABLK/MCON1,MCON2,MCON3,MCON4
      COMMON/VAERO/VAEROC(10,19),M18,M19,MBS18,MES19
      COMMON/VEHSTE/VEHSTE
```

```
      COMMON/XTRA/XMD(15),CFR(15),F(15),CISP(15)
      DIMENSION ANE(2)
C*****   THIS ROUTINE TAKES SPECIAL FORINT COMMON*****
      IF(ITHR.NE.JUMP-1) CALL APRTN(1)
   10 JT=ITHR
      IPRT=0
      IF(JT.EQ.JUMP-1)I=IZERO
      ITHR=ITHR+1
      IF(ITHR.EQ.1) GO TO 11
      IF(TNE(3,ITHR).EQ.0..AND.TNE(3,ITHR-1).LT.1.E-05) IPRT=13
      IF(TNE(3,ITHR).LT.1.E-05.AND.TNE(3,ITHR-1).GT.0.) IPRT=12
   11 CONTINUE
      L=1
      IF(ITHR.LT.ICONSL) L=2
      MMIN=0
      DO 204 I=1,15
      IF(NSYST(I).NE.L) GO TO 204
      IF(MMIN.EQ.0) MMIN=I
      MMAX=I
  204 CONTINUE
      IF(INT(TNE(L,ITHR)+.1).LT.MMAX-MMIN+1) MMAX=INT(TNE(L,ITHR)+.1)+
     *MMIN-1
      XGPA=0.
      YGPA=0.
      ZGPA=0.
      DO 206 I=MMIN,MMAX
      XGPA=XGPA+ENEDAT(3,I)
      YGPA=YGPA+ENEDAT(1,I)
  206 ZGPA=ZGPA+ABS(ENEDAT(2,I))
      XGPA=XGPA/TNE(L,ITHR)
      YGPA=YGPA/TNE(L,ITHR)
      ZGPA=ZGPA/TNE(L,ITHR)
      NTRG12=-1
      IF(MPSFLG(ITHR).NE.2) GO TO 210
      NTRG12=1
      DO 208 I=1,15
      IF(TIMMPS(I).GT.T) GO TO 209
  208 CONTINUE
  209 ITHRO=I
      TV13=TIMMPS(ITHRO)
  210 CONTINUE
      IF(ITHR .EQ. 16) GO TO 786
      TIME(1,ITHR)=T
      TIME(1,ITHR+1)=T+TAUT(ITHR)
      IF(NVNT.EQ.0) GO TO 201
      DO 200 I=1,NNVNT
      J=NOWE(I)
  200 TIME(2,I)=TIME(1,J)+TAUW(I)
      TV12=TIME(2,IWO+1)
  201 IF(ITHR.EQ.1) GO TO 785
      IF(ITHR.NE.IFOL+1) GO TO 50
      IPOLY=1
      KP1=KFI(1)
      IF(KDB(KP1+18).EQ.0.AND.KDB(KP1+19).EQ.1) CFTBL(KP1+2)=ATAN2(SCHIP
     *,CCHIF)*RAD
      KP=KP1(2)
      KY1=KYI(1)
      IF(KDB(KY1+25).EQ.0.AND.KDB(KY1+26).EQ.1) CYTBL(KY1+2)=ATAN2(SCHIY
     *,CCHIY)*RAD
      KY=KYI(2)
      KP1=KFI+2
      KY1=KY1+2
      CALL FIND(CFTBL(KP1),AP,TTBL(KP1),KP)
      CALL FIND(CYTBL(KY1),AY,TTBL(KY1+15),KY)
      DO 49 I=1,10
      API(I,2)=AP(I)
   49 AYI(I,2)=AY(I)
   50 CONTINUE
      IF(JUMP.GT.NCEVNT(1).AND.ITHR.EQ.JUMP) GO TO 7579
```

```
      IF(LSTGE(ITHR).NE.LSTGE(ITHR-1)) GO TO 7979
      GO TO 786
 7979 CONTINUE
      IPRT=9
      IF(ITHR.EQ.JUMP) IPRT=3
      IF(LSB.EQ.1)GO TO 785
      DO 203 I=1,NFB
  203 XIV(I)=VAR(I)
      XIV(7)=0.
      ISAVET=T
      SCHI=CHISAV
      SYCHI=CHIY
      JUPPST=ITHR
  785 LSTG=LSTGE(ITHR)
      IF(ITHR.EQ.NCEVNT(1)+1) VCHSTG=VAR(15)-VAR(16)-VAR(17)
      PCON1=1
      PCON2=1
      PCON3=1
      PCON4=1
      P1=1
      P2=0
      P5=1
      P6=0
      P7=1
      P8=1
      P9=0
      P10=1
      P11=1
      P15=1
      P16=1
      P17=1
      P18=1
      P19=0
      P20=1
      P21=1
      P22=1
      P23=1
      P24=1
      P30=1
      P31=0
      ETFUEL=0.
      ANE(1)=0.
      ANE(2)=0.
      PSRB=1
      PSG=1
      PS1=1
      PSRW=0
      PLBM1=1
      PLBM2=0
      PPS1=1
      PPS2=1
      PTAU=1
      XLEN=AXLEN(LSTG)
      XREF=AXREF(LSTG)
      YREF=AYREF(LSTG)
      DO 20 I=1,15
      TBEWT(I)=ATBEWT(I,LSTG)*PTWG
      TXCG(I)=ATXCG(I,LSTG)
      TYCG(I)=ATYCG(I,LSTG)
   20 CONTINUE
  786 CONTINUE
      IF(NMAX.NE.0.OR.JUPP.NE.1) GO TO 788
      IF(JT.NE.NCEVNT(1).AND.JT.NE.NVNT) GO TO 788
      LST=1
      IF(JT.EQ.NVNT) LST=2
      J=4*LST-3
      DO 80 I=1,8
      SAVE(J,I)=CTIME(I)
      SAVE(J+1,I)=CLOAD(I)
```

```
      CLOAD(I)=OVALUE(I)
      CTIME(I)=T
      K=I+8
      SAVE(L+2,I)=CTIME(K)
      SAVE(L+3,I)=CLOAD(K)
      CLOAD(K)=OVALUE(I)
   80 CTIME(K)=T
  788 CONTINUE
      IF(ITHR .EQ. 16) GO TO 40
      NTRG2=2
      TV2=T+TAUT(ITHR)
      NTRG9=-2
      IF(MPSFLG(ITHR).EQ.4) NTRG9=2
      HNM=STEP(ITHR)
      HMX=PRINT(ITHR)
      TV1=AINT(T+1.E-03)+PRINT(ITHR)
      IF(ITHR.NE.JLMP)XMAUG=XMAUG-WJET(JT)
      IF(NTRG9.LT.0) GO TO 40
      DO 38 I=1,15
      IF(NSYST(I).EQ.1) GO TO 39
   38 CONTINUE
   39 NMPS=I
      THR=THROT(ITHR)*TNE(1,ITHR)*ENGDAT(6,NMPS)*FTN
      WDOT=THR/(GZERO*VARISP(THROT(ITHR),T))
      TV9=T+(XMIAD+XMALG-THR/(GZERO*GLIM(ITHR)))/WDOT
      IF(TV9.GT.T.AND.TV9.LT.TV2) GO TO 41
      TV9=0.
      NTRG9=-2
      GO TO 40
   41 TGLIM=TV9
   40 CONTINUE
      IF(ITHR.NE.ISTART (1).AND.ITHR.NE.ISTART (2)) GO TO 301
      IF(NMAX.NE.0) GO TO 301
      L=2
      IF(ITHR.EQ.ISTART (1)) J=1
      DO 300 I=1,25
  300 STARTV(I,J)=VAR(I)
      STARTV(7,J)=0.
      STARTV(26,J)=T
      STARTV(27,J)=XMIAD+XMAUG+WJET(JT)
      STARTV(28,J)=CHIF
      STARTV(29,J)=CHIT
      STARTV(30,J)=GLCB
  301 CONTINUE
      IF(NMAX.EQ.0.AND.JT.EQ.NCOAST.AND.NCOAST.NE.0) GO TO 116
      IF(JT.EQ.NVNT) GO TO 9
C
C***  CHECK FOR FIRST POINT
      IF(JUMP.GT.NOEVNT(1).AND.ITHR.EQ.JLMP) GO TO 199
    1 IF(JT.EQ.JUMP-1)RETURN
      IF(ITHR.EQ.NENDCT(KCYTAB))KCYTAB=KCYTAB+1
      IF(LSTGE(ITHR).EQ.1.OR.LSTGE(ITHR).LE.LSTGE(ITHR-1))GO TO 99
  199 CONTINUE
      IF(JUMP.GT.NOEVNT(1).AND.JSTART.NE.0) GO TO 99
      WOORB=XMIAD+XMALG
      IF(LSP.NE.1)WZERST=WOORB
      GLCW=WOORB
   99 CONTINUE
      DO 2 I=1,7
    2 DVARS(I)=DVAR(I)
      DO 995 K=1,4
      IF(NVEST(K).NE.L) GO TO 999
      IF(JT.GT.NOEVNT(1))IPRT=20
      IF(JUMP.EQ.1.AND.JT.GT.NOEVNT(1)) IPRT=21
      DO 787 I=1,7
      DELXDR(I,K)=DVAR(I)
  787 RESTX(I,K)=VAR(I)
      RESTX(7,K)=XMIAD+XMAUG+WJET(JT)
```

```
      RESTX(8,K)=T
      RESTX(9,K)=VAR(14)
      RESTX(10,K)=C.
      DO 998 I=1,6
      IF(KCCRES(I,K).EQ.16) DELXP(K)=DVAR(14)
  998 CONTINUE
  999 CONTINUE
      IF(NVAST(1).EQ.C.AND.JT.EQ.IRTAOA) IPRT=21
      CALL ACER1
      CALL *OER
      DO 7 I=1,7
    7 DELXD(KT,I)=DVARS(I)-DVAR(I)
      IF(JUPP.GT.NCEVAT(1).AND.ITHR.EQ.JLMP) GO TO 102
C
C***  CALCULATE PERF RES AND TAUT(IPR)
C
      IF(IRTLS.EQ.C.OR.ITHR.NE.IRTLS) GO TO 102
      XM=XMIAD+XMALG
      XM1=XP
      WY=PRCF(1)*PTKG
      WOMS=FROP(4)*PTKG
      IRTLS=IRTLS-1
      K=NOEVAT(1)+1
      DO 101 I=K,IRTLS1
      WOMS=WOMS-TALT(I)*(WOOMS(I)+WOOUMP(I))
  101 WY=WY+TNE(1,I)*CFR(I)*TAUT(I)
      L=IRTLS
      L=IRTLS+1
      IF(MSWCH(L).EQ.1) GO TO 150
      DO 148 I=L,NWNT
      IF(MSWCH(I).EQ.1) GO TO 149
      WY=WY+TNE(1,I)*CFR(I)*TAUT(I)
      XM1=XP1-TAUT(I)*(TNE(1,I)*XMO(I)+WCOMS(I)+WCOUPP(I)+WORCS(I))
  148 WOMS=WOMS-TALT(I)*(WOOMS(I)+WOOUMP(I))
  149 L=I
  150 CONTINUE
      WG=(TNE(1,L)*F(L)+FOMS(L)+FRCS(L))/(GZERO*GLIM(L))
      XMTJ=TNE(1,J)*XPC(J)+WOOMS(J)+WORCS(J)+WODLPP(J)
      XMTL=TNE(1,L)*XPC(L)+WOOMS(L)+WORCS(L)+WODLPP(L)
      TALT(J)=(XMTL*WEPS-WOOMS(L)*(XM1-WG))/(XMTL-(WCOMS(J)+WOOUPP(J)))-
     *WOOMS(L)*XPT(J)
      K=J+1
      P=L-1
      WT1=XP-XMTJ*TAUT(J)
      IF(N.GT.M) GO TO 161
      DO 160 I=N,M
  160 WT1=WT1-(TNE(1,I)*XMO(I)+WOOMS(I)+WORCS(I)+WOOUPP(I))*TAUT(I)
  161 WG1=(TNE(1,M)*F(P)+FOMS(M)+FRCS(M))/(GZERO*GLIM(L))
      IRTFLE=0
      IF(WT1.GT.WG1) GO TO 60
      IRTFLE=1
      WT1=WG1
      TALT(J)=XM-WE1
      IF(N.GT.M) GO TO 163
      DO 162 I=N,M
  162 TAUT(J)=TAUT(J)-(TNE(1,I)*XMO(I)+WCOMS(I)+WORCS(I)+WODLMP(I))*TAUT
     *(I)
  163 TAUT(J)=TAUT(J)/XMTJ
   60 TG=(WT1-WG)/XMTL
      WY=WY+TNE(1,J)*CFR(J)*TAUT(J)+TNE(1,L)*CFR(L)*TG
      TV2=T+TAUT(J)
      SPOA=CISP(L)/GLIM(L)
      IF(IPR.NE.C) GO TO 112
  102 IF(ITHR.NE.IPR.OR.IPR.EQ.0.OR.IRTLS.NE.0) GO TO 103
      XM=XMIAD+XMALG
      TEMP=WMX-PRKW*XP*(FR(IPR)/XMO(IPR)
```

```
      K=NOEVNT(1)+1
      IF(GLIM(IPR).GT.5.) GO TO 105
      SPOA=(ISP(IPR)/ELIM(IPR)
      WG=THR/(GLIM(IPR)*GZERO)
      TG=(XP-WG)/XPCOT
      WY=TNE(1,IPR)*CFR(IPR)*TG
      IPRP=IPR-1
      WY=WY+ETFUEL
      DO 108 JA=K,IPRP
1     108 WY=WY+TNE(1,JA)*CFR(JA)*TAUT(JA)
      IF(WY.GE.WPMX) GO TO 105
      112 TEMP1=1.
      IF(MSWCH(IPR+1).EQ.2) TEMP1=EXP(-VRCUT/(GZERO*CISP(IPR+1)))
      TEMP2=WPMX-WY-CFR(IPR)/XMO(IPR)*WG-TNE(1,IPR+2)*XMO(IPR+2)*TAUT(IP
     *R+2)*(1.-PRK4)
      TAUT(IPR)=TG+SPOA*ALOG(WG*(PRK4*TEMP1-CFR(IPR)/XMO(IPR)+(1.-TEMP1)
     *)/TEMP2)
      FMXX=SPOA/TEMP2
      PSW=1
      TIME(1,IPR+1)=TIME(1,IPR)+TAUT(IPR)
      FRTOT=PRK4*(WG*EXP(-GLIM(IPR)*(TAUT(IPR)-TE)/CISP(IPR))*TEMP1-
     *TNE(1,IPR+2)*XMO(IPR+2)*TAUT(IPR+2))
      IF(WY+FRTOT.GE.WPMX) GO TO 105
      GO TO 106
      105 CONTINUE
      PSW=0
      IPRP=IPR-1
      TEMP=TEMP-ETFUEL
      DO 108 JA=K,IPRP
1     108 TEMP=TEMP-TNE(1,JA)*CFR(JA)*TAUT(JA)
      TEMP1=TEMP/(1.-PRK4)
      TAUT(IPR)=TEMP1/(ANE(1)*CFR(IPR))
      TIME(1,IPR+1)=TIME(1,IPR)+TAUT(IPR)
      FRTOT=PRK4*(XM-ANE(1)*XMO(IPR)*TAUT(IPR))
      106 CONTINUE
      IF(ITHR.EQ.IPR) TV2=T+TAUT(IPR)
C
C***  PERF RESERVES AND TAUT(IPR) CALCULATED
C
      103 CONTINUE
      IF(MSWCH(ITHR).NE.2) GO TO 109
      TAUT(ITHR)=(XMIAC+XMAUG)*(1.-EXP(-VRCUT/(GZERO*CISP(ITHR))))/(XMO(
     *ITHR)*TNE(1,ITHR))
      TIME(1,ITHR+1)=TIME(1,ITHR)+TAUT(ITHR)
      TV2=T+TAUT(ITHR)
      109 CONTINUE
      IF(IPRT.NE.2) GO TO 70
C
C     SAVE INCLINATION AND NODE FOR AOA TRAJECTORY CONSTRAINTS FOR
C     VARIABLE IY TARGETING
C
      SVXINC=XINC
      SVXNOD=XNOD
C
C     SAVE EARTH RELATIVE AZIMUTH(HEADING ANGLE) FOR RTLS CONSTRAINT
C
      SVAZRE=AZRE
```

```
      398.        7C CONTINUE
      399.           IF(LSB.EQ.0) CALL APRTN(IPRT)
      400.           IF(ITHR.NE.MINH) RETURN
      401.           IF(LSE.EQ.1) GO TO 104
      402.           DO 107 I=1,7
      403.       107 DELX(I)=VAR(I)
      404.       104 CONTINUE
      405.           TQ=T
      406.           CALL AGER1
      407.           CALL AGER
      408.           CALL APRTN(10)
      409.           IF(JUPP.EQ.1) CALL RTMRK
      410.           RETURN
      411.         9 CONTINUE
      412.           IF(LSB.EQ.1)GO TO 111
      413.       116 IF(NMAX.NE.0.OR.JT.NE.NCOAST) GO TO 115
      414.           XM=XMIAO+XMALG
      415.           CALL AGER1
      416.           CALL AGER
      417.           CALL APRTN(11)
      418.           IF(ITHR.EQ.ISTART(1)) STARTV(22,1)=XM
      419.           IF(ITHR.EQ.ISTART(2)) STARTV(22,2)=XM
      420.           CALL RTMRK
      421.           RETURN
      422.       115 CONTINUE
      423.           TL=0.
      424.           CALL AEOSR
      425.           CALL APRTN(15)
      426.           DO 110 I=1,7
      427.       110 DELXO(JT,I)=DVAR(I)
      428.       111 IHR=0.
      429.           XMDOT=0.
      430.           IF(LSB.EQ.0) OGCW=XMIAO+XMAUG
      431.           XMAUG=XMAUG-FRTC1
      432.           XM=XMIAO+XMALG
      433.           CALL AGER1
      434.           CALL AGER
      435.           CALL APRTN(1)
      436.           CHVEL=VAR(15)-VAR(16)-VAR(17)
      437.           CALL RTMRK
      438.           RETURN
      439.           END
```

```
      SUBROUTINE ATILT
C
C      TRIGGER ROUTINE FOR TIME RELATED EVENTS DIRECTLY AFTER LIFTOFF
C
      CHARACTER*60 HEAD
      CHARACTER*6 MISION
      LOGICAL QFLG,GLIMIT
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *DTZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WG(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAI,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVARS(13),DELXDW(15,3),DELXDR(7,4),DELXD(15,7),WZERO,ALT,
     *XMACH,GDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEL,
     *EYL,BEL,BHMAX(15),BHMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITHR,IWD,IXR,JUMP,KAT,KCYTAB,KPAGE,KS1,KWTA(4),LINES,
     *NBGCT(4),NENDCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *MINH,MPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINDB,KRDERB,MISION,
     *IRTFLG,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),MSWCH(15)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,AMX,AMY,AMZ,
     *TM2,THMFA,SIDE,WMAG,AZW,FANC,FLATC,FOM,VISC
      COMMON/BGEN3/NOSS,JTB,NENT,NACT,NOM1,NF(4)
      COMMON/BODY/ATBCWT(15,2),ATXCG(15,2),ATYCG(15,2),AXLEN(2),AXREF(2)
     *,AYREF(2),KPI(2),KYI(2),API(10,2),AYI(10,2)
      COMMON/BODYF/TBCWT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,M1,M2,M5,
     *M6,M7,M8,M9,M50,M51,MSRW,AP(10),AY(10)
      COMMON/CONST/RAC,PI,RE,FLAT,CJ,H,DL,CMLE,OMEGA,ALT1,ALT2,PSL,
     *GZERO,PTN,PTKG,PSFTM,PFN,PFKG,PSFFM
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/FORINT/HEANM(2),T,TT,W,U,V,X,Y,Z,XM1,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,HMX,HMN,HAM
      COMMON/ITHRO/ITHRO
      COMMON/KP/KP,KY
      COMMON/MPS/TIMMPS(15),FTBL(15),THROT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MPS1,MPS2,TGLIM,GLIMIT
      COMMON/NTILT/NTILT
      COMMON/QMAX/QMAX,QFLG,QPEN,MAXQ
      COMMON/REST/LSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NCPAR,WIBT(15)
      COMMON/TAULIM/TALLIM,NTAU,NBETA,TIMTAU
      NTILT=0
      IF(IFACT.EQ.3) GO TO 4
      IF(ABS(T-TBEGR).GT.1.E-04.OR.NROLL.EQ.-1) GO TO 3
      CALL APRTN(18)
      NROLL=1
      IF(TLIFT.LT.TENDR) GO TO 22
      TV4=TENDR
      RETURN
   22 CONTINUE
      IF(ABS(T-TLIFT).LT.1.E-04) GO TO 4
      IF(T.GT.TLIFT) GO TO 5
      TV4=TLIFT
      RETURN
    3 IF(ABS(T-TLIFT).GT.1.E-04) GO TO 6
    4 CALL APRTN(7)
      MS1=2
      NTRG6=1
      NTRG4=-2
      IF(IFACT.EQ.3) RETURN
      NTRG4=2
      IF(NROLL.EQ.-1) GO TO 8
      IF(NROLL.EQ.0) GO TO 9
    5 TV4=TTILT
      IF(TTILT.GE.TENDR) TV4=TENDR
      IF(T.GT.TLIFT) RETURN
      GO TO 2
    9 TV4=TBEGR
```

```
      GO TO 2
    6 IF(ABS(T-TENCR).GT.1.E-04) GO TO 10
      IF(TENCR.GE.TTILT) NTILT=1
      CALL APRTN(19)
      NRCLL=0
      IF(TENOR.GE.TLIFT) GO TO 23
      IV4=TLIFT
      RETURN
   23 CONTINUE
      IF(ABS(T-TTILT).GT.1.E-04) GO TO 10
      IF(TENOR.GE.TTILT) GO TO 11
      IV4=TTILT
      RETURN
   11 NTRG4=-2
      RETURN
   10 CALL APRTN(8)
      IV4=TENOR
      IF(TTILT.GE.TENCR) NTRG4=-2
      NTILT=1
      CALL ADER1
      CALL ADER
      CALL APRTN(1)
      RETURN
    8 IV4=TTILT
      GO TO 2
    2 CPTBL(1)=CHPBAS*FAC
      CYTBL(1)=CHYBAS*FAC
      AP=XPI(1)
      AY=XYI(1)
      CALL FIND(CPTBL,AP,TTBL,KP)
      CALL FIND(CYTBL,AY,TTBL(16),KY)
      DO 20 I=1,10
      API(I,1)=AP(I)
   26 AYI(I,1)=AY(I)
      RETURN
      ENTRY AQMAX
C
C      TRIGGER ROUTINE FOR MAXIMUM DYNAMIC PRESSURE PRINTOUT
C
      CFLG=.TRUE.
      NTRG6=-1
      CALL ADER1
      CALL ADER
      IF(NTAU.NE.0) GO TO 24
      CALL APRTN(14)
      RETURN
   24 CONTINUE
      IV13=1+TIMTAU
      IIPMPS(NTAU+2)=IV13
      IIPMPS(NTAU+3)=IV13+(FTBL(NTAU+3)-FTBL(NTAU+2))/TAULIM
      NTRG13=1
      RETURN
      ENTRY AXPRT
C
C      TRIGGER ROUTINE FOR NORMAL PRINTOUT
C
      IF(NMAX.EQ.0) CALL APRTN(1)
      IV1=AINT(T+.1E-03)*PRINT(ITHR)
      RETURN
      ENTRY GLIMT
C
C      TRIGGER ROUTINE FOR ACCELERATION LIMIT PRINTOUT
C
      NTRG9=-2
      CALL APRTN(22)
      NMP=STEP(ITHR)
      GLIMIT=.TRUE.
      RETURN
      ENTRY AMDEV
```

```
C
C       TRIGGER ROUTINE FOR WEIGHT DROP EVENTS
C
        IWD=IWD+1
        IF(LSB.NE.1.AND.WD(IWD).GT.0.) CALL APRTN(1)
        XMAUG=XMAUG-WD(IWD)
        IF(LSB.EQ.1) GO TO 14
        DO 12 I=1,3
   12   DVARS(I)=DVAR(I)
        CALL ADER1
        CALL ADER
        DO 13 I=1,3
1  13   DELXDW(IWD,I)=DVARS(I)-DVAR(I)
        IF(WD(IWD).GT.0.) CALL APRTN(17)
   14   IF(IWD.EQ.NWVAT) GO TO 15
        IV12=TIME(2,IWD+1)
        GO TO 16
   15   NTRG12=-2
   16   CONTINUE
        RETURN
        ENTRY ALIFT
C
C       TRIGGER ROUTINE TO FLAG LIFTOFF BASED ON F/P=1
C
        NTRG7=-2
        CALL ADER1
        CALL ADER
        CALL APRTN(2)
        RETURN
        ENTRY ATHRO
C
C       TRIGGER ROUTINE FOR SSME THROTTLE TABLE
C
        CALL APRTN(25)
        IF(TIMMPS(ITHRO+1).LT.TIMMPS(ITHRO)) NTRG13=-1
        IF(NTRG13.LT.0) RETURN
        ITHRO=ITHRO+1
        TV13=TIMMPS(ITHRO)
        IF(ITHRO.EQ.NTAB+2) NTRG13=-1
        RETURN
        END
```

```
      SUBROUTINE BAOLX
C
C      DETERMINES THE INFLUENCE COEFFICIENTS(PARTIAL DERIVATIVES)OF THE
C      PAYOFF AND CONSTRAINTS WITH RESPECT TO THE PARAMETERS
C
      CHARACTER*60 HEAD
      CHARACTER*6 MISION
      COMMON/AERODI/AERO(30,16),AEROB(50,5,3),TCPACH(20),DELCA(20),
     *BNPONE(15),BNFAET(50,2),BNFTHR(50,2),DELCN(20),DELCM(20)
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *ET2,TG,TL,AMAUG,TNE(6,15),ENGDAT(8,15),S(15),WG(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHROOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,OPHIZ,RTHE,R,VR,XM,SW,SU,SV,Q,
     *NIV(25),OVARS(12),CELXDW(15,3),DELXOR(7,4),CELXO(15,7),WZERO,ALT,
     *MACH,GBBT,PAA,FAN,A(3,3),CASE,ET2,UMF,A12L,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHPA)(15),BHMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITER,IWO,IXR,JUMP,KAT,KCYTAB,KPAGE,KS1,KWTA(4),LINES,
     *NBECT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *PINH,PFSFLG(15),ASYST(15),ICONSW,IFTLS,IPOLY,KINOB,KRDERB,MISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/AUTC/NCB(40),SAVCP(40),XLAMB(40,10),CP2
      COMMON/BAKE/BISS(10,10),YL(7,10)
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NOP1,NF(4)
      COMMON/BLAMB/YLBT(15,10),YLBW(15,10),GAPHI(10),PROD(2,10)
      COMMON/BODY/TBOD(15,2),TXCG(15,2),TYCG(15,2),XLEN(2),XREF(2),
     *YREF(2),XP(2),XY(2),AP(10,2),AY(10,2)
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,DJ,CMLE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,PTN,PTNG,PSFTM,PFN,PFNG,PSFFP
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPOTBL(91),
     *CYOTBL(91)
      COMMON/DDELX/DDELX(10)
      COMMON/DELPAR/SFRACB,SYRACB
      COMMON/DELX/DELX(7)
      COMMON/FMXX/FMXX,MSW,WG,SPOA
      COMMON/FORINT/HBANK(2),T,TT,VAR(25)                    ,OVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,NINC,KRDER,EU,EL,AYL,HMX,HPN,HAP
      COMMON/IPR/IPR,LPMX,ILAST,KOT(15),PRK4,PRTCT
      COMMON/LBM/TIMLBP(30),THRLBM(30),WCTLBP(30),PLBM(2),MLBM1,MLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MPS/TIMMFS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MFS1,MPS2,TELIP,GLIMIT
      COMMON/OMSRCS/FCPS(15),FRCS(15),WOCMS(15),WERCS(15),WOCUMP(15)
      COMMON/QMAX/QMAX,QFLG,QPEN,MAXQ
      COMMON/NALTW/NALTW
      COMMON/REST/NSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NCPAR,WIBT(15)
      COMMON/TAULIM/TALLIM,NTAU,NBETA,TIMTAU
      COMMON/XTRA/XMD(15),CFR(15),F(15),CISP(15)
      COMMON/YLI/YLINT(7,10),PHITE(10),PSIREC(10),KCCPHI(10),PSTR(10)
      DATA CTIME/.1/,CIAL/.1/,DBETA/.1/
      DATA CELWZ/100./,DAA/.125/
C
      LSB=1
      IF(NALTW.EQ.1)GO TO 95
      CPENS=CPEN
C
C     GET LAMBDA BETA 16 FOR LIFTOFF WEIGHT
C
      IF(NBB(16).EQ.0)GO TO 4
      IF(JUMP.EQ.1)GO TO 101
      DO 100 I=1,NCSS
  100 XLAMB(16,I)=YL(7,I)
      GO TO 4
  101 CONTINUE
      WZERO=WZERO+CELWZ
      CALL AFORUN
```

```
      WZERO=WZERO-DELWZ
      DO 102 I=1,7
102   DDELX(I)=(DELX(I)-VAR(I))/(-DELWZ)
      DO 20 I=1,NOSS
      XLAMB(16,I)=0.
      DO 20 J=1,7
20    XLAMB(16,I)=XLAMB(16,I)+DDELX(J)*YL(J,I)
      IF(MAXQ.NE.0) XLAMB(16,MAXQ)=(QPEN-QPENS)/DELWZ
C
C     GET LAMBDA BETAS 18-24 FOR CPTBL
C
4     DO 75 K=1,7
      IF(KDB(K+17).EQ.0) GO TO 75
      CPTBL(K+1)=CPTBL(K+1)+SPRADB
      CALL AFORUN
      CPTBL(K+1)=CPTBL(K+1)-SPRADB
      DO 105 I=1,7
105   DDELX(I)=(DELX(I)-VAR(I))/(-SPRADB)
      DO 7 I=1,NOSS
      XLAMB(K+17,I)=0.
      DO 7 J=1,7
7     XLAMB(K+17,I)=XLAMB(K+17,I)+DDELX(J)*YL(J,I)
      IF(MAXQ.NE.0) XLAMB(K+17,MAXQ)=(QPEN-QPENS)/SPRADB
C
C     GET LAMBDA BETAS 25-28 FOR CYTBL
C
75    CONTINUE
      DO 76 K=1,4
      IF(KDB(K+24).EQ.0) GO TO 76
      CYTBL(K+1)=CYTBL(K+1)+SYRADB
      CALL AFORUN
      CYTBL(K+1)=CYTBL(K+1)-SYRADB
      DO 108 I=1,7
108   DDELX(I)=(DELX(I)-VAR(I))/(-SYRADB)
      DO 11 I=1,NOSS
      XLAMB(K+24,I)=0.
      DO 11 J=1,7
11    XLAMB(K+24,I)=XLAMB(K+24,I)+DDELX(J)*YL(J,I)
      IF(MAXQ.NE.0) XLAMB(K+24,MAXQ)=(QPEN-QPENS)/SYRADB
76    CONTINUE
C
C     GET LAMBDA BETA 17 FOR AZIMUTH
C
      IF(KDB(17).EQ.0) GO TO 15
      NALTW=1
      AA=AA+DAA/RAD
      DO 12 I=1,10
12    DDELX(I)=PHITE(I)
      RETURN
96    DO 13 I=1,NOSS
13    XLAMB(17,I)=(DDELX(I)-PHITE(I))/(-DAA)
      IF(MAXQ.NE.0) XLAMB(17,MAXQ)=(QPEN-QPENS)/DAA
      DO 14 I=1,10
14    PHITE(I)=DDELX(I)
      AA=AA-DAA/RAD
      NALTW=0
C
C     GET LAMBDA BETAS 29-31 FOR THROTTLE CONTROL
C
15    CONTINUE
      IF(NTAU.EQ.0) GO TO 153
      DO 172 K=1,3
      IF(KDB(K+28).EQ.0) GO TO 172
      IF(K.EQ.1) TIMMPS(NTAU)=TIMMPS(NTAU)+DTIME
      IF(K.EQ.2) TIMTAL=TIMTAU+DTIME
      IF(K.EQ.3) FTBL(NTAU+1)=FTBL(NTAU+1)+DTAU/100.
      CALL AFORUN
```

```
      IF(K.EQ.1) TIMMPS(NTAU)=TIMMPS(NTAU)-OTIME
      IF(K.EQ.2) TIMTAU=TIMTAU-OTIME
      IF(K.EQ.3) FTBL(NTAU+1)=FTBL(NTAU+1)-OTAU/ICO,
      OEL=OTIME
      IF(K.EQ.3) OEL=OTAU
      DO 170 I=1,7
  170 DDELX(I)=(DELX(I)-VAR(I))/(1-OEL)
      DO 171 I=1,NCSS
      XLAMB(K+28,I)=0.
      DO 171 J=1,7
  171 XLAMB(K+28,I)=XLAMB(K+28,I)+DDELX(J)*YL(J,I)
      IF(MAXO.NE.0) XLAMB(K+28,MAXO)=(OPEN-OPENS)/OEL
  172 CONTINUE
C
C     GET LAMBDA BETA 32-34 FOR SIDESLIP CONTROL
C
  153 CONTINUE
      IF(NBETA.EQ.0) GO TO 157
      DO 156 K=1,3
      IF(NOB(K+31).EQ.0) GO TO 156
      L=NBETA+K-1
      AERCD(L,15)=AERCD(L,15)+DBETA/RAD
      CALL AFORUN
      AERCD(L,15)=AERCD(L,15)-DBETA/RAD
      DO 154 I=1,7
  154 DDELX(I)=(DELX(I)-VAR(I))/(1-DBETA)
      DO 155 I=1,NCSS
      XLAMB(K+31,I)=0.
      DO 155 J=1,7
  155 XLAMB(K+31,I)=XLAMB(K+31,I)+DDELX(J)*YL(J,I)
      IF(MAXO.NE.0) XLAMB(K+31,MAXO)=(OPEN-OPENS)/DBETA
  156 CONTINUE
C
C
  157 CONTINUE
      IF(IPR.EQ.0.OR.IRTLS.EQ.0) GO TO 163
      IF(NOB(16).EQ.0) GO TO 17
      L=IRTLS
      L=IPR
      XMTL=TNE(1,L)*XPC(L)+WDOMS(L)+WDRCS(L)+WDDLPP(L)
      XMTJ=TNE(1,J)*XPC(J)+WDOMS(J)+WDRCS(J)+WDDLPP(J)
      FTLWO=-WDOMS(L)/(XPTL*(WDOMS(J)+WDDUMP(J))-WDOMS(L)*XMTJ)
      IF(IRTFLG.EQ.1) FTLWO=1./XMTJ
      FTGWO=1./XPTL*(1.-XMTJ*FTJWO)
      TEMP=TNE(1,J)*CFR(J)*FTJWO+TNE(1,L)*CFR(L)*FTGWO
      FTLWO=FTGWO+FMXX*TEMP
      DO 161 I=1,NCSS
      IF(NCCFHI(I).EQ.15) GO TO 161
      IF(NCCFHI(I).NE.1) GO TO 160
      XLAMB(16,I)=(1.-FRKW)/(CFR(L)/XMO(L)-FRKW)*TEMP
      GO TO 161
  160 XLAMB(16,I)=XLAMB(16,I)+FTJWO*YLBT(J,I)+FTLWO*YLBT(L,I)
  161 CONTINUE
      GO TO 17
  163 IF(IPR.EQ.0) GO TO 17
C***  ADJUST LAMB1 FOR PERF RES
      IF(MSK.EQ.0) GO TO 151
      TEMP=1./(TNE(1,IPR)*XMO(IPR))+FMXX*(CFR(IPR)/XMO(IPR))
      TEMP1=1.
      IF(MSKCH(IPR+1).EQ.2) TEMP1=EXP(-VRCUT/(GZERO*CISP(IPR+1)))
      TEMP2=0.
      IF(MSKCH(IPR+1).EQ.2) TEMP2=-TAUT(IPR+1)*GLIM(IPR)/CISP(IPR)*(TEMP
     *-1./(TNE(1,IPR)*XMO(IPR)))
      DO 150 I=1,NCSS
      IF(NCCFHI(I).EQ.15) GO TO 150
      IF(NCCFHI(I).NE.1) GO TO 152
      XLAMB(16,I)=(1.-FRKW)/(CFR(IPR)/XMO(IPR)-FRKW*TEMP1-(1.-TEMP1))*CF
     *R(IPR)/XMO(IPR)*TEMP1
```

```
      GO TO 150
  152 XLAMB(16,I)=XLAMB(16,I)+TEMP*YLBT(IPR,I)+TEMP2*YLBT(IPR+1,I)
  150 CONTINUE
      GO TO 17
  151 CONTINUE
      TEMP=FRK4/(1.-FRK4)*XMO(IPR)*TNE(1,IPR))
      DO 16 I=1,NOSS
      IF(KCEPHI(I).EQ.1)YLBT(IPR,I)=0.
      IF(KCEPHI(I).EQ.15) GO TO 16
      XLAMB(16,I)=XLAMB(16,I)-TEMP*YLBT(IPR,I)
   16 CONTINUE
C*****
C     END OF LAMB1 AND LAMB2
   17 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BAKRN
C
C     CONTROLS THE SETUP LOGIC FOR THE BACKWARD TRAJECTORY INTEGRATION
C     AND CALLS THE INTEGRATION PACKAGE
C
      CHARACTER*60 HEAD
      CHARACTER*6 MISION,HEADER
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CT2,TG,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BG(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,OMIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVARS(13),DELXOW(15,3),DELXOR(7,4),DELXC(15,7),WZERO,ALT,
     *MACH,GDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMA)(15),BHMN,BSTEP(15),HAMB,HMXB,IPOLY,XMIAD
      COMMON/AGEN3/ETHF,IWB,IXR,JUMP,KAT,KGYTAB,KPAGE,KS1,KWTA(41),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWO(15),NVNT,NWVNT,IHEAD,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IFTLS,IPOLY,KINOB,KRDERB,MISION,
     *IRTFLG,NROLL,NCCAS1,IFACT,NOBASE,LSTGE(15),MSWCH(15)
      COMMON/AGEN4/QALPHA,QBETA,QCN,STH1,CTH1,XJX(3),XJV(3)
      COMMON/BAKE/WISS(10,10),YLF(7,10)
      COMMON/BAKIN1/BBANK(2),TBK,TBKO,YLC(7,10),YLC(7,10),BSAVE(1135),
     *NBTRG1,TBV1,NBTRG2,TBV2,NBTRG3,TBV3,NBTRG4,TBV4,NBTRG5,TBV5,TEND,
     *CT84,CL3,D23,C43,WISSO(10,10)
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NOP1,NP(4)
      COMMON/BLAMB/YLBT(15,10),YLBW(15,10),GAPHI(10),PROD(2,10)
      COMMON/BODYB/TBCLT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,MBS1,MBS2,
     *MBS3,MBS4,MBS5,MBS6,MBS7,MBS8,MBS9,MBS10
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,DW,CMLE,OMEGA,ALT1,ALT2,PSL,
     *GZERO,PTN,PTMG,PSFTM,PFN,PFMG,PSFFM
      COMMON/CONTRC/TTBL(30),CPTBL(30),CYTBL(30),TOBL(91),CPCTBL(91),
     *CYCTBL(91)
      COMMON/DELXH/DELXH(4)
      COMMON/DRNGDT/DRNGDT(2)
      COMMON/ENPUT/PSIRST(6,4),HEADER(20),NVRST(4),KCDRES(6,4),LAST
      COMMON/FORINT/MBANK(2),T,TT,W,U,V,X,Y,Z,XM1,ZAF(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,HMX,HMN,HAP
      COMMON/HEAT/PAXHAT
      COMMON/IPR/IFR,NPMX,ILAST,KOT(15),PRK4,PRTCT
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TALTBL
     *(5),MPS1,MPS2,TELIP,GLIMIT
      COMMON/NOMOR/NOMCR
      COMMON/REST/JSTR,NCNRS(4),NTCN,TR(4),LSB,NFB,NOPAR,WIBT(15)
      COMMON/TABLK/MCON1,MCON2,MCON3,MCON4
      COMMON/VAERO/VAEROC(10,19),M18,M19,MBS18,MBS19
      COMMON/YLI/YLINT(7,10),PWITE(10),PSIREC(10),KCDFHI(10),PSTR(10)
C     INITIALIZE BACKWARD RUN
      EXTERNAL BDR11,BCR1,BWDEV,BSAVEG,BWEND,BGLIP
      MBS1=1
      MBS2=0
      MBS3=1
      MBS4=0
      MBS7=1
      MBS8=0
      MBS10=1
      MBS19=0
      LLDSOL=5
      IF(KINOB.NE.2)LLDSOL=3*(KRDERB+2)+4
      IENB=1
      TBK=0.
      TBKO=0.
      NOMCR=0
      NAT=0
      NCTN=1
      DO 38 I=1,4
```

```
1          IF(NVRST(I).NE.0) NCTN=NCTN+1
1          TR(I)=0.
1       30 CONTINUE
           NCYTAB=NCYTAB+1
           LTB=1
           ITHR=NVNT+1
           IWD=NNVNT+1
           NENT=IXR
           IF(IXR.EQ.0) NENT=100
           NBSS=NENT
           NBS6=0
           XMAUG=XMAUG+FRTGT
     C     ZERO OUT LAMBDAS
           DO 10 I=1,7
1          DO 10 J=1,10
2       10 YL(I,J)=0.
           DO 1 J=1,7
           DO 1 I=1,NGSS
2        1 YL(J,I)=YLIN1(J,I)
           DO 2 I=1,10
1          DO 2 J=1,I
2        2 WISS(I,J)=0.
           DO 7 I=1,15
           DO 7 J=1,10
2        7 YLBT(I,J)=0.
2    C
2    C     STOP TRIGGER
           NBTRG1=1
           TBV1=TEND-TO
     C
     C*2*  THRUST EVENT TRIGGER  (BTHREV)
           NBTRG2=2
           TBV2=0.
     C
     C*3*  WEIGHT DROP EVENT TRIGGER  (BWDEV)
           NBTRG3=2
           IF(NWVNT.EQ.0) NBTRG3=-2
           TBV3=TEND-TIME(2,NVNT)
     C
     C**   SAVE IMPULSE RESPONSE FUNCTIONS  (BSAVEG)
           NBTRG4=-1
           TBV4=0.
     C
     C*5*  G LIMIT
           NBTRG5=-2
           TBV5=TEND-TGLIM
     C
     C     INITIALIZE CPI SPACING
     C     INITIALIZE THRUST,STEP,HNOM,HMAX,MCOT,ETC.
           CALL BTHREV
     C     INERTIAL INTEGRATION
        3  NACT=7*NOSS
           NOPOR=0
           CALL DESOLV(TC,NACT,BDRII,BDRI,0,6,KERR,HNPR,BBANK,BSAVE,
          *KINOB,LLDSOL,BEU,BEL,HMXB,BHMN,BYL,5,
          *NBTRG4,BSAVEG,TBK,TBV4,NBTRG3,BWDEV,TBK,TBV3,
          *NBTRG2,BTHREV,TBK,TBV2,NBTRG5,BGLIM,TBK,TBV5,
          *NBTRG1,BKEND,TBK,TEV1,   0,0,0,0,   0,0,0,0,   0,0,0,0,
          *0,0,0,0,  0,0,0,0,  0,0,0,0)
           IF(KERR.NE.0)GO TO 4
           IF(NOPOR.EQ.1) GO TO 40
     C**** NOMOR IS SET TO 1 IN BTHREV BEFORE CALL TO RTPRK IF RESTART
           IF(ITHR.EQ.MINH.AND.ITHR-1.EQ.NVRST(1)) NOMOR=3
           IF(NOPOR.EQ.0)RETURN
           IF(NOPOR.EQ.2) ITHR=ITHR-1
        40 NCTN=NCTN-1
           TR(NCTN)=TEND-TBK
           K=NOSS+1
```

