(3)

Copy No. **10**

NASA General Working Paper No. 10,036

A TWO-DIMENSIONAL DIGITAL COMPUTER PROGRAM FOR CALCULATION OF OPTIMUM TRAJECTORIES FROM LAUNCH TO INJECTION

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
MANNED SPACECRAFT CENTER
Houston, Texas

September 21, 1964

N70-35768

A TWO-DIMENSIONAL DIGITAL COMPUTER PROGRAM F CALUCULATION OF OPTIMUM TRAJECTORIES FROM LAUNCH TO INJECTION

National Aeroanutics and Space Administration
Houston, Texas

21 September 1964

NASA General Working Paper No 10,036

A TWO-DIMENSIONAL DIGITAL COMPUTER PROGRAM FOR CALCULATION OF OPTIMUM TRAJECTORIES FROM LAUNCH TO INJECTION

Prepared by: Paul G Thomas
for Lem B Fortenberry, Jr
ASTD, Flight Mechanics Branch

Eric D Bone
Eric D. Bone
Computation and Analysis Division

Authorized for Distribution:

Warren Gillespie, Jr
for Maxime A. Faget
Assistant Director for
Engineering and Development

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

MANNED SPACECRAFT CENTER

Houston, Texas

September 21, 1964

TABLE OF CONTENTS

FIGURE

A TWO-DIMENSIONAL DIGITAL COMPUTER PROGRAM FOR CALCULATION OF OPTIMUM TRAJECTORIES FROM LAUNCH TO INJECTION

SUMMARY

A two-dimensional digital computer program has been developed for computing optimum trajectories from launch to injection An optimum trajectory is the trajectory that allows the maximum payload to be placed into the desired orbit Assuming a fixed thrust, the optimum trajectory is a minimum flight time trajectory The program equations of motion describe the flight of a point-mass multistage vehicle over a spherical planet with the effect of planet rotation taken into account

This program was written to replace two existing trajectory programs an atmospheric preset angle-of-attack trajectory program and a vacuum calculus of variations trajectory program Preliminary results indicate that use of the new program will effect at least a 50 percent reduction in the total time required to complete a launch vehicle performance study using the two existing programs

INTRODUCTION

Most preliminary performance trajectory studies necessitate the use of large storage, high speed, digital computers Due to the high rental cost of these computers, it is highly beneficial to use the most efficient programs available Consequently, the Performance Analysis Section of the Advanced Spacecraft Technology Division is conducting a continuing search for new and improved methods of calculation

One area in which this continuing study is yielding significant progress is that concerned with the calculation of optimum trajectories In the past, optimum trajectory calculations have required the use of two computer programs: an atmospheric preset angle-of-attack program and a vacuum calculus of variations program The former program was used to calculate the atmospheric segment of the trajectory as well as the initial conditions for the calculus of variations program Using these initial conditions, the latter program was subsequently used to calculate an optimum trajectory to a desired flight path angle and altitude As used here, an optimum trajectory is that trajectory which yields the maximum payload at injection

The above process for calculating optimum trajectories was time consuming in terms of both man hours and machine time Using this process, a minimum of five preset angle-of-attack trajectories were required necessitating 10 separate machine runs with considerable intermediate manual data preparation At least 81 calculus of variations trajectories were required, or, under less favorable conditions, an average of 137 computed trajectories per study

The program described in this paper was written to replace the two programs described above and represents a significant improvement over the above technique Using the present program, the average number of preset angle-of-attack trajectories computed is seven and the average number of calculus of variations trajectories computed is 80, all in one computer run This program permits a saving in IBM 7094 machine time of almost 50 percent and a saving in man hours of approximately 60 percent

The present paper presents a detailed description of the new program The appendixes include the equations of motion and auxiliary equations, input locations, print schedule, and Fortran listing of the program

SYMBOLS

Symbol	Fortran	Print	Definition
AAA	AAA	AAA	Integration constant
A	AREA		Cross sectional area of the first stage (ft^2)
Ae	AE	AEX	Exhaust area of the first stage engines (ft^2)
h	ALT	ALT	Altitude (ft).
a			Semi-major axis (ft)
C_D	CD	CD	Axial drag coefficient
C_{Z_α}	CL	CL	Normal lift coefficient/radian
D	DRAG	DRAG	Drag (lbs).
F	F	THRUST	Vehicle thrust (lbs).

Symbol	Fortran	Print	Definition
g	GRAV	GRAV	Vehicle acceleration due to the gravity of the planet (ft/sec^2)
g_{oe}	GRAVO		Factor for converting weight to mass (ft/sec^2)
I_{sp}	SIP	ISP	Specific impulse (sec)
L	ALIFT	LIFT	Lift (lbs).
m	SAM	MASS	Mass (slugs).
mach	MACH	MACH	Mach number
P	PRES	PRES	Atmospheric pressure (lbs/ft^2)
q	QD	Q	Dynamic pressure (lbs/ft^2)
R	R	RRR	Distance from center of planet to center of vehicle (ft)
t	T	TIME	Time (sec)
V	V	VEL	Vehicle velocity (ft/sec)
W	WGT	WLBS	Weight (lbs)
X	X	XXX	Ground range
α	ALPHA	ALPHA	Angle-of-attack (deg).
θ	TATER	THETA	Angle between local vertical and velocity vector (deg)
λ	ALAM	LAMB	Lagrange Multiplier
ρ	RHO	RHO	Atmospheric density (lbs/ft^3)
Φ	PHI	PHI	Range angle (deg)
Φ'	PHIPR		Latitude (deg)
ω	OMEGA		Rotation velocity of planet (rad/sec).

Subscripts

DES	Desired
M	Maximum
o	Initial or planet surface conditions
P	Predicted
S	Speed of sound
SF	Space fixed
SL	Sea level

PROGRAM DESCRIPTION

Atmospheric Preset Angle-of-Attack Computations

The preset angle-of-attack section of the program is designed to predict the point mass trajectory through an atmosphere over a non-rotating celestial sphere The equations of motion are written assuming two degrees of freedom with motion in the pitch plane only. These equations together with the necessary auxiliary equations are presented in appendix A

The preset angle-of-attack section simulates the vertical flight of the vehicle until some preselected time when the vehicle will have obtained sufficient altitude to clear all ground obstructions At this time a small angle-of-attack is ramped in and held at a constant value until it is ramped out at some preselected time before the high dynamic pressure region A no-lift (zero angle-of-attack, gravity tilt) trajectory is then flown through the high dynamic pressure region until cut-off time of the first stage

Capability of instantaneous changes in flow rate, thrust, and exhaust area are incorporated into this section of the program to simulate engine cut-off, coast period, engine failure, and other sudden shifts that can occur during flight

The effect of planet rotation on the velocity of the ascending vehicle is accounted for by a space-fixing equation introduced after cut-off of the first stage This rotational effect is estimated by vectorially adding the rotational velocity vector of the planet to the vehicle velocity vector This calculation produces a change in both the magnitude and direction of the vehicle velocity vector

The magnitude of the angle-of-attack introduced during the first stage flight is the only free variable of the first stage and is controlled by a maximization routine

Vacuum Calculus of Variations Computations

The vacuum calculus of variations section of the program is a two degree-of-freedom trajectory simulation which assumes flight in vacuo over a non-rotating celestial sphere The equations of motion for this section are identical to those for the preset angle-of-attack section with the lift and drag terms omitted The Euler-Lagrange equations resulting from the variational calculus theory, are written in such a manner as to produce the optimum instantaneous steering angle throughout the trajectory This section of the program is employed at the beginning of second stage burning at which time the vehicle has acquired sufficient altitude that the effects of atmospheric drag and lift can be considered negligible

After ignition of the second stage engines a preselected amount of burn time is allotted for the required stabilization of the vehicle At the end of this time, the launch escape system is jettisoned. The resulting weight loss causes a discontinuity in the state variable The Lagrange multipliers are held constant across this discontinuity to insure an optimum trajectory This can be done since the problem is one of minimum time There are no other discontinuities in this section of the program

The Euler-Lagrange equations are concerned with three variables The first of these equations results in the free variable λ_1 which can be used to control the range When λ_1 is set equal to zero, this section of the program will compute the range that will allow the maximum payload to be placed in the desired orbit.

The remaining two variables are the angle-of-attack, α_0, and the time rate of change of angle-of-attack, $\dot{\alpha}_0$, at second stage ignition The Euler-Lagrange equations are manipulated so these two variables may be satisfied by either a maximization or an isolation scheme. In this program, α_0 and α_0 are controlled by an isolation routine and no control is exercised over the variable λ_1

The calculus of variations trajectory computations are terminated when the cut-off equation is satisfied

The integration throughout the program is performed by the SHARE subroutine RW-INT

Isolation Routine

The isolation routine uses the variables α_0, α_0, and the altitude and flight path angle at the time of satisfaction of the cut-off equation to insure injection into the desired orbit. For example, if the desired orbit is a 100-nautical mile circular orbit, the cut-off equation will be satisfied at local circular velocity. The initial time rate of change of angle-of-attack will be varied until the desired flight path angle (90 deg) is reached, and the initial angle-of-attack will be varied until the desired altitude (100 nautical miles) is obtained.

This routine is designed to isolate on the desired end conditions in a minimum number of trajectories. However, in cases where the initial conditions are unfavorable, the routine will use as many as five points in order to insure an isolation. In all cases, the equations of isolation are the same with the following exceptions: for the isolation of flight path angle, the equations of isolation over α_0 use all trajectories, for the isolation of altitude, the equations of isolation over α_0 use only those trajectories in which the desired path angle is already obtained.

An initial guess for α_0 and α_0 and the magnitude of their step sizes must be fed into the computer as input. These guesses must not be equal to zero.

The first trajectory is computed using the input values for the initial α_0 and α_0. A second trajectory is then computed, using for its initial conditions the input α_0 and the input α_0 summed to the step to be taken on α_0. That is, α_0 is the same for both trajectories, while α_0 is changed by a small amount for the second trajectory. The direction of the step taken on α_0 is fixed by the program so that the resulting path angle changes in the direction of the desired path angle.

After the first two trajectories are run, the two-point interpolation/extrapolation equation

$$\alpha_3 = \alpha_2 + \left(\theta_{DES} - \theta_2\right) \left(\frac{\alpha_2 - \alpha_1}{\theta_2 - \theta_1}\right) \tag{1}$$

is used to arrive at the desired α_0 used to compute the third trajectory. The α_0's from these three trajectories are then used to construct a conic type curve which closely duplicates the actual plot of α_0 versus the cut-off path angle An accurate three-point fit is obtained using the conic equation

$$\alpha = \frac{-B \pm \sqrt{B^2 - 4A\left(C\theta_{DES} - 1\right)}}{2A} \tag{2}$$

where

$$\begin{bmatrix} A \\ B \\ C \end{bmatrix} = \begin{bmatrix} \alpha_1^{\ 2} & \alpha_1 & \theta_1 \\ \alpha_2^{\ 2} & \alpha_2 & \theta_2 \\ \alpha_3^{\ 2} & \alpha_3 & \theta_3 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{3}$$

Equations (2) and (3) are used to determine the α_0 for the next trajectory This type of conic fit works equally as well with four or five points For example, for four and five points, equation (2) becomes

$$\alpha = \frac{-BB \pm \sqrt{BB^2 - 4ACC}}{2A} \tag{4}$$

where

$$A = A \tag{5}$$

$$BB = B\theta_{DES} + D \tag{6}$$

$$CC = C\theta^2_{DES} + E\theta_{DES} - 1 \tag{7}$$

For four points, equation (3) becomes

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \alpha_1^2 & \alpha_1\theta_1 & \theta_1^2 & \alpha_1 \\ \alpha_2^2 & \alpha_2\theta_2 & \theta_2^2 & \alpha_2 \\ \alpha_3^2 & \alpha_3\theta_3 & \theta_3^2 & \alpha_3 \\ \alpha_4^2 & \alpha_4\theta_4 & \theta_4^2 & \alpha_4 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{8}$$

For five points this equation becomes

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} \alpha_1^2 & \alpha_1\theta_1 & \theta_1^2 & \alpha_1 & \theta_1 \\ \alpha_2^2 & \alpha_2\theta_2 & \theta_2^2 & \alpha_2 & \theta_2 \\ \alpha_3^2 & \alpha_3\theta_3 & \theta_3^2 & \alpha_3 & \theta_3 \\ \alpha_4^2 & \alpha_4\theta_4 & \theta_4^2 & \alpha_4 & \theta_4 \\ \alpha_5^2 & \alpha_5\theta_5 & \theta_5^2 & \alpha_5 & \theta_5 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{9}$$

The conic curve fit can use no more than five points; therefore, if more than five trajectories are required to obtain the desired value of α, the last five computed points are retained

When the path angle obtained is within tolerance of the desired path angle, the trajectory is said to be converged on path angle

After the routine is converged on path angle, the altitude obtained from the converged trajectory is checked against the desired altitude A trajectory is then calculated using the α_0 from the converged path angle in conjunction with the input α_0 summed to the step to be taken on α_0 The direction of the step taken on α_0 is fixed by the program so that the resulting altitude of the second converged-on path angle trajectory changes in the direction of the desired altitude

The method of isolating on all subsequent converged-on path angle trajectories and converged-on altitude trajectories is the same with the exception that the first α_0 and α_0 are computed using the knowledge gained in converging the previous trajectories Also, the step sizes and the direction of the steps on these variables are computed as a function of the variations in earlier trajectories

The information computed from the converged-on altitude trajectory is fed into the maximization routine and the isolation routine is re-initialized

Maximization Routine

The maximization routine optimizes the complete trajectory using the preset angle-of-attack of the first stage trajectory and the cut-off weight of the isolated second stage trajectory As used here, an optimum trajectory is that trajectory which will allow the maximum payload to be placed into the desired orbit

