NASA CR-66515

TOLIP - TRAJECTORY OPTIMIZATION AND LINEARIZED PITCH COMPUTER PROGRAM

By Robert E. Willwerth, Jr., and Richard C. Rosenbaum

GPO PRICE $ ______

CFSTI PRICE(S) $ ______

Hard copy (HC) 3.00

Microfiche (MF) .65

ff 653 July 65

Prepared under Contract No. NAS 1-5106 by
LOCKHEED MISSILES & SPACE COMPANY
Sunnyvale, Calif.

for Langley Research Center

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

FACILITY FORM 602

N68-18185 (ACCESSION NUMBER)

(THRU) 1

266 (PAGES)

(CODE)

CR-66515 (NASA CR OR TMX OR AD NUMBER)

30 (CATEGORY)

NASA CR-66515

TOLIP - Trajectory Optimization and Linearized Pitch Computer Program

FOREWORD

This report was prepared by the Advanced Flight Mechanics department of the Navigation Guidance and Control Division of the Lockheed Missiles and Space Company, Sunnyvale, California. It presents the final documentation for the Scout Operational Performance and Dispersion computer program developed by Lockheed for the Langley Research Center under NASA Contract NAS 1-5106. FORTRAN source listings, and symbolic decks in both FORTRAN II and IV, included with the master copy of this report complete the program documentation. Mr. R. E. Willwerth was responsible for program development. Optimization and pitch program linearization techniques were developed by R. C. Rosenbaum. Initial development of body dynamics simulation was done by C. W. Edwards, and the dispersion and range-safety modules were developed by R. L. Moll and John Slimick. The major portion of the programming was done by Miss Zoe Taulbee. The work was performed under the cognizance of R. L. Nelson and D. I. Kepler.

TABLE OF CONTENTS

PART I THEORY

Section Page

PART II PROGRAMMING OF THE EQUATIONS

ABSTRACT

The Scout Trajectory Optimization and Linearized Pitch computer program (TOLIP) obtained from the Lockheed Missiles and Space Company under contract NAS 1-5106 is designed to perform the trajectory-related calculations necessary in mission-planning and preflight analysis with a large reduction in computer time. It uses a closed-loop steepest descent optimization procedure to obtain flight trajectories that maximize payload, inertial or aerodynamic velocity, or altitude for up to five-stage vehicles. Following optimization, the pitch program is automatically linearized to enable its mechanization in the autopilot system. Control system and body dynamics effects are included in the simulation. A large number of constraints on the trajectory and pitch program can be included in the optimization and linearization solutions, so that a great deal of flexibility as well as recognition of vehicle limitations are possible in the use of the program.

Following the solution for the optimum trajectory and pitch program, the computer program can proceed, by option, directly into calculating: (1) radar tracking coordinate histories for up to twenty stations; (2) the locus of nominal impact points of the spent stages; (3) nominal dispersion envelopes; and (4) failure-mode hardover turns. Finally, for planning purposes, first-order performance exchange ratios for several vehicle/motor characteristics are calculated.

Particular attention was devoted during program development to computing speed. The convergence scheme, the general program arrangement, and the subroutines have been modified to provide significant increases in speed for special applications.

INTRODUCTION

During the past few years procedures have been developed for evaluating the launch vehicle tilt programs that result in maximum performance ascent trajectories. A variety of digital computer routines have been mechanized to perform these computations. The early versions of these routines were used primarily for applied research studies and to a limited extent for preliminary design studies. Typically, the derivation of tilt programs with these routines required a sequence of trial computations for which the operator provided estimates of initial conditions for a number of mathematical parameters. The time required to achieve an acceptable result with these routines and the associated cost of the analysis were such that their application was very limited.

More recently significant advances have been achieved in both the analytical techniques applied to trajectory shaping problems and in the mechanization of these techniques in suitable computer routines. An example of these advances is the digital computer program PRESTO* which solves a complete trajectory shaping problem for multistage vehicles in a single pass at the computer with total computing times of the order of one minute. From further study of these techniques, it became apparent that with certain refinements similar methods could be applied to operational performance and trajectory dispersion problems. The advantage would be a substantial reduction in cost and reaction time (i.e., time from receipt of input data to transmittal of output).

---

*PRESTO - Program for Rapid Earth-to-Space Trajectory Optimization, NASA Contractor Report NASA CR-158, February 1965.

The computer program documented herein brings application of the new technology to operational performance and dispersion problems for the Scout launch system. The report sections are sequenced into three groups. In the first part, a discussion of the theoretical methods used in the program is provided. All equations of motion, constraint parameters, optimization and pitch program linearization equations are fully documented. Part two is an extensive description of the programming, ranging from a discussion of the overall computation flow to the details of some of the more complicated subroutines. The third part of the report is a users' manual in which the data input and output formats are described and detailed instructions are given for using the various program options.

PART I

THEORY

SECTION 4

POINT-MASS EQUATIONS OF MOTION

DEFINITION OF SYMBOLS

A	Aerodynamic reference area, ft^2
$[A]$	A matrix (see page 4-11)
$\bar{a}$	Net vehicle acceleration in local coordinate system
A_e	Rocket engine nozzle exit area, ft^2
$[B]$	B matrix (see page 4-11)
$[C]$	C matrix conversion from P to L systems
C_D	Drag coefficient
C_L	Lift coefficient
C_{xy}	Component of C matrix, x row and y column
D	Aerodynamic drag on vehicle, lb
E	Orbit energy per unit mass
F	dV/dt
FL	Aerodynamic lift on vehicle, lb
G	$d\gamma/dt$
g	μ/r^2 gravitational acceleration on spherical Earth
H	$d\psi/dt$
$\bar{H}$	Angular momentum
I	dr/dt
i	Inclination of geocentric orbit plane
$\bar{i}_x, \bar{i}_y, \bar{i}_z$	Unit vectors for L (local) coordinate system
J	$d\lambda/dt$
J_2	Oblate Earth gravitational constant = 1.0827×10^{-3}
$\bar{J}_x, \bar{J}_y, \bar{J}_z$	Unit vectors for platform (P) coordinate system
K	$d\tau_I/dt$

$\bar{k}_x, \bar{k}_y, \bar{k}_z$	Unit vectors for Earth-centered inertial (I) coordinate system
$\bar{l}_x, \bar{l}_y, \bar{l}_z$	Unit vectors for G (launch geocentric) coordinate system
m	Instantaneous mass of vehicle, slugs
p	Atmospheric pressure, psia
r	Radial distance from Earth-center to vehicle, ft
r_e	Earth equatorial radius, ft
r_p	Perigee radius
T	Vehicle net thrust force, lb
T_G	Thrust vector in geocentric (G) system
t_G	Space Age Date = Julian Date - 2,436,934.5
T_I	Thrust vector in inertial (I) system
$T_{L_x}, T_{L_y}, T_{L_z}$	Components of thrust along axes of local system
$T_{P_x}, T_{P_y}, T_{P_z}$	Components of thrust along axes of platform system
V	Velocity in local system, ft/sec
V_I	Velocity in inertial system, ft/sec
α_e	Right ascension
β	In-plane angle from ascending node
β_p	Argument of perigee
γ	Flight path angle of V in local system, rad
γ_I	Flight path angle of V_I in inertial system, rad
$\delta\lambda$	Angle between geocentric and geodetic vertical
ζ	True anomaly of position
θ	Vehicle pitch attitude in P system (see Fig. 4-2)
λ	Geocentric latitude of vehicle position, rad
μ	Gravity constant, ft^3/sec^2
ρ	Atmospheric density, $slugs/ft^3$

τ	Earth longitude angle, rad
τ_I	Inertial longitude angle, rad
χ	Vehicle yaw attitude in platform system (see Fig. 4-2)
ψ	Azimuth of V in local system, rad
ψ'	$90^\circ - \psi$
ψ_I	Azimuth of V_I in inertial system, rad
ω	Earth rotation rate, rad/sec
Ω_e	Longitude of ascending node

EQUATIONS OF MOTION

The equations of motion employed in the Scout computer program are based on the equations used in PRESTO (see reference, page 3-1). However, improved accuracy in the simulation has been achieved through an oblate Earth gravity model and representation of an inertially-oriented vehicle pitch plane. The trajectory variables of integration are the velocity relative to a spherical rotating Earth and position relative to the center of the Earth. These coordinates are shown in Figure 4-1. The local (or L) system is made up of the $\bar{i}_x$, $\bar{i}_y$ and $\bar{i}_z$ vectors. The origin of the L system moves with the vehicle center of gravity. The $\bar{i}_y$ axis lies along the aerodynamic velocity vector, and the $\bar{i}_x$ axis is perpendicular to the instantaneous plane of motion. The $\bar{i}_z$ vector points generally away from the Earth's center. The $\bar{k}_x - \bar{k}_y - \bar{k}_z$ system will be referred to as the inertial, or I system. Its origin is at the center of the Earth. The I system is fixed in space.

The point-mass acceleration, $\bar{a}$, in the local system, is written (see reference, page 3-1) as follows:

$$\bar{a} = \bar{i}_x \Big[V \cos \gamma \dot{\psi} - \frac{V^2}{r} \frac{\cos^2 \gamma \sin \psi \sin \lambda}{\cos \lambda} - 2 V \cos \gamma \omega \sin \lambda$$

$$- r\omega^2 \sin \lambda \cos \lambda \sin \psi + 2 V \omega \cos \lambda \cos \psi \sin \gamma \Big]$$

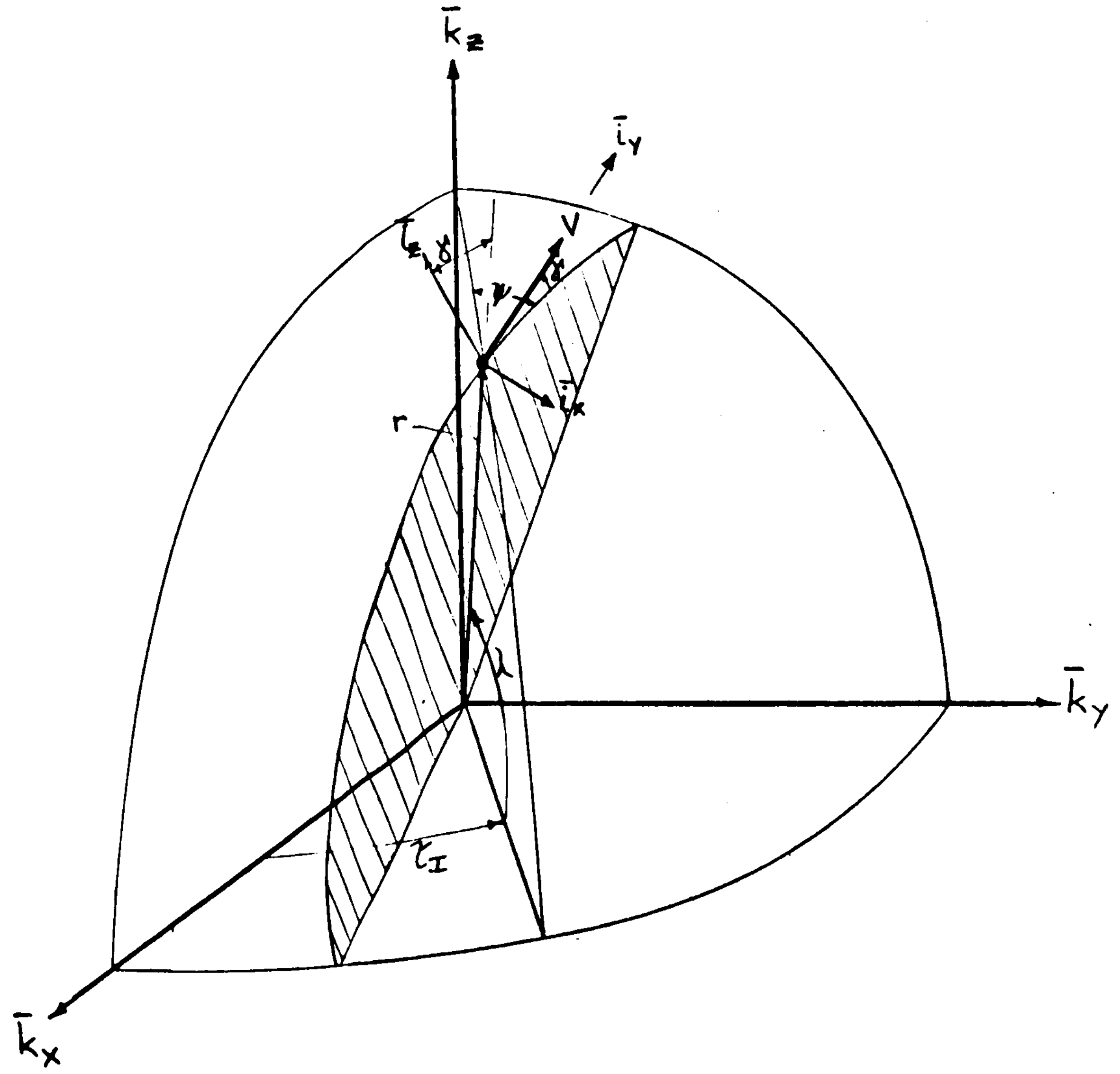
$$+ \bar{i}_y \Big[\dot{V} - r\omega^2 (\cos^2 \lambda \sin \gamma - \sin \lambda \cos \lambda \cos \psi \cos \gamma \Big]$$

$$+ \bar{i}_z \Big[V \dot{\gamma} - \frac{V^2}{r} \cos \gamma - 2 V \omega \cos \lambda \sin \psi$$

$$- r\omega^2 (\cos^2 \lambda \cos \gamma + \sin \lambda \cos \lambda \cos \psi \sin \gamma) \Big]$$

Figure 4-1

TRAJECTORY VARIABLES AND COORDINATE SYSTEMS



The forces acting on the body may be divided into three groups: aerodynamic, gravitational, and thrust. The components of these forces along the L axes are:

Aerodynamic

It will be assumed in evaluation of the point mass motion that the vehicle will be controlled in such a way that no significant aerodynamic side force will appear along the $\bar{i}_x$ axis. Thus,

$$\bar{F}_A = -\bar{i}_y D + \bar{i}_z (FL)$$

where the drag (D) and lift (FL) are computed from input aerodynamic coefficients.

Gravitational

The gravity model used in the Scout program is derived from differentiating the Earth geopotential function*

$$\bar{F}_G = g \left\{ \bar{i}_x \left[J_2 \left(\frac{r_e}{r} \right)^2 3 \sin \lambda \cos \lambda \sin \psi \right] - \bar{i}_y \left[\sin \gamma - \frac{J_2}{2} \left(\frac{r_e}{r} \right)^2 \cdot \right. \right.$$

$$\left. \cdot (3 - 9 \sin \lambda) \sin \gamma + J_2 \left(\frac{r_e}{r} \right)^2 3 \sin \lambda \cos \lambda \cos \psi \cos \gamma \right]$$

$$- \bar{i}_z \left[\cos \gamma + \frac{J_2}{2} \left(\frac{r_e}{r} \right)^2 (3 - 9 \sin^2 \lambda) \cos \gamma - J_2 \left(\frac{r_e}{r} \right)^2 \cdot \right.$$

$$\left. \left. \cdot 3 \sin \lambda \cos \lambda \cos \psi \sin \gamma \right] \right\}$$

The Earth radius as a function of latitude is computed from relationships given on page 4-21.

---

*Solution of the Problem of Artificial Satellite Theory Without Drag. Dick Brouwer. American Astronomical Journal, November 1959.