```
      NOSS=NCSS+NCARS(NCTN)
      DO 5 J=1,7
      DO 5 I=K,NOSS
    5 YL(J,I)=YLINT(J,I)
      DO 6 I=K,NOSS
      YLBT(ITHR,I)=0.
      L=I-K+1
      LL=KCORES(L,NCTN)
      IF(LL.GE.18.AND.LL.LE.20) YLBT(ITHR,I)=DRNECT(2)
      DO 6 J=1,7
    6 YLBT(ITHR,I)=YLBT(ITHR,I)+DELXOR(J,NCTN)*YL(J,I)
      IF(MAXHAT.NE.0.AND.ABS(DELXH(NCTN)).GT.0.) YLBT(ITHR,MAXHAT)=
     1YLBT(ITHR,MAXHAT)+DELXH(NCTN)
      IF(NOPCR.EQ.2) GO TO 20
      GO TO 3
   20 CALL EXEND
      RETURN
    4 WRITE(6,9310)
 9310 FORMAT(//1X,24HINTEGRATION ERROR.DUMPED)
      STOP
      END
```

```
      SUBROUTINE BERI
C
C      COMPUTES TOTAL ADJOINT EQUATIONS
C
      CHARACTER*60 HEADE
      CHARACTER*6 MISION
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *LT2,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVARS(13),DELXDW(15,3),DELXDR(7,4),DELXD(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHPA(15),BHMN,BSTEP(15),HAMB,HMXB,IPOLY,XMIAD
      COMMON/AGEN3/ITER,IWD,IXR,JUMP,KAT,KCYTAB,KFAGE,KS1,KWTA(4),LINES,
     *NBGET(4),NENEET(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IMEAD,
     *PIAH,PPSFLG(15),NSYST(15),ICONSW,IFTLS,IPOLY,KINDB,KRDERB,MISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),MSWCH(15)
      COMMON/BANKINT/BBANK(2),TBK,TBKD,YL(7,10),YLD(7,10),BSAVE(1350),
     *NBTRG1,TBV1,NBTRG2,TBV2,NBTRG3,TBV3,NBTRG4,TBV4,NBTRG5,TBV5,TEND,
     *CTB4,DU3,D23,C43,WISSD(10,10)
      COMMON/BGEN3/NOSS,TB,NENT,NACT,NOM1,NF(4)
      COMMON/CALIF/PXX,PYX,PZX,PXY,PYY,PZY,PXZ,PYZ,PZZ,PXW,PYW,PZW,PXU,
     *PYU,PZU,PXV,PYV,PZV,PXM,PYM,PZM,PDXCP,PDYCP,PDZCP,PDXCY,PDYCY,PDZC
     *Y,PDXCPP,PDYCPP,PDZCPP,PDXCPY,PDYCPY,PDZCPY,PDXCYY,PDYCYY,PDZCYY
      COMMON/CONST/RAD,PI,RE,FLAT,GJ,H,CL,CMUE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,FTN,PTNG,FSFTM,PFN,PFNG,PSFF
      COMMON/HEAT/MAXHAT
      COMMON/MPS/TIMMPS(15),FTBL(15),THROT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MFS1,MPS2,TELIM,GLIMIT
      COMMON/PHRS/PHRS(6)
C     LAMBDA DOTS
      IF(NBTRG5.NE.2.OR.MPSFLG(ITHR).NE.4) GO TO 10
      XM=XMIAD+XMAUG
      THR=ENGDAT(6,1)*TNE(1,ITHR)*PTN
      TAU=XM*GZERO*GLIM(ITHR)/THR
      FTAUM=GZERO*GLIM(ITHR)/THR
      XISP=VARISP(TAU,TGLIM)
      CISP=CEVISP(TAU,FTAUM)
   10 CONTINUE
      DO 1 I=1,NOSS
      YLD(1,I)=PXW*YL(1,I)+PYW*YL(2,I)+PZW*YL(3,I)+YL(4,I)
      YLD(2,I)=PXU*YL(1,I)+PYU*YL(2,I)+PZU*YL(3,I)+YL(5,I)
      YLD(3,I)=PXV*YL(1,I)+PYV*YL(2,I)+PZV*YL(3,I)+YL(6,I)
      YLD(4,I)=PXX*YL(1,I)+PYX*YL(2,I)+PZX*YL(3,I)
      YLD(5,I)=PXY*YL(1,I)+PYY*YL(2,I)+PZY*YL(3,I)
      YLD(6,I)=PXZ*YL(1,I)+PYZ*YL(2,I)+PZZ*YL(3,I)
      YLD(7,I)=PXM*YL(1,I)+PYM*YL(2,I)+PZM*YL(3,I)
      IF(NBTRG5.NE.2.OR.MPSFLG(ITHR).NE.4) GO TO 1
      IF(TAU.LT..65) GO TO 1
      YLD(7,I)=YLD(7,I)-GLIM(ITHR)*(XISP-XM*CISP)/(XISP*XISP)*YL(7,I)
    1 CONTINUE
      IF(MAXHAT.EQ.0) RETURN
      DO 3 I=1,NOSS
      IF(I.NE.MAXHAT) GO TO 3
      DO 2 J=1,6
    2 YLD(J,MAXHAT)=YLD(J,MAXHAT)+PHRS(J)
    3 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BCRII
C
C     COMPLETES TIME DEPENDENT PORTION OF ADJOINT EQUATIONS
C
      CHARACTER*60 HEADE
      CHARACTER*6 PISICN,ATMOSN
      COMMON/AERO1/AERO(30,16),AEROB(50,5,3),TCMACH(20),DELCA(20),
     *FNPONE(15),BKFAB1(50,2),BKFTHR(50,2),DELCN(20),CELCM(20)
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),T2ERG,TLIFT,TTILT,TMINH,
     *CTZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *EAO,FRINT(15),STEP(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FAZ,BC(1,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AZ,SA,DA,ALF,ALFY,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SU,SV,Q,
     *XIV(25),DVARS(13),CELXDW(15,3),DELXCR(7,4),CELXC(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMA(15),EHMN,BSTEP(15),HMB,HMXB,TPOLY,XMIAO
      COMMON/AGEN3/ITHR,IWD,IXR,JUMP,KAT,KCYTAB,KFAGE,KS1,KWTA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IHEAD,
     *NINH,NPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KIADB,KRDERB,PISICN,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),MSWCH(15)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,APX,APY,AMZ,
     *TMZ,TMFA,SIEE,BPAG,AZW,FANG,FLATC,FOM,VISC
      COMMON/AROC/FAT(14),ATMOSN
      COMMON/ALAT/ALAT,ALONG,ALTLS,NTABLE
      COMMON/AVGGP/XGPA,YGPA,ZGPA
      COMMON/BANKIN1/BBANK(2),TBK,TBKD,YL(7,10),YLE(7,10),BSAVE(1350),
     *NBTRG1,TBV1,NBTRG2,TBV2,NBTRG3,TBV3,NBTRG4,TBV4,NBTRG5,TBV5,TEND,
     *CT84,CT3,D23,D43,WISSD(10,10)
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NOM1,NF(4)
      COMMON/BODYB/TBCNT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,MBS1,MBS2,
     *MBS3,MBS4,MBS5,MBS6,MBS7,MBS8,MBS9,MBS10
      COMMON/CALIF/PXX,PYX,PZX,PXY,PYY,PZY,PXZ,PYZ,PZZ,PXW,PYW,PZW,PXU,
     *PYL,PZL,PXV,PYV,PZV,PXM,PYM,PZM,PDXCP,PDYCP,PDZCP,PDXCY,PDYCY,PDZC
     *Y,PDXCPP,PDYCPP,PDZCPP,PDXCPY,PDYCPY,PDZCPY,PDXCYY,PDYCYY,PDZCYY
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,DL,CMLE,OMEGA,ALT1,ALT2,PSL,
     *GZERO,FTN,FTKG,FSFTM,PFN,PFKG,PSFFP
      COMMON/DELFAB/DELALT(25),DELFAB(25),M3C,M31,ITCL(15)
      COMMON/DELMBS/MBS21,MBS30,MBS31
      COMMON/FORINT/WBANK(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,XINC,KRDER,EU,EL,AYL,HPX,HPN,HAP
      COMMON/GRAV/GTO,G11,G22,G23,G33
      COMMON/HEAT/PAXHAT
      COMMON/MLAST/MLAST,MMMM,KKKK
      COMMON/LBM/TIMLEP(30),TLBM(30,2)          ,PLBM(2),PLBM1,PLBM2,
     *ULBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/LBMMBL/MBLB1,MBLB2
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MPS1,MPS2,TGLIP,GLIMIT
      COMMON/OMSRCS/FCPS(15),FRCS(15),WCOMS(15),WCRCS(15),WODUMP(15)
      COMMON/OLOK/CLOK,WINT
      COMMON/PHRS/FHRL,PHRU,PHRV,PHRX,PHRY,PHRZ
      COMMON/TABS/TBL(101),WBL(101),UBL(101),VBL(101),XBL(101),YBL(101),
     *ZBL(101),XMBL(101)
      COMMON/VAERO/VAEROC(10,19),M18,M19,MBS18,MBS19
      COMMON/WIND/ALTTBL(75),WTBL(75,2),NWING
      DIMENSION DFAB(1),CDFAB(1),AF(1),BF(1),CF(1)
      DIMENSION XHAT(3),YHAT(3),ZHAT(3)
      DIMENSION STBL(101,7),STATE(7),SDC(7),AA(7),BB(7),CC(7)
      DIMENSION FAB(3),AFAB(3),BFAB(3),CFAB(3),AERC(12),AERCDD(15),
     *AAERO(12),BAERO(12),CAERO(12)
      DIMENSION CG(2),CGBL(15,2),PCG(2),ACG(2),BCG(2),CCG(2)
      DIMENSION WARG(2),WARGD(2),AW(2),BW(2),CW(2),WIND(3)
      EQUIVALENCE (WMAE,WARG(1)),(AZW,WARG(2)),(FWMOP,WARGD(1)),(PAZWOH,
     *WARGD(2))
      EQUIVALENCE (W,STATE(1)),(U,STATE(2)),(V,STATE(3)),(X,STATE(4)),(Y
     *,STATE(5)),(Z,STATE(6)),(XMI,STATE(7)),(WBL(1),STBL(1,1)),(UBL(1),
     *STBL(1,2)),(VBL(1),STBL(1,3)),(XBL(1),STBL(1,4)),(YBL(1),STBL(1,5)
```

```
     *),(ZBL(1),STBL(1,6)),(XMBL(1),STBL(1,7))
      EQUIVALENCE (XCG,CG(1)),(YCG,CG(2)),(TXCG(1),CGBL(1,1)),(TYCG(1),C
     *GBL(1,2)),(PXCGM,PCG(1)),(PYCGM,PCG(2))
      EQUIVALENCE (PCAAM,AERODD(1)),(PCACM,AERODD(2)),(PCLBM,AERODD(3)),
     *(PCMAM,AERODD(4)),(PCMOM,AERODD(5)),(PCNAM,AERODD(6)),(PCNBM,AEROD
     *D(7)),(PCNOM,AERODD(8)),(PCYBM,AERODD(9)),(PCLOM,AERODD(10)),
     *(PCYOM,AERODD(11)),(PCNBOM,AERODD(12)),(PFABM,AERODD(13)),(PFNBM,
     *AERODD(14)),(PFMBM,AERODD(15))
      EQUIVALENCE (CAALP,AERO(1)),(CAO,AERO(2)),(CLBETA,AERO(3)),(CMALP,
     *AERO(4)),(CMO,AERO(5)),(CNALP,AERO(6)),(CNBETA,AERO(7)),(CNO,AERO(
     *8)),(CYBETA,AERO(9)),(CLO,AERO(10)),(CYO,AERO(11)),
     *(CNBO,AERO(12)),(FAB,FABT(1)),(FNB,FABT(2)),(FMB,FABT(3))
      DIMENSION VAERO(18),VAERDD(18),VA(18),VB(18),VC(18)
      EQUIVALENCE(ACACF,VAERO(1)),(BCACF,VAERO(2)),(CCACF,VAERO(3)),
     *(ACNCF,VAERO(4)),(BCNCF,VAERO(5)),(CCNCF,VAERO(6)),
     *(ACMCF,VAERO(7)),(BCMCF,VAERO(8)),(CCMCF,VAERO(9)),
     *(ACAMF,VAERO(10)),(BCAMF,VAERO(11)),(CCAMF,VAERO(12)),
     *(ACNMF,VAERO(13)),(BCNMF,VAERO(14)),(CCNMF,VAERO(15)),
     *(ACMMF,VAERO(16)),(BCMMF,VAERO(17)),(CCMMF,VAERO(18))
      LOGICAL VISC,PONLY,NOWIND
      DATA XK1,XK2,XK3,XK4,XK5,XK6,XK7/17700.,.0019403,.3048,.3048.,
     *110.4,.432,.50063./
      PONLY=KWTA(KCYTAB).EQ.2
      T=TENC-TBK
C*******
C     GETS STATE VARIABLES FROM FORWARD RUN
      IF(T.GE.TBL(1))GO TO 1
      IF(T.LT.TO+1.)GO TO 1
      NENT=101
      MBS5=NENT
      MBS6=0
C     EQUATE TABLES(101) TO TABLES(1)
      TBL(101)=TBL(1)
      WBL(101)=WBL(1)
      UBL(101)=UBL(1)
      VBL(101)=VBL(1)
      XBL(101)=XBL(1)
      YBL(101)=YBL(1)
      ZBL(101)=ZBL(1)
      XMBL(101)=XMBL(1)
      BACKSPACE 3
      READ(3)(TBL(I),WBL(I),UBL(I),VBL(I),XBL(I),YBL(I),ZBL(I),XMBL(I),
     *I=1,100)
      BACKSPACE 3
      IF(TBL(100).GT.TBL(101)) NENT=100
    1 CONTINUE
      CALL SPLINE(7,MBS5,101,T,TBL,STATE,STBL,SDG,1,MBS6,AA,BB,CC)
C*******
      XMIAD=XMI
      XM=XMIAD+XMALG
C
C     CALCULATE ALTITUDE AND PARTIAL DERIVATIVES W.R.T STATE
C
      R2=X*X+Y*Y+Z*Z
      R=SQRT(R2)
      CTH=(A(1,2)*X+A(2,2)*Y+A(3,2)*Z)/R
      CT2=CTH*CTH
      CALL AEEO(2)
      TEMP1=G22*X+G23*A(1,2)
      TEMP2=G22*Y+G23*A(2,2)
      TEMP3=G22*Z+G23*A(3,2)
      TEMP4=G33*A(1,2)+G23*X
      TEMP5=G33*A(2,2)+G23*Y
      TEMP6=G33*A(3,2)+G23*Z
C
      GXX=G11+TEMP1*X+TEMP4*A(1,2)
      GYX=     TEMP1*Y+TEMP4*A(2,2)
      GZX=     TEMP1*Z+TEMP4*A(3,2)
      GYY=G11+TEMP2*Y+TEMP5*A(2,2)
```

```
      CZY=     TEMP2*Z+TEMP5*A(3,2)
      CZZ=G11+TEMP3*Z+TEMP6*A(3,2)
C
      RTHE=UMF*RE/SQRT(UMF*UMF*(1.-CT2)+CT2)
      UMFC=FLAT*(2.-FLAT)/(RE*RE*UMF*UMF)
      ALT=R-RTHE
      ORTHE=RTHE**3*UMFC*CTH/R2
      FHX=X/R-(X*CTH-R*A(1,2))*ORTHE
      FHY=Y/R-(Y*CTH-R*A(2,2))*ORTHE
      FHZ=Z/R-(Z*CTH-R*A(3,2))*ORTHE
C
C
C     CALCULATE RELATIVE VELOCITY AND PARTIAL DERIVATIVES W.R.T STATE
C
    2 CMX=A12W
      CMY=A22W
      CMZ=A32W
      TEMP=CMY*Z-CMZ*Y
      TEMP1=CMZ*X-CMX*Z
      TEMP2=CMX*Y-CMY*X
      WIND(1)=0.
      WIND(2)=0.
      WIND(3)=0.
      NOWINC=.TRUE.
      IF(NWIND.EQ.0) GO TO 200
      IF(ALT.LT.ALTTBL(1).OR.ALT.GT.ALTTBL(NWIND)) GO TO 200
      NOWINC=.FALSE.
      TEMP3=TEMP*TEMP+TEMP1*TEMP1+TEMP2*TEMP2
      F1=(TEMP1*CMZ-TEMP2*CMY)/TEMP3
      F2=(TEMP2*CMX-TEMP *CMZ)/TEMP3
      F3=(TEMP *CMY-TEMP1*CMX)/TEMP3
      TEMP4=1./SQRT(TEMP3)
C
      E1=TEMP *TEMP4
      E2=TEMP1*TEMP4
      E3=TEMP2*TEMP4
      PE1X=-E1*P1
      PE1Y=-E1*P2-CMZ*TEMP4
      PE1Z=-E1*P3+CMY*TEMP4
      PE2X=-E2*P1+CMZ*TEMP4
      PE2Y=-E2*P2
      PE2Z=-E2*P3-CMX*TEMP4
      PE3X=-E3*P1-CMY*TEMP4
      PE3Y=-E3*P2+CMX*TEMP4
      PE3Z=-E3*P3
C
      TEMP5=1./R
      TEMP6=TEMP5*TEMP5*TEMP5
      PRXX=TEMP5-X*X*TEMP6
      PRXY=      -X*Y*TEMP6
      PRXZ=      -X*Z*TEMP6
      PRYY=TEMP5-Y*Y*TEMP6
      PRYZ=      -Y*Z*TEMP6
      PRZZ=TEMP5-Z*Z*TEMP6
C
      CALL SPLINE(2,MBS9,75,ALT,ALTTBL,WARG,WTBL,WARGC,2,MBS10,AW,BW,CW)
      SAZW=-SIN(AZW)
      CAZW=-COS(AZW)
C
      TEMP4=(E3*Y-E2*Z)*TEMP5
      TEMP3=SAZW*E1+CAZW*TEMP4
      WIND(1)=WMAG*TEMP3
      FW1X=FWMOM*PHX*TEMP3+WMAG*(SAZW*(PE1X-PAZWCH*PHX*TEMP4)+CAZW*((PE3
     *)*Y-PE2X*Z)*TEMP5+(E3*PRXY-E2*PRXZ)+PAZWOM*FHX*E1))
      FW1Y=FWMOM*PHY*TEMP3+WMAG*(SAZW*(PE1Y-PAZWCH*PHY*TEMP4)+CAZW*((PE3
     *Y*Y-PE2Y*Z)*TEMP5+(E3*PRYY-E2*PRYZ)+PAZWOM*FHY*E1))
      FW1Z=FWMOM*PHZ*TEMP3+WMAG*(SAZW*(PE1Z-PAZWCH*PHZ*TEMP4)+CAZW*((PE3
     *Z*Y-PE2Z*Z)*TEMP5+(E3*PRYZ-E2*PRZZ)+PAZWOM*FHZ*E1))
C
      TEMP4=(E1*Z-E3*X)*TEMP5
```

```
      TEMP3=SAZW*E2+CAZW*TEMP4
      WIND(2)=WMAG*TEMP3
      FW2X=FWMDH*PHX*TEMP3+WMAG*(SAZW*(PE2X-PAZWDH*PHX*TEMP4)+CAZW*((PE1
     *X*Z-PE3X*X)*TEMP5+(E1*PRXZ-E3*PRXX)+PAZWDH*FHX*E2))
      FW2Y=FWMDH*PHY*TEMP3+WMAG*(SAZW*(PE2Y-PAZWDH*PHY*TEMP4)+CAZW*((PE1
     *Y*Z-PE3Y*X)*TEMP5+(E1*PRYZ-E3*PRXY)+PAZWDH*FHY*E2))
      FW2Z=FWMDH*PHZ*TEMP3+WMAG*(SAZW*(PE2Z-PAZWDH*PHZ*TEMP4)+CAZW*((PE1
     *Z*Z-PE3Z*X)*TEMP5+(E1*PRZZ-E3*PRXZ)+PAZWDH*FHZ*E2))
C
      TEMP4=(E2*X-E1*Y)*TEMP5
      TEMP3=SAZW*E3+CAZW*TEMP4
      WIND(3)=WMAG*TEMP3
      FW3X=FWMDH*PHX*TEMP3+WMAG*(SAZW*(PE3X-PAZWDH*PHX*TEMP4)+CAZW*((PE2
     *X*X-PE1X*Y)*TEMP5+(E2*PRXX-E1*PRXY)+PAZWDH*FHX*E3))
      FW3Y=FWMDH*PHY*TEMP3+WMAG*(SAZW*(PE3Y-PAZWDH*PHY*TEMP4)+CAZW*((PE2
     *Y*X-PE1Y*Y)*TEMP5+(E2*PRXY-E1*PRYY)+PAZWDH*FHY*E3))
      FW3Z=FWMDH*PHZ*TEMP3+WMAG*(SAZW*(PE3Z-PAZWDH*PHZ*TEMP4)+CAZW*((PE2
     *Z*X-PE1Z*Y)*TEMP5+(E2*PRXZ-E1*PRYZ)+PAZWDH*FHZ*E3))
  200 SW=W-TEMP-WIND(1)
      SU=U-TEMP1-WIND(2)
      SV=V-TEMP2-WIND(3)
      VR=SQRT(SW*SW+SU*SU+SV*SV)
      TEMP=1./VR
      FVRX=(SV*OMY-SU*OMZ)*TEMP
      FVRY=(SW*OMZ-SV*OMX)*TEMP
      FVRZ=(SU*OMX-SW*OMY)*TEMP
      IF(NOWIND) GO TO 201
      FVRX=FVRX+(-SW*FW1X-SU*FW2X-SV*FW3X)*TEMP
      FVRY=FVRY+(-SW*FW1Y-SU*FW2Y-SV*FW3Y)*TEMP
      FVRZ=FVRZ+(-SW*FW1Z-SU*FW2Z-SV*FW3Z)*TEMP
  201 FVRW=SW*TEMP
      FVRU=SU*TEMP
      FVRV=SV*TEMP
C
      PAT(1)=ALT
      CALL FRA63(PAT,TMESN,IER)
      PAM=PAT(2)*10000.
      PHG=PAT(6)
      SOS=PAT(9)
      DPDH=PAT(3)*10000E-
      CRCH=PAT(4)
      CSGH=PAT(5)
      XMU=PAT(10)
      CMUH=PAT(11)
      REYNO=RHO*VR*XLEN/XMU
C
C     PARTIAL OF Q W.R.T STATE
C
      Q=.5*RHO*VR*VR
      TEMP1=RHO*VR
      TEMP2=.5*VR*VR*CRDH
      FQX=TEMP1*PVRX+TEMP2*PHX
      FQY=TEMP1*PVRY+TEMP2*PHY
      FQZ=TEMP1*PVRZ+TEMP2*PHZ
      FQW=TEMP1*PVRW
      FQU=TEMP1*PVRU
      FQV=TEMP1*PVRV
C
C     MACH NUMBER AND PARTIAL W.R.T STATE
C
      XMACH=VR/SOS
      TEMP1=1./SOS
      TEMP2=XMACH*CSDH
      FMX=TEMP1*(PVRX-TEMP2*PHX)
      FMY=TEMP1*(PVRY-TEMP2*PHY)
      FMZ=TEMP1*(PVRZ-TEMP2*PHZ)
      FMW=TEMP1*PVRW
      FMU=TEMP1*PVRU
      FMV=TEMP1*PVRV
```

```
C
C       DETERMINE AERO COEFFICIENTS AND PARTIALS W.R.T MACH NUMBER
C
        SMACH=XMACH
        IF(XMACH.GT.10.) XMACH=10.
        CALL SPLINE(12,MBS1,30,XMACH,AERO0(1,1,1),AERO,XERO,AERODD,2,MBS2,
       *AAERO,BAERO,CAERO)
        DMACH=SMACH
C
C       DETERMINE BASE AXIAL FORCE AND PARTIAL W.R.T ALTITUDE
C
C       CALCULATE ALTITUDE FOR BASE FORCE TABLE LOOKUP
C
        DALTB=AERO0(1,1,1)-ALTLS
        HGT=ALT+DALTB
        IF(ITHR.GE.NCBASE) GO TO 31
        L=LSTEE(ITHR)
        IF(L.EQ.2.AND.IAS(1,ITHR).LT.2.9) L=3
        CALL SPLINE(1,MBS3,50,HGT,AERO0(1,1,L),FAB1,AERO0(1,2,L),
       *AERODD(13),2,MBS4,AFAB,BFAB,CFAB)
        FYB=0.
        FLB=0.
        FNMB=0.
        FFYBH=0.
        FFLBH=0.
        FFNMBH=0.
        GO TO 32
   31   FAB=DEAXIL
        FFABH=0.
        FNB=0.
        FYB=0.
        FMB=0.
        FLB=0.
        FNMB=0.
        FFNBH=0.
        FFYBH=0.
        FFMBH=0.
        FFLBH=0.
        FFNMBH=0.
   32   CONTINUE
        IF(ITCL(ITHR).EQ.0) GO TO 320
        CALL AMULG(2,MBS21,20,XMACH,TOMACH,DCAX,DELCA,DCNP,DELCN)
        CALL AMULG(1,MBS21,20,XMACH,TOMACH,CCMZ,DELCM,C,D)
        CAC=CAC+DCAX
        CNC=CNC+DCNP
        CMC=CMC+DCMZ
        CALL SPLINE(1,MBS30,25,HGT,DELALT,DFAB,DELFAB,DDFAB,2,MBS31,AF,BF,
       *CF)
        FAB=FAB+DFAB(1)
        FFABH=FFABH+DDFAB(1)
  320   CONTINUE
C
C       DETERMINE ATTITUDE ANGLES
C
        CALL ACNTRC
        SCHIP=SIN(CHIP)
        CCHIP=COS(CHIP)
        SCHIY=SIN(CHIY)
        CCHIY=COS(CHIY)
        XHAT(1)=CCHIR*CCHIP
        XHAT(2)=-CCHIR*SCHIP
        YHAT(1)=CCHIY*SCHIP
        YHAT(2)=CCHIY*CCHIP
        YHAT(3)=SCHIY
        ZHAT(1)=-CCHIR*SCHIY*SCHIP
        ZHAT(2)=-CCHIR*SCHIY*CCHIP
        ZHAT(3)=CCHIR*CCHIY
C       PARTIAL OF DIRECTION COSINES W.R.T CONTROL
C
        FX1CP=XHAT(2)
```

```
      FX2CP=-XHAT(1)
      FY1CP=YHAT(2)
      FY2CP=-YHAT(1)
      FY3CP=0.
      FZ1CP=ZHAT(2)
      FZ2CP=-ZHAT(1)
      FZ3CP=0.
C
      FX1CPF=-XHAT(1)
      FX2CPF=-XHAT(2)
      FY1CPF=-YHAT(1)
      FY2CPF=-YHAT(2)
      FY3CPF=0.
      FZ1CPF=-ZHAT(1)
      FZ2CPF=-ZHAT(2)
      FZ3CPF=0.
C
      IF(PONLY) GO TO 59
      FX1CY=0.
      FX2CY=0.
      FY1CY=-SCHIP*SCHIY
      FY2CY=-SCHIY*CCHIP
      FY3CY=CCHIY
      FZ1CY=-CCHIR*YHAT(1)
      FZ2CY=-CCHIR*YHAT(2)
      FZ3CY=-CCHIR*SCHIY
C
      FX1CYY=0.
      FX2CYY=0.
      FY1CYY=-YHAT(1)
      FY2CYY=-YHAT(2)
      FY3CYY=-YHAT(3)
      FZ1CYY=-ZHAT(1)
      FZ2CYY=-ZHAT(2)
      FZ3CYY=-ZHAT(3)
C
      FX1CPY=0.
      FX2CPY=0.
      FY1CPY=-SCHIY*CCHIF
      FY2CPY=-PY1CY
      FY3CPY=0.
      FZ1CPY=PZ2CY
      FZ2CPY=-PZ1CY
      FZ3CPY=0.
   99 CONTINUE
C
C     COMPUTE ANGLE OF ATTACK AND PARTIAL W.R.T STATE
C
      TEMP1=XHAT(1)*SU+XHAT(2)*SU
      TEMP2=YHAT(1)*SU+YHAT(2)*SU+YHAT(3)*SV
      ALP=ATAN2(TEMP1,TEMP2)
      CALP=COS(ALP)
      TEMP3=CALP*CALP/TEMP2
      TEMP4=TEMP1/TEMP2
      FAX=TEMP3*(-XHAT(2)*OMZ+TEMP4*(YHAT(2)*OMZ-YHAT(3)*OMY))
      FAY=TEMP3*(+XHAT(1)*OMZ+TEMP4*(XHAT(3)*OMX-YHAT(1)*OMZ))
      FAZ=TEMP3*(-XHAT(1)*OMY+XHAT(2)*OMX+TEMP4*(YHAT(1)*OMY-YHAT(2)*OMX
     *))
      IF(NOLIND) GO TO 29
      TEMP5=XHAT(1)-YHAT(1)*TEMP4
      TEMP6=XHAT(2)-YHAT(2)*TEMP4
      TEMP7=        -YHAT(3)*TEMP4
      FAX=FAX-TEMP3*(TEMP5*PW1X+TEMP6*PW2X+TEMP7*PW3X)
      FAY=FAY-TEMP3*(TEMP5*PW1Y+TEMP6*PW2Y+TEMP7*PW3Y)
      FAZ=FAZ-TEMP3*(TEMP5*PW1Z+TEMP6*PW2Z+TEMP7*PW3Z)
   29 CONTINUE
      FAU=(XHAT(1)-YHAT(1)*TEMP4)*TEMP3
      FAU=(XHAT(2)-YHAT(2)*TEMP4)*TEMP3
      FAV=(        -YHAT(3)*TEMP4)*TEMP3
```

```
      FACP=TEMP3*(SW*XHAT(2)-SU*XHAT(1)-TEMP4*(SW*YHAT(2)-SU*YHAT(1)))
      FACPCF=-2.*PACP*(TEMP4*PACP*(YHAT(2)*SW-YHAT(1)*SU)/TEMP2)+TEMP3*(
     *-TEMP1+TEMP4*(YHAT(1)*SW+YHAT(2)*SU))
      IF(PONLY) GO TO 10
      TEMP5=-SCHIY*(SW*SCHIP+SU*CCHIP)+SV*CCHIY
      FACY=-TEMP3*TEMP*TEMP5
      FACYCY=-2.*PACY*(TEMP4*PACY+TEMP5/TEMP2)+TEMP2*TEMP3*TEMP4
      FACPCY=-2.*TEMP4*PACP*PACY-PACY*(YHAT(2)*SW-YHAT(1)*SU)/TEMP2-FACP
     *TEMP5/TEMP2-TEMP3*TEMP4*SCHIY*(-SW*CCHIP+SU*SCHIP)
   30 CONTINUE
C
C     CALCULATE SIDESLIP ANGLE AND PARTIALS
C
      TEMP1=(ZHAT(1)*SW+ZHAT(2)*SU+ZHAT(3)*SV)/VR
      ALFY=ASIN(TEMP1)
      TEMP2=1./(VR*COS(ALFY))
      FBX=TEMP2*(-ZHAT(2)*OMZ+ZHAT(3)*OMY-TEMP1*FVRX)
      FBY=TEMP2*(-ZHAT(3)*OMX+ZHAT(1)*OMZ-TEMP1*FVRY)
      FBZ=TEMP2*(-ZHAT(1)*OMY+ZHAT(2)*OMX-TEMP1*FVRZ)
      IF(NOWIND) GO TO 33
      FBX=FBX-TEMP2*(ZHAT(1)*PW1X+ZHAT(2)*PW2X+ZHAT(3)*PW3X)
      FBY=FBY-TEMP2*(ZHAT(1)*PW1Y+ZHAT(2)*PW2Y+ZHAT(3)*PW3Y)
      FBZ=FBZ-TEMP2*(ZHAT(1)*PW1Z+ZHAT(2)*PW2Z+ZHAT(3)*PW3Z)
   33 CONTINUE
      FBW=TEMP2*(ZHAT(3)-TEMP1*PVRW)
      FBU=TEMP2*(ZHAT(2)-TEMP1*PVRU)
      FBV=TEMP2*(ZHAT(1)-TEMP1*PVRV)
      FBCP=TEMP2*(SW*ZHAT(2)-SU*ZHAT(1))
      FBCPCF=TAN(ALFY)*FBCP*FBCP-TEMP2*(ZHAT(1)*SW+ZHAT(2)*SU)
      IF(PONLY) GO TO 11
      FBCY=TEMP2*(-CCHIR*(CCHIY*(SW*SCHIP+SU*CCHIP)+SV*SCHIY))
      FBCYCY=TAN(ALFY)*FBCY*FBCY-TEMP2*(ZHAT(1)*SW+ZHAT(2)*SU+ZHAT(3)*SV
     *)
      FBCPCY=TAN(ALFY)*FBCP*FBCY-TEMP2*CCHIY*(XHAT(1)*SW+XHAT(2)*SU)
   11 CONTINUE
      DELTE=0.
      DELM=0.
      DNBELE=0.
      DAELE=0.
      DMDELE=0.
      FDADM=0.
      FDNDM=0.
      FDMDM=0.
C
C     PARTIAL OF AERO COEFFICIENTS W.R.T STATE
C
      SQ=S(ITHR)*Q
      VISC=XMACH.GT.10.
      IF(.NOT.VISC) GO TO 110
      CAG=0.
      CAALP=0.
      CNQ=0.
      CNALP=0.
      CMQ=0.
      CMALP=0.
      DO 290 I=1,11
      AERODD(I)=0.
  290 CONTINUE
  110 CONTINUE
      CN=CNC+CNALP*ALF+CNDELE*DELTE+FNB/SQ
      TEMP1=FDNDM+ALF*FDNAM
      FDNX=TEMP1*PPX+CNALP*PAX+(PFNBH*PHX-FNB/Q*FDX)/SQ
      FDNY=TEMP1*PPY+CNALP*PAY+(PFNBH*PHY-FNB/Q*FDY)/SQ
      FDNZ=TEMP1*PPZ+CNALP*PAZ+(PFNBH*PHZ-FNB/Q*FDZ)/SQ
      FDNW=TEMP1*PPW+CNALP*PAW+(          -FNB/Q*FDW)/SQ
      FDNU=TEMP1*PPU+CNALP*PAU+(          -FNB/Q*FDU)/SQ
      FDNV=TEMP1*PPV+CNALP*PAV+(          -FNB/Q*FDV)/SQ
C
      CA=CAG+CAALP*ALF+CADELE*DELTE+FAB/SQ
```

```
      TEMP1=FCADM+ALP*FCAAM
      FCAX=TEMP1*PPX+CAALP*PAX+(PFABH*PHX-FAB/Q*FCX)/SQ
      FCAY=TEMP1*PPY+CAALP*PAY+(PFABH*PHY-FAB/Q*FCY)/SQ
      FCAZ=TEMP1*PPZ+CAALP*PAZ+(PFABH*PHZ-FAB/Q*FCZ)/SQ
      FCAW=TEMP1*PPW+CAALP*PAW+(          -FAB/Q*FCW)/SQ
      FCAU=TEMP1*PPU+CAALP*PAU+(          -FAB/Q*FCU)/SQ
      FCAV=TEMP1*PPV+CAALP*PAV+(          -FAB/Q*FCV)/SQ
C
C     VISCOUS DRAG CORRECTIONS WHEN MACH GT 10.0
      SREYNC=SQRT(REYNC)
      IF(.NOT.VISC) GO TO 6C
      CALL SPLINE(18,PBS18,10,XMACH,VAERDD(1,19),VAERDC,VAERDC,VAERDD,2,
     *PBS19,VA,VB,VC)
C
C     PARTIAL OF CONTINUUM FLOW COEFFICIENTS W.R.T STATE
C
      ALPHA=ALP*RAD
      CACF=ACACF+(BCACF+CCACF*ALPHA)*ALPHA
      TEMP1= VAERDD(1)+(VAERDD(2)+VAERDD(3)*ALPHA)*ALPHA
      TEMP2=BCACF+2.*CCACF*ALPHA
      FCACFX=TEMP1*PMX+TEMP2*PAX*RAD
      FCACFY=TEMP1*PMY+TEMP2*PAY*RAD
      FCACFZ=TEMP1*PMZ+TEMP2*PAZ*RAD
      FCACFW=TEMP1*PMW+TEMP2*PAW*RAD
      FCACFU=TEMP1*PMU+TEMP2*PAU*RAD
      FCACFV=TEMP1*PMV+TEMP2*PAV*RAD
C
      CNCF=ACNCF+(BCNCF+CCNCF*ALPHA)*ALPHA
      TEMP1=VAERDD(4)+(VAERDD(5)+VAERDD(6)*ALPHA)*ALPHA
      TEMP2=BCNCF+2.*CCNCF*ALPHA
      FCNCFX=TEMP1*PMX+TEMP2*PAX*RAD
      FCNCFY=TEMP1*PMY+TEMP2*PAY*RAD
      FCNCFZ=TEMP1*PMZ+TEMP2*PAZ*RAD
      FCNCFW=TEMP1*PMW+TEMP2*PAW*RAD
      FCNCFU=TEMP1*PMU+TEMP2*PAU*RAD
      FCNCFV=TEMP1*PMV+TEMP2*PAV*RAD
C
      CMCF=ACMCF+(BCMCF+CCMCF*ALPHA)*ALPHA
      TEMP1=VAERDD(7)+(VAERDD(8)+VAERDD(9)*ALPHA)*ALPHA
      TEMP2=BCMCF+2.*CCMCF*ALPHA
      FCMCFX=TEMP1*PMX+TEMP2*PAX*RAD
      FCMCFY=TEMP1*PMY+TEMP2*PAY*RAD
      FCMCFZ=TEMP1*PMZ+TEMP2*PAZ*RAD
      FCMCFW=TEMP1*PMW+TEMP2*PAW*RAD
      FCMCFU=TEMP1*PMU+TEMP2*PAU*RAD
      FCMCFV=TEMP1*PMV+TEMP2*PAV*RAD
C
C     PARTIAL OF MOLECULAR FLOW COEFFICIENTS W.R.T STATE
C
      CAMF=ACAMF+(BCAMF+CCAMF*ALPHA)*ALPHA
      TEMP1=VAERDD(10)+(VAERDD(11)+VAERDD(12)*ALPHA)*ALPHA
      TEMP2=BCAMF+2.*CCAMF*ALPHA
      FCAMFX=TEMP1*PMX+TEMP2*PAX*RAD
      FCAMFY=TEMP1*PMY+TEMP2*PAY*RAD
      FCAMFZ=TEMP1*PMZ+TEMP2*PAZ*RAD
      FCAMFW=TEMP1*PMW+TEMP2*PAW*RAD
      FCAMFU=TEMP1*PMU+TEMP2*PAU*RAD
      FCAMFV=TEMP1*PMV+TEMP2*PAV*RAD
C
      CNMF=ACNMF+(BCNMF+CCNMF*ALPHA)*ALPHA
      TEMP1=VAERDD(13)+(VAERDD(14)+VAERDD(15)*ALPHA)*ALPHA
      TEMP2=BCNMF+2.*CCNMF*ALPHA
      FCNMFX=TEMP1*PMX+TEMP2*PAX*RAD
      FCNMFY=TEMP1*PMY+TEMP2*PAY*RAD
      FCNMFZ=TEMP1*PMZ+TEMP2*PAZ*RAD
      FCNMFW=TEMP1*PMW+TEMP2*PAW*RAD
      FCNMFU=TEMP1*PMU+TEMP2*PAU*RAD
      FCNMFV=TEMP1*PMV+TEMP2*PAV*RAD
C
```

```
      CMMF=ACMMF+(BCMMF+CCMMF*ALPHA)*ALPHA
      TEMP1=VAERDD(16)+(VAERDD(17)+VAERDD(18)*ALPHA)*ALPHA
      TEMP2=BCMMF+2.*CCMMF*ALPHA
      PCMMFX=TEMP1*PMX+TEMP2*PAX*RAD
      PCMMFY=TEMP1*PMY+TEMP2*PAY*RAD
      PCMMFZ=TEMP1*PMZ+TEMP2*PAZ*RAD
      PCMMFW=TEMP1*PMW+TEMP2*PAW*RAD
      PCMMFU=TEMP1*PMU+TEMP2*PAU*RAD
      PCMMFV=TEMP1*PMV+TEMP2*PAV*RAD
C
C     PARTIAL OF KNUDSEN NUMBER W.R.T STATE
C
      XLAMDA=2.332353E-08*PAT(12)/PAM
      XKNUD=.1407*XLAMDA/VR*(PAM/RHO)**.5
      TEMP=EPDH/PAM-DREH/RHO+2.*2.332353E-08*PAT(13)/XLAMDA
      PKNUDX=XKNUD*(.5*TEMP*PHX-PVRX/VR)
      PKNUDY=XKNUD*(.5*TEMP*PHY-PVRY/VR)
      PKNUDZ=XKNUD*(.5*TEMP*PHZ-PVRZ/VR)
      PKNUDW=XKNUD*(-PVRW/VR)
      PKNUDU=XKNUD*(-PVRU/VR)
      PKNUDV=XKNUD*(-PVRV/VR)
C
      IF(XKNUD.GE..01.AND.XKNUD.LE.10.) GO TO 601
      PKNUDX=0.
      PKNUDY=0.
      PKNUDZ=0.
      PKNUDW=0.
      PKNUDU=0.
      PKNUDV=0.
  601 CONTINUE
C
      IF(XKNUD.LT..01) XKNUD=.01
      IF(XKNUD.GT.10.) XKNUD=10.
      TEMP1=SIN(9.009*(XKNUD-.01)/RAD)
      TEMP2=COS(9.009*(XKNUD-.01)/RAD)*(CAMF-CACF)*9.009/RAD
C
      PCAX=(PCAMFX-PCACFX)*TEMP1+TEMP2*PKNUDX+PCACFX+PCAX
      PCAY=(PCAMFY-PCACFY)*TEMP1+TEMP2*PKNUDY+PCACFY+PCAY
      PCAZ=(PCAMFZ-PCACFZ)*TEMP1+TEMP2*PKNUDZ+PCACFZ+PCAZ
      PCAW=(PCAMFW-PCACFW)*TEMP1+TEMP2*PKNUDW+PCACFW+PCAW
      PCAU=(PCAMFU-PCACFU)*TEMP1+TEMP2*PKNUDU+PCACFU+PCAU
      PCAV=(PCAMFV-PCACFV)*TEMP1+TEMP2*PKNUDV+PCACFV+PCAV
C
      TEMP2=COS(9.009*(XKNUD-.01)/RAD)*(CNMF-CNCF)*9.009/RAD
      PCNX=(PCNMFX-PCNCFX)*TEMP1+TEMP2*PKNUDX+PCNCFX+PCNX
      PCNY=(PCNMFY-PCNCFY)*TEMP1+TEMP2*PKNUDY+PCNCFY+PCNY
      PCNZ=(PCNMFZ-PCNCFZ)*TEMP1+TEMP2*PKNUDZ+PCNCFZ+PCNZ
      PCNW=(PCNMFW-PCNCFW)*TEMP1+TEMP2*PKNUDW+PCNCFW+PCNW
      PCNU=(PCNMFU-PCNCFU)*TEMP1+TEMP2*PKNUDU+PCNCFU+PCNU
      PCNV=(PCNMFV-PCNCFV)*TEMP1+TEMP2*PKNUDV+PCNCFV+PCNV
C
      TEMP2=COS(9.009*(XKNUD-.01)/RAD)*(CMMF-CMCF)*9.009/RAD
      PCMX=(PCMMFX-PCMCFX)*TEMP1+TEMP2*PKNUDX+PCMCFX
      PCMY=(PCMMFY-PCMCFY)*TEMP1+TEMP2*PKNUDY+PCMCFY
      PCMZ=(PCMMFZ-PCMCFZ)*TEMP1+TEMP2*PKNUDZ+PCMCFZ
      PCMW=(PCMMFW-PCMCFW)*TEMP1+TEMP2*PKNUDW+PCMCFW
      PCMU=(PCMMFU-PCMCFU)*TEMP1+TEMP2*PKNUDU+PCMCFU
      PCMV=(PCMMFV-PCMCFV)*TEMP1+TEMP2*PKNUDV+PCMCFV
C
C     CORRECTIONS TO CA AND CN
C
      CA=(CAMF-CACF)*TEMP1+CACF+CA
      CN=(CNMF-CNCF)*TEMP1+CNCF+CN
C
   60 CONTINUE
      CS=CYC+CYBETA*ALFY
      TEMP4=PCYOM+PCYBM*ALFY
      PCSX=TEMP4*PMX+CYBETA*PBX
      PCSY=TEMP4*PMY+CYBETA*PBY
      PCSZ=TEMP4*PMZ+CYBETA*PBZ
      PCSW=TEMP4*PMW+CYBETA*PBW
```