The first isolated second stage trajectory is computed from the end point of the first stage trajectory This first stage trajectory is computed using the input preset angle-of-attack estimate for the first stage flight A second first stage trajectory is then computed using the input angle-of-attack step added to the preset angle-of-attack estimate The preset angle-of-attack step for the third isolated trajectory is taken in the direction that will result in an increase in cut-off weight in orbit Due to the nature of a plot of preset angle-of-attack as a function of cutoff weight in orbit, an extrapolation to the maximum cut-off weight is extremely dangerous in that it can result in trajectories that cannot be advantageously used by the maximization routine Therefore, additional trajectories are computed using a fixed step in the preset angle-of-attack until the maximum cut-off weight in orbit has been enclosed

A knowledge of the range of preset angles-of-attack that can be used by the vehicle will result in a shorter computer running time However, knowledge of this range is not absolutely necessary if small values are input for the preset angle-of-attack and the preset angle-of-attack step

After the maximum has been enclosed, it is seen that a conic equation closely simulates the plot of preset angle-of-attack as a function of cut-off weight in orbit The conic equation used in this routine is

$$\alpha_m = \frac{-BW_p + D}{2A} \tag{10}$$

where

$$W_p = \frac{- b \pm \sqrt{b^2 - 4ac}}{2A} \tag{11}$$

$$a = C - B^2/4A \tag{12}$$

$$b = E - BD/2A \tag{13}$$

$$c = - D^2/4A - 1 \tag{14}$$

and for a three-point fit,

$$\begin{bmatrix} A \\ D \\ E \end{bmatrix} = \begin{bmatrix} \alpha_1^{\,2} & \alpha_1 & W_{c_1} \\ \alpha_2^{\,2} & \alpha_2 & W_{c_2} \\ \alpha_3^{\,2} & \alpha_3 & W_{c_3} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \tag{15}$$

For a four-point fit,

$$\begin{bmatrix} A \\ B \\ C \\ D \end{bmatrix} = \begin{bmatrix} \alpha_1^{\,2} & \alpha_1 W_{c_1} & W_{c_1}^{\,2} & \alpha_1 \\ \alpha_2^{\,2} & \alpha_2 W_{c_2} & W_{c_2}^{\,2} & \alpha_2 \\ \alpha_3^{\,2} & \alpha_3 W_{c_3} & W_{c_3}^{\,2} & \alpha_3 \\ \alpha_4^{\,2} & \alpha_4 W_{c_4} & W_{c_4}^{\,2} & \alpha_4 \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{16}$$

For a five-point fit,

$$\begin{bmatrix} A \\ B \\ C \\ D \\ E \end{bmatrix} = \begin{bmatrix} \alpha_1^2 & \alpha_1 W_{c_1} & W_{c_1}^2 & \alpha_1 & W_{c_1} \\ \alpha_2^2 & \alpha_2 W_{c_2} & W_{c_2}^2 & \alpha_2 & W_{c_2} \\ \alpha_3^2 & \alpha_3 W_{c_3} & W_{c_3}^2 & \alpha_3 & W_{c_3} \\ \alpha_4^2 & \alpha_4 W_{c_4} & W_{c_4}^2 & \alpha_4 & W_{c_4} \\ \alpha_5^2 & \alpha_5 W_{c_5} & W_{c_5}^2 & \alpha_5 & W_{c_5} \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{17}$$

Equation (10) is used to compute the preset angle-of-attack that will result in the maximum cut-off weight in orbit. If more than five preset angle-of-attack trajectories must be computed in order to insure an optimum trajectory, the maximization routine will retain and use only the five points most likely to produce the maximum

The cut-off weight in orbit is checked against the maximum cut-off weight computed by the maximization routine When these weights are within tolerance, the program is said to be in a converged state The program will then print a history of the optimum trajectory variables and check to see if additional trajectories are to be computed

Cut-off Equation

The conditions which terminate a trajectory calculation program are among the limiting factors of the program Hence, the cut-off condition should be as versatile as possible Since the computed velocity for a conic section is the most adaptable to the computation of a local (as a function of altitude) velocity requirement, the conic velocity will be employed as a cut-off condition

The equation for the velocity requirement of all conic sections except the hyperbola can be expressed as

$$V^2 = g_o R_o^2 \left(\frac{2}{R} - \frac{1}{a}\right) \tag{18}$$

The velocity requirement for a hyperbola is

$$V^2 = g_o R_o^{\,2}\left(\frac{2}{R} + \frac{1}{a}\right) \tag{19}$$

One of the most common requirements is that the vehicle pass through some point in space in a given direction If the point in space is point 2(R_2, θ_2) and the instantaneous position of the vehicle is point 1 (R_1, θ_1),

$$V^2 = \left(\frac{g_o R_o^{\,2}}{R_1}\right)\left[\frac{2(R_2 - R_1)\, R_2 \sin^2 \theta_2}{R^2_{\,2} \sin^2 \theta_2 - R^2_{\,1} \sin^2 \theta_1}\right] \tag{20}$$

When the required point in space is the apogee of an ellipse, then R_2 becomes the radius of the apogee of the ellipse, the flight path angle becomes 90°, and the velocity requirement becomes

$$V^2 = \left(\frac{g_o R_o^{\,2}}{R_1}\right)\left[\frac{2(R_2 - R_1)\, R_2}{R^2_{\,2} - R^2_{\,1} \sin^2 \theta_1}\right] \tag{21}$$

The most widely used velocity cut-off requirements are local circular and local parabolic escape velocity To compute the local circular velocity requirement, the quantity in the brackets in equation (21) must equal one, and to compute the local parabolic escape velocity requirement the quantity in the brackets must equal two

Equation (21) can be written in the following form such that the calculus of variations section of the program can be terminated on time or velocity

$$\text{CUTOFF} = Z_1 \sqrt{\frac{\left(Z_2 + 1\right) Z_3 \left(1 + Z_4 \frac{R}{R_2}\right)}{R\left[1 + Z_4\left(\frac{R^2}{R_2^{\,2}}\right) \sin^2 \theta\right]} + Z_5 + Z_6} \tag{22}$$

where Z_1, Z_2, Z_6, and R_2 are input constants

In order to terminate the trajectory on some preselected time, the following values are input, and the trajectory time is compared to CUTOFF

$$Z_1 = 0 \qquad Z_3 = 0 \qquad Z_5 = 0$$

$$Z_2 = 0 \qquad Z_4 = 0 \qquad Z_6 = \text{desired cutoff time}$$

$$R_2 = 1 \tag{23}$$

For velocity termination of the trajectory, the following values are input, and the vehicle velocity is compared to CUTOFF

For local circular

$$Z_1 = 1 \qquad Z_3 = g_o R_o^2 \qquad Z_5 = 0 \qquad R_2 = 1$$

$$Z_2 = 0 \qquad Z_4 = 0 \qquad Z_6 = 0 \tag{24}$$

For parabolic escape

$$Z_1 = 1 \qquad Z_3 = g_o R_o^2 \qquad Z_5 = 0 \qquad R_2 = 1$$

$$Z_2 = 1 \qquad Z_4 = 0 \qquad Z_6 = 0 \tag{25}$$

For hyperbolic escape

$$Z_1 = 1 \qquad Z_3 = g_o R_o^2 \qquad Z_5 = V^2_{He} \qquad R_2 = 1$$

$$Z_2 = 1 \qquad Z_4 = 0 \qquad Z_6 = 0 \tag{26}$$

For elliptic

$$Z_1 = 1 \qquad Z_3 = g_o R_o^2 \qquad Z_5 = 0 \qquad R_2 = R_2$$

$$Z_2 = 0 \qquad Z_4 = 1 \qquad Z_6 = 0 \tag{27}$$

For <u>constant velocity</u>

$$Z_1 = 0 \qquad Z_3 = 0 \qquad Z_5 = 0 \qquad R_2 = 1$$

$$Z_2 = 0 \qquad Z_4 = 0 \qquad Z_6 = \text{Desired velocity} \tag{28}$$

Areas for Future Revision

The program in its present form is quite limited and a number of improvements and expansions are possible These include

1 An increase in the number of stages that can be calculated using the calculus of variations theory

2 A provision for weight loss table and sea level thrust table to be input as a function of time

3 Addition of the ability to simulate thrust decay

4 Modification of the isolation and maximization routine such that for escape trajectories one variable will be used for isolation and two variables will be used for maximization

5 Addition of the ability to optimize the propellant loading of each stage to determine the optimum tank size on predesign stages

6 Revision of the calculus of variations equations so that the atmospheric effects may be taken into account

APPENDIX A

PRESET ANGLE-OF-ATTACK EQUATIONS OF MOTION AND AUXILIARY EQUATIONS

The atmospheric preset angle-of-attack equations of motion are

$$X = \frac{R_o}{R} V \sin \theta \tag{A1}$$

$$R = V \cos \theta \tag{A2}$$

$$V = \frac{F}{m} \cos \alpha - \frac{D}{m} - g \cos \theta \tag{A3}$$

$$\theta = \frac{F}{mV} \sin \alpha + \frac{L}{mV} + \left(\frac{g}{V} - \frac{V}{R}\right) \sin \theta \tag{A4}$$

$$m = \frac{- F_{SL}}{I_{sp}\ g_{oe}} \tag{A5}$$

where

$$F = F_{SL} + Ae\ (P_o - P) \tag{A6}$$

$$g = g_o \left(\frac{Ro}{R}\right)^2 \tag{A7}$$

$$D = Aq\ (C_D \cos \alpha + \alpha\ C_Z \sin \alpha) \tag{A8}$$

$$L = Aq\ (-C_D \sin \alpha + \alpha\ C_Z \cos \alpha) \tag{A9}$$

$$M = \frac{W}{g_{oe}} \tag{A10}$$

also

$$\text{mach} = \frac{V}{V_S} \tag{A11}$$

$$q = \frac{1}{2} \rho V^2 \tag{A12}$$

$$\chi = \Phi + \theta + \alpha \tag{A13}$$

Altitude and range angle are given by

$$H = R - Ro \tag{A14}$$

and

$$\Phi = \frac{X}{Ro} \tag{A15}$$

respectively

A vector diagram of the summation of the forces and the coordinate system is given in figure 1 on page 17

The preset angle-of-attack program is given by

$$\alpha = \alpha_o \qquad t_o \le t \le t_1 \tag{A16}$$

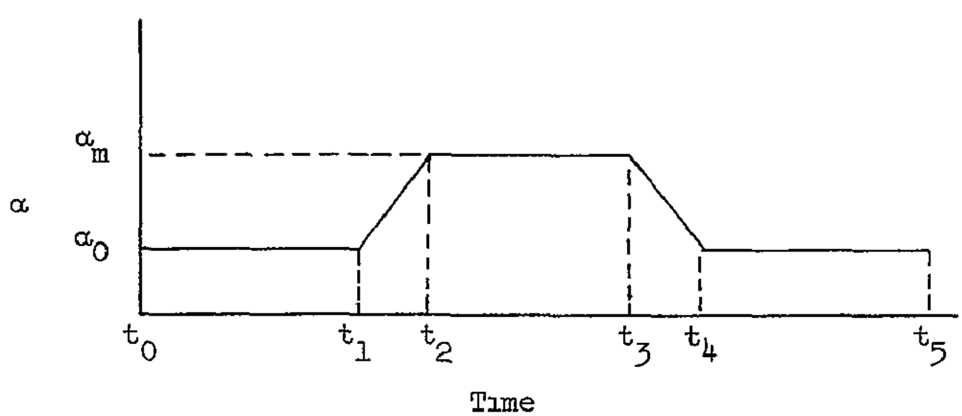
$$\alpha = \alpha_o + (\alpha_m - \alpha_o)\left(\frac{t - t_1}{t_2 - t_1}\right) \qquad t_1 \le t \le t_2 \tag{A17}$$

$$\alpha = \alpha_m \qquad t_2 \le t \le t_3 \tag{A18}$$

$$\alpha = \alpha_m - (\alpha_m - \alpha_o)\left(\frac{t - t_4}{t_3 - t_4}\right) \qquad t_3 \le t \le t_4 \tag{A19}$$