Thrust

The thrust acts along the center line of the vehicle which is oriented with respect to an onboard inertial platform coordinate system aligned with the local vertical a few seconds before launch. During flight, the vehicle pitches and yaws about the axes of this system in a pre-programmed manner. For simulation purposes, it is necessary to resolve the thrust into the trajectory coordinates of integration (local system). In the notation used here, T_{L_x} indicates the component of thrust along the x axis of the local system. Thus,

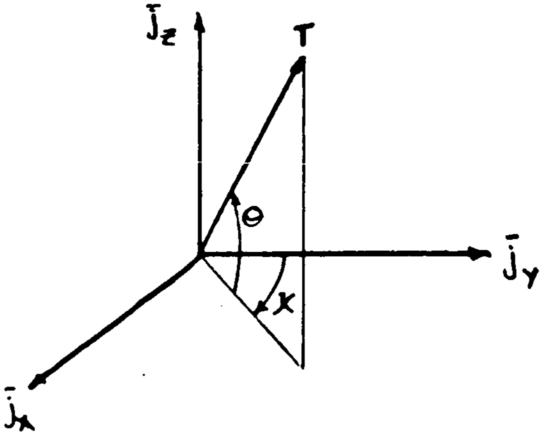
$$\bar{F}_T = \bar{I}_x T_{L_x} + \bar{I}_y T_{L_y} + \bar{I}_z T_{L_z}$$

and T_{L_x}, T_{L_y} and T_{L_z} must be evaluated. It will be necessary to make several coordinate transformations to go from the platform system to the local system. The remainder of this section will be concerned with these transformations.

The platform, or P coordinate system, is shown in Figure 4-2. The orientation of the P system is inertially fixed, and the origin moves with the vehicle center of gravity. At the start of the flight, the $\bar{J}_z$ axis is along the geodetic vertical pointing away from the Earth's center. The $\bar{J}_y$ axis lies in the desired trajectory plane. The orientation of the thrust vector with respect to the platform system is shown here.

Figure 4-2

PLATFORM (P) COORDINATE SYSTEM



The components of thrust along the platform axes are

$$T_{P_x} = T \cos \theta \sin \chi$$

$$T_{P_y} = T \cos \theta \cos \chi$$

$$T_{P_z} = T \sin \theta$$

It is necessary to introduce a geocentric coordinate system to be denoted by G. The G system is similar to the P system except that its Z axis is along the geocentric, rather than the geodetic vertical. The platform and geocentric systems are shown in Figures 4-3 and 4-4 on the following page.

To go from the platform system to the local system, the following sequence of transformations is required:

1. Platform → Geocentric [B']
2. Geocentric → Inertial [B]
3. Inertial → Local [A]

The letters in brackets indicate the name of the matrix that will be used to represent the transformation.

Platform-Geocentric Transformation [B']

The following sequence of rotations is required (see Figure 4-4):

a. Rotate about $\bar{J}_z$ by the angle ψ' so that $\bar{J}_y$ is pointing east.

b. Rotate about $\bar{J}_y$ by the angle $\delta\lambda$ so that $\bar{J}_z$ is along the geocentric vertical.

c. Rotate about $\bar{J}_z$ by $-\psi'$ so that $\bar{J}_y$ is back in the desired trajectory plane.

Figure 4-3

ORIENTATION OF PLATFORM COORDINATE SYSTEM

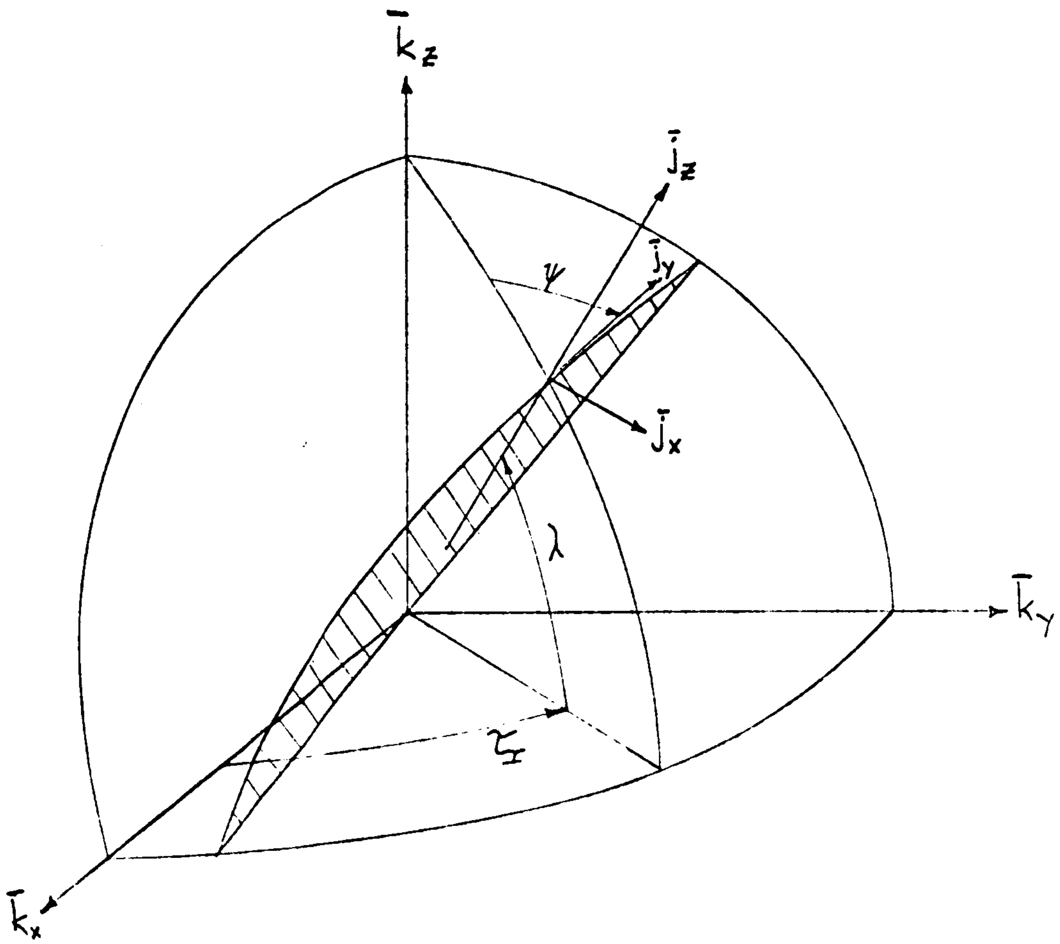
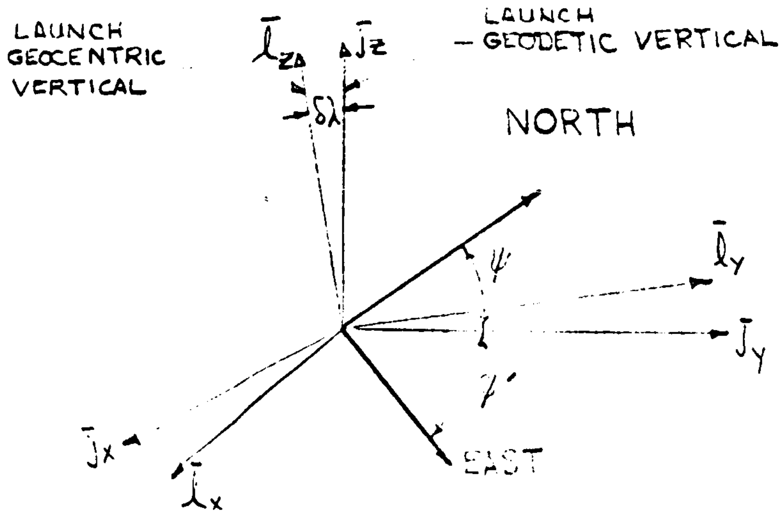


Figure 4-4

GEOCENTRIC AND PLATFORM COORDINATE SYSTEMS



This transformation is represented by the equation

$$\bar{T}_G = [B'] \; \bar{T}_p$$

where [B'] is a 3x3 rotation matrix. Note that $\psi' = 90^\circ - \psi$.

<u>Geocentric-Inertial Transformation [B]</u>

(See Figures 4-3 and 4-4. G and P systems are nearly aligned.)

a. Rotate about $\bar{\ell}_z$ by $-\psi$

b. Rotate about $\bar{\ell}_x$ by $-\lambda$

c. Rotate about $\bar{\ell}_y$ by $-\tau_I$

This transformation is represented by the equation

$$\bar{T}_I = [B] \; \bar{T}_G$$

<u>Inertial-Local Transformation [A]</u>

(See Figure 4-1.)

a. Rotate about $\bar{k}_z$ by $-\tau_I$

b. Rotate about $\bar{k}_y$ by $90^\circ - \lambda$

c. Rotate about $\bar{k}_x$ by $90^\circ - \psi$

d. Rotate about $\bar{k}_y$ by $-\gamma$

This transformation is represented by the equation

$$\bar{T}_L = [A] \; \bar{T}_I$$

The three transformation matrices appear on the following page.

THRUST DIRECTION MATRICES

$[C] = [A]\,[B]\,[B']$

Let s → sine, c → cosine

$$[A] = \begin{bmatrix} \begin{array}{l} - s\tau\, c\psi \\ + c\tau\, s\lambda\, s\psi \end{array} & \begin{array}{l} + c\tau\, c\psi \\ + s\tau\, s\lambda\, s\psi \end{array} & \begin{array}{l} - c\lambda\, s\psi \\ \end{array} \\[2ex] \begin{array}{l} - c\tau\, s\lambda\, c\psi\, c\gamma \\ - s\tau\, s\psi\, c\gamma \\ + c\tau\, c\lambda\, s\gamma \end{array} & \begin{array}{l} - s\tau\, s\lambda\, c\psi\, c\gamma \\ + c\tau\, s\psi\, c\gamma \\ + s\tau\, c\lambda\, s\gamma \end{array} & \begin{array}{l} + c\lambda\, c\psi\, c\gamma \\ + s\lambda\, s\gamma \\ \end{array} \\[2ex] \begin{array}{l} + c\tau\, c\lambda\, c\gamma \\ + c\tau\, s\lambda\, c\psi\, s\gamma \\ + s\tau\, s\psi\, s\gamma \end{array} & \begin{array}{l} + s\tau\, c\lambda\, c\gamma \\ + s\tau\, s\lambda\, c\psi\, s\gamma \\ - c\tau\, s\psi\, s\gamma \end{array} & \begin{array}{l} + s\lambda\, c\gamma \\ - c\lambda\, c\psi\, s\gamma \\ \end{array} \end{bmatrix}$$

$$[B] = \begin{bmatrix} s\psi\, s\lambda\, c\tau - c\psi\, s\tau & - s\psi\, s\tau - c\psi\, s\lambda\, c\tau & c\lambda\, c\tau \\ c\psi\, c\tau + s\psi\, s\lambda\, s\tau & s\psi\, c\tau - c\psi\, s\lambda\, s\tau & c\lambda\, s\tau \\ - s\psi\, c\lambda & c\psi\, c\lambda & s\lambda \end{bmatrix}$$

$$[B'] = \begin{bmatrix} s^2\psi\, c\,\delta\lambda + c^2\psi & s\psi\, c\psi\,(1 - c\,\delta\lambda) & - s\psi\, s\,\delta\lambda \\ s\psi\, c\psi\,(1 - c\,\delta\lambda) & s^2\psi + c^2\psi\, c\,\delta\lambda & c\psi\, s\,\delta\lambda \\ s\psi\, s\,\delta\lambda & - c\psi\, s\,\delta\lambda & c\,\delta\lambda \end{bmatrix}$$

$$\delta\lambda = \left(0.001638 + \frac{\omega^2 r}{2g}\right) \sin 2\lambda$$

MATRICES B AND B' ARE EVALUATED ONLY AT TIME = 0

(See page 4-14)

In terms of these matrices one may write

$$\bar{\bar{T}}_L = [A]\,[B]\,[B']\,\bar{\bar{T}}_p = [C]\,\bar{\bar{T}}_p$$

This equation can be expanded to give the components of thrust in the local system. One obtains

$$T_{L_x} = T\left[C_{11} \cos\theta \sin\chi + C_{12} \cos\theta \cos\chi + C_{13} \sin\theta\right]$$

$$T_{L_y} = T\left[C_{21} \cos\theta \sin\chi + C_{22} \cos\theta \cos\chi + C_{23} \sin\theta\right]$$

$$T_{L_z} = T\left[C_{31} \cos\theta \sin\chi + C_{32} \cos\theta \cos\chi + C_{33} \sin\theta\right]$$

All of the forces acting on the body have now been obtained in the local coordinate system. The equations of motion can then be written and appear on the next page.

3-D EQUATIONS OF MOTION

$$F = \dot{V} = r\omega^2 \left[\cos^2\lambda \sin\gamma - \sin\lambda \cos\lambda \cos\psi \cos\gamma\right] - g\left[\sin\gamma - \frac{J_2}{2}\left(\frac{r_e}{r}\right)^2 \cdot\right.$$

$$\left.\cdot (3. - 9 \sin\lambda) \sin\gamma + J_2 \left(\frac{r_e}{r}\right)^2 3 \sin\lambda \cos\lambda \cos\psi \cos\gamma\right]$$

$$+ \frac{T}{m}\left[C_{21} \cos\theta \sin\chi + C_{22} \cos\theta \cos\chi + C_{23} \sin\theta\right] - \frac{D}{m}$$

$$G = \dot{\gamma} = 2\omega \cos\lambda \sin\psi + \frac{V}{r}\cos\gamma - \frac{g}{V}\left[\cos\gamma + \frac{J_2}{2}\left(\frac{r_e}{r}\right)^2 (3. - 9 \sin^2\lambda) \cos\gamma\right.$$

$$\left. - J_2 \left(\frac{r_e}{r}\right)^2 3 \sin\lambda \cos\lambda \cos\psi \sin\gamma\right] + \frac{(FL)}{mV}$$

$$+ \frac{r\omega^2}{V}\left[\cos^2\lambda \cos\gamma + \sin\lambda \cos\lambda \cos\psi \sin\gamma\right]$$

$$+ \frac{T}{mV}\left[C_{31} \cos\theta \sin\chi + C_{32} \cos\theta \cos\chi + C_{33} \sin\theta\right]$$

$$H = \dot{\psi} = \frac{V}{r}\,\frac{\cos\gamma \sin\psi \sin\lambda}{\cos\lambda} + \frac{r\omega^2 \sin\lambda \cos\lambda \sin\psi}{V\cos\gamma}$$

$$+ 2\omega \sin\lambda - \frac{2\omega \cos\lambda \cos\psi \sin\gamma}{\cos\gamma} + g J_2 \left(\frac{r_e}{r}\right)^2 \cdot$$

$$\cdot \frac{3 \sin\lambda \cos\lambda \sin\psi}{V \cos\gamma}$$

$$+ \frac{T}{mV\cos\gamma}\left[C_{11} \cos\theta \sin\chi + C_{12} \cos\theta \cos\chi + C_{13} \sin\theta\right]$$

$$I = \dot{r} = V \sin\gamma$$

$$J = \dot{\lambda} = \frac{V \cos\gamma \cos\psi}{r}$$

$$K = \dot{\tau} = \frac{V \cos\gamma \sin\psi}{r \cos\lambda} + \omega$$

where: $g = \mu/r^2$

$T = T_V - p\,A_e$

$D = \frac{1}{2}\rho V^2 C_D A$

$FL = \frac{1}{2}\rho V^2 C_L A$

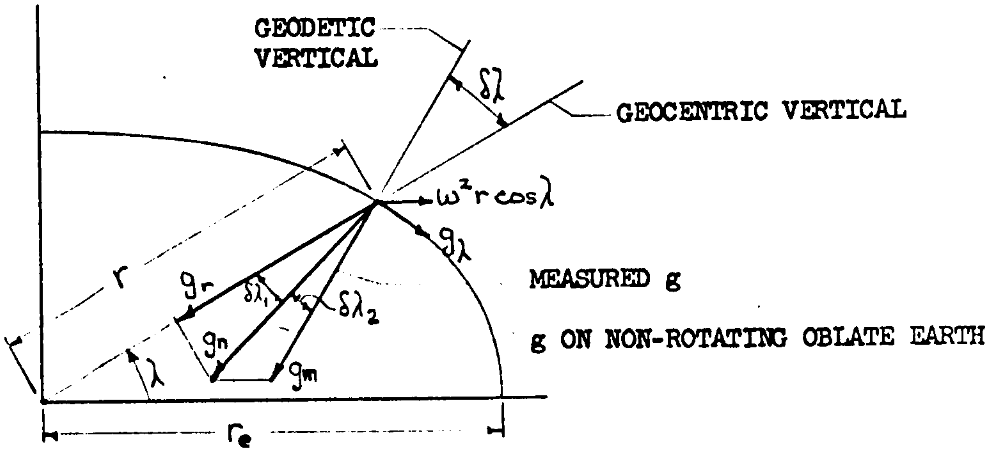
r_e = Equatorial radius

$J_2 = 1.0827 \times 10^{-3}$

θ, χ = Control variables

COMPUTATION OF ANGLE BETWEEN GEODETIC AND GEOCENTRIC VERTICALS

The angle between the two verticals can be found with the aid of the following diagram.