```
      FCSU=TEMP4*PPU+CTBETA*PBU
      FCSV=TEMP4*PPV+CTBETA*PBV
C
C     AERO MOMENTS
C
      DELWT=DLOW-XP
      IF(DELWT.LT.-G1)DELWT=-.01
      CALL SPLINE(2,MBS7,15,DELWT,TBDWT,CG,CGBL,FCG,2,MBSB,ACG,BCG,CCG)
      FXCGM=-PXCGM
      FYCGM=-PYCGM
C
C     FIXED ENGINE  FORCES AND MOMENTS
C
      FTMZX=0.
      FTMZY=0.
      FTMZZ=0.
      FTMZM=0.
      TFXX=0.
      TFYY=0.
      TFZZ=0.
      FFXX=0.
      FFXY=0.
      FFXZ=0.
      FFYX=0.
      FFYY=0.
      FFZY=0.
      FFZX=0.
      FFZY=0.
      FFZZ=0.
      TMZ=0.
      DO 500 N=3,6
      IF(TNE(N,ITHR).LT..001) GO TO 500
      MMIN=0
      DO 400 I=1,15
      IF(NSYST(I).NE.N) GO TO 400
      IF(MMIN.EQ.0) MMIN=I
      MMAX=I
  400 CONTINUE
      NM2=N-2
      GO TO (410,420,430,440),NM2
C
C     LBM SYSTEM
C
  410 CALL AMULG(1,MBLE1,30,T,TIMLBM,THRVAC,TLBM(1,1),0,0)
      THRUST=THRVAC*PTN-ENGDAT(8,MMIN)*PAM
      GO TO 450
C
C     OMS SYSTEM
C
  420 THRUST=FOMS(ITHR)*FTN-ENGDAT(8,MMIN)*PAM
      GO TO 450
C
C     RCS SYSTEM
C
  430 THRUST=FRCS(ITHR)*FTN-ENGDAT(8,MMIN)*PAM
      GO TO 450
C
C     FUTURE SYSTEM
C
  440 GO TO 450
C
  450 IF(INT(TNE(N,ITHR)+.1).GT.MMAX-MMIN+1) MMAX=INT(TNE(N,ITHR)+.1)+
     *MMIN-1
      DO 460 I=MMIN,MMAX
      TANP=TAN(ENGDAT(4,I)/RAD)
      TANY=TAN(ENGDAT(5,I)/RAD)
      TEMP=1./SQRT(1.+TANP*TANP+TANY*TANY)
```

```
      TFX=-THRUST*TEMP+TANP
      FTXXT=ENGDAT(8,I)*TEMP*TANP*OPOH*PHX
      FTXYT=ENGDAT(8,I)*TEMP*TANP*OPOH*PHY
      FTXZT=ENGDAT(8,I)*TEMP*TANP*OPOH*PHZ
      TFY=THRUST*TEMP
      FTYXT=-ENGDAT(8,I)*TEMP*OPOH*PHX
      FTYYT=-ENGDAT(8,I)*TEMP*OPOH*PHY
      FTYZT=-ENGDAT(8,I)*TEMP*OPOH*PHZ
      TFZZ=THRUST*TEMP+TANY+TFZZ
      FFZX=-ENGDAT(8,I)*TEMP*TANY*OPOH*PHX+PFZX
      FFZY=-ENGDAT(8,I)*TEMP*TANY*OPOH*PHY+PFZY
      FFZZ=-ENGDAT(8,I)*TEMP*TANY*OPOH*PHZ+PFZZ
      TMZ=-TFY*(XCG-ENGDAT(3,I))+TFX*(YCG-ENGDAT(1,I))+TMZ
      FTMZX=-PTYXT*(XCG-ENGDAT(3,I))+PTXXT*(YCG-ENGDAT(1,I))+PTMZX
      FTMZY=-PTYYT*(XCG-ENGDAT(3,I))+PTXYT*(YCG-ENGDAT(1,I))+PTMZY
      FTMZZ=-PTYZT*(XCG-ENGDAT(3,I))+PTXZT*(YCG-ENGDAT(1,I))+PTMZZ
      FFXX=FTXXT+PFXX
      FFXY=FTXYT+PFXY
      FFXZ=FTXZT+PFXZ
      FFYX=FTYXT+PFYX
      FFYY=FTYYT+PFYY
      FFYZ=FTYZT+PFYZ
      FTMZM=-TFY*PXCGM+TFX*PYCGM+PTMZM
      TFXX=TFX+TFXX
      TFYY=TFY+TFYY
  400 CONTINUE
  500 CONTINUE
      CXPR=(XCG-XREF)/XLEN
      CYPR=(YCG-YREF)/XLEN
      AMX=CNBO+CNBETA*ALFY-CS*OYPR
      AMY=CLO+CLBETA*ALFY+CS*OXPR
      AMZ=CPC+CMALF*ALF+CA*CXPR-CN*OYPR
C
C     VISCIOUS EFFECTS TO PITCH MOMENTS ARE INCLUDED
C
      IF(VISC) AMZ=(CMMF-CMCF)*TEMP1+CMCF+CA*OXPR-CN*OYPR
C
      EXX=PCNBOM+PCNBM*ALFY
      EYX=PCLOM+PCLBM*ALFY
      EZX=PCMOM+PCMAM*ALF
      EZY=CMALP+CAALP*CXPR-CNALP*OYPR
      IF(.NOT.VISC) GO TO 603
      TEMP2=(BCAMF+2.*CCAMF*ALPHA)*RAD
      TEMP3=(BCACF+2.*CCACF*ALPHA)*RAD
      TEM1CA=TEMP1*(TEMP2-TEMP3)+TEMP3
      TEM2CA=(2.*(CCAMF-CCACF)*TEMP1+2.*CCACF)*RAD*RAD
      TEMP2=(BCNMF+2.*CCNMF*ALPHA)*RAD
      TEMP3=(BCNCF+2.*CCNCF*ALPHA)*RAD
      TEM1CN=TEMP1*(TEMP2-TEMP3)+TEMP3
      TEM2CN=(TEMP1*2.*(CCAMF-CCACF)+2.*CCACF)*RAD*RAD
      TEMP2=(BCMMF+2.*CCMMF*ALPHA)*RAD
      TEMP3=(BCMCF+2.*CCMCF*ALPHA)*RAD
      TEM1CM=(TEMP2-TEMP3)*TEMP3
      TEM2CM=(TEMP1*2.*(CCMMF-CCMCF)+2.*CCMCF)*RAD*RAD
      EZY=TEM1CM+TEM1CA*CXPR-TEM1CN*OYPR
      EZYY=TEM2CM+TEM2CA*OXPR-TEM2CN*OYPR
  603 CONTINUE
      TEMP2=CNBETA
      TEMP3=CLBETA
      EXZ=TEMP2-CYBETA*OXPR
      EYZ=TEMP2+CYBETA*OXPR
C
      TEMP1=XLEN*SIITHF
      FMAXX=TEMP1*(PQX+APX+Q*(EXX*PMX+TEMP2*FBX-FCSX*CYPR))+FFNMBH*PHX*
     *ALFY
      FMAYX=TEMP1*(PQY+APY+Q*(EYX*PMX+TEMP3*FBX+FCSX*CXPR))+FFLBH*PHX*
     *ALFY
      FMAZX=TEMP1*(PQZ+APZ+Q*(EZX*PMX+CMALP*PAX+FCAX*OXPR-PCAX*OYPR))+
     *FFMBH*PHX
```

```
      FMAXY=TEMP1*(PQY*APX+Q*(EXX*PMY+TEMP2*PBY-FCSY*CYPR))+FFMBH*PHY*
     *ALFY
      FMAYY=TEMP1*(PQY*APY+Q*(EYX*PMY+TEMP3*PBY+FCSY*CXPR))+FFLBH*PHY*
     *ALFY
      FMAZY=TEMP1*(PQY*APZ+Q*(EZX*PMY+CPALP*FAY+FCAY*CXPR-FCNY*DYPR))+
     *FFMBH*PHY
      FMAXZ=TEMP1*(PQZ*APX+Q*(EXX*PMZ+TEMP2*PBZ-FCSZ*CYPR))+FFMBH*PHZ*
     *ALFY
      FMAYZ=TEMP1*(PQZ*APY+Q*(EYX*PMZ+TEMP3*PBZ+FCSZ*CXPR))+FFLBH*PHZ*
     *ALFY
      FMAZZ=TEMP1*(PQZ*APZ+Q*(EZX*PMZ+CPALP*FAZ+FCAZ*CXPR-FCNZ*DYPR))+
     *FFMBH*PHZ
      FMAXW=TEMP1*(PQW*APX+Q*(EXX*PMW+TEMP2*PBW-FCSW*CYPR))
      FMAYW=TEMP1*(PQW*APY+Q*(EYX*PMW+TEMP3*PBW+FCSW*CXPR))
      FMAZW=TEMP1*(PQW*APZ+Q*(EZX*PMW+CPALP*FAW+FCAW*CXPR-FCNW*DYPR))
      FMAXU=TEMP1*(PQL*APX+Q*(EXX*PMU+TEMP2*PBU-FCSU*CYPR))
      FMAYU=TEMP1*(PQL*APY+Q*(EYX*PMU+TEMP3*PBU+FCSU*CXPR))
      FMAZU=TEMP1*(PQL*APZ+Q*(EZX*PMU+CPALP*FAU+FCAU*DXPR-FCNU*DYPR))
      FMAXV=TEMP1*(PQV*APX+Q*(EXX*PMV+TEMP2*PBV-FCSV*CYPR))
      FMAYV=TEMP1*(PQV*APY+Q*(EYX*PMV+TEMP3*PBV+FCSV*CXPR))
      FMAZV=TEMP1*(PQV*APZ+Q*(EZX*PMV+CPALP*FAV+FCAV*CXPR-FCNV*DYPR))
      IF(.NOT.VISC) GO TO 602
      FMAZX=FMAZX+TEMP1*C*PCMX
      FMAZY=FMAZY+TEMP1*C*PCMY
      FMAZZ=FMAZZ+TEMP1*C*PCMZ
      FMAZW=FMAZW+TEMP1*C*PCMW
      FMAZU=FMAZU+TEMP1*C*PCMU
      FMAZV=FMAZV+TEMP1*C*PCMV
  602 CONTINUE
C
      TEMP2=C*TEMP1
      TEMP4=TEMP2*CS/XLEN
      FMAXM=-TEMP4*PYCGM
      FMAYM=TEMP4*FXCGP
      FMAZM=TEMP2*(CA*FXCGP-CN*PYCGM)/XLEN
      TEMP3=TEMP2*EYZ
      FMAXCF=TEMP3*PBCF
      FXCPCF=TEMP3*PBCFCF
      IF(PONLY) GO TO 12
      FMAXCY=TEMP3*PBCY
      FXCYCY=TEMP3*PBCYCY
      FXCPCY=TEMP3*PBCFCY
   12 CONTINUE
      TEMP3=TEMP2*EYZ
      FMAYCF=TEMP3*PBCF
      FYCPCF=TEMP3*PBCFCF
      IF(PONLY) GO TO 13
      FMAYCY=TEMP3*PBCY
      FYCYCY=TEMP3*PBCYCY
      FYCPCY=TEMP3*PBCFCY
   13 CONTINUE
      TEMP3=TEMP2*EZY
      FMAZCF=TEMP3*PACF
      FZCPCF=TEMP3*PACFCF
      EZYY=EZYY*TEMP2
      IF(VISC) PZCPCP=FZCPCP+EZYY*PACP*PACP
      IF(PONLY) GO TO 14
      FMAZCY=TEMP3*PACY
      FZCYCY=TEMP3*PACYCY
      FZCPCY=TEMP3*PACFCY
      IF(.NOT.VISC) GO TO 14
      FZCPCY=PZCPCY+EZYY*PACP*PACY
      FZCYCY=PZCYCY+EZYY*PACY*PACY
   14 CONTINUE
C
      AMX=APX*TEMP2
      AMY=APY*TEMP2
      AMZ=APZ*TEMP2+FME
```

```
      AMZ=APZ+TMZ
      PMAZX=PMAZX+PTMZX
      PMAZY=PMAZY+PTMZY
      PMAZZ=PMAZZ+PTMZZ
      PMAZM=PMAZM+PTMZM
      CX=XGPA-XCG
      CY=YGPA-YCG
      CZ=ABS(ZGPA)
      CYI=1./DY
      CZI=1./DZ
      CXDY=CX*DYI
      UDXCYZ=1.+CXCY*CXDY
      CRHO=1./SQRT(UDXDYZ)
      SRHO=CXDY*CRHO
C
      CXX=DZI*SRHO
      CXY=DZI*CRHO
      CXZ=DYI*CRHO
      CYX=-CXX
      CYY=-CXY
      CYZ=CXZ
      CZX=DYI/UDXDYZ
      CZY=-CXDY*CZX
C
      TEMP4=PXEGM+DXEY*PYEGM
      PCRHOP=-SRHO*CZX*TEMP4
      PSRHOP=DXDY*PCRHOM+CRHO*TEMP4*DYI
      PCXXM=DZI*PSRHOP
      PCXYM=DZI*PCRHOP
      PCXZM=DYI*(PCRHOP+CXZ*PYCGM)
      PCYXM=-PCXXM
      PCYYM=-PCXYM
      PCYZM=PCXZM
      PCZXM=CZX*DYI*PYCGM+2.*CRHO*PCRHOM
      PCZYM=DYI*TEMP4*CZX-DXDY*PCZXM
C
      DO 26 I=1,15
      IF(NSYST(I).EQ.1) GO TO 27
   26 CONTINUE
   27 N=I
      THRUST=TNE(1,ITHR)*(THROT(ITHR)*ENGDAT(6,N)*PTN-ENGDAT(8,N)*PAM)
      PTM=0.
      IF(NBTRGS.NE.2.OR.MPSFLG(ITHR).NE.4) GO TO 3
      PTM=GLIM(ITHR)*EZERO
      EM=XM*PTM
      TAU=GP/(TNE(1,ITHR)*ENGDAT(6,N)*PTN)
      THRUST=TNE(1,ITHR)*(TAU*ENGDAT(6,N)*PTN-ENGDAT(8,N)*PAM)
    3 CONTINUE
      TEMP=-TNE(1,ITHR)*ENGDAT(8,N)*DPDM
      PTX=TEMP*PMX
      PTY=TEMP*PMY
      PTZ=TEMP*PMZ
C
C     GIMBAL ANGLES AND PARTIALS
C
      CTHR=1./THRUST
      TANDP1=-DTHR*(CXX*AMX+CXY*AMY+CXZ*AMZ)
      TANDP2=-DTHR*(CYX*AMX+CYY*AMY+CYZ*AMZ)
      TANDY=-DTHR*(CZX*AMX+CZY*AMY)
      PTP1X=-DTHR*(PTX*TANDP1+CXX*PMAXX+CXY*PMAYX+CXZ*PMAZX)
      PTP1Y=-DTHR*(PTY*TANDP1+CXX*PMAXY+CXY*PMAYY+CXZ*PMAZY)
      PTP1Z=-DTHR*(PTZ*TANDP1+CXX*PMAXZ+CXY*PMAYZ+CXZ*PMAZZ)
      PTP2X=-DTHR*(PTX*TANDP2+CYX*PMAXX+CYY*PMAYX+CYZ*PMAZX)
      PTP2Y=-DTHR*(PTY*TANDP2+CYX*PMAXY+CYY*PMAYY+CYZ*PMAZY)
      PTP2Z=-DTHR*(PTZ*TANDP2+CYX*PMAXZ+CYY*PMAYZ+CYZ*PMAZZ)
      PTYX =-DTHR*(PTX*TANDY +CZX*PMAXX+CZY*PMAYX)
      PTYY =-DTHR*(PTY*TANDY +CZX*PMAXY+CZY*PMAYY)
      PTYZ =-DTHR*(PTZ*TANDY +CZX*PMAXZ+CZY*PMAYZ)
```

```
      FTP1W=-OTHR*(CX)*PMAXW+CXY*PMAYW+CXZ*PMAZW)
      FTP1U=-OTHR*(CX)*PMAXU+CXY*PMAYU+CXZ*PMAZU)
      FTP1V=-OTHR*(CX)*PMAXV+CXY*PMAYV+CXZ*PMAZV)
      FTF2W=-OTHR*(CY)*PMAXW+CYY*PMAYW+CYZ*PMAZW)
      FTF2U=-OTHR*(CY)*PMAXU+CYY*PMAYU+CYZ*PMAZU)
      FTF2V=-OTHR*(CY)*PMAXV+CYY*PMAYV+CYZ*PMAZV)
      FTYW  =-OTHR*(CZ)*PMAXW+CZY*PMAYW)
      FTYU  =-OTHR*(CZ)*PMAXU+CZY*PMAYU)
      FTYV  =-OTHR*(CZ)*PMAXV+CZY*PMAYV)
      TEMP4=CTHR*PTM
      TEMP5=TEMP4*AMX-PMAXM
      TEMP6=TEMP4*AMY-PMAYM
      TEMP7=TEMP4*AMZ-PMAZM
      FTP1M=CTHR*((CX*TEMP5+CXY*TEMP6+CXZ*TEMP7-(PCXXM*AMX+PCXYM*AMY+PCX
     *ZM*AMZ))
      FTP2M=OTHR*((CYX*TEMP5+CYY*TEMP6+CYZ*TEMP7-(PCYXM*AMX+PCYYM*AMY+PCY
     *ZM*AMZ))
      FTYM  =CTHR*((CZX*TEMP5+CZY*TEMP6          -(PCZXM*AMX+PCZYM*AMY))
C
      FTP1CF=-OTHR*(CX)*PMAXCP+CXY*PMAYCF+CXZ*PMAZCP)
      FTF2CF=-OTHR*(CY)*PMAXCP+CYY*PMAYCF+CYZ*PMAZCP)
      FTYCP=-OTHR*(CZ)*PMAXCP+CZY*PMAYCP)
      IF(PONLY) GO TO 15
      FTP1CY=-OTHR*(CX)*PMAXCY+CXY*PMAYCY+CXZ*PMAZCY)
      FTF2CY=-OTHR*(CY)*PMAXCY+CYY*PMAYCY+CYZ*PMAZCY)
      FTYCY=-OTHR*(CZ)*PMAXCY+CZY*PMAYCY)
   15 CONTINUE
      FP1CPF=-OTHR*(CX)*FXCPCP+CXY*PYCPCF+CXZ*PZCFCP)
      FP2CPF=-OTHR*(CY)*FXCPCP+CYY*PYCPCF+CYZ*PZCFCP)
      FYCPCF=-OTHR*(CZ)*FXCPCP+CZY*PYCPCF)
      IF(PONLY) GO TO 16
      FP1CYY=-OTHR*(CX)*FXCYCY+CXY*PYCYCY+CXZ*PZCYCY)
      FP2CYY=-OTHR*(CY)*FXCYCY+CYY*PYCYCY+CYZ*PZCYCY)
      FYCYCY=-OTHR*(CZ)*FXCYCY+CZY*PYCYCY)
      FP1CPY=-OTHR*(CX)*FXCPCY+CXY*PYCPCY+CXZ*PZCFCY)
      FP2CPY=-OTHR*(CY)*FXCPCY+CYY*PYCPCY+CYZ*PZCFCY)
      FYCPCY=-OTHR*(CZ)*FXCPCY+CZY*PYCPCY)
   16 CONTINUE
C
      TEMP1=SQRT(1.+TANDF1*TANDF1+TANDY*TANDY)
      TEMP2=SQRT(1.+TANDF2*TANDF2+TANDY*TANDY)
      TY1=THRUST/2./TEMP1
      TY2=THRUST/2./TEMP2
      TXF=-TANDP1*TY1-TANDP2*TY2
      TYP=TY1+TY2
      TEMP7=TEMP1*TEMP1
      TEMP8=TEMP2*TEMP2
      FTY1X=TY1*(PTX*CTHR-(TANDP1*PTP1X+TANDY*PTYX)/TEMP7)
      FTY1Y=TY1*(PTY*CTHR-(TANDP1*PTP1Y+TANDY*PTYY)/TEMP7)
      FTY1Z=TY1*(PTZ*CTHR-(TANDP1*PTP1Z+TANDY*PTYZ)/TEMP7)
      FTY2X=TY2*(PTX*CTHR-(TANDP2*PTP2X+TANDY*PTYX)/TEMP8)
      FTY2Y=TY2*(PTY*CTHR-(TANDP2*PTP2Y+TANDY*PTYY)/TEMP8)
      FTY2Z=TY2*(PTZ*CTHR-(TANDP2*PTP2Z+TANDY*PTYZ)/TEMP8)
      FTY1M=TY1*(PTM*CTHR-(TANDP1*PTP1M+TANDY*PTYM)/TEMP7)
      FTY2M=TY2*(PTM*CTHR-(TANDP2*PTP2M+TANDY*PTYM)/TEMP8)
      TXFX=-PTP1X*TY1-FTF2X*TY2-TANDP1*PTY1X-TANDF2*FTY2X
      TXFY=-PTP1Y*TY1-FTF2Y*TY2-TANDP1*PTY1Y-TANDF2*FTY2Y
      TXFZ=-PTP1Z*TY1-FTF2Z*TY2-TANDP1*PTY1Z-TANDF2*FTY2Z
      TYFX=FTY1X+PTY2X
      TYFY=FTY1Y+PTY2Y
      TYFZ=FTY1Z+PTY2Z
C
      TEMP3=(TANDP1*PTP1CP+TANDY*PTYCP)/TEMP7
      TEMP4=(TANDP2*PTP2CP+TANDY*PTYCP)/TEMP8
      FTY1CF=-TY1*TEMP3
      FTY2CF=-TY2*TEMP4
      TXFCP=-PTP1CF*TY1-FTP2CP*TY2-TANDP1*PTY1CP-TANDF2*PTY2CP
      TYFCP=FTY1CP+PTY2CF
      IF(PONLY) GO TO 17
      TEMP5=(TANDP1*PTP1CY+TANDY*PTYCY)/TEMP7
```

```
      TEMP6=(TANDP2*PTP2CY+TANDY*PTYCY)/TEMP8
      FTY1CY=-TY1*TEMP5
      FTY2CY=-TY2*TEMP6
      TXPCY=-PTP1CY*TY1-PTP2CY*TY2-TANDP1*PTY1CY-TANDP2*PTY2CY
      TYPCY=PTY1CY+PTY2CY
   17 CONTINUE
      FYCCPP=TY1*(2.*TEMP3*TEMP3-(PTP1CP*PTP1CP+TANDP1*PP1CPP+PTYCP*
     *PTYCP+TANDY*FYCPCP)/TEMP7)
      FYTCPP=TY2*(2.*TEMP4*TEMP4-(PTP2CP*PTP2CP+TANDP2*PP2CPP+PTYCP*
     *PTYCP+TANDY*FYCPCP)/TEMP8)
      IF(PONLY) GO TO 18
      FYCCYY=TY1*(2.*TEMP5*TEMP5-(PTP1CY*PTP1CY+TANDP1*PP1CYY+PTYCY*
     *PTYCY+TANDY*FYCYCY)/TEMP7)
      FYTCYY=TY2*(2.*TEMP6*TEMP6-(PTP2CY*PTP2CY+TANDP2*PP2CYY+PTYCY*
     *PTYCY+TANDY*FYCYCY)/TEMP8)
      FYCCPY=TY1*(2.*TEMP3*TEMP5-(PTP1CY*PTP1CP+TANDP1*PP1CPY+PTYCP*
     *PTYCY+TANDY*FYCPCY)/TEMP7)
      FYTCPY=TY2*(2.*TEMP4*TEMP6-(PTP2CY*PTP2CP+TANDP2*PP2CPY+PTYCP*
     *PTYCY+TANDY*FYCPCY)/TEMP8)
   18 CONTINUE
      TXPCPP=-PP1CPP*TY1-PP2CPP*TY2-2.*(PTP1CP*PTY1CP+PTP2CP*PTY2CP)-
     *TANDP1*PYOCPP-TANDP2*PYTCPP
      TYPCPP=PYOCPP+PYTCPP
      FTZCPP=PYCPCP*TYP+2.*PTYCP*TYPCP+TANDY*TYPCPP
      FTXCPP=TXPCPP*CRHO+TYPCPP*SRHO
      FTYCPP=-TXPCPP*SRHO+TYPCPP*CRHO
      IF(PONLY) GO TO 19
      TXPCYY=-PP1CYY*TY1-PP2CYY*TY2-2.*(PTP1CY*PTY1CY+PTP2CY*PTY2CY)-
     *TANDP1*PYOCYY-TANDP2*PYTCYY
      TYPCYY=PYOCYY+PYTCYY
      FTZCYY=PYCYCY*TYP+2.*PTYCY*TYPCY+TANDY*TYPCYY
      TXPCPY=-PP1CPY*TY1-PP2CPY*TY2-PTP1CP*PTY1CY-PTP2CP*PTY2CY-PTP1CY*
     *PTY1CP-PTP2CY*PTY2CP-TANDP1*PYOCPY-TANDP2*PYTCPY
      TYPCPY=PYOCPY+PYTCPY
      FTZCPY=PYCPCY*TYP+PTYCP*TYPCY+PTYCY*TYPCP+TANDY*TYPCPY
      FTXCYY=TXPCYY*CRHO+TYPCYY*SRHO
      FTXCPY=TXPCPY*CRHO+TYPCPY*SRHO
      FTYCYY=-TXPCYY*SRHO+TYPCYY*CRHO
      FTYCPY=-TXPCPY*SRHO+TYPCPY*CRHO
   19 CONTINUE
C     PARTIAL OF THRUST COMPONENTS W.R.T. POSITION
      FTZX=FTYX*TXP+TANDY*TYPX+PFZX
      FTZY=FTYY*TXP+TANDY*TYPY+PFZY
      FTZZ=FTYZ*TXP+TANDY*TYPZ+PFZZ
      FTXX=TXPX*CRHO+TYPX*SRHO+PFXX
      FTXY=TXPY*CRHO+TYPY*SRHO+PFXY
      FTXZ=TXPZ*CRHO+TYPZ*SRHO+PFXZ
      FTYX=-TXPX*SRHO+TYPX*CRHO+PFYX
      FTYY=-TXPY*SRHO+TYPY*CRHO+PFYY
      FTYZ=-TXPZ*SRHO+TYPZ*CRHO+PFYZ
      FTZCP=PTYCP*TYP+TANDY*TYPCP
      FTXCP=TXPCP*CRHO+TYPCP*SRHO
      FTYCP=-TXPCP*SRHO+TYPCP*CRHO
      IF(PONLY) GO TO 20
      FTZCY=PTYCY*TYP+TANDY*TYPCY
      FTXCY=TXPCY*CRHO+TYPCY*SRHO
      FTYCY=-TXPCY*SRHO+TYPCY*CRHO
   20 CONTINUE
C     PARTIAL OF THRUST COMPONENTS W.R.T. VELOCITY
      FTY1W=-TY1*(TANDP1*PTP1W+TANDY*PTYW)/(TEMP1*TEMP1)
      FTY1U=-TY1*(TANDP1*PTP1U+TANDY*PTYU)/(TEMP1*TEMP1)
      FTY1V=-TY1*(TANDP1*PTP1V+TANDY*PTYV)/(TEMP1*TEMP1)
      FTY2W=-TY2*(TANDP2*PTP2W+TANDY*PTYW)/(TEMP2*TEMP2)
      FTY2U=-TY2*(TANDP2*PTP2U+TANDY*PTYU)/(TEMP2*TEMP2)
      FTY2V=-TY2*(TANDP2*PTP2V+TANDY*PTYV)/(TEMP2*TEMP2)
      TXPW=-PTP1W*TY1-PTP2W*TY2-TANDP1*PTY1W-TANDP2*PTY2W
      TXPU=-PTP1U*TY1-PTP2U*TY2-TANDP1*PTY1U-TANDP2*PTY2U
      TXPV=-PTP1V*TY1-PTP2V*TY2-TANDP1*PTY1V-TANDP2*PTY2V
```

```
      TYPW=FTY1W+PTY2W
      TYPU=FTY1U+PTY2U
      TYPV=FTY1V+PTY2V
      FTZW=FTYW*TYF+TANDY*TYPW
      FTZU=FTYU*TYF+TANDY*TYPU
      FTZV=FTYV*TYF+TANDY*TYPV
      FTXW=TXPW*CRHO+TYPW*SRHO
      FTXU=TXPU*CRHO+TYPU*SRHO
      FTXV=TXPV*CRHO+TYPV*SRHO
      FTYW=-TXPW*SRHO+TYFW*CRHO
      FTYU=-TXPU*SRHO+TYFU*CRHO
      FTYV=-TXPV*SRHO+TYFV*CRHO
      TX=TXF*CRHO+TYP*SRHO+TFXX
      TY=-TXP*SRHO+TYF*CRHO+TFYY
      TZ=TANCY*TYP+TFZZ
      TEMP=G*S(ITHF)
      TEMP1=TX-TEMF*CN
      TEMP2=TY-TEMF*CA
      TEMP3=TZ+TEMF*CS
      CX=TEMP1*XHAT(1)+TEMP2*YHAT(1)+TEMF3*ZHAT(1)
      CY=TEMP1*XHAT(2)+TEMP2*YHAT(2)+TEMF3*ZHAT(2)
      CZ=                TEMP2*YHAT(3)+TEMF3*ZHAT(3)
      TEMP5=TEMP*CNALF
      IF(VISC) TEMF5=TEMF5+TEM1CN*TEMP
      FAACP =PTXCP -TEMP5*PACP
      FAACPF=PTXCPF-TEMP5*PACPCP
      TEMP2CN=TEM2CN*TEMP
      IF(VISC) PAACPP=FAACPP+TEM2CN*PACP*PACF
      IF(PONLY) GO TO 21
      FAACY =PTXCY -TEMP5*PACY
      FAACYY=PTXCYY-TEMP5*PACYCY
      FAACPY=PTXCPY-TEMP5*PACPCY
      IF(.NOT.VISC) GO TO 21
      FAACYY=PAACYY+TEMP2CN*PACY*PACY
      FAACPY=PAACPY+TEMP2CN*PACP*PACY
   21 CONTINUE
      TEMP5=TEMP*CAALF
      IF(VISC) TEMF5=TEMF5+TEM1CA*TEMP
      FABCP =PTYCP -TEMP5*PACP
      FABCPF=PTYCPF-TEMP5*PACPCP
      TEM2CA=TEM2CA*TEMP
      IF(VISC) PABCPP=FABCPP+TEM2CA*PACP*PACF
      IF(PONLY) GO TO 22
      FABCY =PTYCY -TEMP5*PACY
      FABCYY=PTYCYY-TEMP5*PACYCY
      FABCPY=PTYCPY-TEMP5*PACPCY
      IF(.NOT.VISC) GO TO 22
      FABCYY=PABCYY+TEMP2CA*PACY*PACY
      FABCPY=PABCPY+TEMP2CA*PACP*PACY
   22 CONTINUE
      TEMP5=TEMP*CYBETA+FYB
      FACCP =PTZCP +TEMP5*PBCP
      FACCPF=PTZCPF+TEMP5*PBCPCP
      IF(PONLY) GO TO 23
      FACCY =PTZCY +TEMP5*PBCY
      FACCYY=PTZCYY+TEMP5*PBCYCY
      FACCPY=PTZCPY+TEMP5*PBCPCY
   23 CONTINUE
      OMASS=1./XM
      FDXCP=(XHAT(1)*FAACP+YHAT(1)*PABCP+ZHAT(1)*FACCP+TEMP1*PX1CP+TEMP2
     **PY1CF+TEMF3*PZ1CP)*OMASS
      FDYCP=(XHAT(2)*FAACP+YHAT(2)*PABCP+ZHAT(2)*FACCP+TEMP1*PX2CP+TEMP2
     **PY2CF+TEMP3*PZ2CP)*OMASS
      FDZCP=(                YHAT(3)*PABCP+ZHAT(3)*FACCP              +TEMP2
     **PY3CF+TEMP3*PZ3CP)*OMASS
      IF(PONLY) GO TO 24
      FDXCY=(XHAT(1)*FAACY+YHAT(1)*PABCY+ZHAT(1)*FACCY+TEMF1*PX1CY+TEMP2
     **PY1CY+TEMP3*PZ1CY)*OMASS
```

```
      FDYCY=(XHAT(2)*FAACY+YHAT(2)*PABCY+ZHAT(2)*FACCY+TEMP1*PX2CY+TEMP2
     *PY2CY+TEMP3*PZ2CY)*OMASS
      FDZCY=(                YHAT(3)*PABCY+ZHAT(3)*FACCY                +TEMP2
     *PY3CY+TEMP3*PZ3CY)*OMASS
   24 CONTINUE
C
      FDXCPF=(XHAT(1)*PAACPP+YHAT(1)*PABCPP+ZHAT(1)*PACCPP+2.*PX1CP*PAAC
     *F+2.*PY1CP*PABCF+2.*PZ1CP*PACCP+TEMP1*PX1CPF+TEMP2*PY1CPP+TEMP3*
     *PZ1CPF)*OMASS
      FDYCPF=(XHAT(2)*PAACPP+YHAT(2)*PABCPP+ZHAT(2)*PACCPP+2.*PX2CP*PAAC
     *F+2.*PY2CP *PABCF+2.*PZ2CP*PACCP+TEMP1*PX2CPP+TEMP2*PY2CPP+TEMP3*
     *PZ2CPF)*OMASS
      FDZCPF=(                YHAT(3)*PABCPP+ZHAT(3)*PACCPP
     * +2.*PY3CP *PABCF+2.*PZ3CP*PACCP                +TEMP2*PY3CPP+TEMP3*
     *PZ3CPF)*OMASS
C
      IF(PONLY) GO TO 25
      FDXCYY=(XHAT(1)*PAACYY+YHAT(1)*PABCYY+ZHAT(1)*PACCYY+2.*PX1CY*PAAC
     *Y+2.*PY1CY*PABCY+2.*PZ1CY*PACCY+TEMP1*PX1CYY+TEMP2*PY1CYY+TEMP3*
     *PZ1CYY)*OMASS
      FDYCYY=(XHAT(2)*PAACYY+YHAT(2)*PABCYY+ZHAT(2)*PACCYY+2.*PX2CY*PAAC
     *Y+2.*PY2CY*PABCY+2.*PZ2CY*PACCY+TEMP1*PX2CYY+TEMP2*PY2CYY+TEMP3*
     *PZ2CYY)*OMASS
      FDZCYY=(                YHAT(3)*PABCYY+ZHAT(3)*PACCYY
     * +2.*PY3CY*PABCY+2.*PZ3CY*PACCY                +TEMP2*PY3CYY+TEMP3*
     *PZ3CYY)*OMASS
C
      FDXCPY=(XHAT(1)*PAACPY+YHAT(1)*PABCPY+ZHAT(1)*PACCPY+PX1CY*PAACP+
     *PX1CP*PAACY+PY1CY*PABCP+PY1CP*PABCY+PZ1CY*PACCP+PZ1CP*PACCY+TEMP1*
     *PX1CPY+TEMP2*PY1CPY+TEMP3*PZ1CPY)*OMASS
      FDYCPY=(XHAT(2)*PAACPY+YHAT(2)*PABCPY+ZHAT(2)*PACCPY+PX2CY*PAACP+
     *PX2CP*PAACY+PY2CY*PABCP+PY2CP*PABCY+PZ2CY*PACCP+PZ2CP*PACCY+TEMP1*
     *PX2CPY+TEMP2*PY2CPY+TEMP3*PZ2CPY)*OMASS
      FDZCPY=(                YHAT(3)*PABCPY+ZHAT(3)*PACCPY
     *            +PY3CY*PABCP+PY3CP*PABCY+PZ3CY*PACCP+PZ3CP*PACCY
     *            +TEMP2*PY3CPY+TEMP3*PZ3CPY)*OMASS
   25 CONTINUE
      TEMP3=TY1*TANOP1+TY2*TANOP2
      TEMP4=TY1+TY2
      TEMP5=-PTP1M*TY1-PTP2M*TY2-TANOP1*PTY1P-TANOP2*PTY2M
      TEMP6=PTY1M+PTY2P
      PTZM=PTYM*TEMP4+TANOY*TEMP6
      PTXM=-TEMP3*PCRHOM+TEMP4*PSRHOM+CRHO*TEMP5+SRHO*TEMP6
      PTYM=TEMP3*PSRHOM+TEMP4*PCRHOM-SRHO*TEMP5+CRHO*TEMP6
      TEMP4=1./XM
      TEMP5=TEMP4*TEMP4
      FXM=-TEMP5*DX+TEMP4*(PTXM*XHAT(1)+PTYM*YHAT(1)+PTZM*ZHAT(1))
      FYM=-TEMP5*DY+TEMP4*(PTXM*XHAT(2)+PTYM*YHAT(2)+PTZM*ZHAT(2))
      FZM=-TEMP5*DZ+TEMP4*(                PTYM*YHAT(3)+PTZM*ZHAT(3))
    9 CONTINUE
C
      TEMP1=PTXX-S(ITHR)*(PQX*CN+Q*PCNX)
      TEMP2=PTXY-S(ITHR)*(PQY*CN+Q*PCNY)
      TEMP3=PTXZ-S(ITHR)*(PQZ*CN+Q*PCNZ)
      TEMP4=PTYX-S(ITHR)*(PQX*CA+Q*PCAX)
      TEMP5=PTYY-S(ITHR)*(PQY*CA+Q*PCAY)
      TEMP6=PTYZ-S(ITHR)*(PQZ*CA+Q*PCAZ)
      TEMP7=PTZX+S(ITHR)*(PQX*CS+Q*PCSX)
      TEMP8=PTZY+S(ITHR)*(PQY*CS+Q*PCSY)
      TEMP9=PTZZ+S(ITHR)*(PQZ*CS+Q*PCSZ)
C
C     TOTAL DERIVATIVE W.R.T POSITION
      FXX=OPASS*(XHAT(1)*TEMP1+YHAT(1)*TEMP4+ZHAT(1)*TEMP7)+GXX
      FXY=OPASS*(XHAT(1)*TEMP2+YHAT(1)*TEMP5+ZHAT(1)*TEMP8)+GYX
      FXZ=OPASS*(XHAT(1)*TEMP3+YHAT(1)*TEMP6+ZHAT(1)*TEMP9)+GZX
      FYX=OPASS*(XHAT(2)*TEMP1+YHAT(2)*TEMP4+ZHAT(2)*TEMP7)+GYX
      FYY=OPASS*(XHAT(2)*TEMP2+YHAT(2)*TEMP5+ZHAT(2)*TEMP8)+GYY
      FYZ=OPASS*(XHAT(2)*TEMP3+YHAT(2)*TEMP6+ZHAT(2)*TEMP9)+GZY
```

```
      FZX=DMASS*(YHAT(3)*TEMP4+ZHAT(3)*TEMP7)+GZX
      FZY=DMASS*(YHAT(3)*TEMP5+ZHAT(3)*TEMP8)+GZY
      FZZ=DMASS*(YHAT(3)*TEMP6+ZHAT(3)*TEMP9)+GZZ
C
      TEMP1=PTXW-S(ITHR)*(POW*CN+Q*PCNW)
      TEMP2=PTXU-S(ITHR)*(POU*CN+Q*PCNU)
      TEMP3=PTXV-S(ITHR)*(POV*CN+Q*PCNV)
      TEMP4=PTYW-S(ITHR)*(POW*CA+Q*PCAW)
      TEMP5=PTYU-S(ITHR)*(POU*CA+Q*PCAU)
      TEMP6=PTYV-S(ITHR)*(POV*CA+Q*PCAV)
      TEMP7=PTZW+S(ITHR)*(POW*CS+Q*PCSW)
      TEMP8=PTZU+S(ITHR)*(POU*CS+Q*PCSU)
      TEMP9=PTZV+S(ITHR)*(POV*CS+Q*PCSV)
C     TOTAL DERIVATIVE W.R.T VELOCITY
      FXW=DMASS*(XHAT(1)*TEMP1+YHAT(1)*TEMP4+ZHAT(1)*TEMP7)
      FXU=DMASS*(XHAT(1)*TEMP2+YHAT(1)*TEMP5+ZHAT(1)*TEMP8)
      FXV=DMASS*(XHAT(1)*TEMP3+YHAT(1)*TEMP6+ZHAT(1)*TEMP9)
      FYW=DMASS*(XHAT(2)*TEMP1+YHAT(2)*TEMP4+ZHAT(2)*TEMP7)
      FYU=DMASS*(XHAT(2)*TEMP2+YHAT(2)*TEMP5+ZHAT(2)*TEMP8)
      FYV=DMASS*(XHAT(2)*TEMP3+YHAT(2)*TEMP6+ZHAT(2)*TEMP9)
      FZW=DMASS*(YHAT(3)*TEMP4+ZHAT(3)*TEMP7)
      FZU=DMASS*(YHAT(3)*TEMP5+ZHAT(3)*TEMP8)
      FZV=DMASS*(YHAT(3)*TEMP6+ZHAT(3)*TEMP9)
C
      IF(MAHAT.EQ.0) RETURN
      HRATE1=XK1*(XK2*RHO)**.5*(VR/XK4)**3.D7
      HRATE2=XK5/(XK6*PAT(12)+(VR/XK3)**2)/XK7)
      HRATE=HRATE1*(1.-HRATE2)
      HRATE3=HRATE1*HRATE2*HRATE2/XK5
      HRATE4=HRATE*CRCH/RHO+HRATE3*XK6*PAT(13)
      HRATE5=HRATE*3.E-3/VR+HRATE3*2./XK3*VR/(XK3*XK3)
      FHRW=HRATE5*FVRW
      FHRU=HRATE5*FVRU
      FHRV=HRATE5*FVRV
      FHRX=HRATE5*FVRX+HRATE4*PHX
      FHRY=HRATE5*FVRY+HRATE4*PHY
      FHRZ=HRATE5*FVRZ+HRATE4*PHZ
      RETURN
      END
```