$$\alpha = \alpha_o \qquad t_4 \le t \le t_5 \tag{A20}$$

as indicated in the sketch below



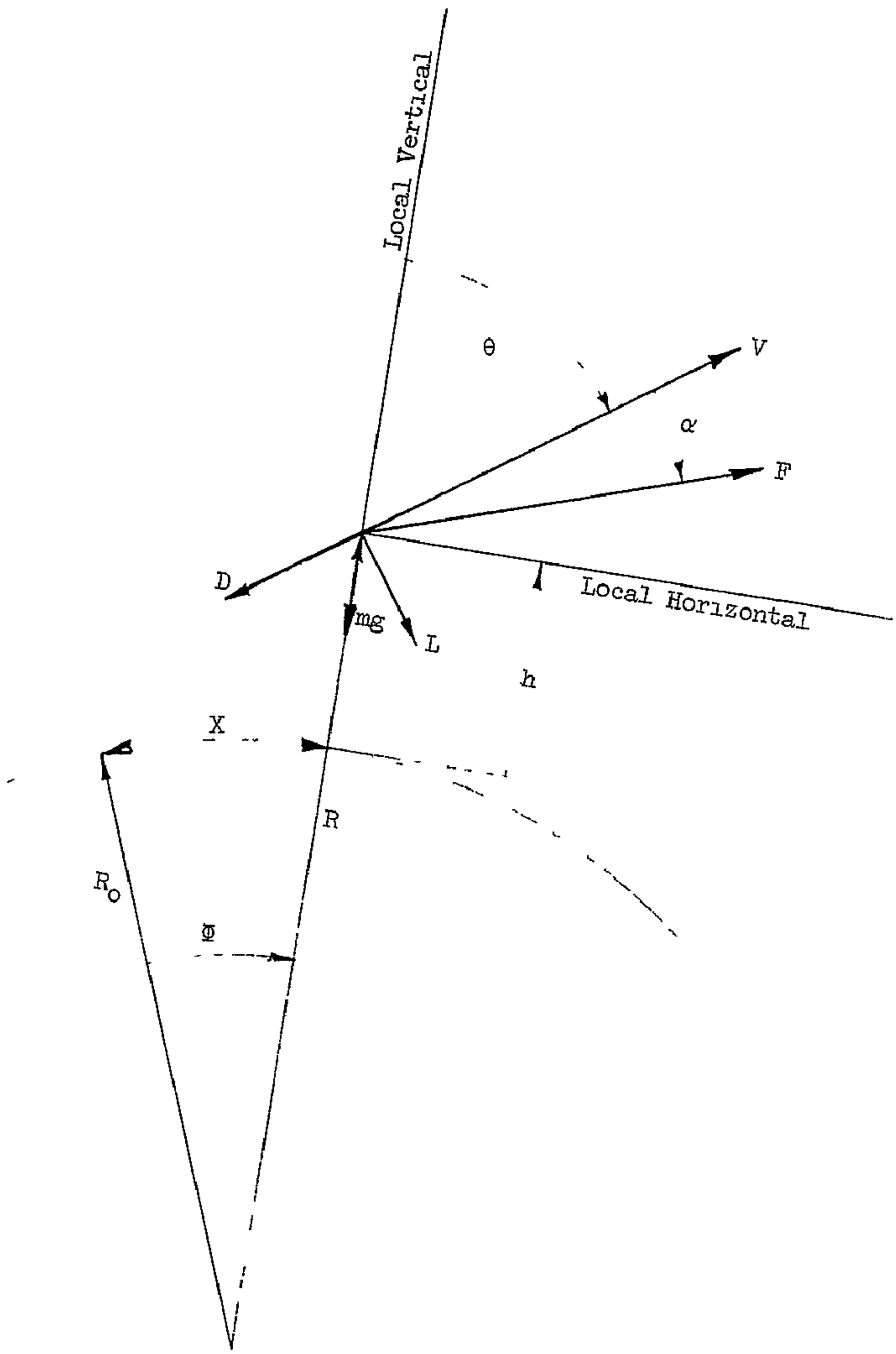


Figure A1.- Vector and coordinate system diagram.

Pressure, density and the velocity of sound are table look-up values as a function of altitude The drag and lift coefficients are table look-up values as a function of Mach number.

At first stage cut-off, the rotation of the planet is taken into account by means of the equations

$$V_{SF} = \sqrt{V^2 + 2\omega R_o V \cos \Phi' \sin \theta \sin A_Z + \omega^2 R_o^{\ 2} \cos^2 \Phi'} \qquad (A21)$$

and

$$\theta_{SF} = \text{Arc} \cos \left(\frac{V}{V_{SF}} \cos \theta \right) \qquad (A22)$$

APPENDIX B

CALCULUS OF VARIATIONS EQUATIONS OF MOTION, EULER-LAGRANGE EQUATIONS, AND AUXILIARY EQUATIONS

The equations of motion for the vacuum calculus of variations are given below The coordinate system and forces are the same as those in figure 1 with the exception that in the calculus of variations there are no aerodynamic forces acting on the vehicle

$$X = \frac{R_o}{R} V \sin \theta \tag{B1}$$

$$R = V \cos \theta \tag{B2}$$

$$V = \frac{F}{m} \cos \alpha - g \cos \theta \tag{B3}$$

$$\theta = \frac{F}{mV} \sin \alpha + \left(\frac{g}{V} - \frac{V}{R}\right) \sin \theta \tag{B4}$$

$$m = \frac{-F}{I_{sp}\, g_{oe}} \tag{B5}$$

where

$$V = V_{SF} \tag{B6}$$

$$\theta = \theta_{SF} \tag{B7}$$

$$g = g_o \left(\frac{R_o}{R}\right)^2 \tag{B8}$$

and

$$m = \frac{W}{g_{o_e}} \tag{B9}$$

The Euler-Lagrange equations result in

$$\dot{\lambda}_1 = 0 \tag{B10}$$

$$\dot{\lambda}_2 = \frac{-2\lambda_3 g}{R} \cos\theta + \frac{\lambda_4}{R}\left(\frac{2g}{v} - \frac{v}{R}\right) \sin\theta \tag{B11}$$

$$\dot{\lambda}_3 = -\lambda_2 \cos\theta + \frac{\lambda_4}{v} \left[\frac{F}{Mv} \sin\alpha + \left(\frac{g}{v} + \frac{v}{R}\right) \sin\theta\right] \tag{B12}$$

$$\dot{\lambda}_4 = \lambda_2 V \sin\theta - \lambda_3 g \sin\theta - \lambda_4 \left(\frac{g}{v} - \frac{v}{R}\right) \cos\theta \tag{B13}$$

$$\dot{\lambda}_5 = \frac{F}{m^2}\left(\lambda_3 \cos\alpha + \frac{\lambda_4}{v} \sin\alpha\right) \tag{B14}$$

and

$$0 = \lambda_3 \sin\alpha - \frac{\lambda_4}{v} \cos\alpha \tag{B15}$$

$$\alpha = \arctan \frac{\lambda_4}{\lambda_3 V} \tag{B16}$$

Differentiating equation (B15) and substituting equation (B-12) and (B-13) for $\dot{\lambda}_3$ and $\dot{\lambda}_4$ leads to

$$\lambda_3 = \frac{\lambda_2 B \cos\alpha}{V} \tag{B18}$$

and

$$\lambda_4 = \lambda_2 B \sin\alpha \tag{B19}$$

where

$$B = \frac{V^2 \sin(\alpha + \theta)}{V \dot{\alpha} + g \sin\theta + \frac{F}{m} \sin\alpha - \frac{V^2}{R} \sin\alpha \cos(\alpha + \theta)} \tag{B20}$$

and

$$\lambda_2 = \frac{B}{|B|} \tag{B21}$$

It can be shown from equation (B14) that

$$\lambda_5 \geq 0 \quad \text{when} \quad F \geq 0$$

Substituting equations (B18) and (B19) into equation (B14) and solving the inequality results in equation (B21)

By solving equations (B18) through (B21) at the time of initialization of the calculus of variations the Lagrange multipliers are expressed as a function of the state variables and the control variables α_0 and α_0

APPENDIX C

INPUT LOCATIONS AND PRINT SCHEDULE

The subroutine used by this program to input information into the machine is RW-FINP The FINP input subroutine uses D, N, J, F, and $ in a unique order and are defined here instead of in the list of symbols

Ni = D is the introduction of the table being entered into the machine where i is the number of the table in the calling sequence Ji = the introduction of each entry in the table, where i is the location number in the table F signals the end of each table and $ signals the end of each calling sequence

COMMENT CARD

N1 = D, (Altitude Table) F

N2 = D, (Density Table) F

N3 = D, (Pressure Table) F

N4 = D, (1 /Velocity of Sound Table) F$

Atmosphere locations, 250 locations per table

N1 = D, (mach Table) F

N2 = D, (C_D Table) F

N3 = D, (C_L Table) F$

N1 = D

J4 = X_o, Initial Range

J5 = $R = R_o + h_o$, Initial Radius to Vehicle

$J_6 = V_o$, Initial Velocity

$J_7 = \theta_o$F, Initial Flight Path Angle (from Vertical)

This block of input may be omitted Built in values are:

$x_o = 0$ $V_o = 0$

$R_o = 20898906$ $\theta = 0$

Input values may be used to override these and other built in values (below)

N2 = D

J2 = $\Delta t^{(I)}$, Step size for 1st stage

J3 = KAIP = 1B or 2B
integration mode for 1st stage

J4 = $\Delta t^{(II)}$, initial step size for
2nd stage

J5 = KAIC = 0, 1B, or 2B integration
mode for 2nd stage

This input may be omitted
Built in values are·

$\Delta t^{(I)} = 1$
KAIP = 2B
$\Delta t^{(II)} = 5$
KAIC = 0

(integration modes are· 0 - Variable step Adams-Moulton

1 - Fixed step Runge-Kutta

2 - Fixed step Adams-Moulton)

J6 = A2 … J11 = A7: Integration Parameters; see writeup on <u>RW-INT</u>
No input necessary; built in values are

$A2 = 10^{-6}$, $A3 = A4 = A5 = A6 = A7 = 0$

J13 = g_{oe} $\frac{\text{Weight}}{\text{Mass}}$ conversion factor

J14 = g_{ob} Surface gravity of body

J15 = R_o Radius of body

J16 = p_o Sea level pressure

J17 = ω Angular velocity of body

J18 = Φ' Latitude

J19 = Az Azimuth

No input necessary,

built in values are

$g_{oe} = 32\ 1849$

$g_{ob} = 32\ 1849$

$R_o = 20898906$

$p_o = 2124\ 214$

$\omega = 7292115 \times 10^{-4}$

$\Phi' = 28\ 28^\circ$

$A_z = 90^\circ$

J21 = Printout Controls; 1 prints every trajectory according to J22; 0 prints only last trajectory

J22 = nB to print every nth integration step when J21 = 1

J23 = nB to print every nth integration step of final trajectory

No input necessary, built in values are

J21 = 0
J22 = 0
J23 = 5B

For both J22 and J23 a 0 causes only 1st and last points to be printed

J24 = ±nB where + or - causes maximization or minimization respectively of velocity if n = 1 or weight if n = 2

J25 = Factor by which J80 is multiplied, when a successful step is taken, to modify step size

Built in value

J25 = 1

J28 = $t_c^{(I)}$, cut-off time on 1st stage

J29 = $W_o^{(I)}$, initial weight on 1st stage

J30 = $W_o^{(I)}$, initial flow rate of 1st stage

J31 = $F_o^{(I)}$, initial sea level thrust of 1st stage

J32 = $A_{E_o}^{(I)}$, initial exhaust area of 1st stage

J33 = $A_{C_S}^{(I)}$, cross sectional area for 1st stage

J34 = t_o, initial time, built in equal to 0

J35 = t_1

J38 = t_4

Ramp function times for tilt program*

*Any time not input will cause the rest of that particular block to be ignored

$J39 = t_{F1}$ ... $J42 = t_{F4}$ — Times during the 1st stage when changes in thrust and exhaust area may be specified*

$F43 = F_1$ ... $J46 = F_4$ — New values for sea level thrust to be brought in at times specified in J39 - J42 respectively

$J47 = A_{E1}$ ... $J50 = A_{E4}$ — New values for exhaust area to be brought in at times specified in J39 - J42 respectively

$J51 = t_{w_1}$ ... $J57 = t_{w_7}$ — Times during the 1st stage when changes in flow rate may be specified *

$J58 = w_1$ ... $J64 = w_7$ — New values for flow rate to be brought in at times specified in J51 - J57 respectively

J66 = nB, 2nd stage cut-off control; n = 1 cuts off on velocity, n = 2 cuts off on time = J67

$J67 = t_c^{(II)}$, cut-off time of 2nd stage if J66 = 2B

J68 = TOL, cut-off tolerance

For velocity cut-off:

J69 = Z1 = 0 or 1

J_{70} = Z2 = 0 or 1

$J_{71} = Z3 = g_{ob}R_o^2$

J_{72} = Z4 = 0 or 1

J73 = Z5 = 0 or 1

J74 = Z6 = 0 or 1

$J75 = R_2$ = 1 or apogee radius

*Any time not input will cause the rest of that particular block to be ignored

CUT-OFF EQUATION

$$V_{cut} = Z1 * \left[(1 + Z2) * Z3 * (1 + Z4 * R/R_2)/(1 + Z4 * R^2/R_2{}^2 * \sin^2 \theta) * R + Z5 \right]^{\frac{1}{2}} + Z6$$

J76 = α_o, initial 1st stage α; 0 if not input

J77 = α_{max}, maximum α 1st stage tilt program

J78 = α_o, initial α of 2nd stage

J79 = α_o, initial α of 2nd stage

(J76–J79:) Independent variables of search, initial guesses ($\neq 0$) must be made

J80 = $\Delta\alpha_{max}$

J81 = $\Delta\alpha_o$

J82 = $\Delta\alpha_o$

(J80–J82:) Initial step size or increment for variables to be used in search procedure

J84 = $W_o^{(II)}$, initial weight of 2nd stage

J85 = $Isp^{(II)}$, specific impulse of 2nd stage

J86 = $F_o^{(II)}$, thrust of 2nd stage

J130 = θ_{DES}, desired final flight path angle (from vertical)

J131 = h_{DES}, desired final altitude

J132 = TOL1, tolerance on θ_{DES}

J133 = TOL2, tolerance of h_{DES}

J134 = TOL3, tolerance on variable to be maximized (velocity or weight) F$

Print schedule for the preset angle-of-attack computations

XXX	XDOT	TIME	ALPHA	ALDOT
RRR	RDOT	ALT	WLBS	WDOT
VEL	VDOT	THRUST	MASS	MDOT
THETA	THDOT	GRAV	PHI	CHI
RHO	PRES	VS	DRAG	LIFT
Q	AEX	MACH	CD	CL

Print schedule for the calculus of variations computations

XXX	XDOT	TIME	ALPHA	ALDOT
RRR	RDOT	ALT	WLBS	WDOT
VEL	VDOT	THRUST	MASS	MDOT
THETA	THDOT	GRAV	PHI	CHI
ISP	LAMB2	LAMB3	LAMB4	LAMB5
AAA	DLAM2	DLAM3	DLAM4	DLAM5