The geodetic vertical is the measured vertical, i.e., the direction of a plumb bob. The geocentric vertical goes through the center of the Earth. For the purpose of this calculation, it is assumed that the Earth is an oblate spheroid. δλ will then be a function only of the latitude.

The measured gravitational acceleration is made up of two parts. One is the acceleration due to the oblate figure of the Earth and the other is the centrifugal acceleration. The radial and tangential components of the acceleration due to the figure of the Earth can be written as[*]

$$g_r = \frac{\mu}{r^2} - \frac{3\sigma \mu r_e^2}{r^4} (1 - 3 \cos 2\lambda)$$

$$g_\lambda = \frac{6\mu \sigma r_e^2}{r^4} \sin 2\lambda$$

---

[*] "3 Dimensional Orbits of Earth Satellites Including Effects of Earth Oblateness and Atmospheric Rotation." NASA Memo 12-4-58A. Nelsen, Goodwin, Mersman.

where r_e is the equatorial radius
and $6\sigma = 1.638 \times 10^{-3}$

The angle between g_n and g_r can be written as

$$\frac{g_\lambda}{g_r} = 6\sigma \sin 2\lambda = \delta\lambda_1$$

This expression neglects the second component of g_r and assumes r_e equal to r.

The tangential component of the centrifugal acceleration is

$$\omega^2 r \cos \lambda \sin \lambda$$

The angle between g_n and the geodetic vertical is then

$$\frac{\omega^2 r \cos \lambda \sin \lambda}{g_n} = \delta\lambda_2$$

$\delta\lambda$ is the sum of these two angles. Therefore,

$$\delta\lambda = \left(6\sigma + \frac{\omega^2 r}{2g_n}\right) \sin 2\lambda = \delta\lambda_1 + \delta\lambda_2$$

TRAJECTORY VARIABLES IN INERTIAL FRAME

The following equations are used in the subroutine INER to compute the magnitude and direction of the vehicle velocity vector relative to a non-rotating Earth. These variables are used in calculating the terminal orbit elements defined on the next page.

$$V_I = \sqrt{V^2 + 2(V \cos \gamma \sin \psi)(\omega r \cos \lambda) + (\omega r \cos \lambda)^2}$$

$$\gamma_I = \tan^{-1} \left[V \sin \gamma \Big/ \sqrt{V_I{}^2 - (V \sin \gamma)^2} \right]$$

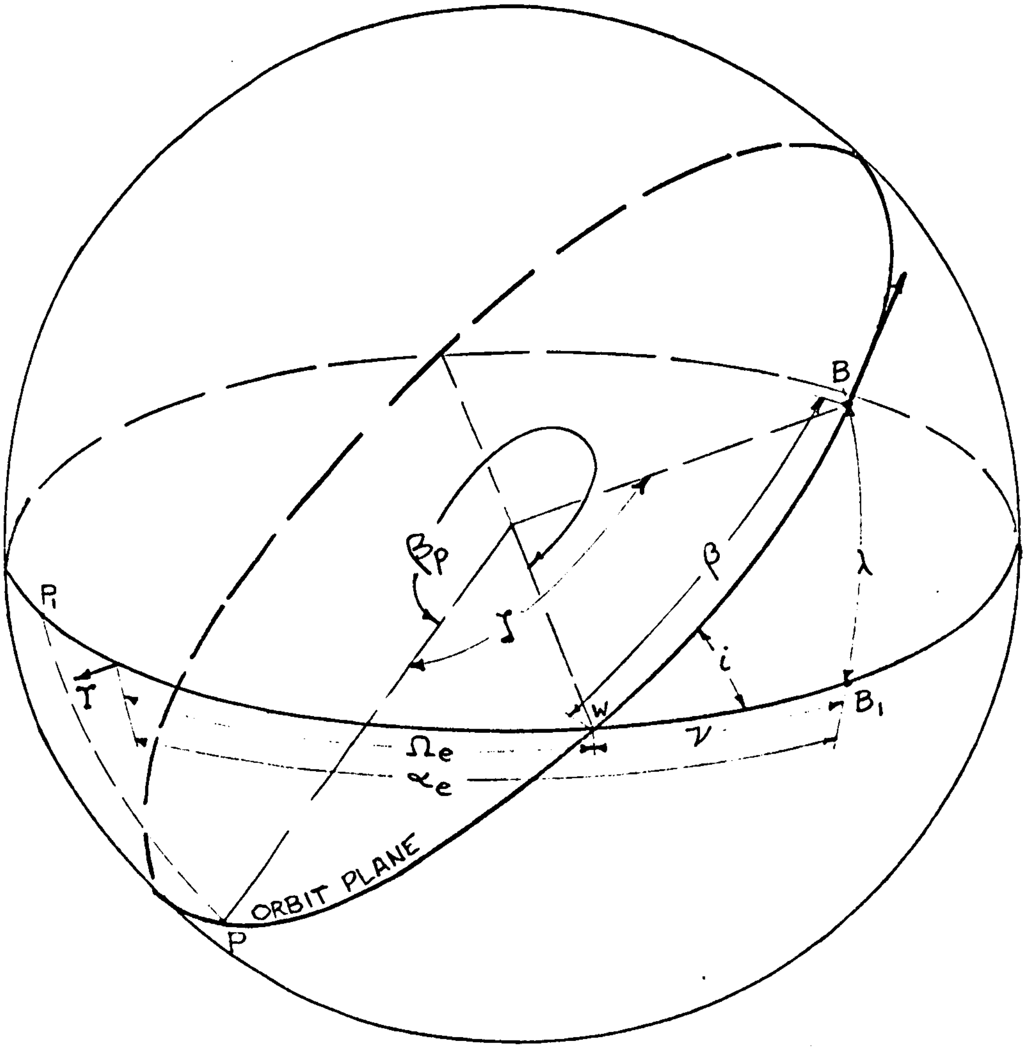
$$\psi_I = \tan^{-1} \left[(V \cos \gamma \sin \psi + \omega r \cos \lambda) \Big/ V \cos \gamma \cos \psi \right]$$

DIAGRAM AND EQUATIONS OF TERMINAL ORBIT ELEMENTS

In the Scout program, terminal constraints can be imposed on the trajectory variables explicitly, or they can be specified in terms of conventional orbit elements which are functions of the trajectory variables. The orbit elements are diagrammed in the figure on the following page. The equator constitutes the basic reference plane. The inertial "longitude" reference is in the direction of the vernal equinox, which is defined here to be the intersection of the plane of the ecliptic at 1950.0 and the equator of date.

The form of the equations used for the orbit elements defined on the following pages is such that they are valid for elliptic, parabolic and hyperbolic orbits. A spherical Earth model is assumed for these relationships.

GEOMETRY USED TO DEFINE THE TERMINAL ORBIT ELEMENTS



B	Booster burnout
B_1	Equatorial projection of B
P	Perigee of orbit
P_1	Equatorial projection of P
Y	Vernal equinox
ζ	True anomaly
α_e	Right ascension

β	In-plane angle from ascending node
β_p	Argument of perigee
i=EYE	Inclination of geocentric orbit plane
ν	Inertial longitude angle from ascending node
Ω_e	Longitude of W
W	Ascending node of the orbit plane

ORBIT ELEMENTS

Energy (x2)

$$2E = V_I^2 - \frac{2\mu}{r}$$

Angular Momentum

$$\bar{H} = rV_I \cos \gamma_I$$

Perigee Radius

$$r_p = \frac{H}{\left(\frac{\mu}{H}\right) + \sqrt{2E + \left(\frac{\mu}{H}\right)^2}}$$

Inclination

$$i = \cos^{-1} (\cos \lambda \sin \psi_I)$$

Longitude of Ascending Node

$$\Omega_e = \alpha_e - \nu$$

where α_e = right ascension of vehicle position

ν = inertial longitude angle from ascending node

$\alpha_e = 99^\circ.659967 + 0^\circ.98564743\ t_G + \tau^\circ + 360^\circ\ (t_G - [t_G])^*$

t_G = Julian Date - 2,436,934.5

$(t_G - [t_G])$ = Greenwich time of day

$0^\circ \leq \alpha_e < 360^\circ$

and $\nu = \tan^{-1} \left(\frac{\sin \psi_I \sin \lambda}{\cos \psi_I}\right) \quad 0 \leq \nu \leq \pi$

and $0 \leq \Omega_e < 360^\circ$

---

*Herget, P., Solar Coordinates 1800-2000. Vol. XIV. American Ephemeris and Nautical Almanac

Argument of Perigee

$$\beta_p = (\beta - \zeta)$$

β = in-plane range angle from ascending node

ζ = true anomaly

where

$$\beta = \tan^{-1}\left(\frac{\sin\lambda}{\cos\lambda\cos\psi_I}\right) \qquad 0 \leq \beta < 2\pi$$

and

$$\zeta = \tan^{-1}\left[\frac{\tan\gamma_I}{1 - \left(\frac{\mu}{H}\right)\left(\frac{r}{H}\right)}\right] \qquad 0 \leq \zeta < 2\pi$$

and

$$0 \leq \beta_p < 360^\circ$$

Additional Computed Orbit Elements

The following quantities are computed and output at the end of each trajectory but are not available as constraint parameters.

Semi-Major Axis

$$r_\alpha = -\frac{\mu}{2E}$$

Orbital Period

$$\tau_p = 2\pi\, r_\alpha \sqrt{\frac{r_\alpha}{\mu}}$$

Eccentricity

$$\epsilon = 1 - r_p/r_\alpha$$

OBLATE EARTH MODEL

The Earth model used in the Scout program is that of an ellipsoid having a flattening factor of 1./298.3. The equations used to compute the radius of the Earth at a given geocentric latitude are

$$\eta = \tan^{-1}\left[\frac{f}{(f-1)} \tan \lambda_c\right]^{*}$$

where $f = 298.3$

λ_c = geocentric latitude

$$Z = \frac{(f-1)}{f} \sin \eta$$

$$X = \cos \eta$$

$$R_{OBL} = R_{EQ} \cdot \sqrt{X^2 + Z^2}$$

where R_{EQ} = Earth equatorial radius

The altitude is then computed from

$$h = r - R_{OBL}$$

where r is the geocentric radius to the vehicle, a variable of integration.

For output purposes, the geodetic latitude, λ_d , is computed from the geocentric latitude, λ_c , from

$$\lambda_d = \lambda_c + C_1 \sin 2\lambda_c + C_2 \sin 4\lambda_c^{**}$$

---

[*] "Physical Constants for Satellite Calculation," R. J. Mercer, Report No. TOR-469(5110-02)-2, Aerospace Corporation, January 1965.

[**] "Relating Geodetic Latitude and Altitude to Geocentric Latitude and Radius Vector," E. W. Purcell and W. B. Cowan, ARS Journal, Vol. 31, #7, pp 932-935, July 1961.

where $C_1 = 3.372672 \times 10^{-3}$

$C_2 = -5.6873 \times 10^{-6}$

Finally, the gravity model has already been defined on page 4-6.

RANGE EQUATIONS

The downrange distance from the launch site to the vehicle is computed and output at every integration step, and is available as a constraint parameter at stage points and final burnout. Downrange is defined as a great-circle distance between the geocentric latitudes (λ) and the longitudes (τ) of the two points. Range is considered a central angle in the equations. (DR)

$$\cos(DR) = \cos(90 - \lambda_o) \cos(90 - \lambda) + \sin(90 - \lambda_o) \sin(90 - \lambda) \cdot \cos(\tau - \tau_o)$$

or

$$\cos(DR) = \sin \lambda_o \sin \lambda + \cos \lambda_o \cos \lambda \cos(\tau - \tau_o).$$

Downrange is then converted to nautical miles by assuming 60 n.m./degree of arc. It is calculated in the subroutine INER.

During computation of the dispersed trajectories (see option 21), the crossrange dispersion from the nominal trajectory is evaluated. Since the crossrange dispersions are generally small, computational accuracy is improved by using a small angle approximation in the solution. This is done with the following equations, where the azimuth angle ψ at the nominal staging point determines the reference plane on which crossrange distance is zero.

$$\tan Q = (\tau_d - \tau_n) / (\lambda_d - \lambda_n)$$

$$\text{Slant range} = \left((\lambda_d - \lambda_n)^2 + (\tau_d - \tau_n)^2 \right)^{\frac{1}{2}}$$

$$\text{Crossrange} = - (\text{slant range}) \cdot \sin (\psi - Q)$$

where ranges are again considered as Earth-central angles and with subscripts n = nominal, d = dispersed.

Crossrange is then converted to nautical miles by assuming 60 n.m. per degree of arc.

The sign convention assumed gives a crossrange dispersion to the right of the nominal trajectory (as viewed looking downrange) a positive value, left crossrange a negative value.