```
      SUBROUTINE BNENC
C
C      TERMINATES ADJOINT RUN AND ADJUSTS INFLUENCE COEFFICIENTS
C
      CHARACTER*60 HEAE
      CHARACTER*6 MISION
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TWINH,
     *CTZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WD(15),WJET(15),
     *HEAD,FRINT(15),STEF(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *CHROOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,GA,ALF,ALFY,CHIF,SCHIP,CCHIP,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVARS(13),DELXOW(15,3),DELXOR(7,4),DELXO(15,7),WZERO,ALT,
     *MACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMAX(15),BHMN,BSTEP(15),HMB,HMXB,IPOLY,XMIAO
      COMMON/AGEN3/ITER,IWO,IXR,JUMP,XAT,KCYTAB,KPAGE,KS1,KWTA(4),LINES,
     *NBGET(4),NENEET(4),NMAX,NOEVNT(5),NOWO(15),NVNT,NWVNT,IHEAD,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINDB,KRDERB,MISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/AUTO/NOB(40),SAVCP(40),XLAMB(40,10),CP2
      COMMON/BAKE/BISS(10,10),YLF(7,10)
      COMMON/BAKINT/BBANK(2),TBK,TBKO,YL(7,10),YLC(7,10),BSAVE(1350),
     *NBTRG1,TBV1,NBTRG2,TBV2,NBTRG3,TBV3,NBTRG4,TBV4,NBTRG5,TBV5,TEND,
     *CTB4,CU3,D23,D43,WISSD(10,10)
      COMMON/BGEN3/NOSS,TB,NENT,NACT,NOM1,NP(4)
      COMMON/BLAMB/YLBT(15,10),YLBW(15,10),GAPHI(10),FROD(2,10)
      COMMON/CONST/RAC,PI,RE,FLAT,GJ,H,DW,CMLE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,PTN,PTNG,FSFTM,PFN,PFNG,PSFFM
      COMMON/FMXX/FMXX,MSW,WG,SPOA
      COMMON/IPR/IFR,LFM,ILAST,KDT(15),FRK4,PRTCT
      COMMON/LBM/TIMLBM(20),THRLBM(30),WCTLBM(30),PLEM(2),MLBM1,MLBM2,
     *ALBM(2),BLBM(2),CLBM(2),VLBM(2)
      COMMON/MPS/TIMMPS(15),FTBL(15),THRCT(15),GLIM(15),TAUALT(5),TAUTBL
     *(5),MPS1,MPS2,TELIP,GLIMIT
      COMMON/NOMGR,NOMGR
      COMMON/OMSRCS/FCMS(15),FRCS(15),WCCMS(15),WCRCS(15),WOCUMP(15)
      COMMON/XTRA/XMD(15),CFR(15),F(15),CISP(15)
      COMMON/YLI/YLINT(7,10),PHITE(10),PSIREC(10),KCCPHI(10),PSTR(10)
      IF(NOPOR.EQ.2) GO TO 40
      IF(JTE.NE.1)CALL BSAVEG
C     ADD WG-LAMBDA-BETA TO THRUST LAMBDA BETA
      IF(NWVNT.EQ.0) GO TO 2
      DO 1 I=1,NWVNT
      K=NOWO(I)
      DO 1 J=1,NOSS
C     ADDS WO LAMBDA BETAS TO APPROPRIATE THRUST LAMBDA BETAS
    1 YLBT(K,J)=YLBT(K,J)+YLBW(I,J)
C *****
C ADD UP ALL LAMBS DOWNSTREAM
    2 KV=NVNT-1
    3 TEMP=1.
      DO 4 K=1,NOSS
    4 YLBT(KV,K)=YLBT(KV,K)+YLBT(KV+1,K)*TEMP
      KV=KV-1
      IF(KV.GT.0)GO TO 3
C*****  TINKER LAMBDA BETAS IF CONNECTED
      DO 5 I=1,NVNT
      IF(NOB(I  ).EQ.0)GO TO 5
      IF(IPR.NE.0.OR.PSWCH(NVNT-1).NE.2) GO TO 30
      INVT=NVNT-2
      TEMP=-TAUT(NVNT-1)*GLIM(INVT)/CISP(INVT)*TNE(1,1)*XMD(I)/(TNE(1,IN
     *VT)*XMD(INVT))
      TEMP1=WG*EXP(-GLIM(INVT)*(TIME(1,INVT+1)-TGLIM)/CISP(INVT))*EXP(-
     *VRCUT/(GZERO*CISP(INVT+1)))
      DO 29 K=1,NOSS
      IF(KCCPHI(K).NE.1) GO TO 28
      YLBT(I,K)=TEMP1*TEMP/TAUT(NVNT-1)
      GO TO 29
   28 YLBT(I,K)=YLBT(I,K)+TEMP*YLBT(NVNT-1,K)
   29 CONTINUE
```

```
1        71.        30 CONTINUE
1        72.           IF(IRTLS.EQ.0.OR.IPR.EQ.0) GO TO 33
1        73.           L=IRTLS
1        74.           L=IPR
1        75.           XMTI=TNE(1,I)*XMC(I)+WDOMS(I)+WDRCS(I)+WDDLMP(I)
1        76.           XMTJ=TNE(1,J)*XMC(J)+WDOMS(J)+WDRCS(J)+WDDLMP(J)
1        77.           XMTL=TNE(1,L)*XMC(L)+WDOMS(L)+WDRCS(L)+WDDLMP(L)
1        78.           FTJTI=(-(WDOMS(I)+WDDUMP(I))+WDOMS(L)/XMTL*XMTI)/((WDOMS(J)+
1        79.          *WDDUMP(J))-WDOMS(L)/XMTL*XMTJ)
1        80.           IF(IRIFLG.EQ.1) FTJTI=-XMTI/XMTJ
1        81.           PTGTI=1./XMTL*(-XMTI-XMTJ*PTJTI)
1        82.           TEMP=TNE(1,I)*CFR(I)+TNE(1,J)*CFR(J)*PTJTI+TNE(1,L)*CFR(L)*PTGTI
1        83.           PTLTI=PTGTI+FMXX*(TEMP)
1        84.           DO 32 K=1,NOSS
2        85.           IF(KDEPHI(K).NE.1) GO TO 31
2        86.           YLBT(I,K)=(1.-PRK4)/(CFR(L)/XMD(L)-PRK4*TEMP
2        87.           GO TO 32
2        88.        31 YLBT(I,K)=YLBT(I,K)+PTJTI*YLBT(J,K)+PTLTI*YLBT(L,K)
2        89.        32 CONTINUE
1        90.           GO TO 5
1        91.        33 CONTINUE
1        92.           IF(IRTLS.EQ.0.OR.IRTLS.GT.IRTLS) GO TO 15
1        93.           J=IRTLS
1        94.           L=IRTLS+1
1        95.           IF(MPSFLG(L).EQ.4) GO TO 36
1        96.           DO 34 K=L,NVAT
2        97.           IF(MPSFLG(K).EQ.4) GO TO 35
2        98.        34 CONTINUE
1        99.        35 L=K
1       100.        36 CONTINUE
1       101.           TEMP=(F(L)*TNE(1,L)+FOMS(L)+FRCS(L))/(GZERO*GLIM(L))*EXP(-GLIM(L)*
1       102.          *(TEND-TGLIM-TAUT(L+1))/CISP(L))*EXP(-GLIM(L+1)*TAUT(L+1)/CISP(L+1)
1       103.          *)
1       104.           XMTJ=TNE(1,J)*XMC(J)+WDOMS(J)+WDRCS(J)+WDDLMP(J)
1       105.           XMTL=TNE(1,L)*XMC(L)+WDOMS(L)+WDRCS(L)+WDDLMP(L)
1       106.           XMTI=TNE(1,I)*XMC(I)+WDOMS(I)+WDRCS(I)+WDDLMP(I)
1       107.           FTJTI=(-(WDOMS(I)+WDDUMP(I))+WDOMS(L)*XMTI/XMTL)/(WDOMS(J)+WDDLMP(
1       108.          *J)-WDOMS(L)*XMTJ/XMTL)
1       109.           IF(IRIFLG.EQ.1) FTJTI=-XMTI/XMTJ
1       110.        16 DO 14 K=1,NOSS
2       111.           IF(KDEPHI(K).NE.1) GO TO 13
2       112.           YLBT(I,K)=-TEMP*GLIM(L)/CISP(L)*(XMTI+XMTJ*FTJTI)/XMTL
2       113.           GO TO 14
2       114.        13 YLBT(I,K)=YLBT(I,K)+PTJTI*YLBT(J,K)
2       115.        14 CONTINUE
1       116.        15 CONTINUE
1       117.           IF(KDT(I   ).EQ.0) GO TO 11
1       118.           J=I+KDT(I)
1       119.           IF(MSW.EQ.0) GO TO 21
1       120.           TEMP=-TNE(1,I)*XMD(I)/(TNE(1,J)*XMC(J))+FMXX*(TNE(1,I)*CFR(I)-CFR(
1       121.          *J)/XMC(J)*TNE(1,I)*XMD(I))
1       122.           TEMP1=1.
1       123.           TEMP2=0.
1       124.           IF(MSWCH(J+1).NE.2) GO TO 26
1       125.           TEMP1=EXP(-VRGUT/(GZERO*CISP(J+1)))
1       126.           TEMP2=-TAUT(J+1)*GLIM(J)/CISP(J)*(TEMP*TNE(1,I)*XMD(I)/(TNE(1,J)*
1       127.          *XMC(J)))
1       128.        26 CONTINUE
1       129.           DO 20 K=1,NOSS
2       130.           IF(KDEPHI(K).NE.1) GO TO 25
2       131.           YLBT(I,K)=(1.-PRK4)/(CFR(J)/XMD(J)-PRK4*TEMP1*(1.-TEMP1)*(TNE(1,I
2       132.          *)*CFR(I)-CFR(J)/XMD(J)*TNE(1,I)*XMC(I))*TEMP1
2       133.           GO TO 20
2       134.        25 YLBT(I,K)=YLBT(I,K)+TEMP*YLBT(J,K)+TEMP2*YLBT(J+1,K)
2       135.        20 CONTINUE
1       136.           GO TO 5
1       137.        21 CONTINUE
1       138.           TEMP=CFR(I)/(CFR(J)*TNE(1,I)/TNE(1,J)
```

```
      IF(I.LT.ILAST.OR.IPR.EQ.0)GO TO 9
      TEMPR=(1.-CFR(J)*XMO(I)/CFR(I)*PRK4/XMC(J))/(1.-PRK4)
      TEMP=TEMP*TEMPR
    9 DO 10 K=1,NOSS
      TEMP1=TEMP*YLBT(J,K)
      IF(KCDPHI(K).NE.1)GO TO 10
      IF(I.EE.ILAST.AND.IPR.NE.0)TEMP1=TEMP1/TEMPR
   10 YLBT(I,K)=YLBT(I,K)-TEMP1
      GO TO 5
C     ADJUST UNBOUNDED LAMBDA BETAS FOR FR
   11 J=IPR
      IF(J.EQ.0)GO TO 5
      IF(MSK.EQ.0) GO TO 23
      TEMP=-TNE(1,I)*XMO(I)/(TNE(1,J)*XMC(J))-FMXX*(CFR(J)/XMO(J)*TNE(1,
     *I))*XMC(I))
      TEMP1=1.
      IF(MSKCH(J+1).EQ.2) TEMP1=EXP(-VRCLT/(GZERC*CISP(J+1)))
      DO 22 K=1,NOSS
      IF(KCDPHI(K).NE.1) GO TO 24
      YLBT(I,K)=(1.-PRK4)/(CFR(J)/XMO(J)-PRK4*TEMP1-(1.-TEMP1))*(-CFR(J)
     */XMO(J)*TNE(1,I)*XMO(I))*TEMP1
      GO TO 22
   24 YLBT(I,K)=YLBT(I,K)+TEMP*YLBT(J,K)
   22 CONTINUE
      GO TO 5
   23 CONTINUE
      TEMP=XMO(I)/XMO(J)*TNE(1,I)/TNE(1,J)
      DO 12 K=1,NOSS
      IF(KCDPHI(K).EQ.1)GO TO 12
      YLBT(I,K)=YLBT(I,K)+TEMP*PRK4*YLBT(J,K)/(1.-PRK4)
   12 CONTINUE
    5 CONTINUE
C     ****
C*** PUT ACTIVE YLBTS INTO XLAMB
    8 DO 7 I=1,NVNT
      DO 6 J=1,NOSS
    6 XLAMB(I,J)=YLBT(I,J)
    7 CONTINUE
      DO 8 I=1,7
      DO 8 J=1,NOSS
    8 YLF(I,J)=YL(I,J)
      CALL RTMRK
      RETURN
      END
```

```
      SUBROUTINE BSAVEE
C
C      SAVES IMPULSE RESPONCE FUNCTIONS AND PERFORMS SIMPSON INTEGRATION
C      ON PRODUCTS OF IMPULSE RESPONCE FUNCTIONS
C
      CHARACTER*60 HEAD
      CHARACTER*6 MISION
      COMMON/AGEN1/TIME(2,16),TALT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CTZ,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WG(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHRDOT,CHIRO,FA2,BG(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,CA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIF,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SU,SV,Q,
     *VIV(25),DVARS(13),DELXDW(15,3),DELXDR(7,4),DELXD(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *BYL,BEL,BHMAX(15),BHMN,BSTEP(15),HNMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITHR,IWD,IXR,JUMP,KAT,KCYTAB,KPAGE,KS1,KWTA(4),LINES,
     *NBGCT(4),NENDCT(4),NMAX,NOEVNT(5),NCWD(15),NVNT,NWVNT,IHEAD,
     *MINH,MPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINDB,KRDERB,MISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),MSWCH(15)
      COMMON/BAKE/WISS(10,10),YLF(7,10)
      COMMON/BAKINT/BBANK(2),TBK,TBKD,YL(7,10),YLD(7,10),BSAVE(1350),
     *NBTRG1,TBV1,NBTRG2,TBV2,NBTRG3,TBV3,NBTRG4,TBV4,NBTRG5,TBV5,TEND,
     *CTB4,DL3,D23,D43,WISSD(10,10)
      COMMON/BGEN3/NOSS,JTB,NENT,NACT,NOM1,NP(4)
      COMMON/BLAMB/YLBT(15,10),YLBW(15,10),GAPHI(10),PROD(2,10)
      COMMON/CALIF/PXX,PYX,PZX,PXY,PYY,PZY,PXZ,PYZ,PZZ,PXW,PYW,PZW,PXU,
     *PYU,PZU,PXV,PYV,PZV,PXM,PYM,PZM,PDXCP,PDYCP,PDZCP,PDXCY,PDYCY,PDZC
     *Y,PDXCPP,PDYCPP,PDZCPP,PDXCPY,PDYCPY,PDZCPY,PDXCYY,PDYCYY,PDZCYY
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,DC,CMLE,OMEGA,ALT1,ALT2,PSL,
     *GZERO,PTN,PTKG,PSFTM,PFN,PFKG,PSFFM
      COMMON/COVL/COVL(10)
      COMMON/FORINT/HBANK(2),T,TT,W,U,V,X,Y,Z,XM1,ZAF(18),DVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,MINC,KRDER,EU,EL,AYL,HMX,HMN,HNM
      COMMON/GNU/GNU(10)
      COMMON/IMP/GAPTB(90,10),GSPTB(90,10),AHTB(90),SITB(90),TS(3)
     *,DELTS(3),KJTB
      DIMENSION GSPHI(10)
C
C  30 FORMAT(3X,8E14.6)
      TBV4=TBV4+CTB4
      JTB=JTB-1
      JJ=KJTB+JTB
C     IMPULSE RESPONSE FUNCTIONS AND I-SY-SY DOTS
      IF(NOM1.NE.0)GO TO 4
      IF(KWTA(KCYTAB).EQ.2) GO TO 1
C     GET HERE ONLY ON NOMINAL WITH KWTA=3
      HV11=1./SQRT(PDXCPP*PDXCPP+PDYCPP*PDYCPP+PDZCPP*PDZCPP)
      HV12=0.
      HV22=1./SQRT(PDXCYY*PDXCYY+PDYCYY*PDYCYY+PDZCYY*PDZCYY)
      GO TO 8
    1 WAH=1./SQRT(PDXCPP*PDXCPP+PDYCPP*PDYCPP+PDZCPP*PDZCPP)
    2 CONTINUE
      DO 3 I=1,NOSS
      GAPHI(I)=YL(1,I)*PDXCP+YL(2,I)*PDYCP+YL(3,I)*PDZCP
      DO 3 J=1,I
    3 WISSD(I,J)=WAH*GAPHI(I)*GAPHI(J)
      GO TO 11
C     GET COV LAMBDAS
    4 NOSSM1=NOSS-1
      DO 5 I=1,7
      COVL(I)=YL(I,1)
      DO 5 J=1,NOSSM1
    5 COVL(I)=COVL(I)+GNU(J)*YL(I,J+1)
C     GOT COV LAMBDAS
```

```
      IF(KWTA(KCYTAB).EQ.3) GO TO 6
      WAH=1./(-XJEXT*(COVL(1)*PDXCPP+COVL(2)*PDYCPP+COVL(3)*PDZCPP))
      TEMP=1./SQRT(COVL(1)**2+COVL(2)**2)
      GO TO 2
   21 CONTINUE
      KAT=1
      WAH=TEMP
      GO TO 2
C     3-D MIN-H
C     CHECK HUU FOR POSITIVE DEFINITE
    6 A11=COVL(1)*PDXCPP+COVL(2)*PDYCPP+COVL(3)*PDZCPP
      A22=COVL(1)*PDXCYY+COVL(2)*PDYCYY+COVL(3)*PDZCYY
      A12=COVL(1)*PDXCPY+COVL(2)*PDYCPY+COVL(3)*PDZCPY
      A11=-A11*XJEXT
      A12=-A12*XJEXT
      A22=-A22*XJEXT
      DET=A11*A22-A12*A12
      TDET=COVL(1)*COVL(1)+COVL(2)*COVL(2)
      IF(A11.GT.0..AND.A22.GT.0.) GO TO 7
      DET=TDET
      A22=(DET+COVL(3)*COVL(3))**.5
      A11=DET/A22
      A12=0.
      KAT=1
C     HUU INVERSE
    7 HV11= A22/DET
      HV12=-A12/DET
      HV22= A11/DET
C
    8 DO 9 I=1,NOSS
      GAPHI(I)=YL(1,I)*PDXCP+YL(2,I)*PDYCP+YL(3,I)*PDZCP
      GSPHI(I)=YL(1,I)*PDXCY+YL(2,I)*PDYCY+YL(3,I)*PDZCY
      PROD(1,I)=HV11*GAPHI(I)+HV12*GSPHI(I)
      PROD(2,I)=HV12*GAPHI(I)+HV22*GSPHI(I)
      DO 9  J=1,I
    9 WISSD(I,J)=GAPHI(J)*PROD(1,I)+GSPHI(J)*PROD(2,I)
      DO 10 I=1,NOSS
      GAPTB(JJ,I)=PROD(1,I)
   10 GSPTB(JJ,I)=PROD(2,I)
      GO TO 13
   11 DO 12 I=1,NOSS
   12 GAPTB(JJ,I)=WAH*GAPHI(I)
   13 AHTB(JJ)=CHIP
      SITB(JJ)=CHIY
C  ** INTEGRATE ISYSY BY SIMPSONS RULE--
C  **                    A=H/3*(Y1+4*Y2+2*Y3+...+4*YK-1+YN)
      IF(JTB.LT.NP(KCYTAB))GO TO 15
C     GET HERE AT FIRST POINT  (MULT BY 1/3)
      LL=2
      DO 14 I=1,NOSS
      DO 14 J=1,I
   14 WISS(I,J)=WISS(I,J)+ D43  *WISSD(I,J)
      RETURN
   15 IF(LL.EQ.1)GO TO 17
C     GET HERE AT AN EVEN POINT  (MULT BY 4/3)
      LL=1
      DO 16 I=1,NOSS
      DO 16 J=1,I
   16 WISS(I,J)=WISS(I,J)+ D43  *WISSD(I,J)
      RETURN
   17 IF(JTB.EQ.1)GO TO 19
C     GET HERE AT AN ODD POINT  (MULT BY 2/3)
      LL=2
      DO 18 I=1,NOSS
      DO 18 J=1,I
   18 WISS(I,J)=WISS(I,J)+ D23  *WISSD(I,J)
      RETURN
C     GET HERE AT LAST POINT (JTB=1)  (MULT BY 1/3)
```

```
      15 DO 20 I=1,NOSS
         DO 20 J=1,I
      20 WISS(I,J)=WISS(I,J)+ DU3  *WISSD(I,J)
         NBIRG4=-1
         TS(KCYTAB)=T
         DELTS(KCYTAB)=DTE4
         RETURN
         END
```

```
      SUBROUTINE BTHREV
C
C     BACKWARD THRUST EVENT CONTROL ROUTINE
C
      CHARACTER*60 HEAE
      CHARACTER*6 PISICN,HEADER
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERO,TLIFT,TTILT,TMINH,
     *CT2,TC,TL,XMAUG,TNE(6,15),ENGDAT(8,15),S(15),WC(15),WJET(15),
     *HEAD,PRINT(15),STEP(15),HNOM(15),HMAX(15),PROP(6),TBEGR,TENDR,
     *CHROOT,CHIRO,FAZ,BC(3,2),CBAXIL,TCHIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN2/AA,SA,GA,ALF,ALFY,CHIF,SCHIP,CCHIF,CHIY,SCHIY,CCHIY,
     *CHIR,SCHIR,CCHIR,STH,CTH,STHL,CTHL,DPHIZ,RTHE,R,VR,XM,SW,SL,SV,Q,
     *VIV(25),DVARS(12),CELXOW(15,3),DELXOR(7,4),CELXO(15,7),WZERO,ALT,
     *XMACH,CDOT,FAA,FAN,A(3,3),CASE,CT2,UMF,A12W,A22W,A32W,XJEXT,BEU,
     *EYL,BEL,BHMAX(15),BHMN,BSTEP(15),HAMB,HMXB,TPOLY,XMIAD
      COMMON/AGEN3/ITHR,IWO,IXR,JUMP,KAT,KCYTAB,KPAGE,KSI,KWTA(4),LINES,
     *NBGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWO(15),NVNT,NWVNT,THEAD,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IFTLS,IPOLY,KINDB,KRDERB,PISION,
     *IRTFLE,NROLL,NCCAST,IFACT,NOBASE,LSTGE(15),PSWCH(15)
      COMMON/AGEN5/CHIBAS,CHPBAS,CHYBAS,CHISAV,TXX,TYY,TZZ,AMX,AMY,AMZ,
     *TMZ,THMFA,SICE,WMAG,AZW,FANC,FLATC,FOM,VISC
      COMMON/AVGGP/XGPA,YGPA,ZGPA
      COMMON/BANKINT/BBANK(2),TBK,TBKO,YL(7,10),YLC(7,10),BSAVE(1125),
     *NBTRG1,TBV1,NBTRG2,TBV2,NBTRG3,TBV3,NBTRG4,TBV4,NBTRG5,TBV5,TEND,
     *CT84,CU3,O23,C43,WISSD(10,10)
      COMMON/BGEN3/NOSS,NTB,NENT,NACT,NOPI,NF(4)
      COMMON/BLAMB/YLBT(15,10),YLBW(15,10),GAPHI(10),PROD(2,10)
      COMMON/BODY/ATBCNT(15,2),ATXCG(15,2),ATYCG(15,2),AXLEN(2),AXREF(2)
     *,AYREF(2),XPI(2),XYI(2),API(10,2),AYI(10,2)
      COMMON/BODYB/TBCNT(15),TXCG(15),TYCG(15),XLEN,XREF,YREF,MBS1,MBS2,
     *MBS3,MBS4,MBS5,MBS6,MBS7,MBS8,MBS9,MBS10
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,CJ,CMUE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,FTN,FTNG,FSFTM,PFN,PFNG,PSFFP
      COMMON/DELMBS/MBS21,MBS3C,MBS31
      COMMON/ORNGDT/ORNGDT(2)
      COMMON/ENPUT/PSIRST(6,4),HEADER(20),NVRST(4),KCCRES(6,4),LAST
      COMMON/FORINT/HBANK(2),T,TT,W,U,V,X,Y,Z,XMI,ZAP(18),OVAR(25),FSAVE
     *(625),NTRG1,TV1,NTRG2,TV2,NTRG3,TV3,NTRG4,TV4,NTRG5,TV5,NTRG6,TV6,
     *NTRG7,TV7,NTRG8,TV8,NTRG9,TV9,NTRG10,TV10,NTRG11,TV11,NTRG12,TV12,
     *NTRG13,TV13,KINC,KRDER,EU,EL,AYL,HMX,HMN,HAP
      COMMON/LBMMBL/MBLB1,MBLB2
      COMMON/IMP/GAPTE(90,10),GSPTB(90,10),APTB(90),SITB(90),TS(3)
     *,DELTS(3),XJTB
      COMMON/NOMCR/NOMCR
      COMMON/OLOK/CLOK,WINT
      COMMON/YLI/YLINT(7,10),PHITE(10),PSIREQ(10),KCCPHI(10),PSTR(10)
C     BACKWARDS THRUST EVENT ROUTINE
      IF(ITHR.EQ.JUMP.OR.ITHR.EQ.MINH) GO TO 6
C***  CHECK FOR END(BEGINNING) OF CHI TABLE BEFORE UPDATING THRUST
C     CHECK ON CHI TABLES
      IF(KCYTAB.EQ.1)GO TO 1
      IF(NENDCT(KCYTAB-1).EQ.0) KCYTAB=KCYTAB-1
      IF(NENDCT(KCYTAB-1).NE.ITHR  )GO TO 1
C *   HAVE ARRIVED AT THE END OF A CHI TABLE
      IF(JTB.NE.1)CALL BSAVEG
      KCYTAB=KCYTAB-1
      NBTRG4=1
      TBV4=TBK
      NTB=NF(KCYTAB)+1
      NJTB=(KCYTAB-1)*10
      SP=NP(KCYTAB)-1
      NN=NBGCT(KCYTAB)
      N=NENDCT(KCYTAB)
      DTB4=(TIME(1,N)-AMAX1(TO,TIME(1,NN)))/SP
      DU3=DTB4/3.
      D23=DU3*2.
      D43=DU3*4.
      1 CONTINUE
      IF(J.EQ.ITHR.AND.JTB.NE.1)CALL BSAVEG
```

```
C       END OF CHI TABLE CHECK
        ITHR=ITHR-1
        HNPB=BSTEP(ITHR)
        HMXB=2.*PRIN1(ITHR)
        IF(MPSFLG(ITHR).EQ.4.AND.TV9.GT.0.AND.TBK.LT.TRV5) NBTRG5=2
        IF(MPSFLG(ITHR).EQ.4.AND.TV9.GT.0.) NBTRG5=2
        XMAUG=XMAUG+WJET(ITHR)
        MMIN=0
        DO 205 I=1,15
        IF(NSYST(I).NE.1) GO TO 205
        IF(MMIN.EQ.0) MMIN=I
        MMAX=I
  205   CONTINUE
        IF(INT(TNE(1,ITHR)+.1).LT.MMAX-MMIN+1) MMAX=INT(TNE(1,ITHR)+.1)+
       *MMIN-1
        XGPA=0.
        YGPA=0.
        ZGPA=0.
        DO 206 I=MMIN,MMAX
        XGPA=XGPA+ENEDAT(3,I)
        YGPA=YGPA+ENEDAT(1,I)
  206   ZGPA=ZGPA+ABS(ENEDAT(2,I))
        XGPA=XGPA/TNE(1,ITHR)
        YGPA=YGPA/TNE(1,ITHR)
        ZGPA=ZGPA/TNE(1,ITHR)
        K=LSTGE(ITHR)
        IF(ITHR.NE.NVNT.AND.K.EQ.LSTGE(ITHR+1)) GO TO 100
        IF(K.EQ.1) WINT=WZERO
        PBS1=1
        PBS2=0
        PBS3=1
        PBS4=0
        PBS7=1
        PBS8=0
        PBS9=1
        PBS10=0
        PBLB1=1
        PBLB2=0
        PBS21=1
        PBS30=1
        PBS31=0
        XREF=AXREF(K)
        YREF=AYREF(K)
        XLEN=AXLEN(K)
        DO 99 I=1,15
        TBOWT(I)=ATBOWT(I,K)*.45359237
        TXCG(I)=ATXCG(I,K)
        TYCG(I)=ATYCG(I,K)
   99   CONTINUE
C       GET LAMBDA-BETAS
  100   DO 2 I=1,NCSS
        YLBT(ITHR,I)=0.
        IF(KCDPHI(I).GE.18.AND.KCDPHI(I).LE.20.AND.ITHR.EQ.NVNT) YLBT(ITHR
       *,I)=DFNGDT(1)
        DO 2 J=1,7
    2   YLBT(ITHR,I)=YLBT(ITHR,I)+DELXD(ITHR,J)*YL(J,I)
C       GET LAMBDA BETAS
        TBV2=TEND-TIME(1,ITHR)
   20   DO 5 K=1,4
        IF(ITHR.NE.NVRST(K)) GO TO 5
        NOMOR=1
        CALL FTMRK
        GO TO 10
    5   CONTINUE
   10   RETURN
C
    6   CALL BKEND
        RETURN
```

```
      ENTRY BWDEV
C
C      TRIGGER ROUTINE FOR BACKWARD WEIGHT DROP EVENT
C
      IWD=IBD-1
      XMAUG=XMAUG+BD(IBD)
C     GET LAMBDA BETAS
      DO 11 I=1,NOSS
      YLBW(IWD,I)=0.
      DO 11 J=1,3
   11 YLBW(IWD,I)=YLBW(IWD,I)+DELXOW(IWD,J)*YL(J,I)
      IF(IWD.EQ.1)GO TO 12
      TBV3=TEND-TIME(2,IBD-1)
      GO TO 13
   12 NBTRG3=-2
   13 RETURN
      ENTRY EGLIM
C
C      TRIGGER ROUTINE FOR ACCELERATION LIMIT IN BACKWARD INTEGRATION
C
      NBTRG5=-2
      RETURN
      END
```

```
      SUBROUTINE BOPTEL
       LOGICAL DEMAND
      CHARACTER*6 SRID,BLK1,AT
      CHARACTER*12 OFFICE,BLK2,DATE
      CHARACTER*48 TITLE,BLK3
      CHARACTER*18 DN
       COMMON/CCC/GRAPH,JOUT,DEMAND
      COMMON/BOPCOM/NCASE,OFFICE,TITLE,DATE,SRID(60),NOTAB,NERR
      COMMON/ALAT/ALAT,ALONGO,ALTLS,NTABLE
      NAMELIST/INPUT2/NCASE,OFFICE,TITLE,DATE,SRID,NOTAB
      DATA BLK1/6H      /,BLK2/12H            /
      DATA BLK3/48H
      DATA DN/18HBADD ELTTBL ;      /
      IF(NTABLE.EQ.0) RETURN
      DO 1 I=1,60
    1 SRID(I)=BLK1
      OFFICE=BLK2
      TITLE=BLK3
      DATE=BLK2
       NERR=10
      AT=6HENDPRT
      WRITE(9) AT,AT,(WORD,I=1,129),AT
      WRITE(JOUT,4)
    4 FORMAT(1H1)
      CALL CSFREQ(DN)
      READ(5,INPUT2)
      WRITE(JOUT,6) NCASE
      WRITE(JOUT,7) OFFICE
      WRITE(JOUT,8) TITLE
      WRITE(JOUT,9) DATE
      WRITE(JOUT,10) (SRID(I),SRID(I+1),I=1,60,2)
      WRITE(JOUT,11) NOTAB
C
    6 FORMAT(1X,6HNCASE=,I3)
    7 FORMAT(1X,7HOFFICE=,A12)
    8 FORMAT(1X,6HTITLE=,A48)
    9 FORMAT(1X,5HDATE=,A12)
   10 FORMAT(1X,5HSRID=/20(1X,2A6/))
   11 FORMAT(1X,6HNOTAB=,I3)
      RETURN
      END
      SUBROUTINE BIPTEL
C
C     OUTPUT TABLE GENERATOR
C
      PARAMETER MT=14,ML=10,ML2=ML+2,ML3=ML+3
      CHARACTER*6 NBK(4),NSTR2(2),NSTOP2,NFOR(ML3),BK2,SRID,PRNT,NAME,
     *NA,NB
      CHARACTER*12 NHEAD,NSTR1,NTBL(ML2),NSTOP1,BK1,OFFICE,DATE
      CHARACTER*12 IUNIT(2,20)
      CHARACTER*48 DEF(128)
      CHARACTER*48 TITLE
      DIMENSION NWORD(128),NNAME(ML,MT),VALUE(ML),PRNT(ML),NA(2),B(129),
     *NUNIT(128)
      DIMENSION UNIT(2,20)
      COMMON/NAME/NAME(128),WORD(129)
      COMMON/CCC/GRAPH,JOUT,DEMAND
      COMMON/BOPCOM/NCASE,OFFICE,TITLE,DATE,SRID(60),NOTAB,NERR
      COMMON/ALAT/ALAT,ALONGO,ALTLS,NTABLE
      DATA NWORD/
C       0    1    2    3    4    5    6    7    8    9
     *      2,   2,   3,   2,   0,   0,   0,   2,   3,
     1  2,   2,   3,   0,   2,   3,   2,   3,   3,   0,
     2  0,   0,   2,   2,   2,   3,   3,   3,   3,   3,
     3  3,   3,   3,   3,   3,   3,   3,   0,   0,   0,
     4  0,   0,   0,   0,   0,   0,   0,   0,   0,   2,
     5  2,   2,   2,   2,   2,   2,   2,   2,   2,   2,
```

```
      6  2,  0,  0,  0,  0,  0,  0,  0,  0,  0,
      7  0,  0,  0,  2,  2,  2,  2,  2,  2,  2,
      8  2,  2,  2,  2,  2,  2,  2,  2,  0,  0,
      9  2,  2,  2,  2,  2,  2,  2,  3,  3,  3,
      *  1,  1,  1,  2,  0,  0,  1,  1,  1,  1,
      1  1,  2,  2,  2,  2,  1,  3,  2,  0,  3,
      2  3,  2,  2,  2,  3,  3,  3,  2,  2/
      DATA NSTR1/12H(12X          /,NSTR2/6H(1X,2A,6H6      /
      DATA NSTOP1/12H)            /,NSTOP2/6H)      /
      DATA NHEAD/12H,5X,A6,1X     /
      DATA BK1/12H,12X          /,BK2/6H,12X   /
      DATA NBK/6H,F12.0,6H,F12.1,6H,F12.2,6H,F12.3/
      DATA NNAME/
     *   1,   7,  13,   8,   9,  10,  11,  17,  12,  34,
     *   1,   2,   3,   4,  14,  15,  16, 111, 112,  35,
     *   1,  19,  20,  21,  22,  23,  24, 124, 125, 126,
     *   1,  25,  26,  27,  31,  32,  33, 119, 120,   0,
     *   1, -37,  85, -38, -39, -40, -41,  -5,   0,   0,
     *   1, -37, -43,  49, -55, -61, -67,  73,  79,  85,
     *   1, -38, -44,  50, -56, -62, -68,  74,  80,  86,
     *   1, -39, -45,  51, -57, -63, -69,  75,  81,   0,
     *   1, -40, -46,  52, -58, -64, -70,  76,  82,   0,
     *   1, -41, -47,  53, -59, -65, -71,  77,  83,   0,
     *   1,  97,  98,  99,-103,-104,-105, 121, 122, 123,
     *   1,  90,  96,  91,  92,  93,  94,  95, -87, -88,
     *   1,-100,-106,-109,-102,-108,-110,-101,-107,-115,
     *   1,  28,  29,  30, 113, 114, 117,   0,   0,   0/
C
C          0       1       2       3       4       5       6       7       8       . 9
C
C    0 ------  TIME    VSUBE   GAME    AZE     THRST   XMLP    R       VSUBI   G . AMI
C
C    1   AZI    GCLAT   LONG    ALT     VSUBR   GAMR    AZR     GDLAT   NODE    Z . 8-X
C
C    2   X8-Y   Y8-Z   ZD8-W   XD8-U   YD8-V   CHIP    CHIY    CHIR    TIMP    LT . IMP
C
C    3  LNGMF   THERI   PSIRI   PHIRI   INCL    RANGE   RNGAN   THMPS   THSRM   TH . LP
C
C    4  THOMS   THRCS   TH      TVMPS   TVSRM   TVLBM   TVOMS   TVRCS   TVAC    IS . PMP
C
C    5  ISPSR   ISPLB   ISPCM   ISPRC   ISP     WOMPS   WDSRM   WDLBM   WDOMS   WD . RCS
C
C    6   WDOT   MPS F   SRM F   LBM F   OMS F   RCS F   FUEL    MPS C   SRM D   LB . M D
C
C    7  OMS C   RCS C   SPENT   DPMPS   DPSRM   DPLBM   DPOMS   DPRCS   DELP    DY . MPS
C
C    8  DYSRM   DYLBM   DYOMS   DYRCS   DELY    TAU     SRMAE   PDOME   SRMET   OR . BET
C
C    9  CH VL   TN L    GV L    OR L    BKP L   GIM L   ID VL   MACH    ALPHA   B . ETA
C
C   10   FAA    FAS     FAN     Q      QALPH   QBETA   FAB     FMB     FNB     D . RAG
C
C   11   LIFT   VWINC   AZW     HEAT    HRATE   PRESS   RHC     TEMP    REYNO   DE . LPC
C
C   12  DELYC   EMACH   INBD    OUTBD   AXACC   NRACC   LTACC   XCG     ZCG
C
      DATA DEF(  1)/48HINSTANTANEOUS TIME FROM SRB IGNITION                  /
      DATA DEF(  2)/48HEARTH-FIXED VELOCITY                                  /
      DATA DEF(  3)/48HEARTH-FIXED FLIGHT PATH ANGLE                         /
      DATA DEF(  4)/48HEARTH-FIXED AZIMUTH                                   /
      DATA DEF(  5)/48HTOTAL LOCAL THRUST                                    /
      DATA DEF(  6)/48HTOTAL VEHICLE WEIGHT                                  /
      DATA DEF(  7)/48HRADIUS FROM EARTH CENTER                              /
      DATA DEF(  8)/48HINERTIAL VELOCITY                                     /
      DATA DEF(  9)/48HINERTIAL FLIGHT PATH ANGLE                            /
      DATA DEF( 10)/48HINERTIAL AZIMUTH                                      /
      DATA DEF( 11)/48HGEOCENTRIC LATITUDE                                   /
      DATA DEF( 12)/48HLONGITUDE                                             /
```

```
DATA DEF( 13)/48HALTITUDE ABOVE REFERENCE ELLIPSOID                /
DATA DEF( 14)/48HWIND RELATIVE VELOCITY                            /
DATA DEF( 15)/48HWIND RELATIVE FLIGHT PATH ANGLE                   /
DATA DEF( 16)/48HWIND RELATIVE AZIMUTH                             /
DATA DEF( 17)/48HGEODETIC LATITUDE                                 /
DATA DEF( 18)/48HDESCENDING NODE-MEASURED CCW FROM LAUNCH SITE     /
DATA DEF( 19)/48HDOWNRANGE POSITION COMPONENT(INERTIAL)            /
DATA DEF( 20)/48HVERTICAL POSITION COMPONENT(INERTIAL)             /
DATA DEF( 21)/48HCROSSRANGE POSITION COMPONENT(INERTIAL)           /
DATA DEF( 22)/48HDOWNRANGE VELOCITY COMPONENT(INERTIAL)            /
DATA DEF( 23)/48HVERTICAL VELOCITY COMPONENT(INERTIAL)             /
DATA DEF( 24)/48HCROSSRANGE VELOCITY COMPONENT(INERTIAL)           /
DATA DEF( 25)/48HPITCH ATTITUDE (INERTIAL)                         /
DATA DEF( 26)/48HYAW ATTITUDE (INERTIAL)                           /
DATA DEF( 27)/48HROLL ATTITUDE (INERTIAL)                          /
DATA DEF( 28)/48HTIME TO VACUUM IMPACT POINT                       /
DATA DEF( 29)/48HLATITUDE OF VACUUM IMPACT POINT                   /
DATA DEF( 30)/48HLONGITUDE OF VACUUM IMPACT POINT                  /
DATA DEF( 31)/48HPITCH ANGLE IN BOOSTER REF SYSTEM
DATA DEF( 32)/48HYAW(HEADING) ANGLE IN BOOSTER REF SYSTEM
DATA DEF( 33)/48HROLL ANGLE IN BOOSTER REF SYSTEM
DATA DEF( 34)/48HORBITAL INCLINATION ANGLE
DATA DEF( 35)/48HRELATIVE SURFACE RANGE
DATA DEF( 36)/48HRANGE ANGLE
DATA DEF( 37)/48HMFS LOCAL THRUST
DATA DEF( 38)/48HSGM LOCAL THRUST
DATA DEF( 39)/48HLEM LOCAL THRUST
DATA DEF( 40)/48HOPS LOCAL THRUST
DATA DEF( 41)/48HRCS LOCAL THRUST
DATA DEF( 42)/48H
DATA DEF( 43)/48HMFS VACUUM THRUST
DATA DEF( 44)/48HSGM VACUUM THRUST
DATA DEF( 45)/48HLEM VACUUM THRUST
DATA DEF( 46)/48HOPS VACUUM THRUST
DATA DEF( 47)/48HRCS VACUUM THRUST
DATA DEF( 48)/48H
DATA DEF( 49)/48HMFS SPECIFIC IMPULSE
DATA DEF( 50)/48HSGM SPECIFIC IMPULSE
DATA DEF( 51)/48HLEM SPECIFIC IMPULSE
DATA DEF( 52)/48HOPS SPECIFIC IMPULSE
DATA DEF( 53)/48HRCS SPECIFIC IMPULSE
DATA DEF( 54)/48H
DATA DEF( 55)/48HMFS FLOWRATE
DATA DEF( 56)/48HSGM FLOWRATE
DATA DEF( 57)/48HLEM FLOWRATE
DATA DEF( 58)/48HOPS FLOWRATE
DATA DEF( 59)/48HRCS FLOWRATE
DATA DEF( 60)/48H
DATA DEF( 61)/48HMFS FUEL ONBOARD
DATA DEF( 62)/48HSGM FUEL ONBOARD
DATA DEF( 63)/48HLEM FUEL ONBOARD
DATA DEF( 64)/48HOPS FUEL ONBOARD
DATA DEF( 65)/48HRCS FUEL ONBOARD
DATA DEF( 66)/48H
DATA DEF( 67)/48HMFS FUEL SPENT
DATA DEF( 68)/48HSGM FUEL SPENT
DATA DEF( 69)/48HLEM FUEL SPENT
DATA DEF( 70)/48HOPS FUEL SPENT
DATA DEF( 71)/48HRCS FUEL SPENT
DATA DEF( 72)/48H
DATA DEF( 73)/48HMFS PITCH GIMBAL ANGLE
DATA DEF( 74)/48HSGM PITCH GIMBAL ANGLE
DATA DEF( 75)/48HLEM PITCH GIMBAL ANGLE
```

```
158.        DATA DEF( 76)/48HOPS PITCH GIMBAL ANGLE
159.        DATA DEF( 77)/48HRCS PITCH GIMBAL ANGLE
160.        DATA DEF( 78)/48H
161.        DATA DEF( 79)/48HMFS YAW GIMBAL ANGLE
162.        DATA DEF( 80)/48HSGM YAW GIMBAL ANGLE
163.        DATA DEF( 81)/48HLEM YAW GIMBAL ANGLE
164.        DATA DEF( 82)/48HOPS YAW GIMBAL ANGLE
165.        DATA DEF( 83)/48HRCS YAW GIMBAL ANGLE
166.        DATA DEF( 84)/48H
167.        DATA DEF( 85)/48HMFS THROTTLE FACTOR
168.        DATA DEF( 86)/48HSGM EXIT AREA
169.        DATA DEF( 87)/48HLCX BULKHEAD PRESSURE INDICATOR
170.        DATA DEF( 88)/48HSRM-ET SHEAR LOAD INDICATOR
171.        DATA DEF( 89)/48HORBITER-ET ATTACH LOAD INDICATOR
172.        DATA DEF( 90)/48HCHARACTERISTIC VELOCITY
173.        DATA DEF( 91)/48HTURNING LOSSES
174.        DATA DEF( 92)/48HGRAVITY LOSSES
175.        DATA DEF( 93)/48HDRAG LOSSES
176.        DATA DEF( 94)/48HBAKPRESSURE LOSSES
177.        DATA DEF( 95)/48HGIMBAL LOSSES
178.        DATA DEF( 96)/48HIDEAL VELOCITY
179.        DATA DEF( 97)/48HMACH NUMBER
180.        DATA DEF( 98)/48HPITCH ANGLE OF ATTACK
181.        DATA DEF( 99)/48HSIDESLIP ANGLE
182.        DATA DEF(100)/48HAXIAL AERODYNAMIC FORCE
183.        DATA DEF(101)/48HSIDE AERODYNAMIC FORCE
184.        DATA DEF(102)/48HNORMAL AERODYNAMIC FORCE
185.        DATA DEF(103)/48HDYNAMIC PRESSURE
186.        DATA DEF(104)/48HQ * ANGLE OF ATTACK
187.        DATA DEF(105)/48HQ * SIDESLIP ANGLE
188.        DATA DEF(106)/48HAXIAL BASE FORCE
189.        DATA DEF(107)/48HBASE PITCHING MOMENT
190.        DATA DEF(108)/48HNORMAL BASE FORCE
191.        DATA DEF(109)/48HAERODYNAMIC DRAG
192.        DATA DEF(110)/48HAERODYNAMIC LIFT
193.        DATA DEF(111)/48HWIND VELOCITY
194.        DATA DEF(112)/48HWIND AZIMUTH
195.        DATA DEF(113)/48HSTAGNATION HEATING INDICATOR
196.        DATA DEF(114)/48HSTAGNATION HEATING RATE
197.        DATA DEF(115)/48HATMOSPHERIC PRESSURE
198.        DATA DEF(116)/48HATMOSPHERIC DENSITY
199.        DATA DEF(117)/48HATMOSPHERIC TEMPERATURE
200.        DATA DEF(118)/48HREYNOLDS NUMBER
201.        DATA DEF(119)/48HCOMPOSITE PITCH GIMBAL ANGLE
202.        DATA DEF(120)/48HCOMPOSITE YAW GIMBAL ANGLE
203.        DATA DEF(121)/48HEARTH-RELATIVE MACH NUMBER
204.        DATA DEF(122)/48HINBOARD ELEVON ANGLE
205.        DATA DEF(123)/48HOUTBOARD ELEVON ANGLE
206.        DATA DEF(124)/48HTOTAL AXIAL ACCELERATION
207.        DATA DEF(125)/48HTOTAL NORMAL ACCELERATION
208.        DATA DEF(126)/48HTOTAL LATERAL ACCELERATION
209.        DATA DEF(127)/48HLONGITUDINAL CENTER OF GRAVITY LOCATION
210.        DATA DEF(128)/48HVERTICAL CENTER OF GRAVITY LOCATION
211.        DATA IUNIT/
212.       *12HSECONDS    ,12HSECONDS    ,12HMETERS     ,12HFEET       ,
213.       *12HMETERS/SEC ,12HFEET/SEC   ,12HKILOGRAMS  ,12HPOUNDS     ,
214.       *12HDEGREES    ,12HDEGREES    ,12HNEWTONS    ,12HPOUNDS     ,
215.       *12HKILOMETERS ,12HNAUT MILES ,12HKG/SEC     ,12HLBS/SEC    ,
216.       *12HNEW/M**2   ,12HLBS/FT**2  ,12HN-DEG/M**2 ,12HLB-DEG/FT**2,
217.       *12HBTU        ,12HBTU        ,12HBTU/SEC    ,12HBTU/SEC    ,
218.       *12HMETERS**2  ,12HFEET**2    ,12HKG/M**3    ,12HSLUGS/FT**3,
219.       *12HRANKIN     ,12HRANKIN     ,12HG'S        ,12HG'S        ,
220.       *12HCENTIMETERS,12HINCHES     ,12HNEWTON-METER,12HFOOT-POUND ,
221.       *12HUNITLESS   ,12HUNITLESS   ,12HN/M**2     ,12HLBS/IN**2  /
222.        DATA NUNIT/
223.   C     0     1     2     3     4     5     6     7     8     9
224.        *     1,    3,    5,    5,   -6,   -4,    2,    3,    5,
225.        1     5,    5,    5,    2,    3,    5,    5,    5,    5,    2,
226.        2     2,    2,    3,    3,    3,    5,    5,    5,    1,    5,
```