APPENDIX D

FORTRAN PROGRAM

```
$IBFTC MAIN
CMAIN CONTROL PROGRAM INCLUDING INPUT AND SEARCH PROCEDURE
      DIMENSION TOP(12), C1(150), T1(150)
      DIMENSION T(150),C(150),ALTA(250),RHOTA(250),PRETA(250),FVSTA(250)
     1,FMTA(30),CDTA(30),CLTA(30)
      DIMENSION C2(150),T2(150)
      COMMON  T      , C
      COMMON ALTA,RHOTA,PRETA,FVSTA,FMTA,CDTA,CLTA
      COMMON  C1     , T1
      COMMON TEST1,TEST2
      COMMON C2,T2
      EQUIVALENCE (TIME,T(2)),(STEP,T(3)),(X,T(4)),(R,T(5)),(V,T(6)),
     1 (TATER,T(7)),(SAM,T(8)),(ALAM2,T(9)),(ALAM3,T(10)),(ALAM4,T(11)),
     2 (ALAM5,T(12)),(XDOT,T(13)),(RDOT,T(14)),(VDOT,T(15)),(TATDOT,
     3 T(16)),(SAMDOT,T(17)),(DLAM2,T(18)),(DLAM3,T(19)),(DLAM4,T(20)),
     4 (DLAM5,T(21))
      EQUIVALENCE (N,C(1)),(STEPP,C(2)),(KA1P,C(3)),(STEPC,C(4)),(KA1C,C
     1(5)),(A2,C(6)),(A3,C(7)),(A4,C(8)),(A5,C(9)),(A6,C(10)),(A7,C(11))
     2
     3,(G0,C(13)),(GRAV0,C(14)),(R0,C(15)),(PRES0,C(16)),(OMEGA,C(17)),(
     4PHIPR,C(18)),(AZ,C(19))
     5
     6,(SAVPAP,C(21)),(IPRNT,C(22)),(NPR,C(23)),(IOPT,C(24)),(STP,C(25))
     7
     8,(TIMPC,C(28)),(W0,C(29)),(WD0,C(30)),(F0,C(31)),(AE0,C(32)),(AREA
     9,C(33)),(TIM0,C(34))
      EQUIVALENCE (TIM1,C(35)),(TIM2,C(36)),(TIM3,C(37)),(TIM4,C(38)),(T
     1IMF1,C(39)),(TIMF2,C(40)),(TIMF3,C(41)),(TIMF4,C(42)),(F1,C(43)),(
     2F2,C(44)),(F3,C(45)),(F4,C(46)),(AE1,C(47)),(AE2,C(48)),(AE3,C(49)
     3),(AE4,C(50)),(TIMW1,C(51)),(TIMW2,C(52)),(TIMW3,C(53)),(TIMW4,C(5
     44)),(TIMW5,C(55)),(TIMW6,C(56)),(TIMW7,C(57)),(WD1,C(58)),(WD2,C(5
     59)),(WD3,C(60)),(WD4,C(61)),(WD5,C(62)),(WD6,C(63)),(WD7,C(64))
     6
     7,(ICUT,C(66)),(TIMCC,C(67)),(TOL,C(68)),(Z1,C(69)),(Z2,C(70)),(Z3,
     8C(71)),(Z4,C(72)),(Z5,C(73)),(Z6,C(74)),(R2,C(75))
     9
      EQUIVALENCE (ALPHAP,C(76)),(ALFMAX,C(77)),(ALPHAC,C(78)),(DALF,C(7
     19)),(DLAMX,C(80)),(DELALF,C(81)),(DELADT,C(82))
     2
     3,(WC,C(84)),(SIP,C(85)),(FC,C(86))
     4
      EQUIVALENCE (ALPHA,C(90)),(SA,C(91)),(CA,C(92)),(ST,C(93)),(CT,C(9
     14)),(SALT,C(95)),(CALT,C(96)),(GRAV,C(97)),(ALT,C(98)),(DRAG,C(99)
     2),(FVS,C(100)),(FMACH,C(101)),(VS,C(102)),(CD,C(103)),(QD,C(104)),
     3(RHO,C(105)),(PRES,C(106)),(AE,C(107)),(PDIFF,C(108)),(FRAC,C(109)
     4),(WGT,C(110)),(WDOT,C(111)),(F,C(112)),(A,C(113)),(PHI,C(114)),(C
     5HI,C(115)),(ISAVE1,C(116)),(ISAVE2,C(117)),(JJ,C(118)),(IMODE,C(11
     69)),(U,C(120)),(KA1,C(121)),(CL,C(122)),(ALIFT,C(123))
     7
     8,(TATD,C(130)),(ALTD,C(131)),(TOL1,C(132)),(TOL2,C(133)),(TOL3,C(1
     934))
      READ  (5,100)TOP
      WRITE (6,100)TOP
  100 FORMAT (12A6)
      DO 463 I=1,150
  463 C(I)=0.
      CALL FINP (4, ALTA, RHOTA, PRETA, FVSTA)
      CALL FINP (3,FMTA,CDTA,CLTA)
      KA1P=2
      KA1C=0
```

```
      STEPP=1.
      STEPC=5.
      A2=1.E-6
      TIMO=0.
      ALPHAP=0.
      G0=32.1849
      GRAV0=32.1849
      R0=20898906.
      PRES0=2124.214
      OMEGA=.7292115E-4
      PHIPR=28.28
      AZ=90.
      R=20898906.
      X=0.
      V=0.
      TATER=0.
      SAVPAP=0.
      IPRNT=0
      NPR=5
      STP=1.
      CASE2=0.
    1 CALL FINP (2,T,C)
      N=9
      COUNT=0.
      Z=57.295775
      ALFMAX=ALFMAX/Z
      ALPHAC=ALPHAC/Z
      DALF=DALF/Z
      DO 2 I=1,150
      C1(I)=C(I)
    2 T1(I)=T(I)
      TEST1=0.0
      TEST2=0.0
      TEST3=0.0
      TEST4=0.
      JGO1=1
      JGO2=2
      JGO3=3
      FAKTOL=0.
      FAKTST=0.
      CASE=0.
      IF (IOPT) 76,76,77
   76 FMNMX=-1.
      GO TO 78
   77 FMNMX=1.
   78 JOPT=IABS(IOPT)
      DADSAV=DELADT/Z
      DALSAV=DELALF/Z
   11 CONTINUE
      AZ=AZ/Z
      PHIPR=PHIPR/Z
      ALPHAP=ALPHAP/Z
      TATER=TATER/Z
      TATD=TATD/Z
      TOL1=TOL1/Z
      DELALF=DELALF/Z
      DELADT=DELADT/Z
      DLAMX=DLAMX/Z
    3 CONTINUE
      CALL FCALC
```

```
      IF (TEST4)40,16,40
   40 DO 41 I=1,150
      C(I)=C1(I)
   41 T(I)=T1(I)
      WRITE (6,105)
  105 FORMAT (1H1)
      CASE2=1.
      ALFMAX=ALFMAX*Z
      ALPHAC=ALPHAC*Z
      DALF=DALF*Z
      GO TO 1
   16 CONTINUE
      IF (TEST1) 20,20,21
   20 ALD1=C2(79)
      TAT1=TATER
      IF (TEST2) 17,17,18
   17 IF (TEST3) 18,19,18
   19 CONTINUE
      IF (CASE2) 80,80,18
   80 CONTINUE
      IF (TATER-TATD) 84,84,72
   84 DELADT=ABS(DELADT)
      GO TO 71
   72 DELADT=-ABS(DELADT)
      GO TO 71
   18 DELADT=(TATD-TATER)*ADINC/TATINC
      IF (DELADT) 71,82,71
   82 DELADT=DADSAV
      GO TO 80
   21 ADINC=C2(79)-ALD1
      TATINC=TATER-TAT1
   71 CONTINUE
      COUNT=COUNT+1.
      IF (COUNT-25.) 300,300,75
   75 WRITE (6,104)
  104 FORMAT (60H025 TRAJECTORIES WITHOUT 1ST ORDER CONVERGENCE. GUESS A
     1GAIN.)
      GO TO 40
  300 CALL ISO (TATER,C2(79),TATD,DELADT,TOL1,TEST1,JGO1)
      IF (TEST1) 3,24,3
   24 IF (TEST2) 26,26,27
   26 ALF1=C2(78)
      ALT1=ALT
      DAL1=C2(79)
      IF (TEST3) 28,12,28
   12 IF (CASE2) 81,81,28
   81 CONTINUE
      IF (ALT-ALTD) 73,85,85
   85 DELALF=ABS(DELALF)
      GO TO 74
   73 DELALF=-ABS(DELALF)
      GO TO 74
   28 DELALF=(ALTD-ALT)*ALINC/ALTINC
      IF (DELALF) 74,83,74
   83 DELALF=DALSAV
      GO TO 81
   27 ALINC=C2(78)-ALF1
      ALTINC=ALT-ALT1
      DALINC=C2(79)-DAL1
      ALFINC=C2(78)-ALF1
```

```
 74 CONTINUE
    WRITE (6,102)
102 FORMAT (22H0CONVERGED FIRST ORDER)
    COUNT=0.
    ASAV=C2(78)
    CALL ISO (ALT,C2(78),ALTD,DELALF,TOL2,TEST2,JGO2)
    IF (TEST2) 14,95,14
 14 CONTINUE
    IF (TEST3) 30,29,30
 29 FAKDEL=0.
    GO TO 4
 30 CONTINUE
    FAKDEL=(C2(78)-ASAV)*DALINC/ALFINC
  4 CONTINUE
    CALL ISO (ALT,C2(79),ALTD,FAKDEL,FAKTOL,FAKTST,JGO3)
    GO TO 3
 95 CB177=C177
    C177=C1(77)
    CB278=C278
    C278=C2(78)
    CB279=C279
    C279=C2(79)
    GO TO (67,68),JOPT
 67 FNC=V
    GO TO 69
 68 FNC=WGT
 69 CONTINUE
    CALL FMXMN (FNC,C1(77),FMNMX,DLAMX,STP,TOL3,TEST3,JGO1)
 13 WRITE (6,101)
101 FORMAT (23H0CONVERGED SECOND ORDER)
    FAKTST=0.
  5 DO 6 I=1,150
    C(I)=C1(I)
  6 T(I)=T1(I)
 91 IF (TEST3) 25,92,25
 25 IF (CASE) 22,23,22
 22 C1(78)=C2(78)+(C1(77)-C177)*(C278-CB278)/(C177-CB177)
    C(78)=C1(78)
    C1(79)=C2(79)+(C1(77)-C177)*(C279-CB279)/(C177-CB177)
    C(79)=C1(79)
    GO TO 11
 92 CONTINUE
    WRITE (6,103)
103 FORMAT (16H0CONVERGED STATE/25H0************************/1H1)
    SAVPAP=1.
    IPRNT=NPR
    TEST4=1.
    ADINC=C2(79)-ALD1
    TATINC=TATD/Z-TAT1
    ALINC=C2(78)-ALF1
    ALTINC=ALTD-ALT1
 23 CASE=1.
    C(79)=C2(79)
    C(78)=C2(78)
    C1(79)=C2(79)
    C1(78)=C2(78)
    GO TO 11
    END
```

```
$IBFTC FCALC
CFCALC SUBROUTINE TO RUN TRAJECTORIES
       SUBROUTINE FCALC
       DIMENSION T(150),C(150),ALTA(250),RHOTA(250),PRETA(250),FVSTA(250)
      1,FMTA(30),CDTA(30),CLTA(30)
       DIMENSION T1(150),C1(150)
       DIMENSION C2(150),T2(150)
       COMMON  T      , C
       COMMON ALTA,RHOTA,PRETA,FVSTA,FMTA,CDTA,CLTA
       COMMON C1,T1,TEST1,TEST2,C2,T2
       EQUIVALENCE (TIME,T(2)),(STEP,T(3)),(X,T(4)),(R,T(5)),(V,T(6)),
      1 (TATER,T(7)),(SAM,T(8)),(ALAM2,T(9)),(ALAM3,T(10)),(ALAM4,T(11)),
      2 (ALAM5,T(12)),(XDOT,T(13)),(RDOT,T(14)),(VDOT,T(15)),(TATDOT,
      3 T(16)),(SAMDOT,T(17)),(DLAM2,T(18)),(DLAM3,T(19)),(DLAM4,T(20)),
      4 (DLAM5,T(21))
       EQUIVALENCE (N,C(1)),(STEPP,C(2)),(KA1P,C(3)),(STEPC,C(4)),(KA1C,C
      1(5)),(A2,C(6)),(A3,C(7)),(A4,C(8)),(A5,C(9)),(A6,C(10)),(A7,C(11))
      2
      3,(G0,C(13)),(GRAV0,C(14)),(R0,C(15)),(PRES0,C(16)),(OMEGA,C(17)),(
      4PHIPR,C(18)),(AZ,C(19))
      5
      6,(SAVPAP,C(21)),(IPRNT,C(22)),(NPR,C(23)),(IOPT,C(24)),(STP,C(25))
      7
      8,(TIMPC,C(28)),(W0,C(29)),(WD0,C(30)),(F0,C(31)),(AE0,C(32)),(AREA
      9,C(33)),(TIM0,C(34))
       EQUIVALENCE (TIM1,C(35)),(TIM2,C(36)),(TIM3,C(37)),(TIM4,C(38)),(T