SECTION 5

DERIVATION OF OPTIMIZATION EQUATIONS

DEFINITION OF SYMBOLS

The equations of Section 5 are closely dependent on those of Section 4. Thus, if a symbol is not defined below, please refer to page 4.1.

a	Local speed of sound, ft/sec
dx	Vector of deviations in trajectory variables
$d\psi$	Vector of constraint corrections
h	Vehicle altitude above Earth surface, ft
jc	Total number of constraints
M	Mach number
S	Array of sensitivity coefficients for adjustable parameters on constraints
Y	Array of weighting constants for adjustable parameters
α	Aerodynamic angle of attack
$\delta\theta$	Change in pitch angle from previous trajectory
$\delta\tau$	Change in adjustable parameter
δx	Vector of deviations in trajectory variables
Λ	Vector of sensitivity coefficients of θ or χ
$\lambda_{V_i}, \lambda_{\gamma_i}, \lambda_{\psi_i}$ etc.	Adjoint variables giving effect of V, γ, ψ, etc., on ith constraint

DERIVATION OF PROPERTIES OF ADJOINT VARIABLES

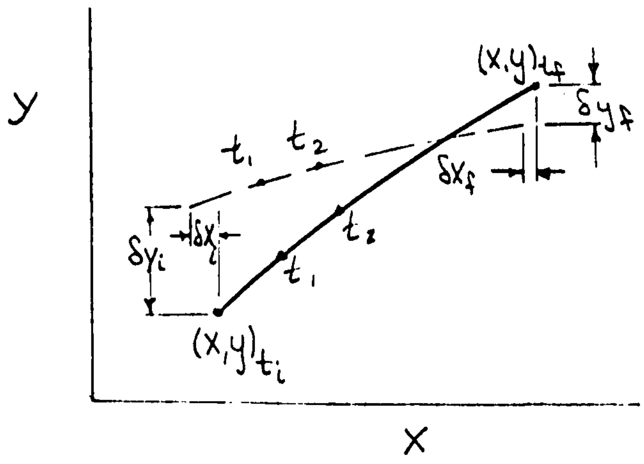
The steepest descent method of trajectory optimization depends on obtaining the effects of small changes in the control and trajectory variables on the trajectory constraints. These effects are provided by solving a set of equations which are adjoint to the linear perturbation equations written about a nominal trajectory. As used here, "adjoint" means that the coefficients of the two sets of equations are the negative transpose of each other. A derivation of the properties of the solution of the adjoint equations is given here.

Assume a two-variable system which is described by the nonlinear differential equations

$$\dot{x} = f(x, y, t, u(t)) \qquad (5\text{-}1)$$

$$\dot{y} = g(x, y, t, u(t)) \qquad (5\text{-}2)$$

where x and y are the dependent variables, t is the independent variable and u is the control variable. Assume that a solution to these equations is given by the solid line in the figure. One is interested in determining the effect of perturbations δx_i, δy_i and δu on a function $Z(x, y)$ at the terminal time t_f. δx_i and δy_i are known at a particular time t_i and δu is a function of time which is known from t_i to t_f.



The straightforward way to solve this problem is to obtain the solution to Eqs. (5-1) and (5-2) with initial conditions

$$x_i = x_{n_i} + \delta x_i$$

$$y_i = y_{n_i} + \delta y_i$$

and a new control variable

$$u = u_n + \delta u$$

where the subscript n denotes the nominal value. The values of x and y at the terminal time are then substituted into Z to determine $\delta Z = Z(x, y) - Z_n (x_n, y_n)$.

If the deviations from the nominal trajectory are small, this process is equivalent to solving the set of linear perturbation equations

$$\delta \dot{x} = \frac{\partial f}{\partial x} \delta x + \frac{\partial f}{\partial y} \delta y + \frac{\partial f}{\partial u} \delta u \qquad (5\text{-}3)$$

$$\delta \dot{y} = \frac{\partial g}{\partial x} \delta x + \frac{\partial g}{\partial y} \delta y + \frac{\partial g}{\partial u} \delta u \qquad (5\text{-}4)$$

Note that if one wanted to change the time t_i at which the perturbations are known, it would be necessary to obtain a new solution to Eqs. (5-3) and (5-4) in order to find the terminal perturbations. Thus, it would be very tedious to determine the effects of perturbations at all times t from t_i to t_f. However, by solving the equations which are adjoint to Eqs. (5-3) and (5-4), it is possible to obtain the desired information with just one solution of a set of differential equations.

To derive the adjoint differential equations, begin with the desired form of the adjoint variables, λ_x and λ_y , as first order sensitivity coefficients.

$$\delta Z \,|_{t_f} = \left[\lambda_x \, \delta x + \lambda_y \, \delta y\right]_t + P(\delta u, t) \quad t_i \le t \le t_f \qquad (5\text{-}5)$$

λ_x and λ_y are functions of time which relate perturbations in x and y to the perturbation in Z at the final time, i.e., $\lambda_x(t) = \partial Z_{t_f}/\partial x(t)$. P is an unknown function which represents the influence of δu on $\delta Z|_{t_f}$.

Consider the perturbed trajectory represented by the dotted line in the figure. At time t_1 all the quantities on the right-hand side of Eq. (5-5) will have some value. At t_2 they will have, in general, a slightly different value. However, $\delta Z|_{t_f}$ is always the same for a given perturbed trajectory. Therefore, the quantities on the right-hand side of Eq. (5-5) must change in such a way that $\delta Z|_{t_f}$ remains constant. The time derivative of $\delta Z|_{t_f}$ is zero.

$$\delta \dot{Z}|_{t_f} = 0 = \dot{\lambda}_x \delta x + \lambda_x \delta \dot{x} + \dot{\lambda} \delta_y + \lambda_y \dot{\delta}_y + \dot{P} \qquad (5\text{-}6)$$

Substituting Eqs. (5-3) and (5-4) into (5-6) gives

$$0 = \delta x \left(\dot{\lambda}_x + \lambda_x \frac{\partial f}{\partial x} + \lambda_y \frac{\partial g}{\partial x} \right) + \delta y \left(\dot{\lambda}_y + \lambda_x \frac{\partial f}{\partial y} + \lambda_y \frac{\partial g}{\partial y} \right)$$

$$+ \left[\dot{P} + \delta u \left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u} \right) \right] \qquad (5\text{-}7)$$

λx, λy, and P depend only on the nominal trajectory. One is, therefore, free to pick any perturbed trajectory which will produce the desired results. In particular, assume that δu is zero (and therefore $\dot{P} = 0$) and that at some arbitrary time, $\delta y = 0$.

δx is the only remaining perturbation. In order to satisfy Eq. (5-7), the coefficient of δx must equal zero. Similarly, if δx is assumed to be zero while δy has some value, Eq. (5-7) is satisfied only if the coefficient of δy is zero. One is, therefore, led to the differential equations

$$\dot{\lambda}_x = -\lambda_x \frac{\partial f}{\partial x} - \lambda_y \frac{\partial g}{\partial x} \qquad (5\text{-}8)$$

$$\dot{\lambda}_y = -\lambda_x \frac{\partial f}{\partial y} - \lambda_y \frac{\partial g}{\partial y} \qquad (5\text{-}9)$$

Eqs. 5-8) and (5-9) are adjoint to the linear perturbation equations (Eqs. (5-3) and (5-4)) with the forcing function δu set to zero. λ_x and λ_y are referred to as adjoint variables. One solution of the adjoint equations provides the effect of perturbations at any time t on $\delta Z|_{t_f}$.

To solve Eqs. (5-8) and (5-9), a set of initial conditions is required. These initial conditions are specified at the terminal time, because it is at this point that values are known for λ_x and λ_y. Referring to Eq. (5-5), it is seen that at t_f

$$\lambda_x = \frac{\partial z}{\partial x}\bigg|_{t_f} \qquad \lambda_y = \frac{\partial z}{\partial y}\bigg|_{t_f}$$

For Z = x,

$$\lambda_x = 1 \qquad \lambda_y = 0$$

Note that the adjoint equations do not depend on Z. Only the initial conditions depend on the form of the constraint parameter.

Now consider a perturbed trajectory for which δu is not zero. The terms in parenthesis in Eq. (5-7) have been shown to be zero. The term in square brackets must also be zero. Therefore,

$$\dot{P} = -\left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u}\right) \delta u = \Lambda_u \qquad (5\text{-}10)$$

Integrating Eq. (5-10) from t_f to t

$$P_t - P_{t_f} = -\int_{t_f}^{t} \delta u \left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u}\right) dt$$

At the final time t_f a change in the control, δu, can have no effect on the terminal constraint. P_{t_f} is therefore zero and

$$P_t = -\int_{t_f}^{t} \delta u \left(\lambda_x \frac{\partial f}{\partial u} + \lambda_y \frac{\partial g}{\partial u} \right) dt \qquad (5\text{-}11)$$

Thus, to find the influence of changes in the control variable, one evaluates the integral of Eq.(5-11) where λ_x and λ_y are solutions of Eqs. (5-8) and (5-9), and the partial derivatives are evaluated along the nominal trajectory.

One is often interested in finding the perturbation in Z at the time that another function S(x, y) reaches a certain value. S is referred to as the stopping condition. Let T be the unknown time at which the desired value of S is reached. Assuming that T is close to t_f, one may write

$$\delta S_T = \delta S_{t_f} + \dot{S}_{t_f} \delta t \qquad (5\text{-}12)$$

where $\delta t = T - t_f$. δS_T must equal zero if the stopping condition is to be met. Therefore,

$$\delta t = - \left. \frac{\delta S}{\dot{S}} \right|_{t_f} \qquad (5\text{-}13)$$

Similarly,

$$\delta Z_T = \delta Z_{t_f} + \dot{Z}_{t_f} \delta t \qquad (5\text{-}14)$$

Substituting Eq. (5-13) into Eq. (5-14) gives

$$\delta Z_T = \delta Z_{t_f} - \left. \frac{\dot{Z}}{\dot{S}} \right|_{t_f} \delta S_{t_f} \qquad (5\text{-}15)$$

δZ and δS at t_f are given by

$$\delta Z_{t_f} = \left[\frac{\partial Z}{\delta x} \delta x + \frac{\partial Z}{\partial y} \delta y \right]_{t_f} \qquad (5\text{-}16)$$

$$\delta S_{t_f} = \left[\frac{\partial S}{\delta x} \delta x + \frac{\partial S}{\partial y} \delta y \right]_{t_f} \qquad (5\text{-}17)$$

Substituting (5-16) and (5-17) into (5-15) gives

$$\delta Z_T = \left(\frac{\partial Z}{\partial x} - \frac{\dot{Z}}{\dot{S}} \frac{\partial S}{\partial x} \right)_{t_f} \delta x_{t_f} + \left(\frac{\partial Z}{\partial y} - \frac{\dot{Z}}{\dot{S}} \frac{\partial S}{\partial y} \right)_{t_f} \delta y_{t_f} \qquad (5\text{-}18)$$

Referring to Eq. (5-5) with $t = t_f$, it is seen that λ_x and λ_y should be given the following values at t_f.

$$\lambda_x = \left(\frac{\partial Z}{\partial x} - \frac{\dot{Z}}{\dot{S}} \frac{\partial S}{\partial x} \right)_{t=t_f}$$

$$\lambda_y = \left(\frac{\partial Z}{\partial y} - \frac{\dot{Z}}{\dot{S}} \frac{\partial S}{\partial y} \right)_{t=t_f}$$

With these initial conditions for backward integration, the solution of the adjoint equations will determine the effects of perturbations in x and y on Z at the unknown time when S reaches a desired value.

LINEAR PERTURBATION EQUATIONS

The linear perturbation equations written about the equations of motion, defined in Section 4, are as follows.

$$\frac{d\delta V}{dt} = \frac{\partial F}{\partial V}\delta V + \frac{\partial F}{\partial \gamma}\delta\gamma + \frac{\partial F}{\partial \psi}\delta\psi + \frac{\partial F}{\partial r}\delta r + \frac{\partial F}{\partial \lambda}\delta\lambda + \frac{\partial F}{\partial m}\delta m + \frac{\partial F}{\partial \tau}\delta\tau + \frac{\partial F}{\partial \theta}\delta\theta + \frac{\partial F}{\partial \chi}\delta\chi$$

$$\frac{d\delta \gamma}{dt} = \frac{\partial G}{\partial V}\delta V + \frac{\partial G}{\partial \gamma}\delta\gamma + \frac{\partial G}{\partial \psi}\delta\psi + \frac{\partial G}{\partial r}\delta r + \frac{\partial G}{\partial \lambda}\delta\lambda + \frac{\partial G}{\partial m}\delta m + \frac{\partial G}{\partial \tau}\delta\tau + \frac{\partial G}{\partial \theta}\delta\theta + \frac{\partial G}{\partial \chi}\delta\chi$$

$$\frac{d\delta \psi}{dt} = \frac{\partial H}{\partial V}\delta V + \frac{\partial H}{\partial \gamma}\delta\gamma + \frac{\partial H}{\partial \psi}\delta\psi + \frac{\partial H}{\partial r}\delta r + \frac{\partial H}{\partial \lambda}\delta\lambda + \frac{\partial H}{\partial m}\delta m + \frac{\partial H}{\partial \tau}\delta\tau + \frac{\partial H}{\partial \theta}\delta\theta + \frac{\partial H}{\partial \chi}\delta\chi$$

$$\frac{d\delta r}{dt} = \frac{\partial I}{\partial V}\delta V + \frac{\partial I}{\partial \gamma}\delta\gamma \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$$\frac{d\delta \lambda}{dt} = \frac{\partial J}{\partial V}\delta V + \frac{\partial J}{\partial \gamma}\delta\gamma + \frac{\partial J}{\partial \psi}\delta\psi + \frac{\partial J}{\partial r}\delta r \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

$$\frac{d\delta \tau}{dt} = \frac{\partial K}{\partial V}\delta V + \frac{\partial K}{\partial \gamma}\delta\gamma + \frac{\partial K}{\partial \psi}\delta\psi + \frac{\partial K}{\partial r}\delta r + \frac{\partial K}{\partial \lambda}\delta\lambda \quad 0 \quad 0 \quad 0 \quad 0$$

$$\frac{d\delta m}{dt} = 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0$$

ADJOINT EQUATIONS

The adjoint differential equations, of the form shown in Eqs. (5-8) through (5-10) and adjoint to the perturbation equations of the preceding page, are as follows:

$$\frac{d\lambda_V}{dt} = -\lambda_v \frac{\partial F}{\partial V} - \lambda_\gamma \frac{\partial G}{\partial V} - \lambda_\psi \frac{\partial H}{\partial V} - \lambda_r \frac{\partial I}{\partial V} - \lambda_\lambda \frac{\partial J}{\partial V} - \lambda_\tau \frac{\partial K}{\partial V}$$

$$\frac{d\lambda_\gamma}{dt} = -\lambda_v \frac{\partial F}{\partial \gamma} - \lambda_\gamma \frac{\partial G}{\partial \gamma} - \lambda_\psi \frac{\partial H}{\partial \gamma} - \lambda_r \frac{\partial I}{\partial \gamma} - \lambda_\lambda \frac{\partial J}{\partial \gamma} - \lambda_\tau \frac{\partial K}{\partial \gamma}$$

$$\frac{d\lambda_r}{dt} = -\lambda_v \frac{\partial F}{\partial r} - \lambda_\gamma \frac{\partial G}{\partial r} - \lambda_\psi \frac{\partial H}{\partial r} - \lambda_\lambda \frac{\partial J}{\partial r} - \lambda_\tau \frac{\partial K}{\partial r}$$

$$\frac{d\lambda_\psi}{dt} = -\lambda_v \frac{\partial F}{\partial \psi} - \lambda_\gamma \frac{\partial G}{\partial \psi} - \lambda_\psi \frac{\partial H}{\partial \psi} - \lambda_\lambda \frac{\partial J}{\partial \psi} - \lambda_\tau \frac{\partial K}{\partial \psi}$$

$$\frac{d\lambda_\lambda}{dt} = -\lambda_v \frac{\partial F}{\partial \lambda} - \lambda_\gamma \frac{\partial G}{\partial \lambda} - \lambda_\psi \frac{\partial H}{\partial \lambda} - \lambda_\tau \frac{\partial K}{\partial \lambda}$$

$$\frac{d\lambda_\tau}{dt} = -\lambda_v \frac{\partial F}{\partial \tau} - \lambda_\gamma \frac{\partial G}{\partial \tau} - \lambda_\psi \frac{\partial H}{\partial \tau}$$

$$\frac{d\lambda_m}{dt} = -\lambda_v \frac{\partial F}{\partial m} - \lambda_\gamma \frac{\partial G}{\partial m} - \lambda_\psi \frac{\partial H}{\partial m}$$

$$\Lambda_\theta = -\lambda_v \frac{\partial F}{\partial \theta} - \lambda_\gamma \frac{\partial G}{\partial \theta} - \lambda_\psi \frac{\partial H}{\partial \theta}$$

$$\Lambda_\chi = -\lambda_v \frac{\partial F}{\partial \chi} - \lambda_\gamma \frac{\partial G}{\partial \chi} - \lambda_\psi \frac{\partial H}{\partial \chi}$$