```
      3  -5,  -5,  -5,  -5,  -5,  -7,  -8,  -6,  -6,  -6,
      4  -6,  -6,  -6,  -6,  -6,  -6,  -6,  -6,  -6,   1,
      5   1,   1,   1,   1,   1,  -8,  -8,  -8,  -8,  -8,
      6  -8,  -4,  -4,  -4,  -4,  -4,  -4,  -4,  -4,  -4,
      7  -4,  -4,  -4,   5,   5,   5,   5,   5,   5,   5,
      8   5,   5,   5,   5,   5,  19,  13, -20,  -6,  -6,
      9   3,   3,   3,   1,   3,   3,   3,  19,   5,   5,
      *  -6,  -6,  -6,  -9, -10, -10,  -6, -18,  -6,  -6,
      1  -6,   3,   5,  11,  12,  -9,  14,  15,  19,   5,
      2   5,  19,   5,   5,  16,  16,  16, -17, -17/
      DATA UNIT/1.,1.,1.,3.28084,1.,3.28084,1.,.45359236,1.,1.,
     *1.,4.4482216 2,1.,1.852,1.,.45359236,1.,47.8802589,1.,47.8802589,
     *1.,1.,1.,1.,1.,10.76391,1.,.00194C3,1.,1.,1.,1.,1.,.393701,
     *1.,1.2558179,1.,1.,1.,6894.7572/
C
      IF(NTABLE.EQ.0) RETURN
      DO 10 K=1,MT
      NTBL(1)=NSTR1
      NFCR(1)=NSTR2(1)
      NFCR(2)=NSTR2(2)
      IMAX=0
      DO 2 I=1,ML
      JPAR=IABS(NNAME(I,K))
      IF(JPAR.EQ.0) GO TO 1
      NTBL(I+1)=NHEAD
      JFCR=NWORD(JPAR)+1
      NFCR(I+2)=NBK(JFCR)
      FRNT(I)=NAME(JPAR)
      IMAX=IMAX+1
      GO TO 2
    1 NTBL(I+1)=BK1
      NFCR(I+2)=BK2
    2 CONTINUE
      NTBL(ML2)=NSTOP1
      NFCR(ML3)=NSTOP2
      REWIND 9
      KT=0
      KW=0
      LINES=1
    3 READ(9) NA,B,NB
      LINES=LINES-1
      IF(NA(1).EQ.EHENDPRT) GO TO 10
      IF(NA(1).NE.EHTHRUST) GO TO 4
      KT=KT+1
      NA(1)=SRID(2*KT-1)
      NA(2)=SRID(2*KT)
      GO TO 5
    4 IF(NA(1).NE.EHWEIGHT) GO TO 5
      KW=KW+1
      NA(1)=SRID(2*KW+29)
      NA(2)=SRID(2*KW+30)
    5 IF(LINES.NE.0) GO TO 6
      WRITE(LO,100) OFFICE,DATE,TITLE,NCASE,K
      WRITE(LO,NTBL)(FRNT(I),I=1,IMAX)
      WRITE(LO,101)
      LINES=45
    6 DO 7 I=1,ML
      JPAR=IABS(NNAME(I,K))
      IF(JPAR.EQ.0) GO TO 7
      NUNIT=IABS(NUNIT(JPAR))
      JUNIT=1
      IF(NUNIT(JPAR)*NNAME(I,K).LT.0) JUNIT=2
      VALUE(I)=B(JPAR)*UNIT(JUNIT,KUNIT)
    7 CONTINUE
      WRITE(LO,NFCR) NA,(VALUE(I),I=1,IMAX)
      GO TO 3
   10 CONTINUE
      LINES=0
      DO 20 K=1,MT
```

```
      WRITE(JO,161)
      DO 20 I=1,ML
      IF(LINES.GT.0) GO TO 11
      WRITE(JO,200)
      LINES=40
   11 JPAR=IABS(NNAME(I,K))
      IF(JPAR.EQ.0) GO TO 20
      JUNIT=1
      IF(NNAME(I,K).LT.0) JUNIT=2
      KUNIT=IABS(NUNIT(JPAR))
      WRITE(JO,201) NAME(JPAR),K,IUNIT(JUNIT,KUNIT),DEF(JPAR)
      LINES=LINES-1
   20 CONTINUE
  100 FORMAT(1H1/5X,A12,6X,A12,6X,A48,32X,6H  CASE,I3//97X,9HTABLE NO.,
     *I3/)
  101 FORMAT(/)
  200 FORMAT(1H1//91X,'DEFINITION OF TABLE OUTPUT'//9X,'PARAMETER',7X,
     *'TABLE',12X,'UNITS',16X,'DEFINITION'//)
  201 FORMAT(10X,A6,10X,I3,10X,A12,10X,A48)
      RETURN
      END
```

```
      SUBROUTINE CAERC(AERO,XMACH,ALPHA,BETA,DINBD,DOUTBD)
      DIMENSION AERO(1),VEC(6)
      DIMENSION CAB(6,2),CNB(6,2),CMB(6,2),CYB(6,2),CXB(6,2),CLB(6,2),
     *C(6,2)
      COMMON/AERO/NTALPH(2),NTBETA(2),NTMACH,TALPH(11,2),TBETA(11,2),
     *TMACH(28),CAA(15,11,6),CNA(15,11,6),CMA(15,11,6),CYA(15,11,6),
     *CXA(15,11,6),CLA(15,11,6)
      CALL SEARCH(XMACH,TMACH,NTMACH,IM1,IM2)
      CALL SEARCH(ALPHA,TALPH(1,1),NTALP(1),IA1,IA2)
      CALL SEARCH(BETA,TBETA(1,2),NTBETA(2),IB1,IB2)
      CALL INTER(CAB,CAA,IA1,IA2,ALPHA,TALPH(1,1),IM1,IM2)
      CALL INTER(CNB,CNA,IA1,IA2,ALPHA,TALPH(1,1),IM1,IM2)
      CALL INTER(CMB,CMA,IA1,IA2,ALPHA,TALPH(1,1),IM1,IM2)
      CALL INTER(CYB,CYA,IB1,IB2,BETA,TBETA(1,2),IM1,IM2)
      CALL INTER(CXB,CXA,IB1,IB2,BETA,TBETA(1,2),IM1,IM2)
      CALL INTER(CLB,CLA,IB1,IB2,BETA,TBETA(1,2),IM1,IM2)
      VEC(1)=1.
      VEC(2)=DINBD
      VEC(3)=DINBD*DINBD
      VEC(4)=DOUTBD
      VEC(5)=DOUTBD*DOUTBD
      VEC(6)=DINBD*DOUTBD
      DO 4 I=1,6
      DO 4 J=1,2
    4 C(I,J)=0.
      DO 5 J=1,2
      DO 5 I=1,6
      C(1,J)=C(1,J)+VEC(I)*CAB(I,J)
      C(2,J)=C(2,J)+VEC(I)*CNB(I,J)
      C(3,J)=C(3,J)+VEC(I)*CMB(I,J)
      C(4,J)=C(4,J)+VEC(I)*CYB(I,J)
      C(5,J)=C(5,J)+VEC(I)*CXB(I,J)
    5 C(6,J)=C(6,J)+VEC(I)*CLB(I,J)
      IF(XMACH.LT.TMACH(1)) GO TO 7
      IF(XMACH.GT.TMACH(NTMACH)) GO TO 9
      DX=0.
      IF(IM1.NE.IM2) DX=(XMACH-TMACH(IM1))/(TMACH(IM2)-TMACH(IM1))
      DO 6 I=1,6
    6 AERO(I)=C(I,1)+(C(I,2)-C(I,1))*DX
      RETURN
    7 DO 8 I=1,6
    8 AERO(I)=C(I,1)
      RETURN
    9 DO 10 I=1,6
   10 AERO(I)=C(I,2)
      RETURN
C
      SUBROUTINE SEARCH(ANG,TAB,N,IA,IB)
      DIMENSION TAB(1)
      TEMP=ANG-TAB(1)
      DO 1 I=2,N
      TEMP1=ANG-TAB(I)
      IF(TEMP*TEMP1.LE.0.) GO TO 2
      TEMP=TEMP1
    1 CONTINUE
```

```
      IF(ANG.LE.TABA(1)) IA=1
      IF(ANG.GE.TABA(N)) IA=N
      IB=IA
      RETURN
    2 IA=I-1
      IB=I
      RETURN
C
      SUBROUTINE INTER(A,C,IA1,IA2,ANG,TABA,IC1,IC2)
      DIMENSION A(6,2),C(15,11,6),TABA(1)
      DO 2 I=1,6
      K=0
      DO 2 J=IC1,IC2
      K=K+1
      IF(IC1.EQ.IC2.AND.IC1.NE.1) K=K+1
      TEMP=C(J,IA1,I)
      TEMP1=C(J,IA2,I)
      IF(IA1.EQ.IA2) GO TO 1
      TEMP=TEMP+(TEMP1-TEMP)/(TABA(IA2)-TABA(IA1))*(ANG-TABA(IA1))
    1 A(I,K)=TEMP
    2 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE CHIPCL(N1,N2,T,Y1,A1,Y2,A2)
C
C      EVALUATES PITCH AND YAW ATTITUDE POLYNOMIALS
C            N1      : ORDER OF POLYNOMIAL IN A1
C            N2      : ORDER OF POLYNOMIAL IN A2
C            T       : INDEPENDENT VARIABLE IN POLYNOMIALS
C            Y1      : DEPENDENT VARIABLE EVALUATED BY A1 POLYNOMIAL AT T
C            A1      : COEFFICIENTS OF POLYNOMIAL 1
C            Y2      : DEPENDENT VARIABLE EVALUATED BY A2 POLYNOMIAL AT T
C            A2      : COEFFICIENTS OF POLYNOMIAL 2
C
      DIMENSION A1(10),A2(10)
      P=N1+1
      Y1=A1(P)
      DO 1 I=1,N1
      J=P-I
    1 Y1=A1(J)+Y1*T
      P=N2+1
      Y2=A2(P)
      IF(N2.EQ.0) RETURN
      DO 2 I=1,N2
      J=P-I
    2 Y2=A2(J)+Y2*T
      RETURN
      END
```

```
      SUBROUTINE CCPLCF
      CHARACTER*12 FILE
      CHARACTER*36 FILDES
      CHARACTER*6 NA(2),NB
C     COPY FROM LOGICAL UNIT 8 TO LIST-DIRECTED FILE
C     UNIT 8 IS FROM STARGO PROGRAM OUTPUT
      COMMON /GCC/ERAFP,LOUT,DEMAND
      DIMENSION WORD(129)
      DATA FILE/12H              /
      DATA FILDES/36HMASTRE CASE
    1 IF(ICCP.EQ.1)CALL PAGIT
      CALL GOUT(' ',1)
      CALL GOUT('SELECT ONE OF THE FOLLOWING',30)
      CALL GOUT('0-TERMINATE',12)
      CALL GOUT('1-WRITE ON A LIST-DIRECTED FILE',31)
      CALL RDATAI(1,ICCP,$1)
      IF(ICCP.GT.1.OR.ICCP.LT.0)GO TO 1
      IF(ICCP .EQ. 0) RETURN
      RE=INC 9
      CONTRL=0.
      NEXT=1
      NWR=129
      CALL GOUT(' ',1)
      CALL LOSWI(FILE,FILDES,CONTRL,NWR,$1)
    2 READ(9,ERR=10,END=10) NA,WORD,NB
      IF(NA(1).EQ.&HENDPRT) GO TO 10
      CALL LOWRIT(FILE,2,CONTRL,NEXT,WORD,$27)
      GO TO 2
   10 CALL LGCLOS(FILE,NEXT)
      CALL GOUT('THE FOLLOWING LIST-DIRECTED FILE WAS SUCCESSFULLY COMPL
     *ETED',59)
      CALL GOUT(FILE,12)
      CALL TWAIT(9000)
      GO TO 1
   27 CALL GOUT('-',1)
      CALL GOUT('* ERROR WRITING LDF *',21)
      CALL TWAIT(8000)
      GO TO 1
      END
```

```
      SUBROUTINE DBANK(A,NR,IM,IN)
      COMPILER(ARGCHK=OFF)
      COMMON/CCC/GRAPH,JG,DEMAND
      COMMON/SBLOCK/BLOCK(1500)
      DIMENSION A(1)
C
      DEFINE FILE 8(65,1500,U,NDUM)
C
      IF(IM.GT.0) READ(8'NR,ERR=2) BLOCK
      IP=IABS(IM)
      J=0
      DO 1 I=IP,IN
      J=J+1
    1 A(J)=BLOCK(I)
      RETURN
C
    2 WRITE(IWO,3) NR
    3 FORMAT(2X,'ERROR IN READING RECORD NO.',I6)
      STOP
      END
```

```
      SUBROUTINE DESOLV (MXEQ,NACT,DR1,DR2,ENDS,NP,KERR,HNOM, HBK,BLK,
C
C        *         *         *         *         *         *         *
C     DESOLV INTEGRATION PACKAGE PROGRAMMED BY JOAN SHEEHAN,          *
C     NOVEMBER 1966, UNDER SUPPORT WORK ORDER WITH MSFC, HUNTSVILLE,  *
C     ALABAMA.                                                        *
C        *         *         *         *         *         *         *
C
     X                    ITYP, LL, EUP,ELO,DTMX,DTMN,YLC,
     X                    NCTRG,
     X                    NTR1, RT1, VAR1, ZV1,  NTR2, RT2, VAR2, ZV2,
     X                    NTR3, RT3, VAR3, ZV3,  NTR4, RT4, VAR4, ZV4,
     X                    NTR5, RT5, VAR5, ZV5,  NTR6, RT6, VAR6, ZV6,
     X                    NTR7, RT7, VAR7, ZV7,  NTR8, RT8, VAR8, ZV8,
     X                    NTR9, RT9, VAR9, ZV9,  NTR10,RT10,VAR10,ZV10,
     X                    NTR11,RT11,VAR11,ZV11 )
      COMPILER(ARGEMK+EFF)
C        WHERE MXEQ =  THE TOTAL NUMBER OF POSSIBLE FIRST ORDER DIFFER-
C                      ENTIAL EQUATIONS TO BE SOLVED
C              NACT =  ACTUAL NUMBER OF DIFFERENTIAL  EQUATIONS TO SOLVE
C              DR1  =  ENTRY POINT TO ROUTINE THAT CARRIES OUT THE
C                      PORTION OF THE DERIVATIVE EVALUATION INVOLVING
C                      THE INDEPENDENT VARIABLE ONLY
C                      DR1 IS OPTIONAL, PUT A ZERO IF NOT DESIRED.
C              DR2  =  ENTRY POINT TO THE ROUTINE THAT CARRIES OUT THE
C                      REMAINING CALCULATIONS NECESSARY TO COMPUTE ALL
C                      NACT DERIVATIVES AND STORES THEM IN HBK
C              ENDS =  ENTRY POINT TO ROUTINE ENTERED AT THE END OF A
C                      FULL INTEGRATION STEP - OPTIONAL.
C                      PUT A ZERO IF NO END-OF-STEP ROUTINE IS DESIRED.
C              NP   =  ONE OF THE FOLLOWING CODE NUMBERS DEPENDING ON
C                      THE OPTIONS DESIRED
C                      ////////////////////////////
C                      VALUES OF NP / DR1 / ENDS /
C                      ////////////////////////////
C                                1  / NO  / YES  /
C                                2  / YES / YES  /
C                                5  / NO  / NO   /
C                                6  / YES / NO   /
C                      ////////////////////////////
C
C              KERR =  ERROR INDICATOR. NORMAL RETURN WHEN KERR=0
C                                       ERROR RETURN WHEN KERR=1
C              HNOM =  NOMINAL STEP SIZE
C              HBK  =  THE FIRST WORD OF A BLOCK OF STORAGE WITH HBK(1)
C                      STARTING IN AN EVEN LOCATION AND -
C                      HBK(1) = CURRENT STEP SIZE
C                      HBK(2) = NOT USED
C                      HBK(3) = INDEPENDENT VARIABLE IN SINGLE
C                               PRECISION (MUST BE PRESET)
C                      HBK(4) = NOT USED
C                      HBK(5) TO HBK(MXEQ+4) = DEPENDENT VARIABLES
C                      HBK(MXEQ+5) TO HBK(2*MXEQ+4) = DERIVATIVES
C              BLK  =  THE FIRST WORD OF A BLOCK OF STORAGE THAT MUST BE
C                      EQUIVALENT TO HBK(2*MXEQ+5) AND MUST HAVE TWO
C                      DIMENSIONS (MXEQ,LL) WHERE LL IS DEFINED BELOW
C              ITYP =  INDICATOR FOR THE TYPE OF INTEGRATION TO BE USED
C                      ITYP=1, VARIABLE STEP ADAMS-MOULTON
C                      ITYP=2, FIXED STEP RUNGE-KUTTA
C                      ITYP=3, FIXED STEP ADAMS PREDICTOR ONLY
C                      (AN EXTERNAL CHANGE OF HNOM CAUSES A RESTART )
C              LL   =  THE VARIABLE DIMENSION USED IN BLK(MXEQ,LL)
C                      LL=5 WHEN ITYP = 2
C                      LL = 3*(M+2) +4 WHEN ITYP = 1 OR 3
C                      M IS THE ORDER OF THE DIFFERENCES USED
C              EUP  =  UPPER LIMIT FOR STEP SIZE CONTROL
C              ELO  =  LOWER LIMIT FOR STEP SIZE CONTROL
C              DTMX =  MAXIMUM STEP SIZE ALLOWED
```

```
69.   C                 DTMA =  MINIMUM STEP SIZE ALLOWED
70.   C                 YLO  =  THE SMALLEST VALUE OF A DEPENDENT VARIABLE THAT
71.   C                         WILL AFFECT THE AUTOMATIC STEP SIZE LOGIC
72.   C                 NOTRG=  THE NUMBER OF TRIGGERS TO FOLLOW
73.   C                         THE MAXIMUM NO OF TRIGGERS IS SET AT 11
74.   C                 NTR  =  PLUS OR MINUS AN INTEGER 1 OR 2
75.   C                         PLUS SIGN MEANS TRIGGER IS ON
76.   C                         MINUS SIGN MEANS TRIGGER IS OFF
77.   C                         INTEGER 1 INDICATES NO DISCONTINUITY AFTER
78.   C                                   EXECUTING THE TRIGGER ROUTINE
79.   C                         INTEGER 2 INDICATES A DISCONTINUITY AND FORCES
80.   C                                   A RESTART
81.   C                 RT-  =  ENTRY POINT OF THE TRIGGER INTERRUPT ROUTINE
82.   C                 VAR- =  NAME OF THE VARIABLE BEING TESTED
83.   C                 ZV-  =  THE VALUE OF VAR AT WHICH THE TRIGGER INTERRUPT
84.   C                         ROUTINE IS TO BE EXECUTED
85.         COMMON/CCC/GRAPH,JC,DEMAND
86.         COMMON/MOD1/ACO(10),BCO(10)
87.         COMMON /DES1/
88.        *   KR(1),  NPX,  NEQ, NOLD,    M,  MP1,  MF2,   NN, KIND,  KON,
89.        *     IFL, NTRG, NIVT, NDVT,  NTR,NSTART,NBCIF,  NNT,  NTT,IDL(11),
90.        *    ICL2, JJSS,.IOCO,J2000,J4000,J7000, IRST, ECST,   HC,  SMA,
91.        *   ERROR,REDCT,STEPNG,
92.        *   EU(1),   EL,  HPX,  HMN,   YL,    A, ENF1,    D,    B,  SKP,
93.        *      SK,   HC, CELU, DUSC, HOLD,  VAR,   ZV,   HM,   HF,    R,
94.        *       Q, CELT,  HPU,  FMU,CK(10),CO(10),XL(11), RJ(11), WJ(11)
95.         COMMON /DES2/     T,  TSV,  TGO, TMIN, TRIP,   TL,   TR
96.         DOUBLE PRECISION  T,  TSV,  TGO, TMIN, TRIP,   TL,   TR
97.   C
98.         LOGICAL ECST, SMA, HD, IRST, ERROR
99.         LOGICAL  STEPNG
100.  C
101.        DIMENSION HBK(1), BLK(MXEQ,LL)
102.  C
103.  C
104.        DATA EIG/1.0E30/
105.  C
106.        GO TO 400
107.    402 CONTINUE
108.        PXEQ=PXEQ
109.        NACT=NACT
110.        NP=NP
111.        NERR=NERR
112.        PNOM=PNOM
113.        IDEI=2*MXEQ
114.  C
115.        DO 152 ID=1,IDD
116.    152 HBK(ID)=HBK(ID)
117.        BLK(1,1)=BLK(1,1)
118.        ITYP=ITYP
119.        LL=LL
120.        EUP=ELP
121.        ELO=ELO
122.        DTMX=DTMX
123.        DTMN=DTMN
124.        YLO=YLO
125.        NOTRG=NOTRG
126.        NTR1=NTR1
127.        NTR2=NTR2
128.        NTR3=NTR3
129.        NTR4=NTR4
130.        NTR5=NTR5
131.        NTR6=NTR6
132.        NTR7=NTR7
133.        NTR8=NTR8
134.        NTR9=NTR9
135.        NTR10=NTR10
136.        NTR11=NTR11
137.        VAR1=VAR1
```

```
      VAR2=VAR2
      VAR3=VAR3
      VAR4=VAR4
      VAR5=VAR5
      VAR6=VAR6
      VAR7=VAR7
      VAR8=VAR8
      VAR9=VAR9
      VAR10=VAR10
      VAR11=VAR11
      ZV1=ZV1
      ZV2=ZV2
      ZV3=ZV3
      ZV4=ZV4
      ZV5=ZV5
      ZV6=ZV6
      ZV7=ZV7
      ZV8=ZV8
      ZV9=ZV9
      ZV10=ZV10
      ZV11=ZV11
  400 CONTINUE
C
      ACC(1)=1.
      ACC(2)=.5
      ACC(3)=5./12.
      ACC(4)=3./8.
      ACC(5)=251./720.
      ACC(6)=95./288.
      ACC(7)=19087./60480.
      ACC(8)=36799./120960.
      ACC(9)=1070017./3628800.
      ACC(10)=.287675448
      ECC(1)=1.
      ECC(2)=-.5
      ECC(3)=-1./12.
      ECC(4)=-1./24.
      ECC(5)=-19./720.
      ECC(6)=-3./160.
      ECC(7)=-863./60480.
      ECC(8)=-275./24192.
      ECC(9)=-33953./3628800.
      ECC(10)=-.78925542E-2
      NMX = MXEQ
      KINO = ITYP
      NTRG = NOTRG
      EU(1)=EUP
      EL = ELO
      HMX = DTMX
      HMN = DTMN
      YL = YLO
C
   50 MXNTRG = 11
      IF (NTRG .LE. MXNTRG)  GO TO  52
   51 FORMAT (// 52H MAXIMUM NUMBER OF TRIGGERS THIS DECK WILL HANDLE IS
     *,I4/5X,14HRECOMPILE F4IR)
      WRITE (10,51)  MXNTRG
      KERR = 1
      RETURN
   52 CONTINUE
C INITIALIZATION
C     CALCULATE DELTA L SCALE FACTOR = 0.596E-7
      TEMP=24*24
      DUSC = 1.0/TEMP
      STEPNG = .FALSE.
      IF (KERR .EQ. 100)  STEPNG = .TRUE.
      KERR = 0
      KR(1)=0
```

```
      100 CONTINUE
          DO  55  J=1,MXNTRG
   55     IDL(J) = 0
          HBK(1) = HNOM
          HOLD = HNOM
C
C
C POINT FOR RESTART
  102     FORMAT (///5H NF =I3, 21H - UNACCEPTABLE VALUE)
  103     FORMAT (///7H ITYP =, I3, 11HNOT ALLOWED)
          IF (NF .EQ. 1  .OR.  NP .EQ. 2  .OR.  NP .EQ. 5  .OR.  NP .EQ. 6)
         >      GO TO  105
          WRITE (JO,102) NF
          KERR = 1
          RETURN
  105     IF (KIND .EQ. 1  .OR.  KIND .EQ. 2) GO TO  106
          IF (KIND .EQ. 3)  GO TO  106
          WRITE (JO,103)  KIND
          KERR = 1
          RETURN
C
  106     CONTINUE
          HOLD = HACT
          HEC = HACT
          EOST = .FALSE.
          HN = NF-4
          IF (NF .GT. 2)  GO TO  108
          EOST = .TRUE.
          HN = NP
  108     IRST = .FALSE.
C
C    INITIALIZE FOR RUNGE-KUTTA OR ADAMS-MOULTON START
C            KIND = 1 FOR ADAMS-MOULTON
C            KIND = 2 FOR RUNGE-KUTTA
C            KIND = 3 FOR FIXED STEP ADAMS OPEN FORMULA
C
          GO TO (110, 112, 110), KIND
C         VARIABLE STEP ADAMS-MOULTON
  110     TRIP = HBK(3)
          NSTART = 0
          M=(LL-10)/3
          MP1 = M+1
          MP2 = M+2
C         COUNTER FOR SWITCHING TO A-M AFTER M+2 R-K STEPS
          NBCIF = 0
          GO TO  115
C
C         FIXED STEP SIZE RUNGE-KUTTA THROUGHOUT
  112     TRIP = BIG
          NSTART = 100
          M = 0
          MP1=0
          MP2=0
  115     HON = 2
C            HON IS CONTROL FOR R-K LOGIC OR A-M IN MAIN LOOP
C
          ERROR = .FALSE.
          MMP2 = MP1+1
          CALL DPIR  (HBK(3),HBK(5),HBK(NMX+5),HBK(2*NMX+5),HBK(4*NMX+5),
         *             BLK(1,5),BLK(1,MP2+5),BLK(1,2*MP2+5), MXEQ,MMP2,
         >            HBK, HNOM, NACT,  DR1, DR2, ENDS,
         >                   NTR1, RT1, VAR1, ZV1,  NTR2, RT2, VAR2, ZV2,
         >                   NTR3, RT3, VAR3, ZV3,  NTR4, RT4, VAR4, ZV4,
         >                   NTR5, RT5, VAR5, ZV5,  NTR6, RT6, VAR6, ZV6,
         >                   NTR7, RT7, VAR7, ZV7,  NTR8, RT8, VAR8, ZV8,
         >                   NTR9, RT9, VAR9, ZV9,  NTR10,RT10,VAR10,ZV10,
         >                   NTR11,RT11,VAR11,ZV11 )
C
C                                  TO RESTART
```

```
      IF (IFST) GO TO  100
C                                   SUBROUTINE INTERNAL ERROR
      IF (ERROR)  KERR = 1
C
      IF(KR(1).EQ.100) KERR=100
C     (KR IS RESET TO ZERO IN DESOLV INITIALIZATION)
 9999 RETURN
      END
```

```
      SUBROUTINE DFIR (TC,Y,DY,PY,YN,DELZ,DELY,DELX, NNN,LOR,
     *                 H, HNOM, N,  DER1, DER2, ECS,
     *                 NTR1, RT1, VAR1, ZV1,  NTR2, RT2, VAR2, ZV2,
     *                 NTR3, RT3, VAR3, ZV3,  NTR4, RT4, VAR4, ZV4,
     *                 NTR5, RT5, VAR5, ZV5,  NTR6, RT6, VAR6, ZV6,
     *                 NTR7, RT7, VAR7, ZV7,  NTR8, RT8, VAR8, ZV8,
     *                 NTR9, RT9, VAR9, ZV9,  NTR10,RT10,VAR10,ZV10,
     *                 NTR11,RT11,VAR11,ZV11 )
      COMPILER(ARGCHK=OFF)
      COMMON/CCC/GRAPH,JC,DEMAND
      DIMENSION TC(2),Y(NNN),DY(NNN),PY(NNN),YN(NNN)
      DOUBLE PRECISION  PY, YN
      DIMENSION DELX(NNN,LOR),  DELY(NNN,LOR), DELZ(NNN,LOR)
C
      COMMON/MOD1/ACO(10),BCO(10)
C
      COMMON /DES1/
     *   KR(1),   NPX,   NEQ, NOLD,    M,   MP1,   MF2,   NN, KIND,  KON,
     *     IFL, NTFG, NINT, NDVT,  NTR,NSTART,NBCIF,  NNT,  NTT,IDL(11),
     *    ICL2, JJSS,.1000,J2000,J4000,J7000, IRST, ECST,   HD,  SMA,
     *   ERROR,REDCT,STEPNG,
     *   EU(1),   EL,  HPX,  HMN,   YL,    A, ENF1,    D,    B,  SKP,
     *      SK,   HG, DELU, DUSC, HOLD,  VAR,   ZV,   HM,   HP,    R,
     *       Q, DELT,  HPU,  FMU,CK(10),CQ(10),XL(11), RJ(11), BJ(11)
      COMMON /DES2/     T,  TSV,  TGO, TMIN, TRIF,   TL,   TR
      DOUBLE PRECISION  T,  TSV,  TGO, TMIN, TRIF,   TL,   TR
      LOGICAL EOST, SMA, HD, IRST, ERROR
      LOGICAL  STEPNG
      LOGICAL  ODD, REDCT
      DATA BIG/1.0E30/
C
      GO TO  118
 3525 TC(1)=TC(1)
      Y(1)=Y(1)
      DY(1)=DY(1)
      PY(1)=PY(1)
      YN(1)=YN(1)
      DELZ(1,1)=DELZ(1,1)
      DELY(1,1)=DELY(1,1)
      DELX(1,1)=DELX(1,1)
      NNN=NNN
      LOR=LOR
      H=H
      HNOM=HNOM
      N=N
      NTR1=NTR1
      NTR2=NTR2
      NTR3=NTR3
      NTR4=NTR4
      NTR5=NTR5
      NTR6=NTR6
      NTR7=NTR7
      NTR8=NTR8
      NTR9=NTR9
      NTR10=NTR10
      NTR11=NTR11
      VAR1=VAR1
      VAR2=VAR2
      VAR3=VAR3
      VAR4=VAR4
      VAR5=VAR5
      VAR6=VAR6
      VAR7=VAR7
      VAR8=VAR8
      VAR9=VAR9
      VAR10=VAR10
      VAR11=VAR11
      ZV1=ZV1
```

```
      ZV2=ZV2
      ZV3=ZV3
      ZV4=ZV4
      ZV5=ZV5
      ZV6=ZV6
      ZV7=ZV7
      ZV8=ZV8
      ZV9=ZV9
      ZV10=ZV10
      ZV11=ZV11
  100 IRST = .TRUE.
      RETURN
C
C     SET INITIAL VALUES OF STORAGE AREAS TO ZERO
  118 CONTINUE
      DO  120  I=1,NMX
      FY(I)=Y(I)
  120 YN(I) = 0.
      HO = .TRUE.
      SMA = .TRUE.
      GO TO (121, 126, 121),  KIND
C     FOR A-M MUST ZERO OUT DIFFERENCE TABLES
  121 DO  125  I=1,NMX
      DO  125  J=1,MP2
      DELX(I,J) = 0.
      DELY(I,J) = 0.
  125 DELZ(I,J) = 0.
      A = -ACO(MP1)/BCO(MP2)
  126 HC = H
C
C     TO SET UP INDICATOR BLOCK FOR TRIGGERS THAT ARE DEPENDENT
C           (IDL = 1)  OR INDEPENDENT (IDL = 0),  ALSO CALCULATES THE
C           NUMBER OF EACH
      NIVT = 0
      IF (NTRG)  131, 131, 128
  128 TEMP=TC(1)
      TC(1)=-BIG
      JJSS = 2
      DO  130  J=1,NTRG
      IDL(J) = 1
      JNT=J
      GO TO  8000
   58 VAR=VAR
      IF(VAR.NE.-BIG) GO TO 60
C     THIS INDICATES A T-STOP, SINCE T(1)
C     WAS SET EQUAL TO -BIG AT THE BEGINNING OF THIS LOOP
      NIVT = NIVT+1
      IDL(J)=0
   60 CONTINUE
      DLJ(J) = 0.
      RJ(J) = 0.
  130 WJ(J) = 0.
      TC(1) = TEMP
  131 T = TC(1)
   62 NOVT = NTRG - NIVT
C
C     CALC DERIVATIVES AT THE CURRENT TIME
      GO TO  (172, 171), NN
  171 CALL  DER1
  172 CALL  DER2
C     GO TO ROUTINE TO GET THE CURRENT TMIN AND TRIGGER ROUTINE
C      AND NUMBER THAT GOES WITH IT.  ROUTINE IS AT 1000
      ICOD = 1
      GO TO  1000
  205 CONTINUE
C
C     GO TO END-OF-STEP CALCULATIONS, IF ANY
      IF (ECST)  CALL EOS
      IF(KR(1).NE.0) GO TO 4910
```

```
      DELU = T
      IF (HC .GT. DELU)  DELU=HC
      DELU = DUSC*DELU
C
C*************** MAIN LOOP ****************************************
C
  300 CONTINUE
      INDIS = 0
C     TEST FOR CALCULATION ERROR
  310 IF (TMIN + DELU .LT. T) GO TO 9000
      TS = DABS(TMIN-T)
  315 IF(TS.GT.DELU)GO TO 350
C
C   EXECUTE INDEPENDENT VARIABLE TRIGGER ROUTINE
      NNT = NTT
      JCOD = 1
      GO TO  4000
C     TEST FOR A DISCONTINUITY CAUSING A RESTART
  320 IF (INDIS)  100, 325, 100
C     AFTER EXECUTING TRIGGER, RE-CALC. NEW TMIN
  325 JCOD = 2
      GO TO  1000
  330 GO TO  310
C
C     CALCULATE L'S FOR EACH Y-STOP
  350 IF (NCVT .EQ. 0)  GO TO  380
      IL = 1
      JJSS = 3
      DO  375  J=1,NTFE
      JNT=J
      IF (IDL(J))  360, 375, 360
  360 GO TO  8000
  365 NTR=NTR
      ZV=ZV
      VAR=VAR
      IF(NTR) 375,370,370
  370 XL(J) = VAR - ZV
      IF(XL(J)) 375,371,375
  371 IDL(J)=3
      GO TO 3500
  375 CONTINUE
C
C   MOVE ONE STEP FORWARD,  METHOD DEPENDS ON KON
  380 GO TO ( 5000 , 6000 ) , KON
C
C   ADAMS - MOULTON CONTROL LOGIC
C
 5000 CONTINUE
      IF (DABS(T-TEQ) .GT. DELU)  GO TO  5400
C     MOVE FORWARD ONE HC STEP
      JSSG = 1
      GO TO  5500
C     ON RETURN FROM ADAMS TEST FOR STEP-SIZE FLAG
 5020 GO TO (5100, 5200, 5300),  IFL
C
C     NORMAL RETURN,  LEAVE STEP SIZE ALONE
 5100 CONTINUE
      IF (STEPNG) GO TO  5150
 5105 IGO = T
      DELU = T
      IF (HC .GT. DELU)  DELU=HC
      DELU = DUSC*DELU
      GO TO  5400
C
C     AT END OF FIRST HC AFTER SWITCHING TO ADAMS MODE, TEST HC TO SEE
C     IF INITIAL STEP WAS TOO BIG.  IF SO RETURN CONTROL TO CALLING PROG
 5110 FORMAT (//22H  INITIAL STEP SIZE OF, 1PE12.5,
     *          17H  HAS BEEN CUT TO, E12.5)
 5150 STEPNG = .FALSE.
```

```
      IF (HC .GE. HSWICH)  GO TO  5105
C     STEP HAS BEEN CUT DOWN
      WRITE (JO,5110)  HSWICH, HC
      HR(1)=100
      GO TO  9999
C
C     STEP SIZE IS TOO BIG.  HALVE STEP SIZE AND RESET CONDITIONS AT
C          BEGINNING OF STEP.  DIFFERENCE TABLES MUST BE REGENERATED FOR .  HC/2.
 5200 HC = HC/2.0
      H = HC
C
C     USING DIFFERENCES IN DELX GET (M+1) DERIVATIVES BACK FROM THIS
C     TIME AND STORE IN DELZ.   (DELX STORAGE DESTROYED)
      DO 5220  I=1,NEQ
      DO 5210  J=1,MP1
      MM= MP2-J
      DELZ(I,J)=DELX(I,1)
      DO 5210  K=1,MM
 5210 DELX(I,K) = DELX(I,K) - DELX(I,K+1)
 5220 DELZ(I,MP2) = DELX(I,1)
C
C     USING THESE DERIVATIVES IN DELZ AND DERIVATIVES AT STEPS HALF WAY
C     BETWEEN THESE (WHICH ARE CALCULATED USING BACKWARD DIFFERENCE
C     INTERPOLATION), SET UP A TABLE OF (M+1) BACK DERIVATIVES
C     USING HC/2 AND STORE IN DELY
      FM=M
      FMP1=MP1
C          CALC.  (M+1) FACTORIAL
      FFMP1=1.0
      DO 5230  J=2,MP1
      FJ=J
 5230 FFMP1=FFMP1*FJ
C
      CDD= .FALSE.
      DO 5270  NM=1,MP2
      IF  (CDD)  GO TO 5250
C
C      USE Y DOT FROM DELZ       (N=0,2,4,---)
      NMM=NM/2 + 1
      DO 5240 I=1,NEQ
 5240 DELY (I,NM) = DELZ(I,NMM)
      GO TO 5270
C
C        CALC.  Y DOT HALF WAY BETWEEN  (N=1,3,5,  ---)
 5250 FN=NM-1
      FMU=-FN/2.0
      PROD=1.
      DO 5255  J=1,MP1
      FJ=J
 5255 PROD = PROD*(FMU+FJ)
C          CALC.  A(0)
      DO 5265  I=1,NEQ
      AK = PROD/FFMP1
      SUM = AK*DELZ(I,1)
C
      DO 5260 K=2,MP2
      FK=K-1
      AK=-AK*(FMU+FK)/(FMU+FK+1.) * (FMP1-FK)/(FK+1.)
 5260 SUM = SUM+ AK*DELZ(I,K)
 5265 DELY (I,NM) = SUM
C
 5270 CDD = .NOT. CDD
C
C     NOW HAVE (M+1) BACK DERIVATIVES AT HC/2 SAVED IN DELY
C     RE-GENERATE NEW BACKWARD DIFFERENCE TABLES IN DELX
      K1=MP2
 5275 DO 5280  I=1,NEQ
      TP1 = DELY(I,K1)
```

```
      DO 5280  K=1,MP2
      TP2=TP1
      TP1=TP1-DELX(I,K)
 5280 DELX(I,K) = TP2
      K1=K1-1
      IF(K1)   5285, 5285, 5275
C
C     DO TO INTEGRATE ONE ADAMS STEP AGAIN WITH HALF THE STEP SIZE.
C           SETTING JSSCD=1 FORCES RETURN TO STATEMENT 5020
 5285 JSSCD=1
      GO TO 5500
C
C     STEP SIZE IS SMALL.  PREPARE TO DOUBLE HC.
 5300 HD = .FALSE.
      MMP1 = MP1
      TRIP = T + FMP1*HC
      ASTART = -1
C
C
C     MOVE DELX TO DEL2 AND USING THE DIFFERENCES FROM HERE GET THE
C     R+1 BACK DERIVATIVES AND SAVE THEM IN DELY
C
C           DEL(T-HC)**R = DEL(T)**R - DEL(T)**(R+1)
      DO 5310  I=1, NEQ
      DO  5310  K=1, MP2
 5310 DEL2(I,K) = DELX(I,K)
C
C
      DO 5350  I=1, NEQ
      DO  5330  J=1, MP1
      MM = MP2 - J
      DELY(I,J) = DEL2(I,1)
      DO  5330  K=1,MM
 5330 DEL2(I,K) = DEL2(I,K)-DEL2(I,K+1)
 5350 DELY (I,MP2) = DEL2(I,1)
C
      GO TO 5100
C
C
 5400 IF (TMIN .LE. (TGO-DELU)) GO TO 5430
C     ALL T-STOPS IN THIS STEP HAVE BEEN EXECUTED
C     SET CONDITIONS BACK TO TGO
      T = TGO
      TC(1)=T
      DO 5410  I=1,NEQ
      Y(I) = PY(I)
 5410 DY(I) = DELX(I,1)
C
      IF (REDOT)  GO TO 5425
      GO TO (5416,5415), NN
 5415 CALL DER1
 5416 CALL DER2
      REDOT = .TRUE.
 5425 IF (ECST)  CALL EOS
      IF(KR(1).NE.0) GO TO 4910
      GO TO  6100
C
C     INTERPOLATE FOR CONDITIONS AT TMIN
 5430 T = TMIN
      TC(1)=T
      JTCOD = 1
      GO TO  7000
 5450 GO TO  6100
C
C  USE RUNGE-KUTTA THIS STEP
C       CALCULATE STEP SIZE TO USE
```