      1IMF1,C(39)),(TIMF2,C(40)),(TIMF3,C(41)),(TIMF4,C(42)),(F1,C(43)),(
      2F2,C(44)),(F3,C(45)),(F4,C(46)),(AE1,C(47)),(AE2,C(48)),(AE3,C(49)
      3),(AE4,C(50)),(TIMW1,C(51)),(TIMW2,C(52)),(TIMW3,C(53)),(TIMW4,C(5
      44)),(TIMW5,C(55)),(TIMW6,C(56)),(TIMW7,C(57)),(WD1,C(58)),(WD2,C(5
      59)),(WD3,C(60)),(WD4,C(61)),(WD5,C(62)),(WD6,C(63)),(WD7,C(64))
      6
      7,(ICUT,C(66)),(TIMCC,C(67)),(TOL,C(68)),(Z1,C(69)),(Z2,C(70)),(Z3,
      8C(71)),(Z4,C(72)),(Z5,C(73)),(Z6,C(74)),(R2,C(75))
      9
       EQUIVALENCE (ALPHAP,C(76)),(ALFMAX,C(77)),(ALPHAC,C(78)),(DALF,C(7
      19)),(DLAMX,C(80)),(DELALF,C(81)),(DELADT,C(82))
      2
      3,(WC,C(84)),(SIP,C(85)),(FC,C(86))
      4
       EQUIVALENCE (ALPHA,C(90)),(SA,C(91)),(CA,C(92)),(ST,C(93)),(CT,C(9
      14)),(SALT,C(95)),(CALT,C(96)),(GRAV,C(97)),(ALT,C(98)),(DRAG,C(99)
      2),(FVS,C(100)),(FMACH,C(101)),(VS,C(102)),(CD,C(103)),(QD,C(104)),
      3(RHO,C(105)),(PRES,C(106)),(AE,C(107)),(PDIFF,C(108)),(FRAC,C(109)
      4),(WGT,C(110)),(WDOT,C(111)),(F,C(112)),(A,C(113)),(PHI,C(114)),(C
      5HI,C(115)),(ISAVE1,C(116)),(ISAVE2,C(117)),(JJ,C(118)),(IMODE,C(11
      69)),(U,C(120)),(KA1,C(121)),(CL,C(122)),(ALIFT,C(123))
      7
      8,(TATD,C(130)),(ALTD,C(131)),(TOL1,C(132)),(TOL2,C(133)),(TOL3,C(1
      934))
       IF (TEST1) 4,5,4
     5 IF (TEST2) 4,8,4
     8 CONTINUE
       ITEST2=1
       STEP=STEPP
       KA1=KA1P
       WGT=W0
       SAM=WGT/G0
       TIME=TIM0
```

```
      WDOT=-WDO
      AE=AE0
      ALAM2=0.0
      ALAM3=0.0
      ALAM4=0.0
      ALAM5=0.0
      DLAM2=0.0
      DLAM3=0.0
      DLAM4=0.0
      DLAM5=0.0
      IF (SAVPAP) 7,7,6
    6 WRITE (6,100)
  100 FORMAT (47H0GIVEN AND PRECOMPUTED VALUES FOR LIFTOFF PHASE)
    7 IMODE=1
      CALL PRESET
      DO 13 I=1,150
      C2(I)=C(I)
   13 T2(I)=T(I)
    4 CONTINUE
      DO 14 I=1,150
      C(I)=C2(I)
   14 T(I)=T2(I)
      IMODE=2
      VELS=SQRT(V*V+2.0*OMEGA*R0*V*COS(PHIPR)*SIN(TATER)*SIN(AZ)+      OM
     1EGA*OMEGA*R0*R0*COS(PHIPR)*COS(PHIPR))
      THETAS=ATAN(SQRT(1.0-V*V*COS(TATER)*COS(TATER)/(VELS*VELS))/     (V
     1*COS(TATER)/VELS))
      V=VELS
      TATER=THETAS
      ALPHA=ALPHAC
      STEP=STEPC
      KA1=KA1C
      F=FC
      WGT=WC
      SAM=WGT/G0
      TIMO=TIME
      SA=SIN(ALPHA)
      CA=COS(ALPHA)
      ST=SIN(TATER)
      CT=COS(TATER)
      SALT=SIN(ALPHA+TATER)
      CALT=COS(ALPHA+TATER)
      FRAC=R0/R
      H=V*V
      WDOT=-F/SIP
      SAMDOT=WDOT/G0
      VDOT=F/SAM * CA -GRAV *CT
      TATDOT=F/(SAM*V)*SA+ST*GRAV/V-ST*V/R
      B=(H*SALT)/(V*DALF+GRAV*ST+SA*F/SAM-SA*H/R*CALT)
      IF (B) 11,12,12
   11 ALAM2=-1.0
      GO TO 15
   12 ALAM2=1.0
   15 ALAM3=ALAM2*B*CA/V
      ALAM4=ALAM2*B*SA
      ALAM5=0.0
      DLAM2=-2.0*ALAM3*GRAV/R*CT+ALAM4/R*ST*2.0*GRAV/V-ALAM4/R*ST*V/R
      DLAM3=-ALAM2*CT+ALAM4/V*((F*SA)/(SAM*V)+(GRAV/V+V/R)*ST)
      DLAM4=ALAM2*V*ST-ALAM3*GRAV*ST-ALAM4*GRAV/V*CT+ALAM4*V/R*CT
      DLAM5=F/SAM**2*(ALAM3*CA+ALAM4/V*SA)
```

```
      IF (SAVPAP) 10,10,9
    9 WRITE (6,200)
  200 FORMAT (62HOGIVEN AND PRECOMPUTED VALUES FOR CALCULUS OF VARIATION
     1S PHASE)
   10 CALL CALVAR
      WGT=SAM*GO
      ALT=R-RO
      RETURN
      END
```

```
$IBFTC PRESE
CPRESET SUBROUTINE FOR PRE SET ALPHA SEGMENT
      SUBROUTINE PRESET
      DIMENSION T(150),C(150),ALTA(250),RHOTA(250),PRETA(250),FVSTA(250)
     1,FMTA(30),CDTA(30),CLTA(30)
      DIMENSION C1(150),T1(150)
      DIMENSION C2(150),T2(150)
      COMMON  T      , C
      COMMON ALTA,RHOTA,PRETA,FVSTA,FMTA,CDTA,CLTA
      COMMON C1,T1,TEST1,TEST2,C2,T2
      EQUIVALENCE (TIME,T(2)),(STEP,T(3)),(X,T(4)),(R,T(5)),(V,T(6)),
     1 (TATER,T(7)),(SAM,T(8)),(ALAM2,T(9)),(ALAM3,T(10)),(ALAM4,T(11)),
     2 (ALAM5,T(12)),(XDOT,T(13)),(RDOT,T(14)),(VDOT,T(15)),(TATDOT,
     3 T(16)),(SAMDOT,T(17)),(DLAM2,T(18)),(DLAM3,T(19)),(DLAM4,T(20)),
     4 (DLAM5,T(21))
      EQUIVALENCE (N,C(1)),(STEPP,C(2)),(KA1P,C(3)),(STEPC,C(4)),(KA1C,C
     1(5)),(A2,C(6)),(A3,C(7)),(A4,C(8)),(A5,C(9)),(A6,C(10)),(A7,C(11))
     2
     3,(G0,C(13)),(GRAV0,C(14)),(R0,C(15)),(PRES0,C(16)),(OMEGA,C(17)),(
     4PHIPR,C(18)),(AZ,C(19))
     5
     6,(SAVPAP,C(21)),(IPRNT,C(22)),(NPR,C(23)),(IOPT,C(24)),(STP,C(25))
     7
     8,(TIMPC,C(28)),(W0,C(29)),(WD0,C(30)),(F0,C(31)),(AE0,C(32)),(AREA
     9,C(33)),(TIM0,C(34))
      EQUIVALENCE (TIM1,C(35)),(TIM2,C(36)),(TIM3,C(37)),(TIM4,C(38)),(T
     1IMF1,C(39)),(TIMF2,C(40)),(TIMF3,C(41)),(TIMF4,C(42)),(F1,C(43)),(
     2F2,C(44)),(F3,C(45)),(F4,C(46)),(AE1,C(47)),(AE2,C(48)),(AE3,C(49)
     3),(AE4,C(50)),(TIMW1,C(51)),(TIMW2,C(52)),(TIMW3,C(53)),(TIMW4,C(5
     44)),(TIMW5,C(55)),(TIMW6,C(56)),(TIMW7,C(57)),(WD1,C(58)),(WD2,C(5
     59)),(WD3,C(60)),(WD4,C(61)),(WD5,C(62)),(WD6,C(63)),(WD7,C(64))
     6
     7,(ICUT,C(66)),(TIMCC,C(67)),(TOL,C(68)),(Z1,C(69)),(Z2,C(70)),(Z3,
     8C(71)),(Z4,C(72)),(Z5,C(73)),(Z6,C(74)),(R2,C(75))
     9
      EQUIVALENCE (ALPHAP,C(76)),(ALFMAX,C(77)),(ALPHAC,C(78)),(DALF,C(7
     19)),(DLAMX,C(80)),(DELALF,C(81)),(DELADT,C(82))
     2
     3,(WC,C(84)),(SIP,C(85)),(FC,C(86))
     4
      EQUIVALENCE (ALPHA,C(90)),(SA,C(91)),(CA,C(92)),(ST,C(93)),(CT,C(9
     14)),(SALT,C(95)),(CALT,C(96)),(GRAV,C(97)),(ALT,C(98)),(DRAG,C(99)
     2),(FVS,C(100)),(FMACH,C(101)),(VS,C(102)),(CD,C(103)),(QD,C(104)),
     3(RHO,C(105)),(PRES,C(106)),(AE,C(107)),(PDIFF,C(108)),(FRAC,C(109)
     4),(WGT,C(110)),(WDOT,C(111)),(F,C(112)),(A,C(113)),(PHI,C(114)),(C
     5HI,C(115)),(ISAVE1,C(116)),(ISAVE2,C(117)),(JJ,C(118)),(IMODE,C(11
     69)),(U,C(120)),(KA1,C(121)),(CL,C(122)),(ALIFT,C(123))
     7
     8,(TATD,C(130)),(ALTD,C(131)),(TOL1,C(132)),(TOL2,C(133)),(TOL3,C(1
     934))
      ISAVE1=1
      ISAVE2=1
      CHOP=0.
      JJ=1
      IW=0
      IA=0
      ISTAGE=0
      TEMP=STEP
      KK=1
```

```
      IPRINT=IPRNT-1
      IF (TIM1) 60,60,61
   60 TMAX=TIMPC
      GO TO 62
   61 TMAX=TIM1
   62 IF (TIMF1) 63,63,64
   63 TMAXF=TIMPC
      GO TO 65
   64 TMAXF=TIMF1
   65 IF (TIMW1) 66,66,67
   66 TMAXW=TIMPC
      GO TO 6
   67 TMAXW=TIMW1
      GO TO 6
    7 GO TO (10,11,12),KK
   10 JJ=JJ+1
      IF (JJ-4) 51,51,22
   51 IF (C(JJ+34)) 22,22,21
   22 TMAX=TIMPC
      GO TO 6
   21 TMAX=C(JJ+34)
      GO TO 6
   11 IA=IA+1
      AE=C(IA+46)
      FO=C(IA+42)
      IF (IA-3) 52,52,24
   52 IF (C(IA+39)) 24,24,23
   24 TMAXF=TIMPC
      IA=3
      GO TO 6
   23 TMAXF=C(IA+39)
      GO TO 6
   12 IW=IW+1
      WDOT=-C(IW+57)
      IF (IW-6) 53,53,26
   53 IF (C(IW+51)) 26,26,25
   26 TMAXW=TIMPC
      IW=6
      GO TO 6
   25 TMAXW=C(IW+51)
    6 IF (TMAX.GT.TMAXF.OR.TMAX.GT.TMAXW) IF (TMAXF-TMAXW) 8,9,9
      TCHEK=TMAX
      KK=1
      GO TO 13
    8 TCHEK=TMAXF
      KK=2
      GO TO 13
    9 TCHEK=TMAXW
      KK=3
   13 STEP=TEMP
      CHOP=0.
      ISTAGE=1
    1 CALL INTG (T,N,KA1,A2,A3,A4,A5,A6,A7)
    2 IF (CHOP) 40,32,33
   40 TIME=TCHEK
      GO TO 7
   33 CHOP=-1.
      GO TO 14
   32 CONTINUE
      TOGO=TCHEK-TIME
```

```
      IF (TOGO-STEP) 30,14,14
   30 IF (TOGO) 5,7,5
    5 CONTINUE
      STEP=TOGO
      CHOP=1.
      GO TO 1
   14 IF (SAVPAP) 4,4,15
   15 IPRINT=IPRINT+1
      IF (IPRINT-IPRNT) 17,19,17
   17 IF (ISTAGE) 4,4,16
   19 IPRINT=0
   16 CALL OWT
      ISTAGE=0
    4 CALL INTM
      IF (AbS(TIME-TIMPC)-.0001) 18,2,2
   18 IF (SAVPAP) 20,20,29
   29 CALL OWT
   20 RETURN
      END
```

```
$IBFTC CALVA
CCALVAR SUBROUTINE TO RUN CALCULUS OF VARIATIONS TRAJECTORY
      SUBROUTINE CALVAR
      DIMENSION T(150),C(150),ALTA(250),RHOTA(250),PRETA(250),FVSTA(250)
     1,FMTA(30),CDTA(30),CLTA(30)
      DIMENSION C1(150),T1(150)
      DIMENSION C2(150),T2(150)
      COMMON  T      , C