ADJOINT EQUATIONS
EVALUATION OF PARTIAL DERIVATIVES

$$\frac{\partial F}{\partial V} = -\frac{1}{m}\rho VA\left[C_D + \frac{1}{2}\frac{V}{a}\frac{\partial C_D}{\partial M}\right]$$

$$\frac{\partial F}{\partial \gamma} = r\,\omega^2\left[\cos^2\lambda\cos\gamma + \sin\lambda\cos\lambda\cos\psi\sin\gamma\right] - g\cos\gamma \quad *$$

$$+ \frac{T}{m}\left[\frac{\partial A_{21}}{\partial\gamma}\;\frac{\partial A_{22}}{\partial\gamma}\;\frac{\partial A_{23}}{\partial\gamma}\right][B]\,[B']\begin{bmatrix}\cos\theta\sin\chi\\ \cos\theta\cos\chi\\ \sin\theta\end{bmatrix}$$

$$\frac{\partial F}{\partial r} = \omega^2\left[\cos^2\lambda\sin\gamma - \sin\lambda\cos\lambda\cos\psi\cos\gamma\right] + \frac{2\mu}{r^3}\sin\gamma \quad *$$

$$- \frac{A_e}{m}\frac{dp}{dh}\left[C_{21}\cos\theta\sin\chi + C_{22}\cos\theta\cos\chi + C_{23}\sin\theta\right]$$

$$+ \frac{AV^2}{2m}\left[\rho\frac{V}{a^2}\frac{da}{dh}\frac{\partial C_D}{\partial M} - C_D\frac{d\rho}{dh}\right]$$

$$\frac{\partial F}{\partial m} = -\frac{T}{m^2}\left[C_{21}\cos\theta\sin\chi + C_{22}\cos\theta\cos\chi + C_{23}\sin\theta\right] + \frac{D}{m^2}$$

$$\frac{\partial F}{\partial \theta} = \frac{T}{m}\left[-C_{21}\sin\theta\sin\chi - C_{22}\sin\theta\cos\chi + C_{23}\cos\theta\right]$$

$$\frac{\partial F}{\partial \chi} = \frac{T}{m}\left[C_{21}\cos\theta\cos\chi - C_{22}\cos\theta\sin\chi\right]$$

$$\frac{\partial F}{\partial \psi} = r\,\omega^2\sin\lambda\cos\lambda\sin\psi\cos\gamma$$

$$+ \frac{T}{m}\left[\frac{\partial A_{21}}{\partial\psi}\;\frac{\partial A_{22}}{\partial\psi}\;\frac{\partial A_{23}}{\partial\psi}\right][B]\,[B']\begin{bmatrix}\cos\theta\sin\chi\\ \cos\theta\cos\chi\\ \sin\theta\end{bmatrix}$$

---

*Neglect oblateness terms in adjoint equations

$$\frac{\partial F}{\partial \lambda} = r\,\omega^2 \left[-2 \cos\lambda \sin\lambda \sin\gamma - \cos^2\lambda \cos\psi \cos\gamma + \sin^2\lambda \cos\psi \cos\gamma\right]$$

$$+ \frac{T}{m} \left[\frac{\partial A_{21}}{\partial \lambda} \quad \frac{\partial A_{22}}{\partial \lambda} \quad \frac{\partial A_{23}}{\partial \lambda} \right] [B][B'] \begin{bmatrix} \cos\theta \sin\chi \\ \cos\theta \cos\chi \\ \sin\theta \end{bmatrix}$$

$$\frac{\partial F}{\partial \tau} = \frac{T}{m} \left[\frac{\partial A_{21}}{\partial \tau} \quad \frac{\partial A_{22}}{\partial \tau} \quad \frac{\partial A_{23}}{\partial \tau} \right] [B][B'] \begin{bmatrix} \cos\theta \sin\chi \\ \cos\theta \cos\chi \\ \sin\theta \end{bmatrix}$$

$$\frac{\partial G}{\partial V} = \frac{\cos\gamma}{r} - \frac{r\,\omega^2}{V^2} \left[\cos^2\lambda \cos\gamma + \sin\lambda \cos\lambda \cos\psi \sin\gamma\right] + \frac{g}{V^2} \cos\gamma$$

$$- \frac{T}{mV^2} \left[C_{31} \cos\theta \sin\chi + C_{32} \cos\theta \cos\chi + C_{33} \sin\theta\right] + \frac{\rho A}{2m} \left[C_L + \frac{V}{a} \frac{\partial C_L}{\partial M}\right]$$

$$\frac{\partial G}{\partial \gamma} = -\frac{V}{r} \sin\gamma + \frac{r\,\omega^2}{V} \left[\sin\lambda \cos\lambda \cos\psi \cos\gamma - \cos^2\lambda \sin\gamma\right]$$

$$+ \frac{g}{V} \sin\gamma + \frac{T}{mV} \left[\frac{\partial A_{31}}{\partial \gamma} \quad \frac{\partial A_{32}}{\partial \gamma} \quad \frac{\partial A_{33}}{\partial \gamma} \right] [B]\,[B'] \begin{bmatrix} \cos\theta \sin\chi \\ \cos\theta \cos\chi \\ \sin\theta \end{bmatrix}$$

$$- \rho \frac{VA}{2m} \left(\frac{\partial C_L}{\partial \alpha}\right)$$

$$\frac{\partial G}{\partial r} = -\frac{V}{r^2} \cos\gamma + \frac{\omega^2}{V} \left[\cos^2\lambda \cos\gamma + \sin\lambda \cos\lambda \cos\psi \sin\gamma\right]$$

$$+ \frac{2\mu}{r^3 V} \cos\gamma - \frac{Ae}{mV} \frac{dp}{dh} \left[C_{31} \cos\theta \sin\chi + C_{32} \cos\theta \cos\chi + C_{33} \sin\theta\right]$$

$$+ \frac{VA}{2m} \left[C_L \frac{d\rho}{dh} - \rho \frac{V}{a^2} \frac{da}{dh} \frac{\partial C_L}{\partial M}\right]$$

$$\frac{\partial G}{\partial m} = -\frac{T}{m^2 V}\left[+C_{31}\cos\theta\sin\chi + C_{32}\cos\theta\cos\chi + C_{33}\sin\theta\right] - \frac{(FL)}{m^2 V}$$

$$\frac{\partial G}{\partial \theta} = \frac{T}{mV}\left[-C_{31}\sin\theta\sin\chi - C_{32}\sin\theta\cos\chi + C_{33}\cos\theta\right] + \frac{\rho VA}{2m}\frac{\partial C_L}{\partial \alpha}$$

$$\frac{\partial G}{\partial \chi} = \frac{T}{mV}\left[C_{31}\cos\theta\cos\chi - C_{32}\cos\theta\sin\chi\right]$$

$$\frac{\partial G}{\partial \psi} = 2\omega\cos\lambda\cos\psi - \frac{r\omega^2}{V}\sin\lambda\cos\lambda\sin\psi\sin\gamma$$

$$+ \frac{T}{mV}\left[\frac{\partial A_{31}}{\partial \psi}\ \frac{\partial A_{32}}{\partial \psi}\ \frac{\partial A_{33}}{\partial \psi}\right][B]\,[B']\begin{bmatrix}\cos\theta\sin\chi\\ \cos\theta\cos\chi\\ \sin\theta\end{bmatrix}$$

$$\frac{\partial G}{\partial \lambda} = \frac{r\omega^2}{V}\left[-2\cos\lambda\sin\lambda\cos\gamma + \cos^2\lambda\cos\psi\sin\gamma - \sin^2\lambda\cos\psi\sin\gamma\right]$$

$$+ 2\omega\sin\lambda\sin\psi + \frac{T}{mV}\left[\frac{\partial A_{31}}{\partial \lambda}\ \frac{\partial A_{32}}{\partial \lambda}\ \frac{\partial A_{33}}{\partial \lambda}\right][B][B']\begin{bmatrix}\cos\theta\sin\chi\\ \cos\theta\cos\chi\\ \sin\theta\end{bmatrix}$$

$$\frac{\partial G}{\partial \tau} = \frac{T}{mV}\left[\frac{\partial A_{31}}{\partial \tau}\ \frac{\partial A_{32}}{\partial \tau}\ \frac{\partial A_{33}}{\partial \tau}\right][B]\,[B']\begin{bmatrix}\cos\theta\sin\chi\\ \cos\theta\cos\chi\\ \sin\theta\end{bmatrix}$$

$$\frac{\partial H}{\partial V} = \frac{\cos\gamma\sin\psi\sin\lambda}{r\cos\lambda} - \frac{r\omega^2\sin\lambda\cos\lambda\sin\psi}{V^2\cos\gamma}$$

$$- \frac{T}{mV^2\cos\gamma}\left[C_{11}\cos\theta\sin\chi + C_{12}\cos\theta\cos\chi + C_{13}\sin\theta\right]$$

$$\frac{\partial H}{\partial \gamma} = - \frac{V \sin \gamma \sin \psi \sin \lambda}{r \cos \lambda} + \frac{r\omega^2 \sin \lambda \cos \lambda \sin \psi \sin \gamma}{V \cos^2 \gamma}$$

$$- \frac{2\omega \cos \lambda \cos \psi}{\cos^2 \gamma} + \frac{T \sin \gamma}{mV \cos^2 \gamma} \left[C_{11} \cos \theta \sin \chi + C_{12} \cos \theta \cos \chi + C_{13} \sin \theta \right]$$

$$+ \frac{T}{mV \cos \gamma} \left[\frac{\partial A_{11}}{\partial \gamma} \; \frac{\partial A_{12}}{\partial \gamma} \; \frac{\partial A_{13}}{\partial \gamma} \right] [B] \, [B'] \begin{bmatrix} \cos \theta \sin \chi \\ \cos \theta \cos \chi \\ \sin \theta \end{bmatrix}$$

$$\frac{\partial H}{\partial r} = - \frac{V \cos \gamma \sin \psi \sin \lambda}{r^2 \cos \lambda} + \frac{\omega^2 \sin \lambda \cos \lambda \sin \psi}{V \cos \gamma}$$

$$- \frac{A_e \, (dp/dh)}{mV \cos \gamma} \left[C_{11} \cos \theta \sin \chi + C_{12} \cos \theta \cos \chi + C_{13} \sin \theta \right]$$

$$\frac{\partial H}{\partial m} = - \frac{T}{m^2 V \cos \gamma} \left[C_{11} \cos \theta \sin \chi + C_{12} \cos \theta \cos \chi + C_{13} \sin \theta \right]$$

$$\frac{\partial H}{\partial \theta} = \frac{T}{mV \cos \gamma} \left[-C_{11} \sin \theta \sin \chi - C_{12} \sin \theta \cos \chi + C_{13} \cos \theta \right]$$

$$\frac{\partial H}{\partial \chi} = \frac{T}{mV \cos \gamma} \left[C_{11} \cos \theta \cos \chi - C_{12} \cos \theta \sin \chi \right]$$

$$\frac{\partial H}{\partial \psi} = \frac{V \cos \gamma \cos \psi \sin \lambda}{r \cos \lambda} + \frac{r\omega^2 \sin \lambda \cos \lambda \cos \psi}{V \cos \gamma} + \frac{2\omega \cos \lambda \sin \psi \sin \gamma}{\cos \gamma}$$

$$+ \frac{T}{mV \cos \gamma} \left[\frac{\partial A_{11}}{\partial \psi} \; \frac{\partial A_{12}}{\partial \psi} \; \frac{\partial A_{13}}{\partial \psi} \right] [B] \, [B'] \begin{bmatrix} \cos \theta \sin \chi \\ \cos \theta \cos \chi \\ \sin \theta \end{bmatrix}$$

$$\frac{\partial H}{\partial \lambda} = \frac{V \cos \gamma \sin \psi}{r \cos^2 \lambda} + \frac{r\omega^2 \sin \psi}{V \cos \gamma}\left[\cos^2 \lambda - \sin^2 \lambda\right] + \frac{2\omega \sin \lambda \cos \psi \sin \gamma}{\cos \gamma}$$

$$+ 2\omega \cos \lambda + \frac{T}{mV \cos \gamma}\left[\frac{\partial A_{11}}{\partial \lambda} \quad \frac{\partial A_{12}}{\partial \lambda} \quad \frac{\partial A_{13}}{\partial \lambda}\right][B][B'] \begin{bmatrix} \cos \theta \sin \chi \\ \cos \theta \cos \chi \\ \sin \theta \end{bmatrix}$$

$$\frac{\partial H}{\partial \tau} = \frac{T}{mV \cos \gamma}\left[\frac{\partial A_{11}}{\partial \tau} \quad \frac{\partial A_{12}}{\partial \tau} \quad \frac{\partial A_{13}}{\partial \tau}\right][B]\,[B'] \begin{bmatrix} \cos \theta \sin \chi \\ \cos \theta \cos \chi \\ \sin \theta \end{bmatrix}$$

$$\frac{\partial I}{\partial V} = \sin \gamma \qquad\qquad \frac{\partial K}{\partial V} = \frac{\cos \gamma \sin \psi}{r \cos \lambda}$$

$$\frac{\partial I}{\partial \gamma} = V \cos \gamma \qquad\qquad \frac{\partial K}{\partial \gamma} = -\frac{V \sin \gamma \sin \psi}{r \cos \lambda}$$

$$\frac{\partial J}{\partial V} = \frac{\cos \gamma \cos \psi}{r} \qquad\qquad \frac{\partial K}{\partial r} = -\frac{V \cos \gamma \sin \psi}{r^2 \cos \lambda}$$

$$\frac{\partial J}{\partial \gamma} = -\frac{V \sin \gamma \cos \psi}{r} \qquad\qquad \frac{\partial K}{\partial \psi} = \frac{V \cos \gamma \cos \psi}{r \cos \lambda}$$

$$\frac{\partial J}{\partial r} = -\frac{V \cos \gamma \cos \psi}{r^2} \qquad\qquad \frac{\partial K}{\partial \lambda} = \frac{V \cos \gamma \sin \psi \sin \lambda}{r \cos^2 \lambda}$$

$$\frac{\partial J}{\partial \psi} = -\frac{V \cos \gamma \sin \psi}{r}$$

ADJOINT EQUATIONS

PARTIAL DERIVATIVES OF $\underline{A}$ MATRIX

$$\frac{\partial A_{11}}{\partial \gamma} = 0$$

$$\frac{\partial A_{12}}{\partial \gamma} = 0$$

$$\frac{\partial A_{13}}{\partial \gamma} = 0$$

$$\frac{\partial A_{11}}{\partial \psi} = \sin \tau \sin \psi + \cos \tau \sin \lambda \cos \psi$$