```
      6000 CONTINUE
           DT=H
           IF(TS.LT.DT) DT=TS
           .2000 = 1
           GO TO  2000
      601C IGC = 1
           DELU = T
           IF (HC .GT. DELU)  DELU=HC
           DELU = DUSC*DELU
C
C          GO TO END-OF-STEP
           IF (ECST)  CALL EOS
           IF(KR(1).NE.(1 GO TO 4910
C          TEST FOR ANY Y-STOPS IN THIS STEP
C
C                    FLAG  ROUTINE
      61CC CONTINUE
           IDL2 = 0
           IF (NEVT) 6165,3600,6105
      6105 .JSS = 4
C          FLAG TRIGGERS THAT HAVE HAD A CHANGE OF SIGN IN THIS STEP BY
C                SETTING THE INDICATOR IDL = 2
           DO 615C  J=1,NTRG
           .NT=J
           IF (IDL(J)) 611C,6150,6118
      611C GO TO  8000
      6120 NTR=NTR
           ZV=ZV
           VAR=VAR
           IF(NTR) 6150,6130,6130
      613C TEMP = VAR - ZV
           IF (SIGN(1.,TEMP) .NE. SIGN(1.,XL.(J)) )  GO TO 6140
           IDL(J) = 1
           GO TO  615C
      614C IDL(J) = 2
           IDL2 = IDL2+1
           BJ(J) = TEMP
           BJ(J) = TEMP
      615C CONTINUE
C
           IF (IDL2 .EQ. 0)  GO TO  300
C
C          SEARCH ROUTINE
C             SEARCH AND FLAG FIRST  Y-STOP IN THIS TIME STEP
      300C CONTINUE
C
           HM = BIG
           HP = BIG
           .JSS = 5
           DO 3100  J=1,NTRG
           IF (IDL(J) .LT. 2)  GO TO 3100
           .NT=J
           GO TO  8000
      3110 NTR=NTR
           ZV=ZV
           VAR=VAR
           IF(NTR) 3100,3115,3115
      3115 BJ(J) = VAR - ZV
           IF (SIGN(1.,BJ(J)) .NE. SIGN(1.,XL.(J)))  GO TO  3120
           IF (SIGN(1.,BJ(J)) .NE. SIGN(1.,R.(J)))  GO TO  3140
C          UNFLAG DEPENDENT VARIABLE J
           IDL(J) = 1
           GO TO  3100
C
      3120 R = W.(J)*(TL-T)/(BJ(J)-XL.(J))
           IF (R .LT. HM)  HM = R
           GO TO  3150
```

```
C
 3140 R = WJ(J)*(TR-T)/(WJ(J)-RJ(J))
      IF (R .LT. HP)  HP = R
 3150 G = R
      IDL(J) = 2
      IF (ABS(G) .LT. DELU)  IDL(J) = 3
 3100 CONTINUE
C
 3180 IF (HP .EQ. BIG)  GO TO 3220
      DO 3200  J=1,NTRE
 3200 RJ(J) = WJ(J)
      TR = T
      DELT = HM
      GO TO 3260
 3220 DO 3240  J=1,NTRE
 3240 XL(J) = WJ(J)
      TL = T
      DELT = HP
 3260 IF (ABS(DELT) .LT. DELU)  GO TO  3500
C
C     NOT CONVERGED
      GO TO  (3300, 3400), KON
C     FOR ADAMS-MOULTON, INTERPOLATE FOR Y'S AT T + DELT
C          GET DERIVATIVES AND EOS CONDITIONS
 3300 T = T + DELT
      IC(1)=1
      LZCOD = 2
      GO TO  7000
 3350 GO TO  3000
C
C     FOR RUNGE-KUTTA, INTEGRATE TO T + DELT
 3400 DT = DELT
      LZCOD = 2
      GO TO  2000
 3450 IF (ECST)  CALL EOS
      GO TO  3000
C
C     SEARCH HAS CONVERGED WITHIN ACCEPTABLE TOLERANCE ON ONE OF
C          DEPENDENT VARIABLE TRIGGERS
 3500 CONTINUE
      INDIS = 0
      DO 3540  J=1,NTRE
      LJSS = 6
      LNT=J
      GO TO 8000
 3520 NTR=NTR
      IF(NTR) 3540,3520,3520
 3530 IF (IDL(J) .NE. 3)  GO TO  3540
C     GO TO EXECUTE TRIGGER
      NNT = J
      IDL(J) = 1
      LZCOD = 2
      GO TO  4000
 3540 CONTINUE
C
C     CALC. NEW TMIN AND NTT
      LIOOD = 5
      GO TO  1000
 3600 CONTINUE
C
      GO TO (3680, 3700), KON
C     TEST FOR DISCONTINUITIES, CAUSES A RESTART IN ADAMS
 3680 IF (INDIS .EQ. 0)  GO TO  350
      GO TO  100
C
C     TEST FOR DISCONTINUITIES, CAUSES DERIVATIVES TO BE RE-CALCULATED
C          IN KUTTA
 3700 TGO = T
```

```
      477.          IF (INDIS .EQ. 0)  GO TO  300
      478.    C
      479.          GO TO (3752, 3751), NN
      480.     3751 CALL DER1
      481.     3752 CALL DER2
      482.          IF (ECST)  CALL EOS
      483.          IF(KR(1).NE.0) GO TO 4910
      484.          GO TO  300
      485.    C
      486.    C
      487.    C*****************************************************************. ********
      488.    C  ROUTINE TO CALCULATE CURRENT TMIN FROM LIST OF T-STOPS
      489.    C     ALSO CHECKS FOR ANY CHANGE IN ARGUMENTS THAT WILL CAUSE A RESTART
      490.    C           STARTS AT 1000 AND RETURNS BY A CALCULATED GO TO USING J1000
      491.    C
      492.     1000 CONTINUE
      493.          TMIN = TRIP
      494.          NTT = NSTART
      495.          IF (NINT)  1060, 1060, 1010
      496.     1010 NJSS = 1
      497.          DO 1050 J=1,NTRE
1     498.          NNT=J
1     499.          IF (ICL(J)) 1050, 1020, 1050
1     500.     1020 GO TO 8000
1     501.     1030 NTR=NTR
1     502.          ZV=ZV
1     503.          IF(NTR) 1050,1040,1040
1     504.     1040 IF(ZV.GT.TMIN)GO TO 1050
1     505.          IF(ABS(ZV-TMIN).LE.DELU)GO TO 1050
1     506.          TMIN = ZV
1     507.          NTT = J
1     508.     1050 CONTINUE
      509.     1060 CONTINUE
      510.    C     CHECK POSSIBILITY OF A RESTART DUE TO A CHANGE IN ARGUMENTS
      511.          IF (NOLD - N)  100, 1075, 1070
      512.     1070 NOLD = N
      513.          NEQ = N
      514.     1075 IF (HNOM .EQ. HOLD)  GO TO 1080
      515.    C     THE NOMINAL STEP SIZE HAS BEEN CHANGED EXTERNALLY
      516.    C           MUST RESTART.
      517.          H = HNOM
      518.          HOLD = HNOM
      519.          GO TO  100
      520.    C
      521.     1080 GO TO(205,330,5505,5570,3600,350),-1000
      522.    C
      523.    C*****************************************************************. ********
      524.    C  FOURTH ORDER RUNGE-KUTTA ROUTINE
      525.    C     TO MOVE ONE STEP FROM T TO T + DT (INITIALIZE DT BEFORE CALLING)
      526.    C           USES YN TO SAVE PY(T)
      527.    C                 PY AND Y CALC. TOGETHER TO T+DT
      528.    C                 DELZ(I,1) TO SAVE SUM OF K'S
      529.    C           RETURN IS BY -2000
      530.    C
      531.     2000 TSV = T
      532.          DOT = DT/2.
      533.    C     SAVE INITIAL Y'S IN YN
      534.          DO 2002  I=1,NEQ
1     535.     2002 YN(I) = PY(I)
1     536.    C
1     537.    C           OUTER LOOP
      538.          DO 2060  L=1,4
      539.    C           INNER LOOP   FOR ALL N EQUATIONS
1     540.          DO 2050  I=1,NEQ
2     541.          SK = DT*DY(I)
2     542.          GO TO (2010, 2020, 2030, 2040), L
2     543.    C     L=1, SK=K1
2     544.     2010 DELZ(I,1) = SK
```

```
2     2015 PY(I) = YN(I) + SK/2.
2          Y(I) = PY(I)
2          GO TO 2050
2    C     L=2, SK=K2
2     2020 DELZ(I,1) = DELZ(I,1)+2.*SK
2          GO TO  2015
2    C     L=3, SK=K3
2     2030 DELZ(I,1) = DELZ(I,1) + 2.*SK
2          PY(I) = YN(I) + SK
2          Y(I) = PY(I)
2          GO TO  2050
2    C     L=4, SK=K4
2     2040 PY(I) = YN(I) + (DELZ(I,1) + SK)/6.
2          Y(I) = PY(I)
2     2050 CONTINUE
2    C
1          GO TO  ( 2051, 2057, 2053, 2057),  L
1     2051 T = TSV + PDT
1          GO TO  2055
1     2053 T = TSV + DT
1     2055 TC(1)=T
1          GO TO  ( 2057, 2056),  NN
1     2056 CALL DER1
1     2057 CALL DER2
1     2060 CONTINUE
1    C
1    C          RETURN
           GO TO ( 6010,  3450),  J2000
     C*****************************************************************
     C*****************************************************************
     C
     C     CALLS TO EXECUTE TRIGGER ROUTINES.  NUMBER IS SET IN NNT
     C                NNT MAY EQUAL ZERO WHEN GETTING THE FIRST M+2
     C                     DIFFERENCES BY RUNGE-KUTTA BEFORE SWITCHING TO ADAMS
     C
     C                NNT MAY EQUAL -1 WHEN ENOUGH DIFFERENCES ARE SAVED
     C                     TO DOUBLE THE STEP SIZE
     C          RETURN IS BY J4000
     C
      4000 CONTINUE
           IF (NNT)  4500,  4400,  4005
     C
      4005 IF (NNT .GT. NTFE)  GO TO  4300
           GO TO  (4010,4020,4030,4040,4050,4060,4070,4080,4090,4100,
          X          4110),  NNT
     C
      4010 CALL RT1
           GO TO  4900
      4020 CALL RT2
           GO TO  4900
      4030 CALL RT3
           GO TO  4900
      4040 CALL RT4
           GO TO  4900
      4050 CALL RT5
           GO TO  4900
      4060 CALL RT6
           GO TO  4900
      4070 CALL RT7
           GO TO  4900
      4080 CALL RT8
           GO TO  4900
      4090 CALL RT9
           GO TO  4900
      4100 CALL RT10
           GO TO  4900
      4110 CALL RT11
           GO TO  4900
```

```
      C
 431C FORMAT (///31H TRYING TO CALL TRIGGER ROUTINE, I3,
     1            21H WHICH IS NOT DEFINED/)
 43CC WRITE (JO,4310)  NRT
      GO TO  990E
      C
      C
      C      ROUTINE TO EXECUTE WHEN TAKING THE FIRST M+2 RUNGE-KUTTA STEPS
      C           AND GENERATING THE NECESSARY DIFFERENCE TABLES.
      C           WHEN NBDIF = M+2 THIS TRIGGER IS SHUT OFF AND CONTROL IS
      C           TRANSFERED TO ADAMS CONTROL.
 44CC CONTINUE
      TRIP = T + HC
      C      UPDATE DIFFERENCES TABLES
      CO 441C  I=1,NEC
      TP1 = DY(I)
      CO 441C  K=1,MP2
      TP2 = TP1
      TP1 = TP1 - DELX(I,K)
 441C DELX(I,K) = TP2
      C
      NBDIF = NBDIF+1
      IF (NBDIF .LT. MF2)  GO TO 4995
      C
      C      WHEN ENOUGH STEPS HAVE BEEN TAKEN, SWITCH TO ADAMS-MOULTON
      NON = 1
      HSWITCH = HC
      TRIP = BIG
      NSTART = 100
      GO TO  325
      C
      C      ROUTINE TO DOUBLE THE STEP SIZE AFTER ENOUGH DIFFERENCES
      C           HAVE BEEN SAVED
      C      TIME OF THIS TRIGGER IS (M+1)*HC AFTER THE TIME WHEN THE DOUBLING
      C      PROCEDURE WAS SET OFF
 45CC CONTINUE
      HC = 2.*HC
      H = HC
      HO = .TRUE.
      TRIP = BIG
      NSTART = 100
      C
      C      USING DIFFERENCES IN DELX GET (M+1) DERIVATIVES BACK FROM THIS
      C      TIME AND STORE THESE IN DELZ
      CO 453C I=1,NEQ
      CO 452C J=1,MP1
      MM = MP2-J
      DELZ(I,J) = DELX(I,1)
      CO 452C K=1,MM
 452C DELX(I,K) = DELX(I,K) - DELX(I,K+1)
 453C DELZ(I,MP2) = DELX(I,1)
      C
      C      FROM DERIVATIVES IN DELY AND DELZ, SET DELY= BACK DERIVATIVES
      C      WITH 2*HC TIME INTERVAL BETWEEN
      ASSIGN 4542 TO ISW
      I1 = MF2+1
      I2 = MF2+2
 4535 I1 = I1-1
      IF (I1) 4560, 4560, 4536
 4536 I2 = I2-2
      IF (I2) 4538, 4538, 4540
      C      SWITCH TO DELZ TABLES
 4538 ASSIGN 4544 TO ISW
      I2 = I2 + MP1
      C
      C      IN SAME LOOP, RE-CALC. NEW DIFFERENCE TABLES IN DELX BASED ON NEW
      C      TIME STEP
 454C CO 455C   I=1,NEC
      GO TO ISW,  (4542,4544)
 4542 DELY(I,I1) = DELY(I,I2)
```

```
      GO TO 4545
 4544 DELY (I,I1) = DELZ(I,I2)
 4545 TP1 = DELY(I,I1)
      DO 4550 K=1,MP2
      TP2 = TP1
      TP1 = TP1-DELX(I,K)
 4550 DELX(I,K) = TP2
C
      GO TO 4535
C
 4560 GO TO 4995
C
C
C     AFTER EXECUTING AN EXTERNAL TRIGGER ROUTINE,  COME HERE
 4900 CONTINUE
C           HAS RTMRK BEEN CALLED
      IF(KR(1).EQ.0) GO TO 4920
C           YES, RETURN TO CALLING PROGRAM
 4910 KR(1)=0
      GO TO 9999
C
C     TEST FOR A DISCONTINUITY AFTER TRIGGER EXECUTION
 4920 NNT = NNT
      JJSS = 7
      GO TO 8000
 4930 CONTINUE
      VAR=VAR
C
C     IF THERE IS A DISCONTINUITY, SET INDICATOR
      IF ( IABS(NTR) .EQ. 2)  INDIS = 1
C
 4995 GO TO ( 320 , 3540 ) , J4000
C
C
C*****************************************************************
C
C ADAMS - MOULTON M-TH ORDER PREDICTOR - CORRECTOR ROUTINE WITH
C     AUTOMATIC STEP SIZE CONTROL
C     MOVES ONE STEP FORWARD USING CURRENT STEP SIZE, HC
C     RETURNS WITH AN INDICATOR,  IFL
C
C
C           IF IFL = 1, NORMAL RETURN
C                       TIME IS UPDATED
C                       Y'S ARE IN Y AND PY
C                       Y DERIVATIVES ARE IN DY
C                       BACK DIFFERENCES UPDATED AND IN DELX
C
C           IF IFL = 2, INDICATES STEP SIZE IS TOO BIG
C                       TIME IS SET BACK TO BEGINNING OF STEP
C                       OLD Y'S ARE PUT BACK INTO Y AND PY
C                       OLD BACK DIFFERENCES ARE STILL IN DELX
C                       OLD DERIVATIVES SET IN DY FROM FIRST COL. OF DELX
C                       END-OF-STEP CONDITIONS ARE RE-CALCULATED
C                       ANY DOUBLING INDICATORS ARE SHUT OFF
C
C           IF IFL = 3, INDICATES STEP SIZE IS TOO SMALL
C                       STATE OF STORAGE IS THE SAME AS FOR NORMAL RETURN
C
C     RETURN IS BY J5500
C
 5500 CONTINUE
      REDOT = .TRUE.
      IF (SMA) GO TO 5505
C     GET NEW TMIN AND NTT
      SMA = .TRUE.
      J1000 = 3
      GO TO 1000
```

```
      5505 TSV = T
C          SAVE CURRENT Y'S IN YN AND CALC PREDICTED Y'S
C
           DO 5520  I=1,NEQ
           YN(I) = PY(I)
           SK = 0.
C               SUM FROM K=0 TO M
           DO 5510  K=1,MP1
      5510 SK = SK + ACC(K)*DELX(I,K)
           PY(I) = YN(I) + HC*SK
      5520 Y(I) = PY(I)
C          CALCULATE DERIVATIVES OF PREDICTED Y'S AT T + HC
           T = TSV + HC
           IC(1)=1
           GO TO (5526, 5525),  NN
      5525 CALL DER1
      5526 CALL DER2
C
           IFL=1
           GO TO (5528, 9900, 5598), KIND
C          LEAVE DIFFERENCES AT BEG. OF STEP IN DELX
C          UPDATE DIFFERENCE TABLES FROM DELX USING PREDICTED
C          DERIVATIVES IN DELZ
      5528 CONTINUE
           DO 5530  I=1,NEQ
           TP1 = DY(I)
           DO 5530  K=1,MP2
           TP2 = TP1
           TP1 = TP1 - DELX(I,K)
      5530 DELZ(I,K) = TP2
C
C          CALCULATE CORRECTED Y'S AND GET LARGEST ERROR TERM
           ENP1 = 0.
           DO 5550  I=1,NEQ
           SKP=0.
           SK = 0.
           DO 5540  K=1,MP1
           SKP = SKP+ACC(K)*DELX(I,K)
      5540 SK = SK + BCC(K)*DELZ(I,K)
           PY(I) = YN(I) + HC*SK
           Y(I) = PY(I)
           D = ABS(Y(I))
           IF (D .LE. YL)  D = YL
           E = ABS(HC*(SKP-SK))/D/A
      5550 IF (E .GT. ENP1)  ENP1 = E
C          CALC. DERIVATIVES OF CORRECTED Y'S
           CALL DER2
C          TEST ERROR LIMITS FOR AUTOMATIC STEP SIZE CONTROL
           IF (ENP1 .LE. EL)  GO TO 5580
C
           IF(ENP1.LE.EL(1)) GO TO 5590
C          STEP SIZE IS TOO BIG
           IF (HC/2. .LT. HMN) GO TO 5590
C          RESET EVERYTHING TO CONDITIONS AT BEGINNING OF STEP
C               INCLUDING END OF STEP CONDITIONS
C               Y(I) AND PY(I) FROM YN(I)
C               DY(I) FROM 1ST COLUMN OF DELX
           T = TSV
           IC(1)=1
           DO 5560  I=1,NEQ
           Y(I) = YN(I)
           PY(I) = YN(I)
      5560 DY(I) = DELX(I,1)
           IF (ECST)  CALL EOS
C
           IFL = 2
           HO = .TRUE.
           TRIP = BIG
```

```
      819.          NSTART = 100
      820.          NCCO = 4
      821.    C                   TO RECALCULATE TMIN AND NTT
      822.          GO TO  1000
      823.     5570 GO TO  5599
      824.    C
      825.    C     STEP SIZE IS TOO SMALL
      826.     5580 IF ( .NOT. HC)  GO TO  5590
      827.          IF (2.*HC .GT. HMX) GO TO 5590
      828.          IFL = 3
      829.          SMA = .FALSE.
      830.    C
      831.    C     UPDATE DIFFERENCE TABLES USING CORRECTED DERIVATIVES
      832.     5590 DO 5595  I=1,NEQ
      833.          TP1 = DY(I)
      834.          DO 5595  K=1,MP2
      835.          TP2 = TP1
      836.          TP1 = TP1 - DELX(I,K)
      837.     5595 DELX(I,K) = TP2
      838.    C
      839.    C                RETURN
      840.     5599 GO TO (5020), JSSOC
      841.    C
      842.    C*****************************************************************
      843.    C
      844.    C  ROUTINE TO INTERPOLATE USING DIFFERENCE FORMULAS FOR THE Y'S AT TIME
      845.    C     SET IN T AND USING DIFFERENCE TABLES EVALUATED AT TGO.  BEFORE
      846.    C     RETURNING THE DERIVATIVES ARE CALCULATED AND STORED IN DY AND
      847.    C     THE END-OF-STEP ROUTINE HAS BEEN EXECUTED.
      848.    C              RETURN IS BY  J7000
      849.    C
      850.     7000 CONTINUE
      851.          REDOT = .FALSE.
      852.          HMU = TGO - T
      853.          FMU = HMU/HC
      854.          FACTJ = 1.0
      855.          FNUM = 1.0
      856.          DO 7020  J=1,M
      857.          FJ = J
      858.          FACTJ=FACTJ*(FJ+1.)
      859.          FNUM = FNUM*(FMU-FJ)
      860.     7020 CO(J) =(-1.)**J *FNUM/FACTJ
      861.    C     CALCULATE COEFFICIENTS OF DIFFERENCES
      862.          DO 7050  K=1,MP1
      863.          CK(K) = BCO(K)
      864.          IF (K .EQ. 1)  GO TO  7050
      865.          KK = K-1
      866.          DO 7040  J=1,KK
      867.          KJ = K-J
      868.     7040 CK(K) = CK(K) + CO(J)*BCO(KJ)
      869.     7050 CONTINUE
      870.    C
      871.    C     CALCULATE NEW Y'S AT T
      872.          DO 7080  I=1,NEQ
      873.          SUM = 0.
      874.          DO 7060  K=1,MP1
      875.     7060 SUM = SUM + CK(K)*DELX(I,K)
      876.     7080 Y(I) = PY(I) - HMU*SUM
      877.    C
      878.    C     CALCULATE DERIVATIVES
      879.          GO TO  (7092, 7091),  NN
      880.     7091 CALL DER1
      881.     7092 CALL DER2
      882.    C              GET EOS CONDITIONS
      883.          IF (ECST)  CALL  ECS
      884.          IF(XR(1).NE.0) GO TO 4910
      885.    C
      886.          GO TO (5450, 3350),  J7000
```

```
C
C
C*****************************************************************************
C       ROUTINE TO SET UP TRIGGER VARIABLES WITH CURRENT VALUES
 8000 GO TO (1,2,3,4,5,6,7,8,9,10,11),  .NT
C       RETURN TO THESE STATEMENT NOS. DEPENDING ON VALUE OF JJSS
C       JJSS =  1  ,  2  ,  3  ,  4  ,  5  ,  6  ,  7
 8100 GO TO (1030,58,365,6120,3110,3520,4930),JJSS
C
    1 NTR = NTR1
      VAR = VAR1
      ZV = ZV1
      GO TO  8100
    2 NTR = NTR2
      VAR = VAR2
      ZV = ZV2
      GO TO  8100
    3 NTR = NTR3
      VAR = VAR3
      ZV = ZV3
      GO TO  8100
    4 NTR = NTR4
      VAR = VAR4
      ZV = ZV4
      GO TO  8100
    5 NTR = NTR5
      VAR = VAR5
      ZV = ZV5
      GO TO  8100
    6 NTR = NTR6
      VAR = VAR6
      ZV = ZV6
      GO TO  8100
    7 NTR = NTR7
      VAR = VAR7
      ZV = ZV7
      GO TO  8100
    8 NTR = NTR8
      VAR = VAR8
      ZV = ZV8
      GO TO  8100
    9 NTR = NTR9
      VAR = VAR9
      ZV = ZV9
      GO TO  8100
   10 NTR = NTR10
      VAR = VAR10
      ZV = ZV10
      GO TO  8100
   11 NTR = NTR11
      VAR = VAR11
      ZV = ZV11
      GO TO  8100
C
C*****************************************************************************
C*****************************************************************************
C
C
 9000 WRITE (JO,9010)  TMIN, T, DELU
 9010 FORMAT (///10H T(MIN) = ,1PE10.3,28H IS LESS THAN CURRENT TIME, ,
     >                 E10.3, 17H, MINUS DELTA U, ,E10.3/)
C
 9900 ERROR = .TRUE.
 9999 RETURN
      END
```

```
      SUBROUTINE ELMENT
      CHARACTER*6 DN3(2)
      CHARACTER*12 DN1(4)
      CHARACTER*60 DN2
      COMMON/IUSE/IUSE
      COMMON/IIELE/II,DN2(2)
      DATA DN3/6H@ADD  ,6H .    /
      II=0
100   CALL PAGIT
      CALL ECUT('  SELECT FROM THE FOLLOWING MENU:',33)
      CALL ECUT('  0 - EXIT.',11)
      CALL ECUT('  1 - READ INPUT FROM ELEMENTS.',31)
      CALL FDATAI(1,IWANT,$100)
      IF(IWANT.EQ.0)GO TO 6
      IF(IWANT.EQ.1)GO TO 10
      GO TO 100
C                    INITIALIZE BY RESIDENT ELEMENT
   10 CONTINUE
5     CALL ECUT(' USE THE FOLLOWING PROCEDURE TO ENTER NON-RESIDENT ELEM
     *ENT',58)
      CALL ECUT(' FILENAME.ELEMENT-NAME',22)
      II=II+1
      IF(II.GT.2)GO TO 100
      CALL FDCHAR(DN1,4,4,$5)
      DN2(II)=DN3(1)//DN1(1)//DN1(2)//DN1(3)//DN1(4)//DN3(2)
      GO TO 100
6      RETURN
      END
```

ELTTBL

```
1        $INPUT2
2        OFFICE='MSFC-EL24'
3        DATE='16 MAR 1981'
4        TITLE='LBP TRAJECTORY - TEST'
5        NCASE=1,  NOTAE=1,
6        $END
```

```
      SUBROUTINE FIND(CHI,A,DT,NORDER)
C
C      DETERMINES POLYNOMIALS BASED ON ATTITUDE TIME HISTORY INFORMATION
C           CHI     : ARRAY OF DEPENDENT VARIABLES
C           A       : POLYNOMIAL COEFFICIENTS
C           DT      : ARRAY OF INDEPENDENT VARIABLES
C           NORDER  : ORDER OF REQUIRED POLYNOMIAL
      DIMENSION B(9,9),CHI(9),DT(9),A(9)
      IF(NORDER.EQ.0) RETURN
      A(1)=CHI(1)
      DO 1 I=1,NORDER
      DO 1 J=1,NORDER
    1 B(I,J)=DT(I+1)**J
      CALL MINV(B,NORDER)
      DO 2 I=1,NORDER
      A(I+1)=0.
      DO 2 J=1,NORDER
    2 A(I+1)=A(I+1)+B(I,J)*(CHI(J+1)-A(1))
      K=NORDER+1
      DO 3 I=1,K
    3 A(I)=A(I)/57.29578
      RETURN
      END
```

FUNISP

```
      FUNCTION FUNISP(TAU)
      COMMON/TABISP/MTAU,TIMISP(20),TRFISP(20)
      DATA A1/-5.5509/,A2/12.7964/,A3/447.9445/,A4/455.15/
      REFISP=455.15
      GO TO 10
      ENTRY VARISP(TAU,T)
      CALL APULG(1,MTAU,20,T,TIMISP,REFISP,TRFISP,Q,G)
   10 FUNISP=REFISP*((A1*TAU+A2)*TAU+A3)/A4
      RETURN
      ENTRY DEVISP(TAU,PTAUM)
      FUNISP=REFISP/A4*(2.*A1*TAU+A2)*PTAUM
      RETURN
      END
```

```
LBMW          /RI

@FTN,LW SDATA,SDATA
-10,12
      DATA(A(I, 4),I=1,3)/2796.,2796.,2796./
      DATA(A(I, 5),I=1,3)/141041.,141041.,141041./
      DATA(A(I, 6),I=1,3)/4259.,4259.,4259./
-14,15
      DATA(A(I, 8),I=1,3)/828.,524.,524./
      DATA(A(I, 9),I=1,3)/579.,579.,579./
-19,19
      DATA(A(I,13),I=1,3)/799.,799.,799./
-25,25
      DATA(A(I,19),I=1,3)/1383.,1383.,1383./
-30,31
      DATA(A(I,24),I=1,3)/73495.,73495.,73495./
      DATA(A(I,25),I=1,3)/4378.,4480.,4480./
-35,35
      DATA(A(I,29),I=1,3)/230.,154.,154./
-42,42
      DATA(A(I,36),I=1,3)/0.,76.,76./
-47,47
      DATA A(1,41)/32653./
-53,55
      DATA A(1,47)/344521./
      DATA A(1,48)/0./
      DATA A(1,49)/0./
-70,71
      DATA A(1,64)/12426./
      DATA A(1,65)/8691./
@ACC,P VRA71
@MAP,N SYM,ABS
@
```

LBMW-1

```
$INPUT
HEAD=' DATA FOR LIQUIE BOOST MODULE(LBM) SIMULATION    '
HEAD='  MFS PROPELLANT INVENTORY APPENDIX 10.12,CHANGE 57 '
MISIGN='ACA    '
HEAD=' SRB-MOF1(TC-121-78)-WTR BURNRATE-52DEG F PMBT '
SRMFTB=
        .00000000    ,    .98132651-01,    .19626530+00,    .29439796+00,
        .39253061+00,    .49066327+00,    .58879592+00,    .68692858+00,
        .78506123+00,    .88319388+00,    .98132651+00,    .19626531+01,
        .29439796+01,    .39253061+01,    .49066327+01,    .58879592+01,
        .68692858+01,    .78506123+01,    .88319387+01,    .98132651+01,
        .10794592+02,    .11775918+02,    .12757245+02,    .13738571+02,
        .14719898+02,    .15701225+02,    .16682551+02,    .17663878+02,
        .18645204+02,    .19626531+02,    .20607857+02,    .21589184+02,
        .22570510+02,    .23551837+02,    .24533163+02,    .25514490+02,
        .26495816+02,    .27477143+02,    .28458469+02,    .29439796+02,
        .30421122+02,    .31402449+02,    .32383775+02,    .33365102+02,
        .34346428+02,    .35327755+02,    .36309081+02,    .37290408+02,
        .38271735+02,    .39253061+02,    .40234388+02,    .41215714+02,
        .42197042+02,    .43178367+02,    .44159694+02,    .45141020+02,
        .46122347+02,    .47103673+02,    .48085001+02,    .49066326+02,
        .50047653+02,    .51028979+02,    .52010306+02,    .52991633+02,
        .53972960+02,    .54954286+02,    .55935612+02,    .56916939+02,
        .57898266+02,    .58879592+02,    .59860919+02,    .60842245+02,
        .61823571+02,    .62804899+02,    .63786225+02,    .64767550+02,
        .65748878+02,    .66730205+02,    .67711531+02,    .68692857+02,
        .69674184+02,    .70655510+02,    .71636837+02,    .72618163+02,
        .73599490+02,    .74580816+02,    .75562142+02,    .76543469+02,
        .77524796+02,    .78506123+02,    .79487450+02,    .80468777+02,
        .81450102+02,    .82431428+02,    .83412756+02,    .84394081+02,
        .85375408+02,    .86356735+02,    .87338061+02,    .88319387+02,
        .89300715+02,    .90282042+02,    .91263368+02,    .92244694+02,
        .93226021+02,    .94207347+02,    .95188674+02,    .96170001+02,
        .97151332+02,    .98132650+02,    .99113979+02,    .10009531+03,
        .10107664+03,    .10205796+03,    .10303929+03,    .10402061+03,
        .10500194+03,    .10598326+03,    .10696459+03,    .10794592+03,
        .10892725+03,    .10990857+03,    .11088990+03,    .11187123+03,
        .11285256+03,    .11383387+03,    .11481520+03,    .11579653+03,
        .11677786+03,    .11775918+03,    .11874051+03,    .11972184+03,
        .12070317+03,    .12168449+03,    .12186015+03,    .12266582+03,
        .12291115+03,    .12340181+03,    .12389248+03,    .12438314+03,
        .12487380+03,    .12536447+03,    .12585513+03,    .12634579+03,
SRMFTB=
        .00000000    ,    .40840229+06,    .15307892+07,    .25408739+07,
        .28941108+07,    .29532320+07,    .30360452+07,    .30213183+07,
        .29827921+07,    .29931443+07,    .30049904+07,    .30178960+07,
        .30437762+07,    .30761441+07,    .30971853+07,    .31252599+07,
        .31499333+07,    .31565520+07,    .31642069+07,    .31764939+07,
        .31669973+07,    .31939455+07,    .31969612+07,    .32017403+07,
        .32087717+07,    .32149675+07,    .32205942+07,    .32266815+07,
        .32328914+07,    .32406701+07,    .32401266+07,    .32352299+07,
        .31539803+07,    .30964024+07,    .30506594+07,    .30092532+07,
        .29750042+07,    .29340718+07,    .28988311+07,    .28636816+07,
        .28271834+07,    .27927752+07,    .27582911+07,    .27267616+07,
        .26974905+07,    .26672794+07,    .26378703+07,    .26086550+07,
        .25789408+07,    .25513143+07,    .25257952+07,    .25037936+07,
        .24625578+07,    .24610937+07,    .24439677+07,    .24277695+07,
        .24143538+07,    .24051758+07,    .24016450+07,    .23913798+07,
        .23684777+07,    .23713599+07,    .23385081+07,    .23183684+07,
        .23240247+07,    .23389735+07,    .23486150+07,    .23592309+07,
        .23716843+07,    .23845179+07,    .23963151+07,    .24121229+07,
        .24274522+07,    .24421211+07,    .24493857+07,    .24568561+07,
        .24728535+07,    .24800917+07,    .24910827+07,    .24949553+07,
        .25066187+07,    .25159762+07,    .25234152+07,    .25307983+07,
        .25449665+07,    .25592370+07,    .25755670+07,    .25767199+07,
        .25878753+07,    .26008762+07,    .25755164+07,    .25600765+07,
        .25314608+07,    .24811301+07,    .24576277+07,    .24376077+07,
```

```
68          .24169467+07,     .23964013+07,     .23793402+07,     .23601628+07,
69          .23435266+07,     .23276235+07,     .23131582+07,     .23002333+07,
70          .22845452+07,     .22651001+07,     .22452169+07,     .22276355+07,
71          .22139237+07,     .22009289+07,     .21873297+07,     .21737731+07,
72          .21538845+07,     .21268618+07,     .20987782+07,     .20627323+07,
73          .20251460+07,     .19732377+07,     .19265567+07,     .19056594+07,
74          .18693487+07,     .17990732+07,     .16844154+07,     .15302284+07,
75          .12511694+07,     .96721630+06,     .78490109+06,     .66737508+06,
76          .47035772+06,     .39124884+06,     .27860486+06,     .21002470+06,
77          .15637630+06,     .10920945+06,     .10196514+06,     .68833651+05,
78          .57572411+05,     .45490980+05,     .36171512+05,     .29711257+05,
79          .22382633+05,     .17107769+05,     .12276064+05,     .71270308+04,
80     SRPWCT=
81          0.,               1540.7296,        5725.0042,        9585.6749,
82          10518.282,        11141.281,        11453.742,        11398.195,
83          11252.775,        11291.874,        11336.556,        11386.314,
84          11464.003,        11606.977,        11686.895,        11793.215,
85          11886.943,        11912.736,        11942.490,        11989.708,
86          12030.328,        12057.339,        12069.627,        12088.515,
87          12118.776,        12140.406,        12162.543,        12186.508,
88          12210.875,        12241.242,        12242.461,        12227.188,
89          11922.485,        11707.306,        11536.605,        11382.454,
90          11255.115,        11102.591,        10971.391,        10840.597,
91          10704.626,        10576.372,        10447.514,        10330.526,
92          10221.666,        10109.152,        9999.5815,        9890.7232,
93          9775.6334,        9676.9650,        9582.0186,        9500.2725,
94          9421.1675,        9341.2495,        9277.5794,        9217.4629,
95          9167.7044,        9133.9915,        9121.9075,        9084.3347,
96          9073.7735,        9007.0568,        8885.4034,        8809.1410,
97          8830.8722,        8887.8397,        8924.6001,        8965.2196,
98          9012.7430,        9061.5883,        9106.4723,        9166.6896,
99          9225.0796,        9280.9300,        9308.7546,        9337.3907,
100         9356.4205,        9426.2446,        9468.2854,        9483.3139,
101         9527.9956,        9563.8417,        9592.5794,        9621.0125,
102         9675.2395,        9729.8715,        9792.3234,        9796.9950,
103         9835.7458,        9889.9107,        9795.0648,        9737.8942,
104         9630.6601,        9440.9698,        9352.9275,        9278.0874,
105         9200.8085,        9123.9372,        9060.3699,        8988.6765,
106         8922.8221,        8833.8763,        8813.7104,        8765.6811,
107         8707.3899,        8634.6826,        8560.3493,        8494.7492,
108         8443.8742,        8395.7404,        8345.2721,        8294.9044,
109         8220.5717,        8119.125,         8013.6170,        7877.646,
110         7735.7836,        7539.188,         7362.3539,        7283.796,
111         7223.3755,        6882.0738,        6453.7468,        5874.2158,
112         4815.279,         3133.4922,        3039.1127,        2447.0907,
113         1977.4335,        1605.4655,        1154.5942,        834.21259,
114         567.24896,        358.87842,        372.65884,        252.24425,
115         212.40738,        166.97296,        132.95752,        109.42331,
116         82.60435,         63.26130 4,       45.505824,        26.49373,
117    NSYST=241,242,243,244,445,
118    ENEDAT(1,1) =-31.3225,-1.3462,- 7.0900,-10.0000,0.,470000.,1032.6266,4.170500,
119    ENEDAT(1,2) =-31.3225, 1.3462,- 7.0900,-10.0000,0.,470000.,1032.6266,4.170500,
120    ENEDAT(1,3) =-36.7340, 0.      ,-9.63920,-16.0000,0.,470000.,1032.6266,4.170500,
121    ENEDAT(1,4) =-36.6319,-6.3627, 0.      , 0.      ,0.,0.      , 0.      ,10.776365,
122    ENEDAT(1,5) =-36.6319, 6.3627, 0.      , 0.      ,0.,0.      , 0.      ,10.776365,
123    ENEDAT(1,6) =-35.4660, 0.0000, 0.73660, -7.5000,0.,264500.,877.60045,1.8129,
124    ENEDAT(1,7) =-35.4660, 0.0000,-0.73660, -1.5000,0.,264500.,877.60045,1.8129,
125    ENEDAT(1,8) =-32.5628,-2.2352,-10.8839,-15.2924,6., 6000.0,19.1571 ,0.,
126    ENEDAT(1,9) =-32.5628, 2.2352,-10.8839,-15.2924,-6.,6000.0,19.1571 ,0.,
127    ENEDAT(1,10)=-33.     ,-3.    ,-10.    ,-10.    ,0., 0.    , 0.    ,0.,
128    ENEDAT(1,11)=-33.     ,-3.    ,-10.    ,-10.    ,0., 0.    , 0.    ,0.,
129    ENEDAT(1,12)=-33.     , 3.    ,-10.    ,-10.    ,0., 0.    , 0.    ,0.,
130    ENEDAT(1,13)=-33.     , 3.    ,-10.    ,-10.    ,0., 0.    , 0.    ,0.,
131    PRCP=155716 4.,2232524.,350000.,250640.,7508.,
132    QE(M(12)=3.,
133    TNE(1,1)=3.,2.,4*0.,  3.,2.,4*0.,  3.,2.,2.,3*0.,  3.,2.,2.,3*0.,
134    TNE(1,6)=3.,0.,2.,3*0.,  3.,0.,2.,3*0.,  3.,5*0.,  3.,5*0.,  3.,5*0.,
135    TNE(1,10)=3.,5*0.,  3.,5*0.,  3.,5*0.,  3.,5*0.,
136    SRMAE=10.744825,0.,
```

```
WTJET=3*0.,344521.,0.,32653.,9*0.,
TALT=.3,4.7,103.77,15.,41.23,40.925,42.9,8.,16.,8.,103.,96.2,6.,2*0.,
IPR=0,
STEP=4*.1,11*4., BSTEP=4*1.,11*4., PRINT=4*1.,11*1000.,
NOEVNT=4,5,
WG1=4843599.,
MPSFLG=1,2,2,8*1,4,1,
IFACT=2,
VRCUT=0.,
KCB=6*0,1,4*0,1,5*0,1,12*0,
S=13*249.50918,
HEAD=' MACH VS ALPHA AND BETA/LBM TRAJ-RI WINTER(5/5/80)'
TMACH=0.,.14,.26,.30,.35,.40,.44,.49,.54,.58,.63,.68,.77,.82,.95,1.0,
      1.14,1.25,1.62,2.01,2.48,3.10,3.82,4.26,5.0,
TALFP=8.11,8.10,8.10,8.67,4.26,2.60,1.57,.40,-.71,-1.88,-2.71,-3.27,
      -3.64,-3.67,-3.31,-3.34,-3.08,-3.20,-1.68,.02,5*2.0,
TBETA=3*-1.35,-1.03,-.3,-.72,-.96,-.72,-.49,-.33,-.24,-.18,-.10,12*0.,
XLEN=2*32.766,
XREF=0.,0.,
ZREF=0.,0.,
HEAD=' TOTAL VEHICLE CG DATA-BOOSTER PHASE-WITH LBM-RI(2/80)'
TBCWT=0.,454249.,790287.,1049464.,1329575.,1731056.,2168109.,2568995.,
      2775195.,2831139.,3000000.,
TXCG=-13.01,-13.05,-12.95,-12.76,-12.36,-11.53,-10.53,-9.52,-9.07,2*-9.05,
TZCG=-.427,-.472,-.512,-.547,-.590,-.666,-.773,-.908,-1.001,2*-1.030,
HEAD=' TOTAL VEHICLE CG DATA-ORBITER PHASE-WITH LBM-RI(2/80) '
TBCWT(1,2)=0.,103611.,349500.,424460.,442495.,642495.,842495.,947894.,
          1086395.,1148572.,1400000.,
TXCG(1,2)=-6.52,-6.44,-6.11,-6.08,-6.099,-7.991,-10.610,-12.581,-16.452,
         2*-18.644,
TZCG(1,2)=-1.245,-1.327,-1.571,-1.667,-1.692,-2.093,-2.799,-3.437,-4.938,
         2*-5.944,
ICONSW=4,
IHEAD=1,
HEAD=' WTR/DECEMBER MEAN .IND/ '
NWIND=70,
ALTTBL=0.,1000.,2000.,3000.,4000.,5000.,6000.,7000.,8000.,9000.,10000.,
11000.,12000.,13000.,14000.,15000.,16000.,17000.,18000.,19000.,20000.,
21000.,22000.,23000.,24000.,25000.,26000.,27000.,28000.,29000.,30000.,
31000.,32000.,33000.,34000.,35000.,36000.,37000.,38000.,39000.,40000.,
41000.,42000.,43000.,44000.,45000.,46000.,48000.,50000.,53000.,56000.,
60000.,65000.,70000.,75000.,80000.,85000.,90000.,95000.,100000.,
100500.,101000.,101500.,102000.,103000.,104000.,106000.,108000.,
110000.,115000.,
AZWTBL=341.03,329.18,308.22,297.66,293.39,290.78,289.46,
       288.55,286.77,285.68,283.97,282.34,279.93,277.59,
       275.81,275.48,275.80,276.39,279.23,286.52,297.35,
       315.00,335.35,352.90,357.33,364.20,360.22,344.50,
       274.65,274.73,274.49,272.81,271.15,269.19,268.64,
       267.24,266.46,266.29,265.84,266.06,265.96,265.35,
       264.39,263.51,262.75,262.05,261.94,261.94,261.94,
       21*261.94,
WTBL=1.12,2.70,5.09,7.93,10.81,13.31,15.52,17.76,19.82,21.87,23.74,
      25.18,25.29,24.45,22.64,20.30,17.12,13.94,10.34,7.28,5.20,3.72,
      2.81,2.59,2.79,2.60,2.62,2.84,10.99,12.97,14.96,16.54,19.95,23.99,
      29.06,34.08,38.92,44.06,47.71,52.06,55.86,58.80,62.30,66.02,
      69.30,71.37,14*72.,36.,9*0.,
ALAT=34.58139,
ALONG=120.72528,
AZ=177.9,
ALTLS=158.16,
JUMP=1,
VIV=20*0.,
NMAX=11,
LAST=1,
KIND=1,
KINDB=3,
ISTART=8,14,
NTABLE=0,
```