      COMMON ALTA,RHOTA,PRETA,FVSTA,FMTA,CDTA,CLTA
      COMMON C1,T1,TEST1,TEST2,C2,T2
      EQUIVALENCE (TIME,T(2)),(STEP,T(3)),(X,T(4)),(R,T(5)),(V,T(6)),
     1 (TATER,T(7)),(SAM,T(8)),(ALAM2,T(9)),(ALAM3,T(10)),(ALAM4,T(11)),
     2 (ALAM5,T(12)),(XDOT,T(13)),(RDOT,T(14)),(VDOT,T(15)),(TATDOT,
     3 T(16)),(SAMDOT,T(17)),(DLAM2,T(18)),(DLAM3,T(19)),(DLAM4,T(20)),
     4 (DLAM5,T(21))
      EQUIVALENCE (N,C(1)),(STEPP,C(2)),(KA1P,C(3)),(STEPC,C(4)),(KA1C,C
     1(5)),(A2,C(6)),(A3,C(7)),(A4,C(8)),(A5,C(9)),(A6,C(10)),(A7,C(11))
     2
     3,(G0,C(13)),(GRAV0,C(14)),(R0,C(15)),(PRES0,C(16)),(OMEGA,C(17)),(
     4PHIPR,C(18)),(AZ,C(19))
     5
     6,(SAVPAP,C(21)),(IPRNT,C(22)),(NPR,C(23)),(IOPT,C(24)),(STP,C(25))
     7
     8,(TIMPC,C(28)),(W0,C(29)),(WD0,C(30)),(F0,C(31)),(AE0,C(32)),(AREA
     9,C(33)),(TIM0,C(34))
      EQUIVALENCE (TIM1,C(35)),(TIM2,C(36)),(TIM3,C(37)),(TIM4,C(38)),(T
     1IMF1,C(39)),(TIMF2,C(40)),(TIMF3,C(41)),(TIMF4,C(42)),(F1,C(43)),(
     2F2,C(44)),(F3,C(45)),(F4,C(46)),(AE1,C(47)),(AE2,C(48)),(AE3,C(49)
     3),(AE4,C(50)),(TIMW1,C(51)),(TIMW2,C(52)),(TIMW3,C(53)),(TIMW4,C(5
     44)),(TIMW5,C(55)),(TIMW6,C(56)),(TIMW7,C(57)),(WD1,C(58)),(WD2,C(5
     59)),(WD3,C(60)),(WD4,C(61)),(WD5,C(62)),(WD6,C(63)),(WD7,C(64))
     6
     7,(ICUT,C(66)),(TIMCC,C(67)),(TOL,C(68)),(Z1,C(69)),(Z2,C(70)),(Z3,
     8C(71)),(Z4,C(72)),(Z5,C(73)),(Z6,C(74)),(R2,C(75))
     9
      EQUIVALENCE (ALPHAP,C(76)),(ALFMAX,C(77)),(ALPHAC,C(78)),(DALF,C(7
     19)),(DLAMX,C(80)),(DELALF,C(81)),(DELADT,C(82))
     2
     3,(WC,C(84)),(SIP,C(85)),(FC,C(86))
     4
      EQUIVALENCE (ALPHA,C(90)),(SA,C(91)),(CA,C(92)),(ST,C(93)),(CT,C(9
     14)),(SALT,C(95)),(CALT,C(96)),(GRAV,C(97)),(ALT,C(98)),(DRAG,C(99)
     2),(FVS,C(100)),(FMACH,C(101)),(VS,C(102)),(CD,C(103)),(QD,C(104)),
     3(RHO,C(105)),(PRES,C(106)),(AE,C(107)),(PDIFF,C(108)),(FRAC,C(109)
     4),(WGT,C(110)),(WDOT,C(111)),(F,C(112)),(A,C(113)),(PHI,C(114)),(C
     5HI,C(115)),(ISAVE1,C(116)),(ISAVE2,C(117)),(JJ,C(118)),(IMODE,C(11
     69)),(U,C(120)),(KA1,C(121)),(CL,C(122)),(ALIFT,C(123))
     7
     8,(TATD,C(130)),(ALTD,C(131)),(TOL1,C(132)),(TOL2,C(133)),(TOL3,C(1
     934))
      IPRINT=IPRNT
      INISH=IPRINT-1
      SEARCH=1.0
      CUTOFF=TIMCC
      JET=1
      TLAST=TIME-1.
    1 CONTINUE
      GO TO (31,32),ICUT
   32 CONTINUE
```

```
      VALUE=TIME
      RATE=1.0
      GO TO 33
   31 CONTINUE
      VALUE=V
      RATE=VDOT
      CUTOFF=Z1*SQRT(((1.0+Z2)*Z3*(1.0+(Z4*R)/R2))/((1.0+(Z4*R*R)/(R2*R2
     1)*ST*ST)*R)+Z5)+Z6
   33 CONTINUE
    9 CONTINUE
      IF (ABS(VALUE -CUTOFF)-TOL) 11,12,12
   12 IF (SEARCH) 14,14,13
   13 GO TO (15,16),JET
   15 CALL INTG (T,N,KA1,A2,A3,A4,A5,A6,A7)
      WHICH=(VALUE-CUTOFF)/ABS(VALUE-CUTOFF)
      JET=2
   16 CHANGE=(VALUE-CUTOFF)/ABS(VALUE-CUTOFF)
      IF(CHANGE/WHICH)14,17,17
   14 STEP=(CUTOFF-VALUE)/RATE
      SEARCH=0.0
      KA1=1
      GO TO 15
   17 IF (SAVPAP) 18,18,42
   42 IF ((TIME-TLAST)*SEARCH) 18,18,43
   43 INISH=INISH+1
      IF (IPRINT-INISH) 18,19,18
   19 CALL OWT
      INISH=0
   18 TLAST=TIME
      CALL INTM
      GO TO 1
   11 IF (SAVPAP) 35,35,34
   34 CALL OWT
   35 CONTINUE
      RETURN
      END
```

```
$IBFTC DAUX
CDAUX SUBROUTINE TO EVALUATE DERIVATIVES
      SUBROUTINE DAUX
      DIMENSION T(150),C(150),ALTA(250),RHOTA(250),PRETA(250),FVSTA(250)
     1,FMTA(30),CDTA(30),CLTA(30)
      DIMENSION C1(150),T1(150)
      DIMENSION C2(150),T2(150)
      COMMON  T      , C
      COMMON ALTA,RHOTA,PRETA,FVSTA,FMTA,CDTA,CLTA
      COMMON C1,T1,TEST1,TEST2,C2,T2
      EQUIVALENCE (TIME,T(2)),(STEP,T(3)),(X,T(4)),(R,T(5)),(V,T(6)),
     1 (TATER,T(7)),(SAM,T(8)),(ALAM2,T(9)),(ALAM3,T(10)),(ALAM4,T(11)),
     2 (ALAM5,T(12)),(XDOT,T(13)),(RDOT,T(14)),(VDOT,T(15)),(TATDOT,
     3 T(16)),(SAMDOT,T(17)),(DLAM2,T(18)),(DLAM3,T(19)),(DLAM4,T(20)),
     4 (DLAM5,T(21))
      EQUIVALENCE (N,C(1)),(STEPP,C(2)),(KA1P,C(3)),(STEPC,C(4)),(KA1C,C
     1(5)),(A2,C(6)),(A3,C(7)),(A4,C(8)),(A5,C(9)),(A6,C(10)),(A7,C(11))
     2
     3,(G0,C(13)),(GRAV0,C(14)),(R0,C(15)),(PRES0,C(16)),(OMEGA,C(17)),(
     4PHIPR,C(18)),(AZ,C(19))
     5
     6,(SAVPAP,C(21)),(IPRNT,C(22)),(NPR,C(23)),(IOPT,C(24)),(STP,C(25))
     7
     8,(TIMPC,C(28)),(W0,C(29)),(WD0,C(30)),(F0,C(31)),(AE0,C(32)),(AREA
     9,C(33)),(TIM0,C(34))
      EQUIVALENCE (TIM1,C(35)),(TIM2,C(36)),(TIM3,C(37)),(TIM4,C(38)),(T
     1IMF1,C(39)),(TIMF2,C(40)),(TIMF3,C(41)),(TIMF4,C(42)),(F1,C(43)),(
     2F2,C(44)),(F3,C(45)),(F4,C(46)),(AE1,C(47)),(AE2,C(48)),(AE3,C(49)
     3),(AE4,C(50)),(TIMW1,C(51)),(TIMW2,C(52)),(TIMW3,C(53)),(TIMW4,C(5
     44)),(TIMW5,C(55)),(TIMW6,C(56)),(TIMW7,C(57)),(WD1,C(58)),(WD2,C(5
     59)),(WD3,C(60)),(WD4,C(61)),(WD5,C(62)),(WD6,C(63)),(WD7,C(64))
     6
     7,(ICUT,C(66)),(TIMCC,C(67)),(TOL,C(68)),(Z1,C(69)),(Z2,C(70)),(Z3,
     8C(71)),(Z4,C(72)),(Z5,C(73)),(Z6,C(74)),(R2,C(75))
     9
      EQUIVALENCE (ALPHAP,C(76)),(ALFMAX,C(77)),(ALPHAC,C(78)),(DALF,C(7
     19)),(DLAMX,C(80)),(DELALF,C(81)),(DELADT,C(82))
     2
     3,(WC,C(84)),(SIP,C(85)),(FC,C(86))
     4
      EQUIVALENCE (ALPHA,C(90)),(SA,C(91)),(CA,C(92)),(ST,C(93)),(CT,C(9
     14)),(SALT,C(95)),(CALT,C(96)),(GRAV,C(97)),(ALT,C(98)),(DRAG,C(99)
     2),(FVS,C(100)),(FMACH,C(101)),(VS,C(102)),(CD,C(103)),(QD,C(104)),
     3(RHO,C(105)),(PRES,C(106)),(AE,C(107)),(PDIFF,C(108)),(FRAC,C(109)
     4),(WGT,C(110)),(WDOT,C(111)),(F,C(112)),(A,C(113)),(PHI,C(114)),(C
     5HI,C(115)),(ISAVE1,C(116)),(ISAVE2,C(117)),(JJ,C(118)),(IMODE,C(11
     69)),(U,C(120)),(KA1,C(121)),(CL,C(122)),(ALIFT,C(123))
     7
     8,(TATD,C(130)),(ALTD,C(131)),(TOL1,C(132)),(TOL2,C(133)),(TOL3,C(1
     934))
      GO TO (1,2), IMODE
    1 CONTINUE
      GO TO (13,20,14,30,13),JJ
   13 ALPHA=ALPHAP
      GO TO 3
   20 ALPHA=ALPHAP+(ALFMAX-ALPHAP)*(TIME-TIM1)/(TIM2-TIM1)
      GO TO 3
   14 ALPHA=ALFMAX
      GO TO 3
```

```
30 ALPHA=ALPHAP+(ALFMAX-ALPHAP)*(TIM4-TIME)/(TIM4-TIM3)
 3 SA=SIN(ALPHA)
   CA=COS(ALPHA)
   GO TO 4
 2 CONTINUE
   U=SQRT(ALAM4**2+V*V*ALAM3**2)
   SA=ALAM4/U
   CA=ALAM3*V/U
   ALPHA=ATAN2(SA,CA)
 4 ST=SIN(TATER)
   CT=COS(TATER)
   FRAC=RO/R
   GRAV=GRAVO*FRAC*FRAC
   XDOT=FRAC*V*ST
   RDOT=V*CT
   GO TO (5,9), IMODE
 5 CONTINUE
   ALT=R-RO
   SAMDOT=WDOT/GO
   CALL TABLE
   PDIFF=PRESO-PRES
   F=FO+AE*PDIFF
   VDOT=(F*CA-DRAG)/SAM-GRAV*CT
   IF (V) 7, 6, 7
 6 TATDOT=0.0
   GO TO 8
 7 TATDOT=(F*SA+ALIFT)/(SAM*V)+(GRAV/V-V/R)*ST
 8 RETURN
 9 SAMDOT=-F/SIP/GO
   VDOT=F/SAM * CA -GRAV *CT
   TATDOT=F/(SAM*V)*SA+ST*GRAV/V-ST*V/R
   DLAM2=-2.0*ALAM3*GRAV/R*CT+ALAM4/R*ST*2.0*GRAV/V-ALAM4/R*ST*V/R
   DLAM3=-ALAM2*CT+ALAM4/V*((F*SA)/(SAM*V)+(GRAV/V+V/R)*ST)
   DLAM4=ALAM2*V*ST-ALAM3*GRAV*ST-ALAM4*GRAV/V*CT+ALAM4*V/R*CT
   DLAM5=F/SAM**2*(ALAM3*CA+ALAM4/V*SA)
   RETURN
   END
```

```
$IBFTC OWT
COWT  OUTPUT SUBROUTINE
      SUBROUTINE OWT
      DIMENSION T(150),C(150),ALTA(250),RHOTA(250),PRETA(250),FVSTA(250)
     1,FMTA(30),CDTA(30),CLTA(30)
      DIMENSION C1(150),T1(150)
      DIMENSION C2(150),T2(150)
      COMMON  T       , C
      COMMON ALTA,RHOTA,PRETA,FVSTA,FMTA,CDTA,CLTA
      COMMON C1,T1,TEST1,TEST2,C2,T2
      EQUIVALENCE (TIME,T(2)),(STEP,T(3)),(X,T(4)),(R,T(5)),(V,T(6)),
     1 (TATER,T(7)),(SAM,T(8)),(ALAM2,T(9)),(ALAM3,T(10)),(ALAM4,T(11)),
     2 (ALAM5,T(12)),(XDOT,T(13)),(RDOT,T(14)),(VDOT,T(15)),(TATDOT,
     3 T(16)),(SAMDOT,T(17)),(DLAM2,T(18)),(DLAM3,T(19)),(DLAM4,T(20)),
     4 (DLAM5,T(21))
      EQUIVALENCE (N,C(1)),(STEPP,C(2)),(KA1P,C(3)),(STEPC,C(4)),(KA1C,C
     1(5)),(A2,C(6)),(A3,C(7)),(A4,C(8)),(A5,C(9)),(A6,C(10)),(A7,C(11))
     2
     3,(G0,C(13)),(GRAV0,C(14)),(R0,C(15)),(PRES0,C(16)),(OMEGA,C(17)),(