$$\frac{\partial A_{12}}{\partial \psi} = -\cos \tau \sin \psi + \sin \tau \sin \lambda \cos \psi$$

$$\frac{\partial A_{13}}{\partial \psi} = -\cos \lambda \cos \psi$$

$$\frac{\partial A_{11}}{\partial \lambda} = \cos \tau \cos \lambda \sin \psi$$

$$\frac{\partial A_{12}}{\partial \lambda} = \sin \tau \cos \lambda \sin \psi$$

$$\frac{\partial A_{13}}{\partial \lambda} = \sin \lambda \sin \psi$$

$$\frac{\partial A_{11}}{\partial \tau} = -\cos \tau \cos \psi - \sin \tau \sin \lambda \sin \psi$$

$$\frac{\partial A_{12}}{\partial \tau} = -\sin \tau \cos \psi + \cos \tau \sin \lambda \sin \psi$$

$$\frac{\partial A_{13}}{\partial \tau} = 0$$

$$\frac{\partial A_{21}}{\partial \gamma} = \cos\tau \sin\lambda \cos\psi \sin\gamma + \sin\tau \sin\psi \sin\gamma + \cos\tau \cos\lambda \cos\gamma$$

$$\frac{\partial A_{22}}{\partial \gamma} = \sin\tau \sin\lambda \cos\psi \sin\gamma - \cos\tau \sin\psi \sin\gamma + \sin\tau \cos\lambda \cos\gamma$$

$$\frac{\partial A_{23}}{\partial \gamma} = -\cos\lambda \cos\psi \sin\gamma + \sin\lambda \cos\gamma$$

$$\frac{\partial A_{21}}{\partial \psi} = \cos\tau \sin\lambda \sin\psi \cos\gamma - \sin\tau \cos\psi \cos\gamma$$

$$\frac{\partial A_{22}}{\partial \psi} = \sin\tau \sin\lambda \sin\psi \cos\gamma + \cos\tau \cos\psi \cos\gamma$$

$$\frac{\partial A_{23}}{\partial \psi} = -\cos\lambda \sin\psi \cos\gamma$$

$$\frac{\partial A_{21}}{\partial \lambda} = -\cos\tau \cos\lambda \cos\psi \cos\gamma - \cos\tau \sin\lambda \sin\gamma$$

$$\frac{\partial A_{22}}{\partial \lambda} = -\sin\tau \cos\lambda \cos\psi \cos\gamma - \sin\tau \sin\lambda \sin\gamma$$

$$\frac{\partial A_{23}}{\partial \lambda} = -\sin\lambda \cos\psi \cos\gamma + \cos\lambda \sin\gamma$$

$$\frac{\partial A_{21}}{\partial \tau} = \sin\tau \sin\lambda \cos\psi \cos\gamma - \cos\tau \sin\psi \cos\gamma - \sin\tau \cos\lambda \sin\gamma$$

$$\frac{\partial A_{22}}{\partial \tau} = -\cos\tau \sin\lambda \cos\psi \cos\gamma - \sin\tau \sin\psi \cos\gamma + \cos\tau \cos\lambda \sin\gamma$$

$$\frac{\partial A_{23}}{\partial \tau} = 0$$

$$\frac{\partial A_{31}}{\partial \gamma} = -\cos\tau \cos\lambda \sin\gamma + \cos\tau \sin\lambda \cos\psi \cos\gamma + \sin\tau \sin\psi \cos\gamma$$

$$\frac{\partial A_{32}}{\partial \gamma} = -\sin\tau \cos\lambda \sin\gamma + \sin\tau \sin\lambda \cos\psi \cos\gamma - \cos\tau \sin\psi \cos\gamma$$

$$\frac{\partial A_{33}}{\partial \gamma} = -\sin\lambda \sin\gamma - \cos\lambda \cos\psi \cos\gamma$$

$$\frac{\partial A_{31}}{\partial \psi} = -\cos\tau \sin\lambda \sin\psi \sin\gamma + \sin\tau \cos\psi \sin\gamma$$

$$\frac{\partial A_{32}}{\partial \psi} = -\sin\tau \sin\lambda \sin\psi \sin\gamma - \cos\tau \cos\psi \sin\gamma$$

$$\frac{\partial A_{33}}{\partial \psi} = \cos\lambda \sin\psi \sin\gamma$$

$$\frac{\partial A_{31}}{\partial \lambda} = -\cos\tau \sin\lambda \cos\gamma + \cos\tau \cos\lambda \cos\psi \sin\gamma$$

$$\frac{\partial A_{32}}{\partial \lambda} = -\sin\tau \sin\lambda \cos\gamma + \sin\tau \cos\lambda \cos\psi \sin\gamma$$

$$\frac{\partial A_{33}}{\partial \lambda} = \cos\lambda \cos\gamma + \sin\lambda \cos\psi \sin\gamma$$

$$\frac{\partial A_{31}}{\partial \tau} = -\sin\tau \cos\lambda \cos\gamma - \sin\tau \sin\lambda \cos\psi \sin\gamma + \cos\tau \sin\psi \sin\gamma$$

$$\frac{\partial A_{32}}{\partial \tau} = \cos\tau \cos\lambda \cos\gamma + \cos\tau \sin\lambda \cos\psi \sin\gamma + \sin\tau \sin\psi \sin\gamma$$

$$\frac{\partial A_{33}}{\partial \tau} = 0$$

INITIAL CONDITIONS FOR INTEGRATION OF THE ADJOINT EQUATIONS

One set of adjoint equations, as defined in the preceding pages is solved for each constraint which the user wishes to impose on the trajectory. As discussed earlier in Section 5.2, the initial conditions given each set of adjoint variables are functions of the constraint parameter. Since the constraints are applied at stage ignition or burnout times, the initial conditions are simply the partial derivatives of the constraint with respect to the trajectory variables evaluated at the point on the trajectory at which the constraint is to be applied. This is the case for all constraints at stage points and for the terminal constraints.

On the next pages are given the various partial derivatives of all the constraint parameters used in the ICS subroutine, where the adjoint variables are initialized. The equations and nomenclature defining the constraint parameters are given in Section 4.

PARTIAL DERIVATIVES OF INERTIAL TRAJECTORY VARIABLES

$$\frac{\partial V_I}{\partial V} = \frac{V + \omega r \cos \lambda \cos \gamma \sin \psi}{V_I}$$

$$\frac{\partial V_I}{\partial r} = \frac{(V \cos \gamma \sin \psi)\,(\omega r \cos \lambda) + (\omega r \cos \lambda)^2}{r\, V_I}$$

$$\frac{\partial V_I}{\partial \gamma} = -(\omega r \cos \lambda)\,(\sin \gamma_I) \sin \psi$$

$$\frac{\partial V_I}{\partial \psi} = \frac{(\omega r \cos \lambda)\, V \cos \gamma \cos \psi}{V_I}$$

$$\frac{\partial V_I}{\partial \lambda} = -\frac{\sin \lambda}{\cos \lambda}\left[\frac{(V \cos \gamma \sin \psi)\,(\omega r \cos \lambda) + (\omega r \cos \lambda)^2}{V_I}\right]$$

$$\frac{\partial \gamma_I}{\partial V} = \left(\frac{\tan \gamma_I}{V_I}\right)\left[\frac{V_I}{V} - \left(\frac{\partial V_I}{\partial V}\right)\right]$$

$$\frac{\partial \gamma_I}{\partial r} = -\left(\frac{\tan \gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial r}\right)$$

$$\frac{\partial \gamma_I}{\partial \gamma} = \frac{V}{V_I}\,\frac{\cos \gamma}{\cos \gamma_I} - \left(\frac{\tan \gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial \gamma}\right)$$

$$\frac{\partial \gamma_I}{\partial \psi} = -\left(\frac{\tan \gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial \psi}\right)$$

$$\frac{\partial \gamma_I}{\partial \lambda} = -\left(\frac{\tan \gamma_I}{V_I}\right)\left(\frac{\partial V_I}{\partial \lambda}\right)$$

$$\frac{\partial \psi_I}{\partial V} = -\frac{(\omega r \cos \lambda)}{V} \left(\frac{\cos^2 \psi_I}{V \cos \gamma \cos \psi} \right)$$

$$\frac{\partial \psi_I}{\partial r} = \frac{(\omega r \cos \lambda)}{r} \left(\frac{\cos^2 \psi_I}{V \cos \gamma \cos \psi} \right)$$

$$\frac{\partial \psi_I}{\partial \gamma} = (\omega r \cos \lambda) (\tan \gamma) \left(\frac{\cos^2 \psi_I}{V \cos \gamma \cos \psi} \right)$$

$$\frac{\partial \psi_I}{\partial \psi} = \frac{\cos^2 \psi_I}{\cos^2 \psi} + \frac{\sin \psi}{\cos \psi} (\omega r \cos \lambda) \left(\frac{\cos^2 \psi_I}{V \cos \gamma \cos \psi} \right)$$

$$\frac{\partial \psi_I}{\partial \lambda} = -\omega r \sin \lambda \left(\frac{\cos^2 \psi_I}{V \cos \gamma \cos \psi} \right)$$

PARTIAL DERIVATIVES OF ORBIT ELEMENTS

ENERGY

$$\frac{\partial E}{\partial V_I} = V_I \qquad \frac{\partial E}{\partial r} = \frac{\mu}{r^2}$$

ANGULAR MOMENTUM

$$\frac{\partial \bar{H}}{\partial V_I} = r \cos \gamma_I$$

$$\frac{\partial \bar{H}}{\partial r} = V_I \cos \gamma_I$$

$$\frac{\partial \bar{H}}{\partial \gamma_I} = -rV_I \sin \gamma_I$$

PERIGEE RADIUS

$$\frac{\partial r_p}{\partial V_I} = \frac{\left(\frac{\partial \bar{H}}{\partial V_I}\right) - \left(\frac{r_p^2}{\bar{H}}\right)\left(\frac{\partial E}{\partial V_I}\right)}{\sqrt{2E + \left(\frac{\mu}{\bar{H}}\right)^2}}$$

$$\frac{\partial r_p}{\partial r} = \frac{\left(\frac{\partial \bar{H}}{\partial r}\right) - \left(\frac{r_p^2}{\bar{H}}\right)\left(\frac{\partial E}{\partial r}\right)}{\sqrt{2E + \left(\frac{\mu}{\bar{H}}\right)^2}}$$

$$\frac{\partial r_p}{\partial \gamma_I} = \frac{\left(\frac{\partial \bar{H}}{\partial \gamma_I}\right)}{\sqrt{2E + \left(\frac{\mu}{\bar{H}}\right)^2}}$$

INCLINATION

$$\frac{\partial i}{\partial \lambda} = \frac{\sin \lambda \sin \psi_I}{\sin i}$$

$$\frac{\partial i}{\partial \psi_I} = - \frac{\cos \lambda \cos \psi_I}{\sin i}$$

LONGITUDE OF ASCENDING NODE

$$\frac{\partial \Omega_e}{\partial \lambda} = - \frac{\cos i \cos \psi_I}{\sin^2 i}$$

$$\frac{\partial \Omega_e}{\partial \psi_I} = - \frac{\sin \lambda}{\sin^2 i}$$

$$\frac{\partial \Omega_e}{\partial \tau} = 1$$

$$\frac{\partial \Omega_e}{\partial T_G} = 4.1068643 \times 10^{-3} \text{ deg/sec}$$

ARGUMENT OF PERIGEE

$$\frac{\partial \beta_p}{\partial V_I} = \frac{2(\sin^2 \zeta)\left(\frac{\mu}{\bar{H}}\right)\left(\frac{r}{\bar{H}}\right)}{V_I \tan \gamma_I}$$

$$\frac{\partial \beta_p}{\partial r} = \frac{\sin^2 \zeta \left(\frac{\mu}{\bar{H}}\right)}{\bar{H} \tan \gamma_I}$$

$$\frac{\partial \beta_p}{\partial \gamma} = -(\sin^2 \zeta) \left[\frac{1 - \left(\frac{\mu}{\bar{H}}\right)\left(\frac{r}{\bar{H}}\right)}{\sin^2 \gamma_I} + 2 \left(\frac{\mu}{\bar{H}}\right)\left(\frac{r}{\bar{H}}\right) \right]$$

$$\frac{\partial \beta_p}{\partial \lambda} = \frac{\sin \beta \cos \beta}{\cos \lambda \cos \lambda}$$

$$\frac{\partial \beta_p}{\partial \psi_I} = \frac{\sin \beta \cos \beta \sin \psi_I}{\cos \psi_I}$$

LOCAL ORBITAL VELOCITY

$$V_{ORB} = V_I - \sqrt{\frac{\mu}{r}}$$

$$\frac{\partial V_{ORB}}{\partial V_I} = 1.$$

$$\frac{\partial V_{ORB}}{\partial r} = \frac{\partial V_I}{\partial r} + \frac{1}{2r} \sqrt{\frac{\mu}{r}}$$

DOWNRANGE DISTANCE

$$\frac{\partial R_D}{\partial \lambda} = \frac{1}{\sin R_D} \left[\cos \lambda_o \sin \lambda \cos (\tau - \tau_o) - \sin \lambda_o \cos \lambda \right]$$

$$\frac{\partial R_D}{\partial \tau} = \frac{1}{\sin R_D} \left[\cos \lambda_o \cos \lambda \sin (\tau - \tau_o) \right]$$

DYNAMIC PRESSURE

$$\frac{\partial \bar{q}}{\partial V} = \rho V$$

$$\frac{\partial \bar{q}}{\partial r} = \frac{V^2}{2} \frac{d\rho}{dh}$$

DERIVATION OF OPTIMIZATION EQUATIONS

The initial type of computation performed by the Scout computer program is one of solving for the boost trajectory characteristics which maximize the desired payoff function (payload, velocity, or altitude) while satisfying various constraints on the trajectory. The technique applied to this optimization problem is known as the "method of steepest descent." It makes possible the optimization of a function of time (pitch program) and discrete trajectory parameters such as coast durations. The trajectory optimization program known as PRESTO* employes the methods of steepest descent and constituted the basic building block upon which the Scout program was developed. The following pages show a derivation of the steepest descent control equations, including the use of the adjoint variables discussed in the first part of this section.

Define the following matrices:

$$d\psi = \begin{bmatrix} d\psi_1 \\ d\psi_2 \\ \cdot \\ \cdot \\ d\psi_{jc} \end{bmatrix}$$

where $d\psi_1$ is the desired change in payoff parameter at the end of the trajectory, and $d\psi_i$, $i = 2, jc$, are the desired changes in the constraints. jc is the number of constraints, including the payoff.

$$\Lambda = \begin{bmatrix} \Lambda\theta_1 \\ \Lambda\theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \Lambda\theta_{jc} \end{bmatrix}$$

where $\Lambda\theta_1 = - \left[\lambda_{v_i} \frac{\partial F}{\partial \theta} + \lambda_{\gamma_i} \frac{\partial G}{\partial \theta} + \lambda_{\psi_i} \frac{\partial H}{\partial \theta} \right]$

$$\Lambda_{\chi_1} = - \left[\lambda_{v_i} \frac{\partial F}{\partial \chi} + \lambda_{\gamma_i} \frac{\partial G}{\partial \chi} + \lambda_{\psi_i} \frac{\partial H}{\partial \chi} \right]$$

---

*See reference on page 3-1.

$$\lambda = \begin{bmatrix} \lambda_{v_1} & \lambda_{\gamma_1} & \lambda_{r_1} & \lambda_{\psi_1} & \lambda_{\lambda_1} & \lambda_{\tau_1} & \lambda_{m_1} \\ \lambda_{v_2} & \lambda_{\gamma_2} & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \lambda_{v_{jc}} & \lambda_{\gamma_{jc}} & \lambda_{r_{jc}} & \lambda_{\psi_{jc}} & \lambda_{\lambda_{jc}} & \lambda_{\tau_{jc}} & \lambda_{m_{jc}} \end{bmatrix}$$

where the $\lambda_{(\ )_i}$'s are the solution of the <u>ith</u> set of adjoint equations

$$\delta x = \begin{bmatrix} \delta V \\ \delta \gamma \\ \delta r \\ \delta \tau^* \\ \delta m \\ \delta \psi \\ \delta \lambda \end{bmatrix}$$

where δV = (V on present trajectory) - (V on nominal trajectory) at the same time.