```
      TZERO=0.,
      TLIFT=7.7,
      TTILT=18.,
      TCHIR=7.7,
      FAZ=270.,
      CHRDOT=8.55,
      KP=1,0,
      KY=1,0,
      IFCLY=20,
      KWTA=3,3,3,3,
      MINH=5,
      TTBL=0.,10.3,13*0.,
      CFTBL=0.,15.5,28*0.,
      TTBL(16)=0.,10.3,13*0.,
      CYTBL=0.,-3.,28*0.,
      KCB(25)=1,
      NBGCT=5,8,0,      NENDCT=8,14,0,      NP=29,29,0,
      TCBL=123.77,237.8,27*0.,          TOBL(31)=237.8,486.,27*0.,
      CPCTBL=64.6,81.7,27*0.,           CPOTBL(31)=81.7,110.2,27*0.,
      CYCTBL=5.4,7.,27*0.,              CYOTBL(31)=7.,9.1,27*0.,
      HEAD=' LBP/WTR/DEC PSEUDO-MECO TARGET--5/5/80 RI TRAJ'
      KCDPHI=1,2,3,4,10,11,
      PSIREQ=7733.9952,.65,648372*.,96.173,-4.02546,
      PSIREQ(5)=-2.7512,
      NVRST=4,7,
      KCDRES(1,1)=8,9,15,              KCDRES(1,2)=7,
      PSIRST(1,1)=-171.3,25.5,0.,       PSIRST(1,2)=2570.6527,
      FFRFAC=.0085
      WIBT=1.,1.,
      NOBASE=20,
      OBAXIL=0.,
      VRINT=0.,0.,   VSLOPE=0.,0.,
      TTABLE=20*0.,
      SIGTAB=20*0.,
      JSTART=0,
      JETET=0,
      IRTLS=0,
      ITCL=6*1,9*0,
      HEAD=' LBP DRAG DATA FUNC(MACH NBR)--MSFC DATA  '
      TCMACH=0.,.6,.8,.9,.95,1.05,1.1,1.15,1.25,1.4,1.55,1.8,2.2,2.5,3.5,4.5,
      DELCA=20*0.,
      DELCN=2*.085,.077,.057,.049,.067,.065,.06,.055,.05,.047,.042,.032,.021,.012,
            .005,
      DELCM=2*-.066,-.061,-.048,-.046,-.06,-.057,-.055,-.053,-.05,-.048,-.043,-.033,
            -.022,-.013,-.0055,
      DELALT=0.,519.,4000.,8000.,12000.,16000.,24000.,30000.,34000.,38000.,42000.,
            46000.,55000.,62500.,70000.,75000.,80000.,85000.,90000.,100000.,110000.,
            160000.,230000.,300000.,
      DELFAB=0.,0.,8600.,11200.,9300.,6200.,-3000.,-9300.,-12000.,-13200.,-12900.,
            -12100.,-9800.,-7500.,-4800.,-3300.,-2100.,-1200.,-600.,100.,400.,900.,
            900.,900.,
      THROT=1.,1.,10*1.09,.65,
      QMAX=650.,
      TALALT=0.,762.,1524.,2651.76,40000.,
      TALTBL=.9,.82,.76,.65,.65,
      TIMLBM=0.,5.722,5.857,5.9,6.3,15.,25.,35.,45.,55.,75.,85.,95.,105.,125.,
            135.,185.,205.,205.6544,205.8374,205.8954,
      THRLBM=0.,1.505,146653.,201807.5,258797.5,265468.,264813.5,264692.,264089.5,
            264122.,265583.,268914.,265725.,265118.,263637.5,263360.,261111.,
            255580.5,200060.,45326.,1.505,
      WOTLBM=0.,.005,874.6,874.6,860.14,882.58,879.98,879.235,876.625,876.32,881.13,
            882.28,881.78,879.7,874.95,873.945,865.89,860.62,876.5,876.5,.005,
      TALT(6)=40.8954,
      TIMISP=0.,20.,40.,60.,80.,100.,120.,140.,160.,180.,200.,220.,240.,260.,280.,
            300.,
      TRFISP=453.20476,453.55262,453.85509,454.11595,454.33898,454.52771,454.68554,
            454.81571,454.92130,455.00525,455.07032,455.11912,455.15413,455.17764,
            455.19180,455.19860,
```

```
PSIRST(1,2)=256.96, PSIFEQ(5)=-2.97,
AZ=180.,
TIPMPS=0.,3.5,4.4,41.,45.4,62.,66.4,200.,7*0.,
FIBL=1.,1.,1.05,1.09,.65,.65,1.09,1.09,7*0.,
LPRINT=1,
HEAD=' LPM(150M)+SRB-MCF1(T(-121-78)-WTR(52 F)/MSN4-WIN/F-AOA/    '
KRCER=5,
STEP=3*.05,
NTAU=4,
NBETA=3,
RATIC=.85714286,
KCB(25)=1, KCB(29)=1,1,1,
KCCRES(1,1)=15,0,0,     PSIRST(1,1)=0.,0.,0.,
STEP=44.1, KINC=2,KINCB=2,
STEP=.1,.25,.5,.25,
NMAX=25,
TAE(1,2)=1.,2.,2.,0.,0.,0.,
TALT=5.,45.,58.77,
MPSFLG=2,2,2,1,
PSIRST=650.,
STEP=.1,.25,.25,.25,
KCB(29)=0,0,0,
NTAU=0,
KCB(31)=0,
NTABLE=1,
TIMTAU=4.,
$END
```

LBMW-2

```
$INPUT
HEAD=' WTR/DEC WIND/52 DEG SRM/AOA/    '
MISION='VAR IY'
JEMP=7,
NOEVNT=4,8,
GLIM=10*.1E+20,3.,4*.1E+20,
WTJET=5*0.,76.,9*0.,
TALT=6*0.,4.5,5.3307,165.5149,124.6518,44.,6.,3*0.,
KDB=10*0,1,19*0,
VRCUT=0.,
IPR=0,
ITOL=15*0,
HEAD='      2ND STAGE(    ACA ) CG HISTORY/RI-MSN4(WTR)-DEC TRAJ/'
TBDWT(1,2)=0.,164678.,442495.,584482.,768615.,880719.,1026341.,1118156.,
1162310.,1200000.,
TXCG(1,2)=-2.9597,-4.049,-6.099,-7.272,-9.150,-10.757,-13.678,-16.515,
-18.070,-18.070,
TZCG(1,2)=-1.156,-1.346,-1.692,-1.900,-2.355,-2.794,-3.752,-4.958,
-5.700,-5.700,
ISTART=7,13,
MINH=7,
NBGCT=7,0,0,
NENDCT=13,0,
NF=29,0,0,
TCBL=237.8,587.8,27*0., TCBL(31)=61*0.,
CPOTBL=73.,107.,27*0., CPOTBL(31)=61*0.,
CYOTBL=.5,2.5,27*0., CYOTBL(31)=61*0.,
NVRST=0,0,
KCDRES=12*0,
PSIRST=12*0.,
PSIREQ=7725.4608,.65,6483724.,
NWIND=0, ALTTBL=75*0., WTBL=75*0., AZWTBL=75*0.,
FPRFAC=.0057,
NCBASE=20,
JSTART=7,
THROT=1.,1.,9*1.09,.65,3*0.,
TNE(1,7)=2.,0.,0.,2.,0.,0.,  2.,0.,0.,2.,4.,0.,  2.,0.,0.,2.,4.,0.,
TNE(1,10)=2.,0.,0.,2.,4.,0.,  2.,5*0.,  2.,5*0.,  6*0.,
MPSFLG(11)=4,1,
FCMS=6*0.,6070.,5471.,5960.,6070.,5*0.,
WDCMS=6*0.,19.37,17.53,15.03,19.37,5*0.,
WDDUMP=7*0.,49.482,12.4,6*0.,
FRCS=7*0.,775.,870.,877.,5*0.,
WDRCS=9*0.,3.132675,5*0.,
HEAD=' LBM(350M)+SRB-MCF1(TC-121-78)-WTR(52 F)/MSN4-WIN/AOA/
ITOL=0,
IRTAOA=0,
$END
```

LBMW-3

```
$INPUT
HEAD=' WTR/DEC WIND/NCM MEGC/'
MISION='NCM    '
GLIM=7*,1E+20,3.,7*,1E+20,
WTLET=15*0.,
TALT=6*0.,160.,137.7,6.,6*0.,
IPR=0,
STEP=4*1.,6*4., BSTEP=4*1.,6*4., PRINT=4*1.,6*16.,
PRINT=15*1000.,
NOEVNT=4,5,
VRCUT=0.,
KOB=7*0,1,22*0,
S=10*249.50918,
HEAD=' 2ND STAGE(NOMINAL) CG HISTORY/RI-MSN4(WTR)-DEC TRAJ/'
TBCWT(1,2)=0.,164678.,442495.,642495.,842495.,947894.,1086395.,
1148572.,1500000.,
TXCG(1,2)=-2.9597,-4.049,-6.099,-7.991,-10.610,-12.581,-16.452,
-18.644,-18.644,
TZCG(1,2)=-1.196,-1.346,-1.692,-2.093,-2.795,-3.437,-4.938,
-5.944,-5.944,
KINOB=3,
JUMP=7,
ISTART=7,10,
KWTA=4*1,
MINH=7,
NBGCT=7,0,0,
NENDCT=10,0,
NP=29,0,0,
TCBL=237.8,481.5,27*0., TCBL(31)=61*0.,
CPCTBL=85.3,103.,27*0., CPOTBL(31)=61*0.,
CYCTBL=10.5,11.5,27*0., CYOTBL(31)=61*0.,
PSIREQ=7733.9952,.65,6483724.,98.,-4.03903,
NVRST=0,0,
KCCRES=12*0,
PSIRST=12*0.,
EPREAC=.0085,
FCMS=15*0., WDCMS=15*0.,
FRCS=15*0., WDRCS=15*0.,
WCCUMP=15*0.,
NCBASE=20,
JSTART=7,
JSTET=0,
MPSFLG(7)=1,4,7*1,
THROT(7)=2*1.09,.65,4*0.,
TNE(1,7)=3.,5*0.,3.,5*0.,  3.,5*0., 6*0.,
HEAD=' LBM(350M)+SRB-MCF1(TC-121-78)-WTR(52 F)/MSN4-WIN/NOM/
$END
```

LPMW-4

```
$INPUT
HEAD=' WTR/DEC/VAR IY/RTLS/'
MISION='RTLS   '
GLIM=10*,1E+20,3.,3.,3*,1E+20,
WTLET=4*0.,76.,10*0.,
TALT=5*0.,4.5,6.65,287.9E46,27.5271,53.4176,37.5,9.,6.,240.,
STEP(6)=944., ESTEP(6)=9*4., PRINT(6)=9*8.,
PRINT=15*1000.,
NCEVNT=4,5,
VRCUT=0.,
KCE=10*0,1,19*0,
S=15*0.,
HEAD='      2ND STAGE(RTLS) CG DATA/RI-MSN4(WTR)'
TBCWT(1,2)=0.,17640.,217059.,106836.,219527.,332218.,449609.,555195.,
           665507.,677779.,701340.,703336.,710869.,725000.,750000.,
TXCG(1,2)=-6.11,-6.23,-6.33,-6.90,-7.86,-9.00,-10.44,-12.49,-15.52,
          -15.68,-16.81,-16.88,3*-17.15,
TZCG(1,2)=-1.75,-1.70,-1.72,-1.82,-2.81,-2.28,-2.67,-3.34,-4.54,-4.62,
          -5.13,-5.16,3*-5.31,
LAST=0,
JUMP=6,
ISTART=6,0,
KINOB=3,
KF=1,1,KI=1,1,IPOLY=6,
KWIA=4*3,
MINM=8,
IFACT=1,
NP=29,3,  NBGCT=8,12,NENCCT=12,14,
TTBL=3*0.,11.15,8*0.,0.,11.15,16*0.,
CFTBL=2*0.,83.,-50.8,26*0.,
CYTBL=2*0.,2*5.,26*0.,
TCBL=237.6,535.,  TCBL(31)=535.,550.,
CPCTBL=3.2,-35.1,27*0., CPCTBL(31)=-60.,-60.5,
CYCTBL=-10.,-8.,27*0., CYCTBL(31)=-8.,-8.,27*0.,
KCDFHI=1,5*0,
PSIREQ=10*0.,
NVFST=11,
KCDRES=4,5,18,8,
PSIRST=6441572.,4.4,0.,10.9,
NGEASE=5,
FPRFAC=.0057,
VRINT=6.,474.62392,
VSLOPE=0.,4.7476635,
JSTART=6,
JETET=0,
IRTLS=0,
THROT(6)=7*1.05,.65,2*0.,
TNE(1,6)=2.,0.,0.,2.,0.,0.,  2.,0.,0.,2.,4.,0.,  2.,0.,0.,2.,4.,0.,
TNE(1,9)=2.,0.,0.,2.,4.,0.,  2.,0.,0.,2.,4.,0.,
TNE(1,11)=2.,5*0.,  2.,5*0.,  2.,5*0.,
FCPS=5*0.,6070.,2*5245.,2*6070.,5*0.,
WCCMS=5*0.,19.37,2*16.84,2*19.37,5*0.,
WCCUMP=6*0.,2*62.64,7*0.,
FRCS=6*0.,2*738.,735.3,911.5,5*0.,
WCRCS=8*0.,7.875,3.25,5*0.,
MRSFLG(6)=5*1,2*4,3*1,
HEAD=' LEM(250MT+SRB+OF-11TE-121-78)WTR(52-F)/MSN4-WIN/RTLS/'
$END
```

```
      SUBROUTINE LLPRT
      CHARACTER*6 HEADER(20),OUT(12),NAME(16),IUNIT(16)
      CHARACTER*60 HEAC
      DIMENSION BLOCK(22),PHITE(10),SAVE(8,8)
      COMMON/LLFLAG/LLFLAG
      IF(LLFLAG.EQ.0)GO TO 11
      LLFLAG=0
C-----------------WRITE END ON 10,11,12,13-----------------
      END FILE 10
      END FILE 11
      END FILE 12
      END FILE 13
11    IHIT10=0
      IHIT11=0
12    CONTINUE
      IHIT12=0
      IHIT13=0
C-----------------------------------------------------------
1     IWANT=0
      REWIND 1
      REWIND 2
      REWIND 10
      REWIND 11
      REWIND 12
      REWIND 13
      CALL PAGIT
      CALL GOUT(' 0 - TERMINATE TO MAIN MENU',27)
      CALL GOUT(' 1 - PRINT CONSTRAINTS AND OPTIMIZED UPDATE VALUES',51
     *)
      CALL GOUT(' 2 - PRINT CONTROL VARIABLE UPDATE',34)
      CALL GOUT(' 3 - PRINT PARAMETER SUMMARY TABLE',34)
      CALL GOUT(' 4 - PRINT LOAD INDICATOR TABLE',31)
      CALL READAI(1,IWANT,51)
      IF(IWANT.EQ.0)LLFLAG=0
      IF(IWANT.EQ.0)RETURN
      IF(IWANT.GT.4) GO TO 1
      IF(IWANT.GT.2)GO TO 6000
      ICOUNT=0
      CALL PAGIT
50    CONTINUE
      IF(IHIT10.EQ.0) READ(10,ERR=500,END=500) JA,K,CASE,HEADER,PHITE
      IF(IHIT10.EQ.0)GO TO 51
500   IHIT10=10
      GO TO 2000
51    CONTINUE
      ENCODE(100,OUT) CASE
100   FORMAT(14H  PASTRE CASE=,F11.4)
      CALL GOUT(OUT,25)
      ICOUNT=ICOUNT+1
      IF(IWANT.NE.1)GO TO 3000
C-----------------------TEST  K-----------------------------
      IF(K.GT.10)GO TO 1000
      ENCODE(101,OUT)(HEADER(I),I=1,K)
101   FORMAT(7X,10A6)
      KP=K*6+7
      CALL GOUT(OUT,KP)
      KHALF=K/2
      ENCODE(102,OUT)(PHITE(I),I=1,KHALF)
102   FORMAT(7X,5E12.5)
      CALL GOUT(OUT,KP)
      ICOUNT=ICOUNT+2
      GO TO 2000
1000  ENCODE(101,OUT)(HEADER(I),I=1,10)
      CALL GOUT(OUT,67)
      ENCODE(102,OUT)(PHITE(I),I=1,5)
      CALL GOUT(OUT,67)
      ENCODE(101,OUT)(HEADER(I),I=11,K)
```

```
      KP=(K-10)*6+7
      CALL EOUT(OUT,KP)
      KHALF=K/2
      ENCODE(102,OUT)(FHITE(I),I=6,KHALF)
      CALL EOUT(OUT,KP)
      ICOUNT=ICOUNT+4
 2000 CONTINUE
C-----------------------------------READ 11-----------------------
 198  IF(IHIT11.EQ.0) READ(11,ERR=1988,END=1988) JA,K,BLOCK(3)
      IF(IHIT11.EQ.0)GO TO 199
 1988 IHIT11=11
      GO TO 2000
 199  CONTINUE
      IF(JA.EQ.8)GO TO 206
      IF(JA.EQ.1)ENCODE(201,OUT)K,BLOCK(3)
 201  FORMAT(2X,5HTAUT(,I2,2H)=E15.8)
      IF(JA.EQ.2)ENCODE(202,OUT)BLOCK(3)
 202  FORMAT(7X,4HGC1=E15.8)
      IF(JA.EQ.3)ENCODE(203,OUT)BLOCK(3)
 203  FORMAT(8X,3HAZ=E15.8)
      IF(JA.EQ.4)ENCODE(204,OUT)K,BLOCK(3)
 204  FORMAT(1X,6HCPTEL(,I2,2H)=E15.8)
      IF(JA.EQ.5)ENCODE(205,OUT)K,BLOCK(3)
 205  FORMAT(1X,6HCYTBL(,I2,2H)=E15.8)
      IF(JA.EQ.6)ENCODE(207,OUT) K,BLOCK(3)
 207   FORMAT(1X,6HT PAR(,I2,2H)=,E15.8)
       IF(JA.EQ.7) ENCODE(208,OUT) K,BLOCK(3)
 208  FORMAT(1X,6HB PAR(,I2,2H)=,E15.8)
      CALL EOUT(OUT,26)
      ICOUNT=ICOUNT+1
      GO TO 198
 206  CONTINUE
      IF(ICOUNT.LE.26)GO TO 3000
      CALL HOLDIT
      ICOUNT=0
      CALL FAGIT
C-----------------------------------READ 12-----------------------
 3000 CONTINUE
      IF(IWANT.NE.2)GO TO 401
      IF(IBACK.EQ.1234)BACKSPACE 12
      IBACK=1234
      CALL EOUT('     TIME      NEW CP     OLD CP    NEW CY    OLD CY    DCP
     *   DCY    ',64)
      ICOUNT=ICOUNT+1
      IF(IHIT12.EQ.12)GO TO 4000
 300  IF(IHIT12.EQ.0) READ(12,ERR=30001,END=30001)(BLOCK(I),I=1,7),JJ
      IF(IHIT12.EQ.0)GO TO 3001
 30001 IHIT12=12
      GO TO 4000
 3001 CONTINUE
      IF((SVT-.1) .GT. BLOCK(1))GO TO 4000
      SVT=BLOCK(1)
      ENCODE(301,OUT)(BLOCK(I),I=1,7),JJ
 301  FORMAT(1X,F9.1,1X,F8.2,1X,F8.2,1X,F8.3,1X,F8.3,1X,F8.3,1X,F8.3,1X,
     *13)
      CALL EOUT(OUT,67)
      ICOUNT=ICOUNT+1
      IF(ICOUNT.LT.34)GO TO 300
      CALL HOLDIT
      ICOUNT=0
      CALL FAGIT
      CALL EOUT('     TIME      NEW CP     OLD CP    NEW CY    OLD CY    DCP
     *   DCY    ',64)
      GO TO 300
C-----------------------------------READ 13-----------------------
 4000 SVT=BLOCK(1)
      IF(IHIT13.EQ.0)READ(13,ERR=4010,END=4010)(BLOCK(I),I=1,4)
```

```
      IF(IHIT13.EQ.0)GO TO 4001
4010  IHIT13=13
      GO TO 5000
4001  CONTINUE
      ENCODE(400,OUT)(BLOCK(I),I=1,4)
400   FORMAT(1X,13HDEL CHIP MAX=,E11.4,14H DEL CHIY MAX=,E11.4,
     *6H BETCON=,L3,6H  KAT=,I3)
      CALL GOUT(OUT,70)
      CALL HOLDIT
      ICOUNT=0
401   IF(IWANT.EQ.1 .AND. (IHIT10+IHIT11).NE.21)GO TO 50
      IF(IWANT.EQ.2)GO TO 50
5000  CONTINUE
      CALL HOLDIT
      CALL PAGIT
      GO TO 11
C-----------------------READ 1--------------READ 1----------------------
C-------------------------PARAMETER SUMMARY TABLE----------------------
6000  CONTINUE
      IF(IWANT.GT.3)GO TO 7000
      CALL PAGIT
      CALL GOUT(' PARAMETER SUMMARY TABLE',24)
      CALL GOUT(' ',1)
      BLOCK(1)=6H
      READ(1) HEAD,(BLOCK(I),I=1,22)
      ENCODE(6001,OUT) HEAD
6001  FORMAT(6X,A60)
      CALL GOUT(OUT,66)
      ENCODE(6002,OUT)(BLOCK(I),I=1,3)
6002  FORMAT(5X,4HTIME,E15.8,7H  RANGE,E15.8,12H RANGE ANGLE,E15.7)
      CALL GOUT(OUT,72)
      ENCODE(6003,OUT)(BLOCK(I),I=4,6)
6003  FORMAT(9H INERTIALE15.8,7H RADIUSE15.8,12H FLIGHT PATHE14.7)
      CALL GOUT(OUT,72)
      ENCODE(6004,OUT)
6004  FORMAT(1X,8HVELOCITY,41X,5HANGLE)
      CALL GOUT(OUT,55)
      ENCODE(6005,OUT)(BLOCK(I),I=7,9)
6005  FORMAT(9H      INCLE15.8,7H   DES.E15.8,12H       FLIGHTE14.7)
      CALL GOUT(OUT,72)
      ENCODE(6006,OUT)
6006  FORMAT(27X,4HNODE,19X,7HAZIMUTH)
      CALL GOUT(OUT,57)
      ENCODE(6007,OUT)(BLOCK(I),I=10,12)
6007  FORMAT(9H     GDLATE15.8,7H  GCLATE15.8,12H          LONGE14.7)
      CALL GOUT(OUT,72)
      ENCODE(6008,OUT)(BLOCK(I),I=13,15)
6008  FORMAT(9H INERTIALE15.8,7H      C1E15.8)
      CALL GOUT(OUT,46)
      CALL GOUT('  AZIMUTH',9)
      ENCODE(6009,OUT)(BLOCK(I),I=16,17)
6009  FORMAT(9H    WEIGHTE15.8,7H      C3E15.8)
      CALL GOUT(OUT,46)
      CALL GOUT(' KEPLERIAN ORBITAL PARAMETERS',25)
      ENCODE(6010,OUT)(BLOCK(I),I=18,20)
6010  FORMAT(9H ECNTRICYE15.8,7H APOGEEE15.8,12H      PERIGEEE14.7)
      CALL GOUT(OUT,72)
      ENCODE(6011,OUT)
6011  FORMAT(25X,6HRADIUS,20X,6HRADIUS)
      CALL GOUT(OUT,57)
      ENCODE(6012,OUT)(BLOCK(I),I=21,22)
6012  FORMAT(9H    APOGEEE15.8,7HPERIGEEE15.8)
      CALL GOUT(OUT,46)
      CALL GOUT('   HEIGHT                    HEIGHT',31)
      CALL HOLDIT
C-----------------------------------------------------------------------
      GO TO 11
C-------------------------------------READ 2-----------READ 2----------
```

```
7000  CONTINUE
      CALL PAGIT
      CALL GOUT(' LOAD INDICATOR TABLE',21)
      CALL GOUT(' ',1)
      READ(2) HEAD,NAME,SAVE,IUNIT
      ENCODE(6001,OUT) HEAD
      CALL GOUT(OUT,66)
      CALL GOUT('                                        FIRST STAGE',44)
      CALL GOUT('                                MINIMUM              MAXIMUM',51)
      CALL GOUT(' LOAD INDICATOR      TIME      VALUE      TIME      VALUE
     *     UNITS',64)
      DO 7001 I=1,8
      N=2*I-1
      ENCODE(7002,OUT)NAME(N),NAME(N+1),(SAVE(K,I),K=1,4),IUNIT(N),
     *IUNIT(N+1)
7001  CALL GOUT(OUT,66)
7002  FORMAT(2X,2A6,4X,2(F6.2,2X,F8.2,2X),2A6)
      CALL GOUT(' ',1)
      CALL GOUT('                                        SECOND STAGE',45)
      DO 7003 I=1,8
      N=2*I-1
      ENCODE(7002,OUT)NAME(N),NAME(N+1),(SAVE(K,I),K=5,8),IUNIT(N),
     *IUNIT(N+1)
7003  CALL GOUT(OUT,66)
      CALL MOLDIT
C-----------------------------------------------------------
      GO TO 11
      END
```

```
      SUBROUTINE MATINV(B,N)
C
C     MATRIX INVERSION
C           B        : INPUT,MATRIX TO BE INVERTED;OUTPUT,INVERTED MATRIX
C           N        : NUMBER OF ROWS
C
      DIMENSION B(9,9)
      DOUBLE PRECISION A(9,9),PIVOT(9),AMAX,T,SWAP
      DIMENSION IPIVOT(9),INDEX(9,2)
      EQUIVALENCE (IROW,JROW), (ICOLUM,JCOLUM), (AMAX, T, SWAP)
      DO 770 I=1,N
      DO 770 J=1,N
  770 A(I,J)=B(I,J)
C
C     INITIALIZATION
C
      DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1) 60, 105, 60
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80, 100, 730
   80 IF(DABS(AMAX)-DABS(A(J,K))) 85,100,100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM) 140, 260, 140
  140 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT(I)=A(ICOLUM,ICOLUM)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0D
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
```

```
      630 JROW=INDEX(L,1)
      640 JCOLUM=INDEX(L,2)
      650 DO 705 K=1,N
      660 SWAP=A(K,JROW)
      670 A(K,JROW)=A(K,JCOLUM)
      700 A(K,JCOLUM)=SWAP
      705 CONTINUE
      710 CONTINUE
      730 CONTINUE
          DO 780 I=1,N
          DO 780 J=1,N
      780 B(I,J)=A(I,J)
          RETURN
          END
```

```
      SUBROUTINE MENU
      LOGICAL ITAPE,DEMAND
      COMMON/TOL/ITOL,ITAPE
      COMMON/CCC/GRAPH,JCUT,DEMAND
      COMMON/IUSE/IUSE
      COMMON/NTIME/NTIME
      IF(.NOT.DEMAND)GO TO 2
C-----------------------------AREA MODE-----------------------------
1     ITAPE=.FALSE.
      CALL AREAIN
C-----------------------------EXECUTE MODE--------------------------
      IF(IUSE.EQ.1)GO TO 3
C-----------------------------PRINT MODE----------------------------
      IF(IUSE.EQ.4)CALL LLPRT
C-----------------------------SELECT-DISPLAY-EDIT-SAVE-MODE---------
      IF(IUSE.NE.2)GO TO 1
      CALL ELMENU
      CALL AINIT
      GO TO 1
C-----------------------------EXECUTE BATCH/DEMAND------------------
2     IUSE=1
      CALL AINIT
      RETURN
C-----------------------ERROR IF TRY RUNNING WITHOUT INPUT----------
3     IF(NTIME.NE.1)RETURN
      GO TO 1
      END
```

```
      SUBROUTINE MINV(B,N)
C     MATRIX INVERSION
      DOUBLE PRECISION A(9,9),PIVOT(9),AMAX,T,SWAP
      DIMENSION IPIVOT(9),B(9,9),INDEX(9,2)
      EQUIVALENCE (IROW,JROW),(ICOLUM,JCOLUM),(AMAX,T,SWAP)
      DO 770 I=1,N
      DO 770 J=1,N
  770 A(I,J)=B(I,J)
C
C     INITIALIZATION
C
      DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J)-1) 60, 105, 60
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80, 100, 730
   80 IF(DABS(AMAX)-DABS(A(J,K))) 85,100,100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF (IROW-ICOLUM) 140, 260, 140
  140 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT(I)=A(ICOLUM,ICOLUM)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.D0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1-ICOLUM) 400, 550, 400
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1)-INDEX(L,2)) 630, 710, 630
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  705 A(K,JCOLUM)=SWAP
```

```
2          69.      705 CONTINUE
1          70.      710 CONTINUE
           71.      730 CONTINUE
           72.          DO 780 I=1,N
1          73.          DO 780 J=1,N
2          74.      780 C(I,J)=A(I,J)
           75.          RETURN
           76.          END
```

PCA63

```
1       @FTN,UW PCA63,PCA63
2       -5,7
3             DIMENSION ATBL(60),PTBL(60),RHOTBL(60),CTBL(60),XT(60),YT(60,3),
4            * Y(3)
5       -14,103
6             DATA N/60/,ATMOS/6H PCA63
7             DATA(ATBL(I),I=1,60)/0.,500.,1000.,1500.,2000.,2500.,3000.,4000.,
8            15000.,6000.,7000.,8000.,9000.,10000.,11000.,12000.,13000.,14000.,
9            21500.,16000.,16500.,17000.,18000.,19000.,20000.,22000.,24000.,
10           32600.,28000.,30000.,32000.,34000.,36000.,38000.,40000.,44000.,
11           44000.,52000.,57000.,62000.,67000.,72000.,78000.,80000.,83000.,
12           84000.,86000.,89000.,90000.,93000.,96000.,98000.,100000.,103000.,
13           61050000.,110000.,114000.,120000.,124000.,130000./
14            DATA(PTBL(I),I=1,60)/10.27,9.6487231,9.0598171,8.5018719,
15           17.9735277,7.4734710,7.0004350,6.1305847,5.3547301,4.6646548,
16           24.0552689,3.5107186,3.0333981,2.6141402,2.2471262,1.9269158,
17           31.6464578,1.4071002,1.1985998,1.0191326,.93920513,.86531102,
18           4.73453635,.62346669,.52929926,.38218424,.27700549,.20160165,
19           5.14748703,.10832073,.07995773,.05931480,.04411646,.03303959,
20           6.02490124,.01439423,8.5065841E-03,5.0981116E-03,2.6690495E-03,
21           71.3722539E-3,6.9297552E-4,3.4146309E-4,1.4385223E-4,1.0599136E-4,
22           86.6650716E-5,5.7106018E-5,4.1339397E-5,2.4955273E-5,.20947456E-04,
23           9 .12240884E-04,.73371013E-05,.52841488E-05,.38426987E-05,
24           1 .24306538E-05,.18315352E-05,.94177632E-06,.58044858E-06,
25           2 .32246879E-06,.23275491E-06,.15686446E-06/
26            DATA(RHOTBL(I),I=1,60)/1.300948,1.2335057,1.1689434,1.1072064,
27           11.0481959,.99181543,.93797119,.83752835,.74616589,.66278387,
28           2.5870627,.51842328,.45620410,.40002236,.34951847,.30436223,
29           3.26421523,.22869238,.19732939,.16953508,.15675060,.14460154,
30           4.12263195,.10417607,.08822921,.06314266,.04516896,.03239635,
31           5.02324539,.01691070,.01230191,8.9853973E-3,6.5724525E-3,
32           64.8153175E-3,3.5523605E-3,1.9658486E-3,1.1187134E-3,6.6395879E-4,
33           73.5713458E-4,1.8679223E-4,9.8073482E-5,4.9815655E-5,2.1701813E-5,
34           81.6231229E-5,1.0700285E-5,9.2763901E-6,6.9222450E-6,4.3857098E-6,
35           9 .27453175E-05,.20847472E-05,.11929696E-05,.83398208E-06,
36           1 .58526755E-06,.34909519E-06,.25103905E-06,.11670334E-06,
37           2 .63217998E-07,.28965425E-07,.17063697E-07,.90170 9E-08/
38            DATA(CTBL(I),I=1,60)/332.43856,330.92403,329.40253,327.87397,
39           1326.33826,324.79528,323.24493,320.12172,316.96774,313.93309,
40           2310.85280,307.87839, 305.08411,302.48192,300.04569,297.73722,
41           3295.53345,293.45514,291.59633,290.15454,289.68724,289.46291,
42           4289.31583,289.44289,289.81148,291.13421,293.00379,295.16011,
43           5297.39355,299.57159,301.66304,303.76093,306.55174,309.92756,
44           6313.26704,319.84138,326.28328,327.87397,323.46687,318.99888,
45           7314.46741,309.86969,304.26072,302.36795,295.81768,293.60179,
46           8289.11906,282.26151,279.93832,286.82681,293.55374,297.59399,
47           93022.29020,312.86789,319.72534,336.25767,361.13665,395.53626,
48           1437.21613,493.19276/
```

```
      @FTN,UW FPA63,PPA63
      -5,7
            DIMENSION ATBL(60),PTBL(60),RHOTBL(60),CTBL(60),XT(60),YT(60,3),
           * Y(3)
      -14,103
            DATA N/60/,ATMOS/6H FPA63
            DATA(ATBL(I),I=1,60)/0.,500.,1000.,1500.,2000.,2500.,3000.,4000.,
           1500C.,6000.,7000.,8000.,9000.,10000.,11000.,12000.,13000.,14000.,
           215000.,16000.,16500.,17000.,18000.,19000.,20000.,22000.,24000.,
           326000.,28000.,30000.,32000.,34000.,36000.,38000.,40000.,44000.,
           448000.,52000.,57000.,62000.,67000.,72000.,78000.,80000.,83000.,
           584000.,86000.,89000.,90000.,93000.,96000.,98000.,100000.,103000.,
           6105000.,110000.,114000.,120000.,124000.,130000./
            DATA(PTBL(I),I=1,60)/10.10,9.5554384,9.0346817,8.5369369,
           18.0614294,7.6074017,7.1741153,6.3668963,5.6342022,4.9770728E,
           24.3714321,3.8319199,3.3464479,2.9119117,2.5238402,2.178E131,
           31.8705877,1.5983606,1.3583010,1.1475476,1.0530908,.96617676,
           4.81349526,.68712577,.58222915,.42201626,.30875086,.22793980,
           5.16572565,.12732124,.096198730,.073279037,.056145477,.043294535,
           6 .33570537E-01,.25577717E-01,.12777672E-01,.80399931E-02,
           7 .42509454E-02,.22383559E-02,.10846496E-02,.48844814E-03,
           8 .16777992E-03,.11290073E-03,.61855316E-04,.50659180E-04,
           9 .32522195E-04,.18634756E-04,.15134811E-04,.88441608E-05,
           1 .83611283E-05,.38178816E-05,.27763888E-05,.17620961E-05,
           2 .13233024E-05,.68044276E-06,.42516E-06,.23370542E-06,
           3 .16815664E-06,.11336136E-06/
            DATA(RHOTBL(I),I=1,60)/1.13537,1.0860059,1.0382755,.99214261,
           1.94757068,.90452383,.86296655,.78418111,.71093516,.64297218,
           2.58001870,.52182367,.46813915,.41872392,.37334363,.33249312,
           3.29459874,.26068215,.22936894,.20088866,.18489258,.16985542,
           4.14173245,.11724374,.097358475,.068172810,.048447550,.034775536,
           5.25159219E-01,.18405121E-01,.13546524E-01,.10065735E-01,
           6.75227375E-02,.56749344E-02,.43068771E-02,.25237846E-02,
           7.15109100E-02,.96910228E-03,.56608540E-03,.31631672E-03,
           8.16770816E-03,.83469629E-04,.32638311E-04,
           9 .33151394E-04,.12674332E-04,.10385513E-04,.69113593E-05,
           1 .38294928E-05,.31070709E-05,.17294816E-05,.98967373E-06,
           2 .69186170E-06,.48875962E-06,.28960520E-06,.20825900E-06,
           3 .56815750E-07,.60640350E-07,.24032692E-07,.14155840E-07,.74970200E-08/
            DATA(CTBL(I),I=1,60)/352.9034,350.97257,349.03106,347.07869,
           1345.11527,343.14062,341.15453,337.14726,333.09181,328.98636,
           2324.62903,320.61379,316.35049,312.02488,307.63837,302.83277,
           3297.54967,292.98520,287.93514,282.79491,282.40316,282.19925,
           4283.50456,286.44251,289.35063,294.39916,298.69059,302.92123,
           5307.09359,311.21002,315.27271,319.28370,323.24493,326.81155,
           6330.33967,337.28520,344.09056,340.79560,328.02715,314.74114,
           7300.66901,286.32557,261.83305,261.37829,261.37829,261.37829,
           8261.37829,261.37829,261.37829,261.37829,261.37829,261.37829,
           9261.37829,261.37829,261.37829,261.37829,261.37829,261.37829,
           1261.37829,261.37829/
```

PRA63

```
      SUBROUTINE PRA63(PR,ATMOSN,ERROR)
      CHARACTER*6 ATMOS,ATMOSN
      COMMON/MEAST/LAST,M,K
      DIMENSION S(9),SM1(9),SP1(9),A(10),B(10),C(10)
      DIMENSION ATBL(89),PTBL(89),RHOTBL(89),CTBL(89)
      DIMENSION  XT(89)
      DIMENSION YT(89,3),Y(3)
      EQUIVALENCE (ATBL,XT)
      EQUIVALENCE (PTBL,YT),(RHOTBL,YT(1,2)),(CTBL,YT(1,3))
      DIMENSION PR(1)
C PR(1)=ALT  PR(2)=PRES  PR(3)=DPDH  PR(4)=DRDH  PR(5)=DSDH  PR(6)=RHO
C     PR(9)=SPDS PR(10)=VMU PR(11)=DMUH
      DATA L/3/
      DATA N/89/,ATMOS/6H PRA63/
      DATA(ATBL(I),I=1,89)/0.,500.,1000.,1500.,2000.,2500.,3000.,4000.,
     15000.,6000.,7000.,8000.,9000.,10000.,11000.,12000.,13000.,14000.,
     215000.,16000.,16500.,17000.,18000.,19000.,20000.,22000.,24000.,
     326000.,28000.,30000.,32000.,34000.,36000.,38000.,40000.,44000.,
     448000.,52000.,57000.,62000.,67000.,72000.,
     *78000.,80000.,83000.,
     *84000.,86000.,89000.,90000.,93000.,96000.,98000.,
     *100000.,103000.,
     *                105000.,110000.,114000.,120000.,124000.,130000.,
     *134000.,140000.,144000.,150000.,160000.,175000.,190000.,210000.,
     6                220000.,240000.,260000.,280000.,300000.,326000.,
     735000.,376000.,400000.,426000.,450000.,476000.,500000.,526000.,
     855000.,576000.,600000.,626000.,650000.,676000.,700000./
C     EASTERN TEST RANGE DATA
      DATA(PTBL(I),I=1,44)/
     * .10170147E02, .96022651E01, .90603417E01, .85438572E01,
     * .80521166E01, .75843000E01, .71395062E01, .63151744E01,
     * .55714346E01, .49008910E01, .42967958E01, .37532038E01,
     * .32649867E01, .28277554E01, .24373143E01, .20909280E01,
     * .17861066E01, .15199026E01, .12892855E01, .10911841E01,
     * .10073656E01, .92252636E00, .78097360E00, .66260085E00,
     * .56315646E00, .40899187E00, .29918756E00, .22038158E00,
     * .16327365E00, .12146274E00, .9090508E-01, .6842991E-01,
     * .5180718E-01, .3944799E-01, .3020518E-01, .1800451E-01,
     * .1091056E-01, .6639319E-02, .3497353E-02, .1781846E-02,
     * .8709643E-03, .4057600E-03, .1511983E-03, .1068430E-03/
      DATA(PTBL(I),I=45,89)/
     * .6235580E-04, .5187821E-04, .3591471E-04, .2069615E-04,
     * .1722443E-04, .1006525E-04, .6033042E-05, .4344971E-05,
     * .3155717E-05, .2005384E-05, .1506007E-05, .7743898E-06,
     * .4838607E-06, .2659771E-06, .1914191E-06, .1289841E-06,
     * .1036962E-06, .7834591E-07, .6656633E-07, .5358494E-07,
     * .3912649E-07, .2594138E-07, .1787156E-07, .1130478E-07,
     * .9106353E-08, .6040650E-08, .4100777E-08, .2841337E-08,
     * .2005731E-08, .1304728E-08, .8946879E-09, .6062792E-09,
     * .4304566E-09, .3014289E-09, .2195289E-09, .1576117E-09,
     * .1173154E-09, .8597494E-10, .6498292E-10, .4833493E-10,
     * .3701989E-10, .2789144E-10, .2157540E-10, .1641584E-10,
     * .1281153E-10/
      DATA(RHOTBL(I),I=1,44)/
     * .11835467E01, .11312044E01, .10793462E01, .10284922E01,
     * .97902799E00, .93122445E00, .88525680E00, .79915661E00,
     * .72084273E00, .64983432E00, .58535150E00, .52651816E00,
     * .47245380E00, .42255460E00, .37638426E00, .33302118E00,
     * .29232217E00, .25432636E00, .21920325E00, .18717684E00,
     * .17239239E00, .15845601E00, .13239217E00, .11096236E00,
     * .9319379E-01, .6619324E-01, .4747889E-01, .3438248E-01,
     * .25119032E-1, .1833406E-01, .1345779E-01, .9930103E-02,
     * .7365417E-02, .5453419E-02, .4122020E-02, .2368456E-02,
     * .1401576E-02, .8652672E-03, .4766351E-03, .2574523E-03,
     * .1343447E-03, .6654933E-04, .2706738E-04, .1967746E-04/
      DATA(RHOTBL(I),I=45,89)/
     * .1202377E-04, .1000425E-04, .6925835E-05, .3991071E-05,
     * .3321580E-05, .1848883E-05, .1057999E-05, .7396272E-06,
```

```
     * .5225459E-06, .1055993E-06, .2226370E-06, .1034998E-06,
     * .5606565E-07, .2569189E-07, .1513314E-07, .8014608E-08,
     * .5638713E-08, .3568138E-08, .2758526E-08, .1943190E-08,
     * .1227308E-08, .7254957E-09, .4609539E-09, .2714797E-09,
     * .2112589E-09, .1322960E-09, .8551031E-10, .5659070E-10,
     * .3816858E-10, .2371704E-10, .1561798E-10, .1014715E-10,
     * .6940371E-11, .4712580E-11, .3338648E-11, .2328286E-11,
     * .1688344E-11, .1215120E-11, .9034760E-12, .6603653E-12,
     * .4976106E-12, .3709640E-12, .2840554E-12, .2138449E-12,
     * .1652610E-12/
      DATA(CTBL(I),I=1,44)/
     * .3468575E03, .3447410E03, .3428197E03, .3410352E03,
     * .3393366E03, .3376800E03, .3360284E03, .3326258E03,
     * .3289594E03, .3249468E03, .3205196E03, .3155102E03,
     * .3110283E03, .3060973E03, .3011537E03, .2965458E03,
     * .2925000E03, .2852283E03, .2869052E03, .2856524E03,
     * .2854472E03, .2855232E03, .2872386E03, .2889707E03,
     * .2907518E03, .2941997E03, .2972150E03, .2996612E03,
     * .3016619E03, .3045482E03, .3075183E03, .3106061E03,
     * .3138052E03, .3170698E03, .3203157E03, .3262285E03,
     * .3301257E03, .3217559E03, .3205093E03, .3112796E03,
     * .3012665E03, .2912898E03, .2796496E03, .2757100E03/
      DATA( CTBL(I),I=45,89)/
     * .2694523E03, .2694412E03, .2694412E03, .2694412E03,
     * .2694412E03, .2760714E03, .2825461E03, .2867813E03,
     * .2909549E03, .3011360E03, .3077263E03, .3236487E03,
     * .3475967E03, .3807045E03, .4208157E03, .4746691E03,
     * .5074058E03, .5528884E03, .5812362E03, .6213376E03,
     * .6680879E03, .7015270E03, .7367437E03, .7635304E03,
     * .7765774E03, .1965255E03, .8193845E03, .8387235E03,
     * .8577243E03, .8775943E03, .8955446E03, .9145933E03,
     * .9318312E03, .9462960E03, .9594545E03, .9735090E03,
     * .9861045E03, .9952685E03, .1003472E04, .1012284E04,
     * .1020350E04, .1025967E04, .1031124E04, .1036832E04,
     * .1041787E04/
C
C*****START SPLINE CALCULATION*****
      IF(M.LT.1.OR.M.GT.NIM)1
      X=FR(1)
      IF(X.LT.0.) X=0.
   98 IF(X.LT.ATBL(N)) GO TO 100
      Y(1)=FTBL(N)
      Y(2)=RHOTBL(N)
      Y(3)=CTBL(N)
   99 FR(3)=0.
      FR(4)=0.
      FR(5)=0.
      GO TO 15
  100 CONTINUE
      IF(X-XT(M).GT.0.)GO TO 2
    1 IF(M.EQ.1)GO TO 4
      P=M-1
      IF(X-XT(M))1,4,4
    2 IF(M.EQ.N)GO TO 5
      P=M+1
      IF(X- XT(M).GT.0.)GO TO 2
    5 P=M-1
    4 MP1=M+1
      DX=X-XT(M)
      DLX=XT(MP1)-XT(P)
      DLXI=1./DLX
      R=DX*DLXI
      R2=R**
      IF(M.EQ.LAST)GO TO 34
      MM1=M-1
      MP2=MP1+1
      DO 6 I=1,L
    6 S(I)=(YT(MP1,I)-YT(M, I))*DLXI
```