     4PHIPR,C(18)),(AZ,C(19))
     5
     6,(SAVPAP,C(21)),(IPRNT,C(22)),(NPR,C(23)),(IOPT,C(24)),(STP,C(25))
     7
     8,(TIMPC,C(28)),(W0,C(29)),(WD0,C(30)),(F0,C(31)),(AE0,C(32)),(AREA
     9,C(33)),(TIM0,C(34))
      EQUIVALENCE (TIM1,C(35)),(TIM2,C(36)),(TIM3,C(37)),(TIM4,C(38)),(T
     1IMF1,C(39)),(TIMF2,C(40)),(TIMF3,C(41)),(TIMF4,C(42)),(F1,C(43)),(
     2F2,C(44)),(F3,C(45)),(F4,C(46)),(AE1,C(47)),(AE2,C(48)),(AE3,C(49)
     3),(AE4,C(50)),(TIMW1,C(51)),(TIMW2,C(52)),(TIMW3,C(53)),(TIMW4,C(5
     44)),(TIMW5,C(55)),(TIMW6,C(56)),(TIMW7,C(57)),(WD1,C(58)),(WD2,C(5
     59)),(WD3,C(60)),(WD4,C(61)),(WD5,C(62)),(WD6,C(63)),(WD7,C(64))
     6
     7,(ICUT,C(66)),(TIMCC,C(67)),(TOL,C(68)),(Z1,C(69)),(Z2,C(70)),(Z3,
     8C(71)),(Z4,C(72)),(Z5,C(73)),(Z6,C(74)),(R2,C(75))
     9
      EQUIVALENCE (ALPHAP,C(76)),(ALFMAX,C(77)),(ALPHAC,C(78)),(DALF,C(7
     19)),(DLAMX,C(80)),(DELALF,C(81)),(DELADT,C(82))
     2
     3,(WC,C(84)),(SIP,C(85)),(FC,C(86))
     4
      EQUIVALENCE (ALPHA,C(90)),(SA,C(91)),(CA,C(92)),(ST,C(93)),(CT,C(9
     14)),(SALT,C(95)),(CALT,C(96)),(GRAV,C(97)),(ALT,C(98)),(DRAG,C(99)
     2),(FVS,C(100)),(FMACH,C(101)),(VS,C(102)),(CD,C(103)),(QD,C(104)),
     3(RHO,C(105)),(PRES,C(106)),(AE,C(107)),(PDIFF,C(108)),(FRAC,C(109)
     4),(WGT,C(110)),(WDOT,C(111)),(F,C(112)),(A,C(113)),(PHI,C(114)),(C
     5HI,C(115)),(ISAVE1,C(116)),(ISAVE2,C(117)),(JJ,C(118)),(IMODE,C(11
     69)),(U,C(120)),(KA1,C(121)),(CL,C(122)),(ALIFT,C(123))
     7
     8,(TATD,C(130)),(ALTD,C(131)),(TOL1,C(132)),(TOL2,C(133)),(TOL3,C(1
     934))
      Z=57.295775
      E2=TATER*Z
      E3=TATDOT*Z
      E4=ALPHA*Z
      PHI=X*Z/R0
      CHI=E2+E4+PHI
      WGT=SAM*G0
      GO TO (1,2),IMODE
    1 CONTINUE
```

```
      GO TO (3,4,3,5,3),JJ
    3 ADOT=0.
      GO TO 6
    4 ADOT=(ALFMAX-ALPHAP)/(TIM2-TIM1)
      GO TO 6
    5 ADOT=(ALFMAX-ALPHAP)/(TIM3-TIM4)
      GO TO 6
    2 ALT=R-R0
      DALF=ALAM2*V*SALT/U-GRAV*ST/V+SA*V*CALT/R-SA*F/(V*SAM)
      ADOT=DALF
      A=ALAM2*RDOT+ALAM3*VDOT+ALAM4*TATDOT+ALAM5*SAMDOT
    6 ADOT=ADOT*Z
      WRITE (6,100) X,XDOT,TIME,E4,ADOT,R,RDOT,ALT,WGT,WDOT,V,VDOT,F,SAM
     1              ,SAMDOT,E2,E3,GRAV,PHI,CHI
      GO TO (11,12),IMODE
   11 WRITE (6,200) RHO,PRES,VS,DRAG,ALIFT,QD,AE,FMACH,CD,CL
      GO TO 13
   12 WRITE (6,300)   SIP,ALAM2,ALAM3,ALAM4,ALAM5,A,DLAM2,DLAM3,DLAM4,
     1              DLAM5
   13 RETURN
  100 FORMAT (8H0XXX    E14.7,12H     XDOT   E14.7,12H     TIME   E14.7,
     112H     ALPHA  E14.7,12H     ALDOT  E14.7/8H RRR    E14.7,12H
     2RDOT   E14.7,12H     ALT    E14.7,12H     WLBS   E14.7,12H     WDO
     3T   E14.7/8H VEL    E14.7,12H     VDOT   E14.7,12H     THRST  E14.
     47,12H     MASS   E14.7,12H     MDOT   E14.7/8H THETA  E14.7,12H
     5  THDOT  E14.7,12H     GRAV   E14.7,12H     PHI    E14.7,12H     C
     6HI    E14.7)
  200 FORMAT (8H RHO    E14.7,12H     PRES   E14.7,12H     VS     E14.7,
     112H     DRAG   E14.7,12H     LIFT   E14.7/8H Q      E14.7,12H
     2AEX    E14.7,12H     MACH   E14.7,12H     CD     E14.7,12H     CL
     3    E14.7)
  300 FORMAT (8H ISP    E14.7,12H     LAMB2  E14.7,12H     LAMB3  E14.7,
     112H     LAMB4  E14.7,12H     LAMB5  E14.7/8H AAA    E14.7,12H
     2DLAM2  E14.7,12H     DLAM3  E14.7,12H     DLAM4  E14.7,12H     DLA
     3M5  E14.7)
      END
```

```
$IBFTC TABLE
CTABLE SUBROUTINE FOR LOOK UP AND AERODYNAMIC PARAMETERS
      SUBROUTINE TABLE
      DIMENSION T(150),C(150),ALTA(250),RHOTA(250),PRETA(250),FVSTA(250)
     1,FMTA(30),CDTA(30),CLTA(30)
      DIMENSION C1(150),T1(150)
      DIMENSION C2(150),T2(150)
      COMMON  T      , C
      COMMON ALTA,RHOTA,PRETA,FVSTA,FMTA,CDTA,CLTA
      COMMON C1,T1,TEST1,TEST2,C2,T2
      EQUIVALENCE (TIME,T(2)),(STEP,T(3)),(X,T(4)),(R,T(5)),(V,T(6)),
     1 (TATER,T(7)),(SAM,T(8)),(ALAM2,T(9)),(ALAM3,T(10)),(ALAM4,T(11)),
     2 (ALAM5,T(12)),(XDOT,T(13)),(RDOT,T(14)),(VDOT,T(15)),(TATDOT,
     3 T(16)),(SAMDOT,T(17)),(DLAM2,T(18)),(DLAM3,T(19)),(DLAM4,T(20)),
     4 (DLAM5,T(21))
      EQUIVALENCE (N,C(1)),(STEPP,C(2)),(KA1P,C(3)),(STEPC,C(4)),(KA1C,C
     1(5)),(A2,C(6)),(A3,C(7)),(A4,C(8)),(A5,C(9)),(A6,C(10)),(A7,C(11))
     2
     3,(G0,C(13)),(GRAV0,C(14)),(R0,C(15)),(PRES0,C(16)),(OMEGA,C(17)),(
     4PHIPR,C(18)),(AZ,C(19))
     5
     6,(SAVPAP,C(21)),(IPRNT,C(22)),(NPR,C(23)),(IOPT,C(24)),(STP,C(25))
     7
     8,(TIMPC,C(28)),(W0,C(29)),(WD0,C(30)),(F0,C(31)),(AE0,C(32)),(AREA
     9,C(33)),(TIM0,C(34))
      EQUIVALENCE (TIM1,C(35)),(TIM2,C(36)),(TIM3,C(37)),(TIM4,C(38)),(T
     1IMF1,C(39)),(TIMF2,C(40)),(TIMF3,C(41)),(TIMF4,C(42)),(F1,C(43)),(
     2F2,C(44)),(F3,C(45)),(F4,C(46)),(AE1,C(47)),(AE2,C(48)),(AE3,C(49)
     3),(AE4,C(50)),(TIMW1,C(51)),(TIMW2,C(52)),(TIMW3,C(53)),(TIMW4,C(5
     44)),(TIMW5,C(55)),(TIMW6,C(56)),(TIMW7,C(57)),(WD1,C(58)),(WD2,C(5
     59)),(WD3,C(60)),(WD4,C(61)),(WD5,C(62)),(WD6,C(63)),(WD7,C(64))
     6
     7,(ICUT,C(66)),(TIMCC,C(67)),(TOL,C(68)),(Z1,C(69)),(Z2,C(70)),(Z3,
     8C(71)),(Z4,C(72)),(Z5,C(73)),(Z6,C(74)),(R2,C(75))
     9
      EQUIVALENCE (ALPHAP,C(76)),(ALFMAX,C(77)),(ALPHAC,C(78)),(DALF,C(7
     19)),(DLAMX,C(80)),(DELALF,C(81)),(DELADT,C(82))
     2
     3,(WC,C(84)),(SIP,C(85)),(FC,C(86))
     4
      EQUIVALENCE (ALPHA,C(90)),(SA,C(91)),(CA,C(92)),(ST,C(93)),(CT,C(9
     14)),(SALT,C(95)),(CALT,C(96)),(GRAV,C(97)),(ALT,C(98)),(DRAG,C(99)
     2),(FVS,C(100)),(FMACH,C(101)),(VS,C(102)),(CD,C(103)),(QD,C(104)),
     3(RHO,C(105)),(PRES,C(106)),(AE,C(107)),(PDIFF,C(108)),(FRAC,C(109)
     4),(WGT,C(110)),(VDOT,C(111)),(F,C(112)),(A,C(113)),(PHI,C(114)),(C
     5HI,C(115)),(ISAVE1,C(116)),(ISAVE2,C(117)),(JJ,C(118)),(IMODE,C(11
     69)),(U,C(120)),(KA1,C(121)),(CL,C(122)),(ALIFT,C(123))
     7
     8,(TATD,C(130)),(ALTD,C(131)),(TOL1,C(132)),(TOL2,C(133)),(TOL3,C(1
     934))
      GLOM=1.0548599/G0
      I=ISAVE1
    3 IF (ALT) 100, 110, 110
  100 I=1
      IL=1
      IU=2
      GO TO 10
  110 IF (ALT-ALTA(I)) 7, 10, 4
    4 IF (I-250) 5,6,6
```

```
  5 I=I+1
    GO TO 110
  6 I=250
    IL=249
    IU=250
    GO TO 10
  7 I=I-1
    IF (ALT-ALTA(I)) 7,10,11
 10 FVS=FVSTA(I)
    RHO=RHOTA(I)*GLOM
    PRES=PRETA(I)
    IF (I-250) 1,12,6
  1 IF (I-1) 100,12,41
 41 IL=I-1
    IU=I+1
    GO TO 12
 11 IL=I
    IU=I+1
    FRAC1=(ALT-ALTA(IL))/(ALTA(IU)-ALTA(IL))
    FVS=FRAC1*(FVSTA(IU)-FVSTA(IL))+FVSTA(IL)
    RHO=(FRAC1*(RHOTA(IU)-RHOTA(IL))+RHOTA(IL))*GLOM
    PRES=FRAC1*(PRETA(IU)-PRETA(IL))+PRETA(IL)
 12 ISAVE1=I
    FMACH=V*FVS
    J=ISAVE2
 16 IF (FMACH) 200, 210, 210
200 J=1
    JL=1
    JU=2
    GO TO 23
210 IF (FMACH-FMTA(J)) 20, 23, 17
 17 IF (J-30) 18,19,19
 18 J=J+1
    GO TO 210
 19 J=30
    JL=29
    JU=30
    GO TO 23
 20 J=J-1
    IF (FMACH-FMTA(J)) 20,23,24
 23 CD=CDTA(J)
    CL=CLTA(J)
    IF (J-30) 14,25,19
 14 IF (J-1) 200,25,51
 51 JL=J-1
    JU=J+1
    GO TO 25
 24 JL=J
    JU=J+1
    K=JL
    FRAC2=(FMACH-FMTA(JL))/(FMTA(JU)-FMTA(JL))
    CD=FRAC2*(CDTA(JU)-CDTA(JL))+CDTA(JL)
    CL=FRAC2*(CLTA(JU)-CLTA(JL))+CLTA(JL)
 25 ISAVE2=J
    VS=1.0/FVS
    QD=.5*RHO*V*V
    DRAG=AREA*QD*(CD*CA+ALPHA*CL*SA)
    ALIFT=AREA*QD*(-CD*SA+ALPHA*CL*CA)
    RETURN
    END
```

```
$IBFTC MATS
      SUBROUTINE MATS(A,X,N,M,NSING)
      DIMENSION A(5,6),X(5,1)
      MM=N+M
      DO 15 I=2,N
   70 II=I-1
    7 DO 15 J=1,II
    8 IF(A(I,J))9,15,9
    9 IF(ABS(A(J,J))-ABS(A(I,J)))11,10,10
   10 R=A(I,J)/A(J,J)
      GO TO 130
   11 R=A(J,J)/A(I,J)
      DO 12 K=1,MM
      B=A(J,K)
      A(J,K)=A(I,K)
   12 A(I,K)=B
  130 JJ=J+1
   13 DO 14 K=JJ,MM
   14 A(I,K)=A(I,K)-R*A(J,K)
   15 CONTINUE
      IF( ABS(A(N,N))-1.E-08)16,16,17
   16 NSING=1
      RETURN
   17 DO 28 J=1,M
      KK=N+J