$\delta \tau^*$ is the deviation in inertial longitude

$$Y = \begin{bmatrix} y_1 & & & & 0 \\ & y_2 & & & \\ & & y_3 & & \\ 0 & & & \ddots & \\ & & & & y_m \end{bmatrix}$$

where y_1 is the weighting constant for the first adjustable parameter

$\delta\theta$ = the change in the pitch plane component of thrust attitude.

$$S = \begin{bmatrix} S_{11} & S_{12} & \cdot & \cdot & \cdot & S_{1_m} \\ S_{21} & \cdot & \cdot & \cdot & \cdot & S_{2_m} \\ S_{31} & \cdot & & & & \\ \cdot & & & & & \\ \cdot & & & & & \\ S_{jc_1} & \cdot & \cdot & \cdot\cdot & \cdot & S_{jc_m} \end{bmatrix}$$

where S_{i_k} is the effect of the <u>kth</u> adjustable parameter on the <u>ith</u> constraint.

$$\delta\tau = \begin{bmatrix} \delta\tau_1 \\ \delta\tau_2 \\ \cdot \\ \cdot \\ \delta\tau_m \end{bmatrix}$$

where $\delta\tau_i$ is the change in the <u>ith</u> adjustable parameter.

The effects of changes in the trajectory variables, control variables, and adjustable parameters on the terminal quantities are given by

$$d\psi = \lambda\delta x + \int_{t_f}^{t_i} \Lambda \, \delta\theta \, dt + S \, \delta\tau \qquad (5\text{-}19)$$

One desires to determine the value of $\delta\theta(t)$ and $\delta\tau$ that will permit (5-19) to be satisfied for a given $d\psi$. There is an infinity of solutions to this problem. In order to determine a unique solution, one further requires that the solution to Eq. (5-19) maximize the following quantity, i.e., make Q, which is a negative number, as large as possible in absolute value.

$$Q = \int_{t_f}^{t_1} (\delta\theta)^2 \, dt - \delta\tau^T Y \, \delta\tau \qquad (5\text{-}20)$$

The superscript T stands for transpose.

The problem, then, is to find $\delta\theta$ and $\delta\tau$ which maximize Q, while satisfying the constraint on $d\psi$. Making use of Lagrange multipliers μ, form the quantity

$$Z = Q + \mu^T \left(d\psi - \lambda\delta x - \int_{t_f}^{t_1} \Lambda\delta\theta \, dt - S \, \delta\tau \right) \qquad (5\text{-}21)$$

Combining terms gives

$$Z = \int_{t_f}^{t_1} \left[(\delta\theta)^2 - \mu^T \Lambda\delta\theta \right] dt + \mu^T (d\psi - \lambda\delta x) - \delta\tau^T Y\delta\tau - \mu^T S\delta\tau \qquad (5\text{-}22)$$

Note that Z is equal to Q. The $\delta\theta$ and $\delta\tau$ that maximize Z will also maximize Q. δZ must equal zero for arbitrary changes in $\delta\theta$ and $\delta\tau$.

$$\delta Z = 0 = \int_{t_f}^{t_1} \left[2\delta\theta \, \delta(\delta\theta) - \mu^T \Lambda \, \delta(\delta\theta) \right] dt - \delta\tau^T Y \, \delta(\delta\tau) - \delta(\delta\tau)^T Y \, \delta\tau - \mu^T S \, \delta(\delta\tau) \qquad (5\text{-}23)$$

or

$$\delta Z = 0 = \int_{t_f}^{t_1} \left[2 \, \delta\theta - \mu^T \Lambda \right] \delta(\delta\theta) \, dt - (2 \, \delta\tau^T Y + \mu^T S) \, \delta(\delta\tau) \qquad (5\text{-}24)$$

The coefficients of $\delta(\delta\theta)$ and $\delta(\delta\tau)$ must equal zero if δZ is to be zero for changes in $\delta\theta$ and $\delta\tau$. Therefore,

$$\delta\theta = 0.5\,\mu^T \Lambda \qquad \text{or} \qquad \delta\theta = 0.5\,\Lambda^T \mu \qquad (5\text{-}25)$$

and

$$\delta\tau^T = -0.5\,\mu^T S\, Y^{-1} \qquad \text{or} \qquad \delta\tau = -0.5\, Y^{-1} S^T \mu \qquad (5\text{-}26)$$

To solve for μ, substitute Eqs. (5-25) and (5-26) into (5-19).

$$d\psi = \lambda\delta x + \frac{1}{2}\left[\int_{t_f}^{t_1} \Lambda\, \Lambda^T dt\right]\mu - \frac{1}{2} S\, Y^{-1} S^T \mu \qquad (5\text{-}27)$$

and

$$\mu = 2\left[\int_{t_f}^{t_1} \Lambda\, \Lambda^T dt - S\, Y^{-1} S^T\right]^{-1} \left[d\psi - \lambda\delta x\right] \qquad (5\text{-}28)$$

Substitute back into Eqs. (5-25) and (5-26) to give

$$\delta\theta = \Lambda^T \left[\int_{t_f}^{t_1} \Lambda\, \Lambda^T dt - S\, Y^{-1} S^T\right]^{-1} (d\psi - \lambda\delta x) \qquad (5\text{-}29)$$

$$\delta\tau = -Y^{-1} S^T \left[\int_{t_f}^{t} \Lambda\, \Lambda^T dt - S\, Y^{-1} S^T\right]^{-1} (d\psi - \lambda\delta x) \qquad (5\text{-}30)$$

At this point it is useful to indicate the difference between the Scout optimization procedure and that of previous programs.[1,2] In References 1 and 2, the matrix in the brackets in Eqs. (5-29) and (5-30) is inverted only once at the initial time t_1. The inverted matrix is then

---

[1] Kelley, H. J., "Gradient Theory of Optimal Flight Paths," ARS Journal, 30, 947-953 (1960).

[2] Bryson, A. E., and Denham, W. F., "A Steepest-Ascent Method for Solving Optimum Programming Problems," J. Applied Mechanics, 29, 247-257 (1962).

multiplied by the vector $d\psi$ (δx is zero, assuming initial conditions to be fixed) to form a vector of constants. This constant vector is then multiplied by Λ^T, which is a function of time, to determine the entire $\delta\theta$ time history for the trajectory. $\delta\tau$ is evaluated in the same manner with $Y^{-1} S^T$ used instead of Λ^T. Thus, the changes in the control and the adjustable parameters are fixed at the beginning of the forward trajectory. In control system terminology, this is an open-loop system.

In the Scout program, the first 20 points are all treated as the initial point. The time t_1 becomes the running variable t and the bracketed matrix is inverted at each of these points of integration while running a backward trajectory. During forward trajectories, the change in the control to be used for the remainder of the trajectory is recomputed at each point, taking into account the deviation from the nominal trajectory at that point. The program operates in a closed-loop fashion because it continuously checks how it is doing in its attempt to satisfy terminal conditions.

The advantage of this closed-loop approach is that larger deviations from the nominal trajectory can be tolerated while still meeting terminal conditions. It is, therefore, possible to move more rapidly from the initial nominal trajectory to the optimum trajectory.

The closed-loop mode of operation can be used with convenience only up to the point at which the first constraint is to be applied. In the Scout program, stage 1 burnout defines that upper limit. For simplicity and core storage requirements, the number of closed loop points was fixed at 20. At this point in stage 1, the computation switches to open-loop for the rest of the trajectory. The $\delta\tau$ for launch azimuth and time of day are computed at the start of the trajectory using Eq. (5-30), and the remaining $\delta\tau$

are computed at the switch point.

Selection of Initial Payoff Improvement

In general, one does not know how far from the optimum a given nominal trajectory will be. It is, therefore, difficult to guess how much payoff improvement to ask for. On the other hand, one can predict reasonable values for the expected changes in the control variable. The procedure to be used, then, is to guess the change in the control and let the computer determine the corresponding change in payoff. The required calculations are given here.

Define dP^2 as

$$dP^2 = \int_{t_f}^{t_1} (\delta\theta)^2 \, dt - \delta\tau^T Y \delta\tau \qquad (5\text{-}31)$$

dP^2 is a measure of the amount of control change. To obtain an estimate of dP^2, an average value of $\delta\theta$ is selected. This is squared and multiplied by the expected burn time to get an approximate value for the integral. Reasonable changes in the adjustable parameters are also selected and their squares, modified by the weighting functions, are added to the integral terms to determine dP^2.

Substitute Eqs. (5-29) and (5-30) with $\delta x=0$ into Eq. (5-31) to obtain dP^2 in terms of the changes in the terminal constraints.

$$dP^2 = d\psi^T \left[\int_{t_f}^{t_1} \Lambda \Lambda^T dt - S Y^{-1} S^T \right]^{-1} d\psi \qquad (5\text{-}32)$$

Let the inverted matrix be denoted as A with components A_{ij}. Split the $d\psi$ vector into two parts, δm_d and $d\phi$.

$$d\psi = \begin{bmatrix} \delta m_d \\ d\Phi \end{bmatrix}$$

where δm_d is the initial change in payoff and

$$d\phi = \begin{bmatrix} d\psi_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ d\psi_{jc} \end{bmatrix}$$

The object now is to solve Eq. (5-32) for δm_d in terms of dP^2 and $d\phi$. Rewrite Eq. (5-32) as

$$dP^2 = \begin{bmatrix} \delta m_d & d\phi \end{bmatrix} \begin{bmatrix} A_{11} & A_{12} & \cdot & \cdot & \cdot & \cdot & \cdot \\ A_{21} & A_{22} & A_{23} & & & & \\ \cdot & A_{32} & \cdot & & & & \end{bmatrix} \begin{bmatrix} \delta m_d \\ d\Phi \end{bmatrix}$$

Let the minor of A_{11} be designated as

$$[M] = \begin{bmatrix} A_{22} & A_{23} & \cdot & \cdot & \cdot \\ A_{32} & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & \\ \cdot & & & & \end{bmatrix}$$

$$\text{Then } dP^2 = A_{11}\, \delta m_d^{\,2} + 2N^T\, d\phi\, \delta m_d + d\phi^T [M] \tag{5-33}$$

where

$$N = \begin{bmatrix} A_{21} \\ A_{31} \\ \cdot \\ \cdot \end{bmatrix}$$

Solving Eq. (5-33) for δm_d gives

$$\delta m_d = \frac{-N^T d\phi}{A_{11}} + \sqrt{\frac{dP^2 - d\phi^T \left([M] - \frac{NN^T}{A_{11}} \right) d\phi}{A_{11}}} \qquad (5\text{-}34)$$

The above derivation explains the type of calculation which the program makes. The actual details of computation which are used are explained more fully in Section 9.6.

SECTION 6

PITCH PROGRAM LINEARIZATION EQUATIONS

LINEARIZATION OF ASCENT TILT PROGRAM

Theory

The objective of the linearization subroutine is to convert the thrust attitude history obtained during the trajectory optimization into a pitch program composed of linear segments. The linear program must satisfy the specified trajectory constraints, and the loss in the payoff function due to linearization should be negligible.

The payoff loss will be small if the linear pitch program closely approximates the optimum program. It is mathematically convenient to obtain this proximity by minimizing the time integral of the square of the difference between the two pitch programs. In equation form, we want to minimize

$$\int_{t_r}^{t_1} (\delta\theta)^2 \, dt \tag{6-1}$$

subject to the constraint

$$d\psi = \int_{t_r}^{t_1} \Lambda_\theta \, \delta\theta \, dt \tag{6-2}$$

where

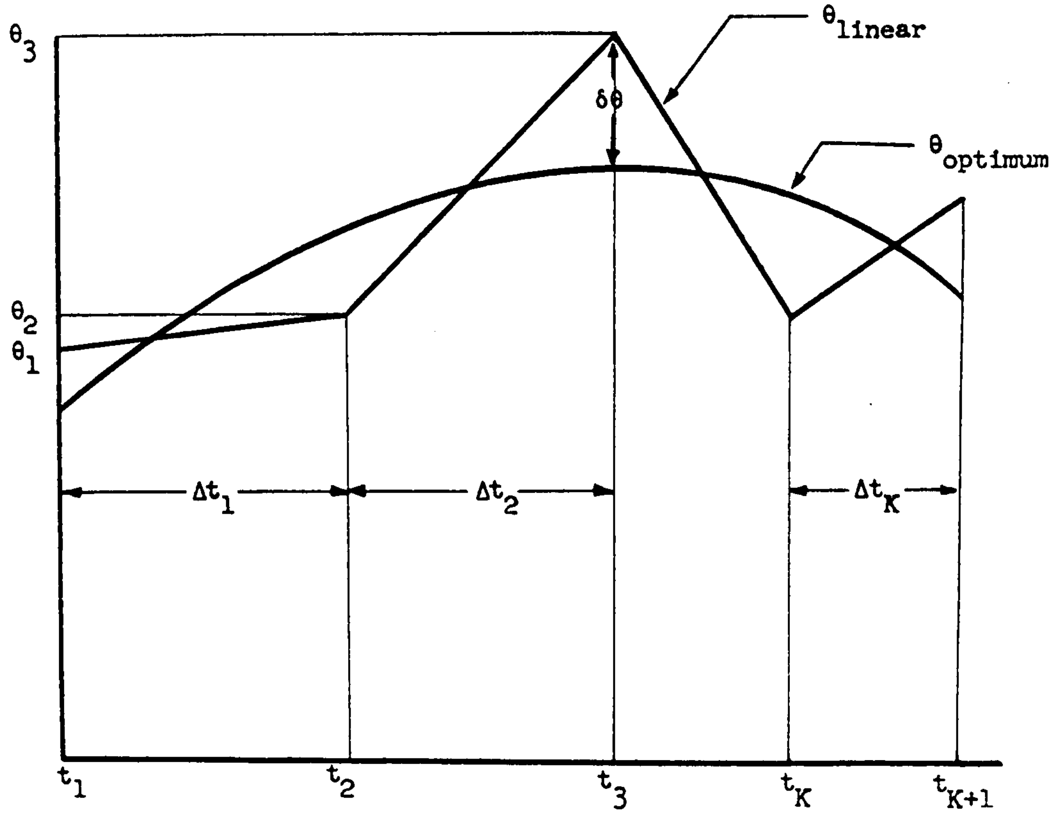
$$\delta\theta = \theta_{linear} - \theta_{optimum} \tag{6-3}$$

$d\psi$ is a vector of trajectory constraint perturbations.

Λ_θ is a vector of influence functions which gives the effect of a change in the control variable on the constraints.

Λ_θ is used during the optimization procedure and is therefore available without any further computation. The $d\psi$ are the errors in the constraints when the optimum pitch program is used.