```
      IF(M.LT.N-1)GO TO 31
      DLXMI=1./(XT(M)-XT(MM1))
      DO 7 I=1,L
      SM1(I)=(YT(M,I)-YT(MM1,I))*DLXMI
    7 SP1(I)=2.*S(I)-SM1(I)
      GO TO 33
   31 CONTINUE
      DLXPI=1./(XT(MP2)-XT(MP1))
      DO 8 I=1,L
    8 SP1(I)=(YT(MP2,I)-YT(MP1,I))*DLXPI
      IF(M.EQ.1)GO TO 32
      DO 9 I=1,L
    9 SM1(I)=2.*S(I)-SP1(I)
      GO TO 33
   32 CONTINUE
      DLXMI=1./(XT(M)-XT(MM1))
      DO 10 I=1,L
   10 SM1(I)=(YT(M,I)-YT(MM1,I))*DLXMI
   33 CONTINUE
      DO 11 I=1,L
      A(I)=.5*(SP1(I)+SM1(I))-S(I)
      B(I)=.5*(S(I)-SP1(I))+S(I)-SM1(I)
   11 C(I)=.5*(S(I)+SM1(I))
   34 IF(K.EQ.2)GO TO 13
      DO 12 I=1,L
   12 Y(I)=YT(M,I)+(A(I)*R2+B(I)*R+C(I))*DX
      GO TO 15
   13 DO 14 I=1,L
      Y(I)=YT(M,I)+(A(I)*R2+B(I)*R+C(I))*DX
   14 FR(I+2)=3.*A(I)*R2+2.*B(I)*R+C(I)
   15 CONTINUE
      LAST=M
C*****END OF SPLINE CALCULATION*****
C
      FR(2)=Y(1)
      FR(6)=Y(2)
      FR(9)=Y(3)
      TN=(Y(3)/20.0467C7)**2
      FR(10)=1.458E-06*(TN**1.5)/(TN+110.4)
      IF(K.EQ.2)PR(11)=FR(10)/Y(3)*(3.-2.*TN/(TN+110.4))*PR(5)
      FR(12)=TN
      IF(K.EQ.2) PR(13)=2.*TN/Y(3)*PR(5)
      ATMOSN=ATMOS
      RETURN
      END
```

```
      SUBROUTINE RTMRK
C
      COMMON /DES1/
     *  KR(1),  NPX,  NEQ, NOLD,    M,  MP1,  MP2,   NN, KIND,  KON,
     *     IFL, NTRG, NIVT, NOVT,  NTR,NSTART,NBCIF,  NNT,  NTT,IOL(11),
     *   ICL2, JJSS, IOCO,J2000,J4000,J1000, IRST, ECST,   PO,  SMA,
     *  ERROR,REDCT,STEPNG,
     *  EU(1),   EL,  HPX,  HMN,   YL,    A, ENF1,    D,    B,  SKP,
     *     SK,   PC, DELU, DUSC, HOLD,  VAR,   2V,   HM,   HP,    R,
     *      Q, DELT,  HPU,  FMU,CX(10),CQ(10),XL(11), RJ(11), BJ(11)
      KR(1)=2
C
      RETURN
      END
```

```
$DATA

      BLOCK DATA
      COMMON/ASTOR/A(1,67)
C
C     DATA FOR SUMMARY TABLE
C           BASED ON MISSION 1 DATA FROM JSC APPENDIX 10.12 UPDATE 31
C
      DATA(A(I, 1),I=1,3)/0.,0.,0./
      DATA(A(I, 2),I=1,3)/0.,0.,0./
      DATA(A(I, 3),I=1,3)/1608.,1608.,1608./
      DATA(A(I, 4),I=1,3)/2644.,2644.,2644./
      DATA(A(I, 5),I=1,3)/140821.,140821.,140821./
      DATA(A(I, 6),I=1,3)/4171.,4171.,4171./
      DATA(A(I, 7),I=1,3)/0.,0.,0./
      DATA(A(I, 8),I=1,3)/898.,568.,568./
      DATA(A(I, 9),I=1,3)/567.,567.,567./
      DATA(A(I,10),I=1,3)/0.,0.,0./
      DATA(A(I,11),I=1,3)/0.,0.,0./
      DATA(A(I,12),I=1,3)/1238.,289.,289./
      DATA(A(I,13),I=1,3)/891.,891.,891./
      DATA(A(I,14),I=1,3)/20484.,20484.,20484./
      DATA(A(I,15),I=1,3)/0.,0.,0./
      DATA(A(I,16),I=1,3)/0.,0.,0./
      DATA(A(I,17),I=1,3)/0.,0.,0./
      DATA(A(I,18),I=1,3)/1337.,2268.,2268./
      DATA(A(I,19),I=1,3)/1524.,1524.,1524./
      DATA(A(I,20),I=1,3)/0.,0.,0./
      DATA(A(I,21),I=1,3)/0.,0.,0./
      DATA(A(I,22),I=1,3)/0.,0.,0./
      DATA(A(I,23),I=1,3)/211.,211.,211./
      DATA(A(I,24),I=1,3)/70990.,70990.,70990./
      DATA(A(I,25),I=1,3)/4285.,4439.,4439./
      DATA(A(I,26),I=1,3)/0.,0.,0./
      DATA(A(I,27),I=1,3)/1047.,1100.,1100./
      DATA(A(I,28),I=1,3)/0.,0.,0./
      DATA(A(I,29),I=1,3)/183.,122.,122./
      DATA(A(I,30),I=1,3)/0.,0.,0./
      DATA(A(I,31),I=1,3)/0.,0.,0./
      DATA(A(I,32),I=1,3)/0.,0.,0./
      DATA(A(I,33),I=1,3)/0.,0.,0./
      DATA(A(I,34),I=1,3)/0.,0.,0./
      DATA(A(I,35),I=1,3)/0.,0.,0./
      DATA(A(I,36),I=1,3)/0.,61.,61./
      DATA A(1,37)/0./
      DATA A(1,38)/0./
      DATA A(1,39)/0./
      DATA A(1,40)/0./
      DATA A(1,41)/0./
      DATA A(1,42)/0./
      DATA A(1,43)/0./
      DATA A(1,44)/0./
      DATA A(1,45)/0./
      DATA A(1,46)/0./
      DATA A(1,47)/354130./
      DATA A(1,48)/0./
      DATA A(1,49)/0./
      DATA A(1,50)/0./
      DATA A(1,51)/0./
      DATA A(1,52)/0./
      DATA A(1,53)/0./
      DATA A(1,54)/0./
      DATA A(1,55)/0./
      DATA A(1,56)/0./
      DATA A(1,57)/0./
      DATA A(1,58)/0./
      DATA A(1,59)/0./
      DATA A(1,60)/0./
      DATA A(1,61)/4./
      DATA A(1,62)/0./
```

```
      DATA A(1,63)/0./
      DATA A(1,64)/12885./
      DATA A(1,65)/8456./
      DATA A(1,66)/51./
      DATA A(1,67)/0./
C
      END
```

SPLINE

```
      SUBROUTINE SPLINE(L,M,N,X,XT,Y,YT,YD,K,LAST,A,B,C)
C
C     THIRD ORDER SPLINE INTERPOLATION ROUTINE
C           L       : NUMBER OF DEPENDENT VARIABLES
C           M       : PLACE IN THE ARRAY(SET=1 INITIALLY;UPDATED BY ROUTIN
C           N       : NUMBER OF POINTS IN INDEPENDENT TABLE
C           X       : VALUE OF INDEPENDENT VARIABLE
C           XT      : INDEPENDENT VARIABLE TABLE
C           Y       : VALUE(S) OF DEPENDENT VARIABLE(S)
C           YT      : DEPENDENT VARIABLE TABLE(S)
C           YD      : PARTIAL DERIVATIVES OF DEPENDENT WRT INDEPENDENT
C           K       : 1,DO NOT CALCULATES PARTIALS;2,CALCULATE PARTIALS
C           LAST    : PREVIOUS PLACE IN ARRAY(SET=0 INITIALLY;UPDATED BY R
C           A       : STORAGE ARRAY FOR SPLINE COEFFICIENTS
C           B       : STORAGE ARRAY FOR SPLINE COEFFICIENTS
C           C       : STORAGE ARRAY FOR SPLINE COEFFICIENTS
C
      DIMENSION XT(1),YT(N,L),S(18),SM1(18),SP1(18),YD(1),Y(1),A(1),
     *B(1),C(1)
      IF(M.LT.1.OR.M.GT.N)M=1
      IF(X-XT(M).GT.0.)GO TO 2
    1 IF(M.EQ.1)GO TO 4
      M=M-1
      IF(X-XT(M))1,4,4
    2 IF(M.EQ.N)GO TO 5
      M=M+1
      IF(X- XT(M).GT.0.)GO TO 2
    5 M=M-1
    4 MP1=M+1
      DX=X-XT(M)
      DLX=XT(MP1)-XT(M)
      DLXI=1./DLX
      R=DX*DLXI
      R2=R*R
      IF(M.EQ.LAST)GO TO 34
      MM1=M-1
      MP2=MP1+1
      DO 6 I=1,L
    6 S(I)=(YT(MP1,I)-YT(M, I))*DLXI
      IF(M.LT.N-1)GO TO 31
      DLXMI=1./(XT(M)-XT(MM1))
      DO 7 I=1,L
      SM1(I)=(YT(M,I)-YT(MM1,I))*DLXMI
    7 SP1(I)=2.*S(I)-SM1(I)
      GO TO 33
   31 CONTINUE
      DLXPI=1./(XT(MP2)-XT(MP1))
      DO 8 I=1,L
    8 SP1(I)=(YT(MP2,I)-YT(MP1,I))*DLXPI
      IF(M.EQ.1)GO TO 32
      DO 9 I=1,L
    9 SM1(I)=2.*S(I)-SP1(I)
      GO TO 33
   32 CONTINUE
      DLXMI=1./(XT(M)-XT(MM1))
      DO 10 I=1,L
   10 SM1(I)=(YT(M,I)-YT(MM1,I))*DLXMI
   33 CONTINUE
      DO 11 I=1,L
      A(I)=.5*(SP1(I)+SM1(I))-S(I)
      B(I)=.5*(S(I)-SP1(I))+S(I)-SM1(I)
      C(I)=.5*(S(I)+SM1(I))
   34 IF(K.EQ.2)GO TO 13
      DO 12 I=1,L
   12 Y(I)=YT(M,I)+(A(I)*R2+B(I)*R+C(I))*DX
      LAST=M
```

```
      RETURN
      DO 14 I=1,L
      Y(I)=YT(M,I)+(A(I)*R2+B(I)*R+C(I))*DX
   14 YD(I)=3.*A(I)*R2+2.*B(I)*R+C(I)
      LAST=P
      RETURN
      END
```

```
      SUBROUTINE SUMARY
      COMMON/CCC/GRAPH,JC,COMANC
      LOGICAL LBM,LUMF
      CHARACTER*42 NAME(67),NTRA(2)
      CHARACTER*6 ASTR(2),NBK(4),NBK1(4),NBK2(4),NBK3(4),NBK4(4),NBK5(4)
     *,NFOR(18),C(10),Z(5),NA(2),NB,NSTOP,NAK1,NAK2,NAK,MISICN,MISSON
      CHARACTER*60 HEAC
      REAL LBMTNK
      DIMENSION NORD(67),NUMB(67),B(129),PRCTET(3),PRCTOR(3)
      COMMON/AGEN1/TIME(2,16),TAUT(15),TAUW(15),TZERC,TLIFT,TTILT,TMINH,
     *ET2,TG,TL,XMAUG,TNE(6,15),ENGDAT(8,16),S(15),WO(15),WJET(15),
     *HEAD,FRINT(15),STEF(15),HNOM(15),HMAX(15),FROP(6),TBEGR,TENDR,
     *(HRDOT,CHIRO,FAZ,B(13,2),CPAXIL,TCMIR,VRCUT,FPRFAC,CHVEL
      COMMON/AGEN3/ITER,IWD,IXR,LUM ,KAT,KCYTAB,KFAGE,KS1,KWTA(4),LINES,
     *ABGCT(4),NENCCT(4),NMAX,NOEVNT(5),NOWD(15),NVNT,NWVNT,IMEAC,
     *PINH,PPSFLG(15),NSYST(15),ICONSW,IRTLS,IPOLY,KINDB,KRDERB,MISSON,
     *IRTFLE,NROLL,NGEAST,IFACT,NOBASE,LSTGE(15),NSWCH(15)
      COMMON/ALAT/ALAT,ALONG,ALTLS,NTABLE
      COMMON/ASTOR/A(4,67)
      COMMON/CONST/RAD,PI,RE,FLAT,CJ,H,DU,CMUE,OMEGA,ALT1,ALT2,PSL,
     *EZERO,FTN,PTKG,FSFTM,PFN,PFKG,PSFFP
      COMMON/LONOFF/LBP
      COMMON/OMSRCS/FOMS(15),FRCS(15),WOOMS(15),WRCS(15),WDCUMP(15)
      EQUIVALENCE(T,B(1)),(XM,B(6)),(DWPFS,B(67)),(CLSRM,B(68)),
     *(DWLBP,B(69)),(CLOMS,B(70)),(DWRCS,B(71)),(CWDLMP,B(125)),
     *(MISICN,NB)
      DATA NUMB/3,3,3,2,3,3,4,3,3,4,4,3,3,3,2,2,2,3,3,3,3,3,3,3,3,3,3,4,
     *3,2,0,2,3,3,3,3,1,0,1,1,1,0,0,1,1,1,1,1,1,0,0,1,1,1,1,0,1,1,1,1,1,
     *1,1,1,1,1,1/
      DATA NORD/1,1,1,1,1,1,6,1,1,6,2,1,1,1,5,1,1,1,1,1,1,5,1,1,1,1,3,4,
     *1,5,0,1,1,1,1,1,5,0,1,5,1,0,0,1,1,5,1,1,1,0,0,1,1,1,5,0,1,1,5,1,1,
     *1,5,1,1,1,5/
      DATA C/6HIGNITI,6HLIFT-O,6HLBM IG,6HSRB SE,6HLBP SE,6HRTLS/A,
     *6HINJECT,6HEADPRI,6HINTERM,6H      /
      DATA ,/6HNOM ,6HAOA ,6HRTLS ,6HPOST N,6HPCST A/
      DATA PRCTET/.598,.618,.553/
      DATA PRCTOR/.402,.382,.447/
      DATA NAME( 1)/42HPAYLOAD                                          /
      DATA NAME( 2)/42HOPS KIT                                          /
      DATA NAME( 3)/42HMISSION KIT                                      /
      DATA NAME( 4)/42HPERSONNEL                                        /
      DATA NAME( 5)/42HORBITER - EMPTY                                  /
      DATA NAME( 6)/42H          - NONPROPELLSIVE CONSUMABLES           /
      DATA NAME( 7)/42H          - USABLE RCS (RCS LOADED)              /
      DATA NAME( 8)/42H          - RCS RESERVES                         /
      DATA NAME( 9)/42H          - RCS RESIDUALS                        /
      DATA NAME(10)/42H          - USABLE OMS (OMS LOADED)              /
      DATA NAME(11)/42H          - OMS PROPELLANT (DEORBIT)             /
      DATA NAME(12)/42H          - OMS RESERVES                         /
      DATA NAME(13)/42H          - OMS RESIDUALS                        /
      DATA NAME(14)/42HSSME EMPTY                                       /
      DATA NAME(15)/42HORBITER AT ORBIT INSERTION                       /
      DATA NAME(16)/42HMECO OMS MANEUVER - OMS PROPELLANT               /
      DATA NAME(17)/42H                  - RCS PROPELLANT               /
      DATA NAME(18)/42HTRAPPED MPS PROPELLANT - ORBITER                 /
      DATA NAME(19)/42H                                   - SSME
      DATA NAME(20)/42HFLIGHT PERFORMANCE RESERVE - ORBITER
      DATA NAME(21)/42H                                   - SSME
      DATA NAME(22)/42HORBITER AT OMS IGNITION
      DATA NAME(23)/42HRCS PROPELLANT
      DATA NAME(24)/42HEXTERNAL TANK - EMPTY
      DATA NAME(25)/42H                RESIDUALS
      DATA NAME(26)/42HFLIGHT PERFORMANCE RESERVE - ET
      DATA NAME(27)/42HFUEL BIAS
      DATA NAME(28)/42HEXCESS MPS PROPELLANT (MPS PROP LOADED)
      DATA NAME(29)/42HEXPELLED MPS PROPELLANT
      DATA NAME(30)/42HINJECTED WEIGHT AT MECO
      DATA NAME(31)/42HPROPELLANT BURNED-AOA/RTLS TO MECO
```

```
      DATA NAME(32)/42H   MPS
      DATA NAME(33)/42H   OMS
      DATA NAME(34)/42H   RCS
      DATA NAME(35)/42HOMS PROPELLANT DUMPED (RCS BURN)
      DATA NAME(36)/42HEXPELLED MPS PROPELLANT
      DATA NAME(37)/42HWEIGHT AT AOA/RTLS
      DATA NAME(38)/42HMPS PROPELLANT BURNED-LBM SEPARATION
      DATA NAME(39)/42H   TO RTLS/AOA
      DATA NAME(40)/42HWEIGHT AT LBM SEPARATION
      DATA NAME(41)/42HLEM INERT WEIGHT
      DATA NAME(42)/42HPROPELLANT BURNED-SRM SEPARATION TO
      DATA NAME(43)/42H   LBM SEPARATION
      DATA NAME(44)/42H   MPS
      DATA NAME(45)/42H   LBM
      DATA NAME(46)/42HE1+ORBITER AFTER SRM SEPARATION
      DATA NAME(47)/42HSRM INERT WEIGHT
      DATA NAME(48)/42HSRM RESIDUAL PROPELLANT
      DATA NAME(49)/42HINERTS CONSUMED
      DATA NAME(50)/42HPROPELLANT BURNED-LBM IGNITION
      DATA NAME(51)/42H   TO SRM SEPARATION
      DATA NAME(52)/42H   MPS
      DATA NAME(53)/42H   LBM
      DATA NAME(54)/42H   SRM
      DATA NAME(55)/42HLBM IGNITION (T+5 SEC)
      DATA NAME(56)/42HPROPELLANT BURNED-LIFTOFF TO LBM IGNITION
      DATA NAME(57)/42H   MPS
      DATA NAME(58)/42H   SRM
      DATA NAME(59)/42HLIFTOFF (T/W=1.0)
      DATA NAME(60)/42HSRB PROPELLANTS
      DATA NAME(61)/42HSRB INERTS
      DATA NAME(62)/42HMPS PROPELLANT
      DATA NAME(63)/42HSRB IGNITION COMMAND (T-0)
      DATA NAME(64)/42HMPS THRUST BUILDUP + SRM IGNITION DELAY
      DATA NAME(65)/42HOVERFILL,DRAINBACK,AND BOILOFF
      DATA NAME(66)/42HNONPROPULSIVE CONSUMABLES
      DATA NAME(67)/42HPRELAUNCH WEIGHT (T-5 MIN)
      DATA NTRA( 1)/42HMPS PROPELLANT BURNED-SRM SEPARATION
      DATA NTRA( 2)/42HPROPELLANT BURNED - LIFTOFF
      DATA NSTR/6H(1X,J4,6H2      /,NSTOP/6H)      /
      DATA NAK1/6H,F8.0 /,NAK2/6H,8X    /
      DATA NBK1/6H,14)  ,6H       ,6H       ,6H       /
      DATA NBK2/6H,2H  ,6HF5.1,5,6HH F/S),6H,2X    /
      DATA NBK3/6H,3X,6H,6HMARGIN,6H,5X    ,6H       /
      DATA NBK4/6H,1X,2H,6H( ,F8.,6HC,1H),,6H3X    /
      DATA NBK5/6H,6X,F8,6H,8     ,6H       ,6H
      CZERO=9.80665
      FTKG=2.2046223
      ETPRW1=PROP(1)
      LBMTNK=PROP(3)
      CMSTNK=PROP(4)
      RCSTNK=PROP(5)
      N=1
   28 L1=1
      L2=37
      IGLIM=10000.
      P=0
      CMS=0.
      RCS=0.
      M=1
      L=1
      CMSDUP=0.
      SUMP=.FALSE.
      IF(N.EQ.1) REWIND 5
   31 READ(9,END=21,ERR=21)NA,B,NB
      IF(NA(1).EQ.C(8)) GO TO 31
      IF(P.NE.0) GO TO 27
      DO 29 I=1,5
      IF(MISION.EQ.Z(I)) GO TO 30
```

```
25 CONTINUE
30 M=1
27 CONTINUE
   IF(M.GT.3.AND.NA(1).EQ.C(7)) GO TO 45
   IF(M.GT.3) GO TO 31
   IF(NA(1).EQ.C(1)) GO TO 31
   DO 25 K=1,7
   IF(NA(1).EQ.C(K)) GO TO 32
25 CONTINUE
   GO TO 31
32 GO TO (33,35,36,37,38,39,40),K
33 STIME=T
   IF(ABS(T).LE.1E-0.31 GO TO 34
   .UPP= .TRUE.
   L=JUM
   SVPPS=DWMPS
   SVOMS=DWOMS
   SVRCS=DWRCS
   GO TO 31
34 A(1,63)=XM
   GO TO 31
35 A(1,55)=XM
   A(1,62)=DWMPS-A(1,64)-A(1,65)
   A(1,60)=DWSRP-A(1,61)
   A(1,67)=A(1,63)+A(1,64)+A(1,65)+A(1,66)
   SVPPS=DWMPS
   SVSRM=DWSRM
   GO TO 31
36 A(1,56)=XM
   A(1,57)=DWPPS-SVPPS
   A(1,58)=DWSRP-SVSRP
   SVPPS=DWMPS
   SVSRM=DWSRM
   LBM=.TRUE.
   GO TO 31
37 A(1,46)=XM
   A(1,52)=DWMPS-SVPPS
   A(1,53)=DWLBP
   A(1,54)=DWSRP-SVSRP-A(1,49)
   SVPPS=DWMPS
   SVLBM=DWLBP
   SVSRM=DWSRM
   GO TO 31
38 A(1,40)=XM
   A(1,44)=DWPPS-SVPPS
   A(1,45)=DWLBM-SVLBP
   SVPPS=DWMPS
   SVLBM=DWLBM
   GO TO 31
39 A(1,37)=XM
   A(1,35)=DWPPS-SVPPS
   SVPPS=DWMPS
   GO TO 31
40 A(M,30)=XM
   A(M,32)=DWPPS-SVPPS
   A(M,33)=DWOMS-DWCUPP
   A(M,34)=DWRCS
   A(M,35)=DWDUPP
   SVOMS=DWOMS
   SVRCS=DWRCS
   FPR=XP*(1.-E)P(-(HVEL*FPRFAC/(GZERC*FUNISP(1.))))
   A(M,26)=PRCTET(M)*FPR
   A(M,20)=PRCTCR(M)*FPR-A(M,21)
   A(M,28)=ETPRAT-(A(M,18)+A(M,19)+A(M,20)+A(M,21)+A(M,25)+A(M,26)+
  *A(M,27)+A(M,29)+A(M,32)+A(M,36)+A(1,39)+A(1,44)+A(1,52)+A(1,57)+
  *A(1,62)+A(1,64)+A(1,65))
```

```
      A(M,22)=A(M,30)-(A(M,23)+A(M,24)+A(M,25)+A(M,26)+A(M,27)+A(M,28)+
     4A(M,29))
      GO TO 46
   45 M=M-3
      A(M,16)=OWCMS-SVCMS
      A(M,17)=OWRCS-SVRCS
   46 A(M,15)=A(M,22)-(A(M,16)+A(M,17)+A(M,18)+A(M,19)+A(M,20)+A(M,21))
      A(M,10)=OMSTNK-(A(M,11)+A(M,12)+A(M,13)+A(M,16)+A(M,33)+A(M,35))
      A(M,7)=RCSTNK-(A(M,8)+A(M,9)+A(M,17)+A(M,23)+A(M,34))
      A(M,1)=A(M,15)
      DO 47 I=2,14
   47 A(M,1)=A(M,1)-A(M,I)
      A(4,7)=A(3,7)
      A(3,7)=A(2,7)
      A(2,7)=RCSTNK
      A(4,10)=A(3,10)
      A(3,10)=A(2,10)
      A(2,10)=OMSTNK
      A(4,11)=A(3,11)
      A(3,11)=A(2,11)
      A(2,11)=0.
      A(4,28)=A(3,28)
      A(3,28)=A(2,28)*1.06
    1 WRITE(JO,101) HEAD
  101 FORMAT(1H1,36X,A60///49X,7HNOMINAL,17X,3HACA,15X,4HRTLS//)
      NFOR(1)=NSTR(1)
      NFOR(2)=NSTR(2)
      NFOR(18)=NSTOP
      DO 5 K=L1,L2
      IF(LBP) GO TO 52
      IF(K.NE.38) GO TO 49
      NAME(38)=NTRA(1)
   49 IF(K.GE.40.AND.K.LE.45) GO TO 5
      IF(K.NE.50) GO TO 51
      NAME(50)=NTRA(2)
   51 IF(K.EQ.53.OR.(K.GE.55.AND.K.LE.58)) GO TO 5
   52 CONTINUE
      NAK=NAK1
      IF(NORD(K).EQ.0.OR.NORD(K).EQ.5) NAK=NAK2
      NFOR(3)=NAK
      NFOR(8)=NAK
      NFOR(13)=NAK
      IF(K.EQ.17.OR.K.GT.36) NFOR(13)=NAK2
      IF(K.GT.36) NFOR(8)=NAK2
      DO 2 I=1,4
      NBK(I)=NBK1(I)
      IF(NORD(K).EQ.2) NBK(I)=NBK2(I)
      IF(NORD(K).EQ.3) NBK(I)=NBK3(I)
      IF(NORD(K).EQ.4) NBK(I)=NBK4(I)
      IF(NORD(K).EQ.5) NBK(I)=NBK5(I)
      IF(NORD(K).EQ.6) NBK(I)=NBK4(I)
    2 CONTINUE
      DO 3 I=1,4
      J=I+3
      NFOR(J)=NBK(I)
      IF(NORD(K).EQ.3) NFOR(J)=NBK1(I)
      J=I+8
      NFOR(J)=NBK(I)
      IF(K.EQ.9.OR.K.EQ.10.OR.K.GT.36) NFOR(J)=NBK1(I)
      IF(NORD(K).EQ.6) NFOR(J)=NBK1(I)
      J=I+13
      NFOR(J)=NBK(I)
      IF(NORD(K).NE.5.OR.K.GT.36) NFOR(J)=NBK1(I)
    3 CONTINUE
      N=NUMB(K)
      IF(NORD(K).EQ.5.AND.K.NE.L1) WRITE(JO,102)
```

```
      IF(N.EQ.0) GO TO 4
      IF(NORD(K).NE.4) GO TO 7
      WRITE(JO,NFOR) NAME(K),A(1,K),ETPRET,A(2,K),A(3,K),A(4,K)
      GO TO 6
    7 CONTINUE
      WRITE(JO,NFOR) NAME(K),(A(J,K),J=1,N)
      GO TO 6
    4 WRITE(JO,NFOR) NAME(K)
    6 IF(NORD(K).EQ.5) WRITE(JO,102)
    5 CONTINUE
      IF(L1.NE.1) GO TO 20
  102 FORMAT( )
      L1=37
      L2=67
      GO TO 1
   20 CONTINUE
      A(2,7)=A(3,7)
      A(3,7)=A(4,7)
      A(2,10)=A(3,10)
      A(3,10)=A(4,10)
      A(2,11)=A(3,11)
      A(3,11)=A(4,11)
      A(3,28)=A(4,28)
      IF(NTABLE.LE.1.OR.N.EQ.NTABLE) RETURN
      N=N+1
      GO TO 28
   21 BACKSPACE 9
      RETURN
      END
```

```
SYP
 1        LIB MIPS*ILIBFTN,SYS$*GRAPHICS$,SYS$*MSFC$.
 2        SEG    MAIN
 3        IN MAIN
 4         IN SDATA
 5        SEG NEWSEG*,(MAIN)
 6        IN SPLINE,FTMRK,CPIR,DESOLV,ACNTRO,AGEO
 7        SEG    AREAC*,(MAIN)
 8        IN AREAIN
 9        IN LLPRT
10        IN ELMENL
11        IN INITIAL,PAGIT,GOUT-GIN,FOATEX
12        SEG    AFCRLO*,(NEWSEG)
13        IN  AFORLN,ATHREV,BACL)
14        IN CAERO
15        SEG    AINITO*,(NEWSEG)
16        IN AINIT,BCPTBL,GBANK,AEBANK
17        SEG    BANKRO*,(NEWSEG)
18        IN BANKRN,BPENO,BSAVEG,BTHREV
19        SEG    ANELCO*,(NEWSEG)
20        IN ANEWCP,AFGANC,AGSTOF
21        SEG    BOFTBO*,(NEWSEG)
22        IN BIPTBL
```

VCA71

```
@FTN,UW PFA63,PFA63
-5,7
      DIMENSION ATBL(60),PTBL(60),RHOTBL(60),CTBL(60),XT(60),YT(60,3),
     * Y(3)
-14,103
      DATA N/60/,ATMOS/6H VCA71/
      DATA(ATBL(I),I=1,60)/0.,500.,1000.,1500.,2000.,2500.,3000.,4000.,
     15000.,6000.,7000.,8000.,9000.,10000.,11000.,12000.,13000.,14000.,
     215000.,16000.,16500.,17000.,18000.,19000.,20000.,22000.,24000.,
     326000.,28000.,30000.,32000.,34000.,36000.,38000.,40000.,44000.,
     448000.,52000.,57000.,62000.,67000.,72000.,
     *78000.,80000.,83000.,
     *84000.,86000.,89000.,90000.,93000.,96000.,98000.,
     *100000.,103000.,
     *                 105000.,110000.,114000.,120000.,124000.,130000./
      DATA(PTBL(I),I=1,60)/10.18,9.5587276,8.9693280,8.4104842,
     A 7.8805162,7.3793908,6.9047074,6.0312686,5.2517769,4.5580398,
     B 3.9423966,3.3976521,2.9171198,2.4993634,2.1409335,1.8334760,
     C 1.5698037,1.3437334,1.1499482,.98387621,.90998229,.84158803,
     D .71970595,.61551163,.52659006,.38584043,.28310973,.20802155,
     E .15306042,.11277507,.83205808E-01,.61610998E-01,.45881753E-01,
     F .34356322E-01,.25862528E-01,.14878758E-01,.87191499E-02,
     G .51359787E-02,.26265177E-02,.13182113E-02,.64858662E-03,
     H .31247647E-03,.12625642E-03,.92675794E-04,.67823921E-04,
     I .49335935E-04,.35914930E-04,.22307069E-04,.19032610E-04,
     J .11114846E-04,.66581474E-05,.47933157E-05,.34841023E-05,
     K .22100782E-05,.16591685E-05,.85248777E-06,.53236849E-06,
     L .29243502E-06,.21037858E-06,.14169341E-06/
      DATA(RHOTBL(I),I=1,60)/1.3004703,1.2340923,1.1704464,1.1094477,
     1 1.0510133,.99506132,.94151267,.84131448,.74981521,.66633385,
     2 .59045721,.52165360,.45941687,.39421853,.33819466,.29006526,
     3 .24672747,.21323105,.18275763,.15660248,.14495125,.13415895,
     4 .11490475,.98044963E-01,.83689297E-01,.61041925E-01,.44586932E-01,
     5 .32613851E-01,.23889470E-01,.17523275E-01,.12871319E-01,
     6 .93481175E-02,.68306486E-02,.50203988E-02,.37107363E-02,
     7 .26601264E-02,.11764019E-02,.69295459E-03,.36415558E-03,
     8 .18795236E-03,.95178066E-04,.47234922E-04,.19806644E-04,
     9 .14717842E-04,.93605867E-05,.79865441E-05,.58134902E-05,
     A .36110878E-05,.30810159E-05,.17138945E-05,.98016301E-06,
     B 68494827E-06,.48368536E-06,.28642162E-06,.20590009E-06,
     C .95645376E-07,.51782146E-07,.23712486E-07,.13961 8E-07,.73908E-08/
      DATA(CTBL(I),I=1,60)/331.04545,329.29882,327.54287,325.77746,
     1 324.00243,322.21762,320.42287,316.80287,313.14103,309.46182,
     2 305.73835,301.96896,298.15192,297.92719,297.70229,297.47722,
     3 297.25198,297.02656,296.80098,296.57523,296.46228,296.34930,
     4 296.12320,296.46228,296.80098,297.47722,298.15192,298.82510,
     5 299.49677,300.16694,300.83561,303.76028,306.65707,309.52674,
     6 312.37005,317.98042,322.12407,322.12407,317.76835,313.35209,
     7 308.87269,304.32737,298.78172,296.91016,294.08048,294.08047,
     8 294.08047,294.08047,294.08047,294.08047,294.08047,294.08047,
     9 294.08047,294.08047,294.08047,294.08047,294.08047,294.08047,
     A 294.08047,294.08047/
```

VHA71

```
@FTN,LW PRA63,PRA63
-5,7
      DIMENSION ATBL(60),PTBL(60),RHOTBL(60),CTBL(60),XT(60),YT(60,3),
     * Y(3)
-14,103
      DATA N/60/,ATMOS/6H VHA71/
      DATA(ATBL(I),I=1,60)/0.,500.,1000.,1500.,2000.,2500.,3000.,4000.,
     15000.,6000.,7000.,8000.,9000.,10000.,11000.,12000.,13000.,14000.,
     215000.,16000.,16500.,17000.,18000.,19000.,20000.,22000.,24000.,
     326000.,28000.,30000.,32000.,34000.,36000.,38000.,40000.,44000.,
     448000.,52000.,57000.,62000.,67000.,72000.,
     *78000.,80000.,83000.,
     *84000.,86000.,89000.,90000.,93000.,96000.,98000.,
     *100000.,103000.,
     *                 105000.,110000.,114000.,120000.,124000.,130000./
      DATA(PTBL(I),I=1,60)/10.1,9.5601089,9.0433739,8.5490524,8.0764172,
     1 7.6247555,7.1933710,6.3887240,5.6672056,4.9937765,4.3936324,
     2 3.8521925,3.3650957,2.9268444,2.5326658,2.1795295,1.8645095,
     3 1.5847846,1.3376399,1.1241236,1.0303371,.94468907,.79564330,
     4 .67212650,.56943204,.41212652,.30141799,.22262406,.16595099,
     5 .12478311,.094554038,.72140493E-01,.55399955E-01,
     6 .42809272E-01,.33276706E-01,.20446185E-01,.12818158E-01,
     7 .80809533E-02,.44444739E-02,.23327743E-02,.11592365E-02,
     8 .53989683E-03,.19438558E-03,.13417355E-03,.75973197E-04,
     9 .62852839E-04,.43018365E-04,.24358324E-04,.20151710E-04,
     A .11762549E-04,.70461420E-05,.50726398E-05,.36871338E-05,
     B .23401629E-05,.17558543E-05,.90216536E-06,.56385584E-06,
     C .30947628E-06,.22263812E-06,.14995041E-06/
      DATA(RHOTBL(I),I=1,60)/1.1152041,1.0772095,1.0307463,.98573248,
     1 .94226637,.90022641,.85957157,.78235450,.71039274,.64344846,
     2 .58128929,.52368784,.47042152,.42389678,.38051989,.34017810,
     3 .30276636,.26817730,.23630347,.20010642,.18271118,.16688595,
     4 .13807836,.11462241,.95455732E-01,.66810926E-01,.47305779E-01,
     5 .33859990E-01,.24483823E-01,.17874341E-01,.13205656E-01,
     6 .98296066E-02,.73688882E-02,.55617586E-02,.42250443E-02,
     7 .24831313E-02,.15075731E-02,.95041952E-03,.56202945E-03,
     8 .31897732E-03,.17253926E-03,.88159774E-04,.35927203E-04,
     9 .25938802E-04,.14687349E-04,.12150895E-04,.83164297E-05,
     A .47090188E-05,.38957845E-05,.21671307E-05,.12393653E-05,
     B .86608154E-06,.61155536E-06,.36216527E-06,.26034998E-06,
     C .12093860E-06,.65475833E-07,.30095125E-07,.17653964E-07,.934E-08/
      DATA(CTBL(I),I=1,60)/354.49409,352.48877,350.47198,348.44352,
     1 346.40318,344.35076,342.28602,338.11873,333.89943,329.62613,
     2 325.29670,320.90885,316.46018,310.90857,305.25600,299.49677,
     3 293.62461,287.63258,281.51303,280.44033,280.97719,281.51303,
     4 284.02747,286.51984,288.99072,293.87016,298.66989,303.39370,
     5 308.04507,312.62725,316.60954,320.54236,324.42751,328.26668,
     6 332.06147,339.52382,345.01468,345.01468,332.73282,319.97822,
     7 306.69451,292.80879,275.22271,269.10542,269.10542,269.10542,
     8 269.10542,296.10542,296.10542,296.10542,296.10542,296.10542,
     9 269.10542,269.10542,269.10542,269.10542,269.10542,269.10542,
     A 269.10542,269.10542/
```

```
VRA71

      @FTN,LW PRA63,PRA63
      -5,7
            DIMENSION ATBL(75),PTBL(75),RHOTBL(75),CTBL(75),XT(75),YT(75,3),
           * Y(3)
      -14,103
            DATA(N/75/,ATMOS/6H VRA71/
            DATA(ATBL(I),I=1,75)/0.,500.,1000.,1500.,2000.,2500.,3000.,4000.,
           15000.,6000.,7000.,8000.,9000.,10000.,11000.,12000.,13000.,14000.,
           215000.,16000.,16500.,17000.,18000.,19000.,20000.,22000.,24000.,
           326000.,28000.,30000.,32000.,34000.,36000.,38000.,46000.,44000.,
           448000.,52000.,57000.,62000.,67000.,72000.,
           *76000.,80000.,83000.,
           *84000.,86000.,89000.,90000.,93000.,96000.,98000.,
           *100000.,103000.,
           *                 105000.,110000.,114000.,120000.,124000.,130000.,
           *134000.,140000.,144000.,150000.,160000.,170000.,190000.,210000.,
           622000.,240000.,260000.,280000.,300000.,326000.,350000./
      C     WESTERN TEST RANGE DATA 40 TO 350 KILOMETERS
            DATA(PTBL(I),I=1,44)/
           1      10.185904,      9.6029909,      9.0477941,      8.5222274,
           2      8.0243502,      7.5523727,      7.1046558,      6.2761809,
           3      5.5286658,      4.8538800,      4.2453086,      3.6977986,
           4      3.2071987,      2.7706815,      2.3796741,      2.0378622,
           5      1.7467523,      1.4839231,      1.2630643,      1.0746324,
           6      .59021667,      .91290847,      .77604645,      .66155133,
           7      .56398303,      .41046343,      .30077469,      .22205890,
           8      .16405777,      .12206691, .91233502E-01, .68522667E-01,
           9 .51176654E-01, .39243655E-01, .30083150E-01, .17895922E-01,
           1 .10638504E-01, .66065708E-02, .35159756E-02, .18156504E-02,
           2 .89556478E-03, .42355436E-03, .15895186E-03, .11243729E-03/
            DATA(PTBL(I),I=45,75)/
           3 .65436167E-04, .54429003E-04, .37664308E-04, .21690308E-04,
           4 .18049200E-04, .10540546E-04, .63141230E-05, .45456466E-05,
           5 .33046795E-05, .20958842E-05, .15734358E-05, .80843999E-06,
           6 .50485582E-06, .27732500E-06, .19950839E-06, .13437216E-06,
           7 .10000000E-06, .81570390E-07, .69292605E-07, .55766306E-07,
           8 .40705497E-07, .26973787E-07, .18574202E-07, .11742803E-07,
           9 .94567372E-08, .62700608E-08, .42545264E-08, .29465578E-08,
           1 .20790502E-08, .13517150E-08, .92648059E-09/
            DATA(RHOTBL(I),I=1,44)/
           1      1.2361775,      1.1632456,      1.0583266,      1.0396111,
           2      .98575564,      .93576941,      .88892961,      .80276189,
           3      .72466088,      .65342553,      .58827869,      .52859970,
           4      .47357411,      .42242641,      .37353754,      .32593435,
           5      .28201088,      .24284485,      .20855008,      .17862758,
           6      .16505515,      .15225517,      .12851184,      .10882416,
           7 .92201911E-01, .65810409E-01, .47498948E-01, .34657364E-01,
           8 .25289117E-01, .18453521E-01, .13544035E-01, .99955356E-02,
           9 .74112084E-02, .55182770E-02, .41277733E-02, .23525465E-02,
           1 .13534194E-02, .84953524E-03, .46833031E-03, .25705139E-03,
           2 .13650376E-03, .68824123E-04, .28236629E-04, .20768398E-04/
            DATA(RHOTBL(I),I=45,75)/
           3 .12618788E-04, .10456153E-04, .72632258E-05, .41827847E-05,
           4 .34606292E-05, .19361501E-05, .11072921E-05, .77378743E-06,
           5 .54642031E-06, .32357107E-06, .23260574E-06, .10815076E-06,
           6 .58498391E-07, .26788033E-07, .15772667E-07, .83453992E-08,
           7 .50127386E-08, .37358155E-08, .28715840E-08, .20222947E-08,
           8 .12767727E-08, .75436669E-09, .47907675E-09, .28199857E-09,
           9 .21553292E-09, .13732034E-09, .88716316E-10, .58634523E-10,
           1 .39563746E-10, .24571145E-10, .16172967E-10/
            DATA(CTBL(I),I=1,75)/
           1 339.70470, 339.57072, 338.78074, 337.59326, 336.14343,
           2 334.50168, 330.82994, 326.80685, 322.49129, 317.86835, 312.95134,
           3 307.85330, 303.03224, 298.65008, 295.84359, 293.95944, 292.50124,
           4 291.20953, 290.13377, 289.79387, 289.70126, 290.77721, 291.65773,
           5 292.84225, 295.53832, 297.80132, 299.43738, 301.43187, 304.31165,
           6 307.05098, 309.61355, 312.72402, 315.53631, 319.42409, 326.34116,
           7 325.99523, 329.88156, 324.19829, 314.46338, 303.73397, 293.52707,
           8 285.73096, 275.20724, 269.44122, 269.44122, 269.44122, 269.44122,
```

```
     9 265.44122, 276.07141, 282.54611, 286.78136, 290.95497, 301.13603,
     1 307.73633, 323.84874, 347.59671, 380.70451, 420.81571, 474.66911,
     2 507.40585, 552.88841, 581.23621, 621.33762, 668.08799, 707.52708,
     3 736.74375, 763.53041, 776.57742, 799.52559, 819.38459, 838.77353,
     4 857.72430, 877.59411, 895.54465/
```

DISTRIBUTION LIST

ORGANIZATION	No. Copies
National Aeronautics and Space Administration George C. Marshall Space Flight Center Marshall Space Flight Center, AL 35812	
ATTN: AS24D	3
AT01	1
EM 12-39	1
EL24/Lott	10 Plus 1 Reproducible
AP29-F	Letter Only
DCASMA, Birmingham 908 South 20th Street Birmingham, AL 35256	Letter Only