      X(N,J)=A(N,KK)/A(N,N)
      DO 28 I=2,N
      JJ=N-I+1
      B=0.0
      II=N-I+2
      DO 25 K=II,N
   25 B=B+A(JJ,K)*X(K,J)
      IF( ABS(A(JJ,JJ))-1.E-08)16,16,28
   28 X(JJ,J)=(A(JJ,KK)-B)/A(JJ,JJ)
      NSING=0
      RETURN
      END
```

```
$IBFTC ISO
CISO
      SUBROUTINE ISO (Q1,Q2,Q3,Q4,Q5,Q6,IQ7)
      DIMENSION X(15),Y(15),Q(5,6),CO(5,1),ICNT(3),R1(3),R2(3),DVAL(3)
      VAL=Q1
      VARY=Q2
      VALD=Q3
      FINC=Q4
      TOL=Q5
      TEST=Q6
      JGO=IQ7
      ICOUNT=ICNT(JGO)
      K=5*JGO
      KO=K-4
      IF (ABS(VAL-VALD)-TOL) 54,54,4
    4 IF (TEST) 1,3,1
    3 ICOUNT=0
      TEST=1.0
      M=1
      R1(JGO)=ABS(1./VARY)
      R2(JGO)=ABS(1./VAL)
      X(KO)=VARY*R1(JGO)
      Y(KO)=VAL*R2(JGO)
      VARY=X(KO)
      VAL=Y(KO)
      DVAL(JGO)=VALD*R2(JGO)
      GO TO 53
    1 X(K)=X(K-1)
      Y(K)=Y(K-1)
      K=K-1
      IF (K-KO) 2,2,1
    2 X(K)=VARY*R1(JGO)
      Y(K)=VAL*R2(JGO)
      K1=K+1
      ICOUNT=ICOUNT+1
      IF (ICOUNT-4) 6,6,5
    5 ICOUNT=4
    6 CONTINUE
      VARY=X(K)
      VAL=Y(K)
      GO TO (12,23,25,31),ICOUNT
   12 KANT=0
   11 VINC=(DVAL(JGO)-VAL)*(VARY-X(K1))/(VAL-Y(K1))
      VARY=VARY+VINC
      IF (KANT) 51,51,22
   22 IF (ABS(V1-VARY)-ABS(V2-VARY)) 16,16,17
   16 VARY=V1
      GO TO 52
   17 VARY=V2
   52 IF ((VARY-X(K))/VINC) 27,27,51
   27 VARY=X(K)+VINC
      GO TO 51
   23 N=3
      DO 24 I=1,N
      L=K+I-1
      Q(I,1)=X(L)**2
      Q(I,2)=X(L)
      Q(I,3)=Y(L)
   24 Q(I,4)=1.0
```

```
      CALL MATS (Q,CO,N,M,NSING)
      IF (NSING) 71,71,12
   71 CONTINUE
      A=CO(1,1)
      B=0.
      C=0.
      D=CO(2,1)
      E=CO(3,1)
      KANT=1
      GO TO 19
   25 N=4
      DO 26 I=1,N
      L=K+I-1
      Q(I,1)=X(L)**2
      Q(I,2)=X(L)*Y(L)
      Q(I,3)=Y(L)**2
      Q(I,4)=X(L)
   26 Q(I,5)=1.0
      CALL MATS (Q,CO,N,M,NSING)
      IF (NSING) 72,72,10
   72 CONTINUE
      A=CO(1,1)
      B=CO(2,1)
      C=CO(3,1)
      D=CO(4,1)
      E=0.
      KANT=2
      GO TO 19
   10 N=4
      DO 13 I=1,N
      L=K+I-1
      Q(I,1)=X(L)**2
      Q(I,2)=X(L)*Y(L)
      Q(I,3)=Y(L)**2
      Q(I,4)=Y(L)
   13 Q(I,5)=1.
      CALL MATS (Q,CO,N,M,NSING)
      IF (NSING) 73,73,23
   73 CONTINUE
      A=CO(1,1)
      B=CO(2,1)
      C=CO(3,1)
      D=0.
      E=CO(4,1)
      KANT=3
      GO TO 19
   31 N=5
      DO 32 I=1,N
      L=K+I-1
      Q(I,1)=X(L)**2
      Q(I,2)=X(L)*Y(L)
      Q(I,3)=Y(L)**2
      Q(I,4)=X(L)
      Q(I,5)=Y(L)
   32 Q(I,6)=1.0
      CALL MATS (Q,CO,N,M,NSING)
      IF (NSING) 74,74,25
   74 CONTINUE
      A=CO(1,1)
      B=CO(2,1)
```

```
      C=CO(3,1)
      D=CO(4,1)
      E=CO(5,1)
      KANT=4
   19 CONTINUE
      BB=B*DVAL(JGO)+D
      CC=C*DVAL(JGO)**2+E*DVAL(JGO)-1.
      DSCRIM=BB**2-4.*A*CC
      IF (DSCRIM) 7,8,8
    8 FORT=SQRT(DSCRIM)
      V1=(-BB+FORT)/(2.0*A)
      V2=(-BB-FORT)/(2.0*A)
      GO TO 11
    7 CONTINUE
      GO TO (9,10,23,25),KANT
    9 VARY=-.5*D/A
      GO TO 51
   53 VARY=VARY+FINC*R1(JGO)
   51 CONTINUE
      VARY=VARY/R1(JGO)
      VAL=VAL/R2(JGO)
   56 CONTINUE
      ICNT(JGO)=ICOUNT
      Q2=VARY
      Q6=TEST
      RETURN
   54 TEST=0.0
      GO TO 56
      END
```

```
$IBFTC FMXMN
CFMXMN    SUBROUTINE FMXMN
      SUBROUTINE FMXMN(VAL,VARY,FMNMX,FINC,STPINC,TOL,TEST,J)
      DIMENSION X(15),Y(15),Q(5,6),CO(5,2),ICNT(3),W1(3),W2(3)
      DIMENSION FKTR(3),R1(3),R2(3)
      ICOUNT=ICNT(J)
      Z1=W1(J)
      Z2=W2(J)
      K1=5*J-4
      K2=K1+1
      K3=K2+1
      K4=K3+1
      K5=K4+1
      IF (TEST) 5,3,5
    3 SC1=1.0
      Z1=0.0
      Z2=0.0
      TEST=1.0
      ICOUNT=-1
      M=1
      R1(J)=ABS(1./VARY)
      R2(J)=ABS(1./VAL)
      X(K1)=VARY*R1(J)
      Y(K1)=VAL*R2(J)
      VARY=X(K1)
      VAL=Y(K1)
      GO TO 52
    5 SC1=FKTR(J)
      VARY=VARY*R1(J)
      VAL=VAL*R2(J)
      IF (ICOUNT-3) 4,6,6
    4 ICOUNT=ICOUNT+1
    6 IF (Z2) 35,7,7
    7 K=5*J
      KO=K-4
    1 X(K)=X(K-1)
      Y(K)=Y(K-1)
      K=K-1
      IF(K-KO)2,2,1
    2 X(K1)=VARY
      Y(K1)=VAL
      IF (Z1) 9,9,8
    8 SC1=SC1/2.0
      Z1=0.0
    9 IF (FMNMX*(Y(K1)-Y(K2)))10,10,50
   10 Y(K1)=Y(K2)
      Y(K2)=VAL
      X(K1)=X(K2)
      X(K2)=VARY
      IF(Z2) 11,11,12
   11 SC1=-2.0*SC1
      Z1=1.0
      Z2=1.0
      GO TO 52
   12 Z2=-1
      GO TO 22
   35 CONTINUE
      DO 55 I=1,4
      L=K1+I-1
```

```
   CO(I,1)=X(L)
55 CO(I,2)=Y(L)
   IF ((X(K1)-X(K2))/(Y(K1)-VARY)) 14,14,17
14 IF (FMNMX*(VAL-Y(K1))) 16,16,15
15 CO(2,1)=X(K1)
   CO(2,2)=Y(K1)
   CO(4,1)=X(K2)
   CO(4,2)=Y(K2)
   GO TO 19
16 CO(4,1)=X(K3)
   CO(4,2)=Y(K3)
   IF (FMNMX*(VAL-Y(K2))) 56,56,57
56 CO(3,1)=VARY
   CO(3,2)=VAL
   GO TO 21
57 CO(2,1)=VARY
   CO(2,2)=VAL
   CO(3,1)=X(K2)
   CO(3,2)=Y(K2)
   GO TO 21
17 IF(FMNMX*(VAL-Y(K1))) 20,20,18
18 CO(3,1)=X(K2)
   CO(3,2)=Y(K2)
   CO(2,1)=X(K1)
   CO(2,2)=Y(K1)
   CO(4,1)=X(K3)
   CO(4,2)=Y(K3)
19 CO(1,1)=VARY
   CO(1,2)=VAL
   GO TO 21
20 CO(2,1)=VARY
   CO(2,2)=VAL
   CO(4,1)=X(K2)
   CO(4,2)=Y(K2)
21 X(K5)=X(K4)
   Y(K5)=Y(K4)
   DO 51 I=1,4
   KI=K1+I
   X(KI-1)=CO(I,1)
   Y(KI-1)=CO(I,2)
51 CONTINUE
22 GO TO (23,25,31),ICOUNT
23 N=3
   KANT=4
   DO 24 I=1,3
   L=K1+I-1
   Q(I,1)=X(L)**2
   Q(I,2)=X(L)
   Q(I,3)=Y(L)
24 Q(I,4)=1.0
   CALL MATS (Q,CO,N,M,NSING)
   IF (NSING) 70,70,58
70 CONTINUE
   A=CO(1,1)
   B=0.
   C=0.
   D=CO(2,1)
   E=CO(3,1)
   GO TO 33
25 N=4
```

```
      KANT=1
      DO 26 I=1,4
      L=K1+I-1
      Q(I,1)=X(L)**2
      Q(I,2)=X(L)*Y(L)
      Q(I,3)=Y(L)**2
      Q(I,4)=X(L)
   26 Q(I,5)=1.0
      CALL MATS (Q,CO,N,M,NSING)
      IF (NSING) 73,73,29
   73 CONTINUE
      A=CO(1,1)
      B=CO(2,1)
      C=CO(3,1)
      D=CO(4,1)
      E=0.
      GO TO 33
   29 N=4
      KANT=2
      DO 30 I=1,4
      L=K1+I-1
      Q(I,1)=X(L)**2
      Q(I,2)=X(L)*Y(L)
      Q(I,3)=Y(L)**2
      Q(I,4)=Y(L)
   30 Q(I,5)=1.0
      CALL MATS (Q,CO,N,M,NSING)
      IF (NSING) 71,71,23
   71 CONTINUE
      A=CO(1,1)
      B=CO(2,1)
      C=CO(3,1)
      D=0.0
      E=CO(4,1)
      GO TO 33
   31 N=5
      KANT=3
      DO 32 I=1,5
      L=K1+I-1
      Q(I,1)=X(L)**2
      Q(I,2)=X(L)*Y(L)
      Q(I,3)=Y(L)**2
      Q(I,4)=X(L)
      Q(I,5)=Y(L)
   32 Q(I,6)=1.0
      CALL MATS (Q,CO,N,M,NSING)
      IF (NSING) 72,72,25
   72 CONTINUE
      A=CO(1,1)
      B=CO(2,1)
      C=CO(3,1)
      D=CO(4,1)
      E=CO(5,1)
   33 CONTINUE
      AA=C-.25*B**2/A
      BB=E-.5*B*D/A
      CC=-.25*D**2/A-1.0
      IF (AA) 27,37,27
   27 QUAN=BB**2-4.*AA*CC
      IF (QUAN) 34,38,38
```

```
 34 GO TO (29,23,25,58),KANT
 37 YPRED=-CC/BB
    GO TO 39
 38 YPRED=.5*(-BB+SQRT(QUAN))/AA
    KAN=1
    GO TO 39
 28 YPRED=.5*(-BB-SQRT(QUAN))/AA
    KAN=2
 39 VAREE=-.5*(B*YPRED+D)/A
    GO TO 60
 58 TERM1=((Y(K3)-Y(K1))/(X(K3)-X(K1))-(Y(K1)-Y(K2))/(X(K1)-X(K2)))/
   1 (X(K3)-X(K2))
    TERM2=(Y(K1)-Y(K2))/(X(K1)-X(K2))
    VAREE=.5*(X(K2)+X(K1)-TERM2/TERM1)
    YPRED=TERM1*VAREE**2+(TERM2-(X(K2)+X(K1))*TERM1)*VAREE+Y(K2)-X(K2)
   1 *TERM2+X(K1)*X(K2)*TERM1
 60 CONTINUE
    WRITE (6,100)A,B,C,D,E,BB,CC,
100 FORMAT (2H0A,E15.7,3H  B,E15.7,3H  C,E15.7,3H  D,E15.7,3H  E,E15
   1.7,4H  BB,E15.7,4H  CC,E15.7)
    VARI=VAREE*57.295775/R1(J)
    Y1PRED=YPRED/R2(J)
    WRITE (6,200) Y1PRED,VARI
200 FORMAT (2E18.7)
    IF(X(K1)-X(K2)) 41,41,42
 41 SIG =1.0
    GO TO 43
 42 SIG =-1.0
 43 IF (SIG*(VAREE-X(K2)))45,44,44
 45 IF(X(K1)-X(K3)) 46,46,47
 46 SIG =1.0
    GO TO 48
 47 SIG =-1.0
 48 IF (SIG*(VAREE-X(K3)))59,44,44
 44 GO TO (28,34),KAN
 59 IF (ABS(VAL-YPRED)-TOL*R2(J)) 54,54,13
 13 VARY=VAREE
    GO TO 53
 50 Z2=1.0
    SC1=STPINC*SC1
 52 VARY=VARY+SC1*FINC*R1(J)
 53 ICNT(J)=ICOUNT
    FKTR(J)=SC1
    W1(J)=Z1
    W2(J)=Z2
    VARY=VARY/R1(J)
    VAL=VAL/R2(J)
    RETURN
 54 TEST=0.0
    GO TO 53
    END
```