The first problem is to write expression (6-1) and Eq. (6-2) in terms of the parameters that determine the linear pitch program. The most convenient set of parameters are the values of the pitch attitude at the break points where the slope changes. Consider the figure shown below.



θ_{linear} is described by a separate equation in each segment. One may write

$$\theta_{linear} = \theta_1 + \frac{\theta_2-\theta_1}{\Delta t_1}(t-t_1) = \theta_1\left(\frac{t_2-t}{\Delta t_1}\right) + \theta_2\left(\frac{t-t_1}{\Delta t_1}\right) ; \; t_1 \le t \le t_2$$

$$= \theta_2 + \frac{\theta_3-\theta_2}{\Delta t_2}(t-t_2) = \theta_2\left(\frac{t_3-t}{\Delta t_2}\right) + \theta_3\left(\frac{t-t_2}{\Delta t_2}\right); \quad t_2 \le t \le t_3$$

$$= \theta_K + \frac{\theta_{K+1}-\theta_K}{\Delta t_K}(t-t_K) = \theta_K\left(\frac{t_{K+1}-t}{\Delta t_K}\right) + \theta_{K+1}\left(\frac{t-t_K}{\Delta t_K}\right); \quad t_K \le t \le t_{K+1}$$

$$(6\text{-}4)$$

$\delta\theta$ is found by substituting for θ_{linear} in Eq. (6-3).

Eq. (6-2) is written in terms of the θ_i by substituting for $\delta\theta$. One obtains

$$d\psi = -\int_{t_{K+1}}^{t_1} \theta_{OPT}\,\Lambda_\theta\,dt + \theta_1\left[\frac{1}{\Delta t_1}\int_{t_2}^{t_1}(t_2-t)\,\Lambda_\theta\,dt\right]$$

$$+ \theta_2\left[\frac{1}{\Delta t_1}\int_{t_2}^{t_1}(t-t_1)\,\Lambda_\theta\,dt + \frac{1}{\Delta t_2}\int_{t_3}^{t_2}(t_3-t)\,\Lambda_\theta\,dt\right]$$

$$+ \cdots + \theta_K\left[\frac{1}{\Delta t_{K-1}}\int_{t_K}^{t_{K-1}}(t-t_{K-1})\,\Lambda_\theta\,dt + \frac{1}{\Delta t_K}\int_{t_{K+1}}^{t_K}(t_{K+1}-t)\,\Lambda_\theta\,dt\right]$$

$$+ \theta_{K+1}\left[\frac{1}{\Delta t_K}\int_{t_{K+1}}^{t_K}(t-t_K)\,\Lambda_\theta\,dt\right] \qquad (6\text{-}5)$$

Eq. (6-5) can be written in a more compact form.

$$d\psi = L + M\theta = \begin{bmatrix} L_1 \\ L_2 \\ \cdot \\ \cdot \\ \cdot \\ L_{JC} \end{bmatrix} + \begin{bmatrix} M_{11} & M_{12} & & M_{1(K+1)} \\ & & & \\ & & & \\ M_{JC1} & & & M_{JC(K+1)} \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \\ \cdot \\ \cdot \\ \cdot \\ \theta_{K+1} \end{bmatrix}$$

where

$$L_i = -\int_{t_{K+1}}^{t_1} \theta_{OPT} \Lambda\theta_i \, dt \qquad i = 1, JC$$

$$M_{i1} = \frac{1}{\Delta t_1} \int_{t_2}^{t_1} (t_2 - t) \Lambda\theta_i \, dt \qquad i = 1, JC$$

$$M_{ij} = \frac{1}{\Delta t_{j-1}} \int_{t_j}^{t_{j-1}} (t - t_{j-1}) \Lambda\theta_i \, dt + \frac{1}{\Delta t_j} \int_{t_{j+1}}^{t_j} (t_{j+1} - t) \Lambda\theta_i \, dt$$

$$i = 1, JC$$
$$j = 2, K$$

$$M_{i(K+1)} = \frac{1}{\Delta t_K} \int_{t_{K+1}}^{t_K} (t - t_K) \Lambda\theta_i \, dt \qquad i = 1, JC$$

Before evaluating expression (6-1), we will break it up into a sum of integrals and introduce a weighting function η_i for each segment. Thus, expression (6-1) is replaced by

$$\eta_1 \int_{t_1}^{t_2} \delta\theta^2 \, dt + \eta_2 \int_{t_2}^{t_3} \delta\theta^2 \, dt + \cdots \qquad \eta_K \int_{t_K}^{t_{K+1}} \delta\theta^2 \, dt \qquad (6\text{-}6)$$

The weighting functions are introduced so that the $\delta\theta$ in any segment can be more closely controlled. If η_1 were to be made much larger than the other η's, for example, the result would be to decrease the $\delta\theta$ in the first segment.

Consider the first segment.

$$\eta_1 \int_{t_1}^{t_2} \delta\theta^2 \, dt = \eta_1 \int_{t_1}^{t_2} \frac{1}{\Delta t_1^2} \left[\theta_1(t_2-t) + \theta_2(t-t_1) - \theta_{OPT}\right]^2 dt$$

After integrating, one obtains

$$\eta_1 \int_{t_1}^{t_2} \delta\theta^2 \, dt = \eta_1 \left[A\theta_1^2 + A\theta_2^2 + 2B\theta_1\theta_2 + D\theta_1 + E\theta_2 + F\right] \qquad (6\text{-}7)$$

where

$$A = \frac{\Delta t_1}{3}$$

$$B = \frac{1}{2} A$$

$$D = \frac{2}{\Delta t_1} \int_{t_2}^{t_1} \theta_{OPT} \, (t_2-t) \, dt$$

$$E = \frac{2}{\Delta t_1} \int_{t_2}^{t_1} \theta_{OPT} \, (t-t_1) \, dt$$

$$F = - \int_{t_2}^{t_1} \theta_{OPT}^2 \, dt$$

The same expression can be written for each of the k linear segments. The integral over all segments can be written as

$$\sum_{i=1}^{k} \eta_i \int_{t_i}^{t_{i+1}} \delta\theta^2 \, dt = \eta_1 \left[A_1\theta_1^2 + 2B_1\theta_1\theta_2 + A_1\theta_2^2 + D_1\theta_1 + E_1\theta_2 + F_1 \right]$$

$$+ \eta_2 \left[A_2\theta_2^2 + 2B_2\theta_1\theta_2 + A_2\theta_3^2 + D_2\theta_2 + E_2\theta_3 + F_2 \right]$$

$$\vdots$$

$$+ \eta_K \left[A_K\theta_K^2 + 2B_K\theta_K\theta_{K+1} + A_K\theta_{K+1}^2 + D_K\theta_K + E_K\theta_{K+1} + F_K \right] \tag{6-8}$$

After collecting terms, one obtains

$$\sum_{i=1}^{k} \eta_i \int_{t_i}^{t_{i+1}} \delta\theta^2 \, dt = \eta_1 A_1 \theta_1^2 + (\eta_1 A_1 + \eta_2 A_2)\theta_2^2 + (\eta_2 A_2 + \eta_3 A_3)\theta_3^2$$

$$+ \ldots . (\eta_{K-1} A_{K-1} + \eta_K A_K)\theta_K^2$$

$$+ \eta_K A_K \theta_{K+1}^2 + 2\eta_1 B_1 \theta_1 \theta_2 + 2\eta_2 B_2 \theta_2 \theta_3 + \ldots 2\eta_K B_K \theta_K \theta_{K+1} + \eta_1 D_1 \theta_1$$

$$+ (\eta_1 E_1 + \eta_2 D_2)\theta_2 + (\eta_2 E_2 + \eta_3 D_3)\theta_3 + \ldots (\eta_{K-1} E_{K-1} + \eta_K D_K)\theta_K$$

$$+ \eta_K E_K \theta_{K+1}$$

$$+ \sum_{j=1}^{K} \eta_j F_j \tag{6-9}$$

This may be written as the sum of a quadratic and a linear expression plus a constant.

$$\sum_{i=1}^{K} \eta_1 \int_{t_i}^{t_{i+1}} \delta\theta^2 \, dt = \theta^T P\theta + N^T \theta + F \tag{6-10}$$

where

$$P = \begin{bmatrix} \eta_1A_1 & \eta_1B_1 & & & 0 \\ \eta_1B_1 & (\eta_1A_1+\eta_2A_2) & \eta_2B_2 & & \\ & \eta_2B_2 & (\eta_2A_2+\eta_3A_3) & \ddots & \\ & & \ddots & \ddots & \eta_KB_K \\ & 0 & & \eta_KB_K & \eta_KA_K \end{bmatrix}$$

P is symmetric with all terms equal to zero except for the middle three diagonals.

$$N^T = \left[\eta_1D_1 \;\; (\eta_1E_1+\eta_2D_2) \quad (\eta_2E_2+\eta_3D_3) \quad \cdots\cdots \quad (\eta_{K-1}E_{K-1}+\eta_KD_K) \quad \eta_KE_K\right]$$

$$F = \sum_{j=1}^{K} \eta_jF_j$$

A representation has now been found for expression (6-1) and Eq. (6-2) in terms of the values of the linear pitch program at the break points. It is now possible to find the values, θ_i , which minimize (6-1), subject to constraint (6-2). This is done with the aid of Lagrange multipliers. One forms the quantity

$$V = \theta^T P\theta + N^T \theta + F + \mu\,(d\psi - L - M\theta) \tag{6-11}$$

where μ is a vector of Lagrange multipliers. We want to minimize V with respect to θ. To do this, evaluate δV.

$$\delta V = (2\theta^T P + N^T - \mu M)\,\delta\theta \tag{6-12}$$

θ is determined by setting the coefficient of $\delta\theta$ equal to zero.

$$\theta = \frac{1}{2} P^{-1} (\mu M - N^T)^T \tag{6-13}$$

To evaluate μ, substitute θ in the equation

$$d\psi = L + M\theta$$

One obtains

$$\mu = (2d\psi^T + N^T P^{-1} M^T - 2L^T) (MP^{-1} M^T)^{-1} \tag{6-14}$$

The final equation for θ is then found by substituting Eq. (6-14) into Eq. (6-13). The result is

$$\theta = \frac{1}{2} P^{-1} \left[(2d\psi^T + N^T P^{-1} M^T - 2L^T) (MP^{-1} M^T)^{-1} M - N^T\right]^T \tag{6-15}$$

THRUST ATTITUDE LINEARIZATION WITH SPECIFIED VALUES FOR SOME θ_i

Because of hardware considerations, it is necessary to take certain practical constraints into consideration when linearizing the thrust attitude history. These constraints are:

1. The pitch rate cannot be too large or too small.
2. The change in pitch rate from one segment to the next cannot be too large or too small.
3. There can be only one pitch rate segment with a sign different from the other segments.

If any pitch rate limit is violated, the thrust attitude at the beginning or end of the offending segment is fixed at an acceptable value and the attitude history linearized once again.

The equation for θ (Eq. 6-15) must be modified to make provision for fixing certain θ_i. A derivation of this modified equation is given here.

Let the vector θ be made up of two components, $\bar{\theta}$ and θ_f, i.e., $\theta = \bar{\theta} + \theta_f$. $\bar{\theta}$ is composed of the adjustable parts of θ while θ_f contains the fixed parts. Both $\bar{\theta}$ and θ_f have ℓ components. If the <u>ith</u> component of θ is fixed, then $\bar{\theta}_i = 0$ and θ_{f_i} is the fixed value of θ at that time point.

With θ in this form, the integral of $\delta\theta^2$ is written as (see Eq. 6-10):

$$\bar{\theta}^T\bar{P}\bar{\theta} + \theta_f{}^T P\bar{\theta} + \bar{\theta}^T P\theta_f + \theta_f{}^T P\theta_f + N^T\bar{\theta} + N^T\theta_f + F$$

where the superscript T means transpose.
$\bar{P}$ is equal to P with the rows and columns corresponding to the fixed θ's set to zero except for the diagonal element which is set to one.

The constraint equation becomes

$$d\psi = L + M\bar{\theta} + M\theta_f \tag{6-16}$$

The derivation now follows that of the previous section, beginning with Eq. (6-11). The quantity V becomes

$$V = \bar{\theta}^T\bar{P}\bar{\theta} + \theta_f{}^T P\bar{\theta} + \bar{\theta}^T P\theta_f + \theta_f{}^T P\theta_f + N^T\bar{\theta} + N^T\theta_f + F$$
$$+ \mu(L + M\bar{\theta} + M\theta_f) \tag{6-17}$$

The differential δV is

$$\delta V = (2\,\bar{\theta}^T\bar{P} + N^T - \mu M + 2\theta_f{}^T P)\,\delta\bar{\theta} \qquad (6\text{-}18)$$

Setting the coefficient of $\delta\bar{\theta}$ equal to zero, one obtains

$$\bar{\theta}^T = \tfrac{1}{2}\,(\mu M - N^T - 2\theta_f{}^T P)\,\bar{P}^{-1} \qquad (6\text{-}19)$$

It should be noted that the diagonal terms in $\bar{P}^{-1}$ corresponding to the fixed θ's are set to zero. Take the transpose of Eq. (6-19) to obtain

$$\bar{\theta} = \tfrac{1}{2}\,\bar{P}^{-1}\,(\mu M - N^T - 2\theta_f{}^T P)^T \qquad (6\text{-}20)$$

Substitute Eq. (6-20) into Eq. (6-16) to determine μ. The result is

$$\mu = \left[2d\psi^T + N^T\bar{P}^{-1}M^T - 2L^T - 2\theta_f{}^T\,(M^T - P\bar{P}^{-1}M^T)\right](M\bar{P}^{-1}M^T)^{-1} \qquad (6\text{-}21)$$

Substitute μ into Eq. (6-20) to obtain the final result.

$$\bar{\theta} = \tfrac{1}{2}\,\bar{P}^{-1}\Big\{\left[2d\psi^T + N^T\bar{P}^{-1}M^T - 2L^T - 2\theta_f{}^T\,(M^T - P\bar{P}^{-1}M^T)\right] (M\bar{P}^{-1}M^T)^{-1}M - N^T - 2\theta_f{}^T P\Big\}^T \qquad (6\text{-}22)$$

This result is identical to Eq. (6-15) when $\theta_f = 0$. Eq. (6-22) is therefore used in place of Eq. (6-15) even when no constraints have been violated.

The complete θ history is then obtained from the equation

$$\theta = \bar{\theta} + \theta_f \qquad (6\text{-}23)$$

SECTION 7

BODY DYNAMICS SIMULATION

DERIVATION OF BODY DYNAMICS EQUATIONS

In developing the Scout program, one of the additions made to the basic PRESTO* point-mass trajectory optimization package was a technique for including body dynamics effects on the boost trajectory. This is necessary in design of the optimum command thrust-attitude history and in evaluation of probable dispersion envelopes. An approximation to the body rotational motion is employed, which yields a suitable compromise between program execution time and overall accuracy. A brief derivation of this technique is given here.

The equations of rotational motion in one plane can be written, under the assumptions of linearization of the overall system and omission of actuator and sensor dynamics, as follows:

$$\ddot{\theta} = M_\delta \delta + M_\alpha \alpha \qquad (7\text{-}1)$$

where θ = pitch attitude angle

δ = control system deflection

α = angle of attack

$$\alpha = \theta - \gamma \qquad (7\text{-}2)$$

where γ = flight path angle

$$\delta = K_\theta(\theta_c - \theta) - K_q \dot{\theta} \qquad (7\text{-}3)$$

where θ_c = commanded θ

K_θ = autopilot position gain

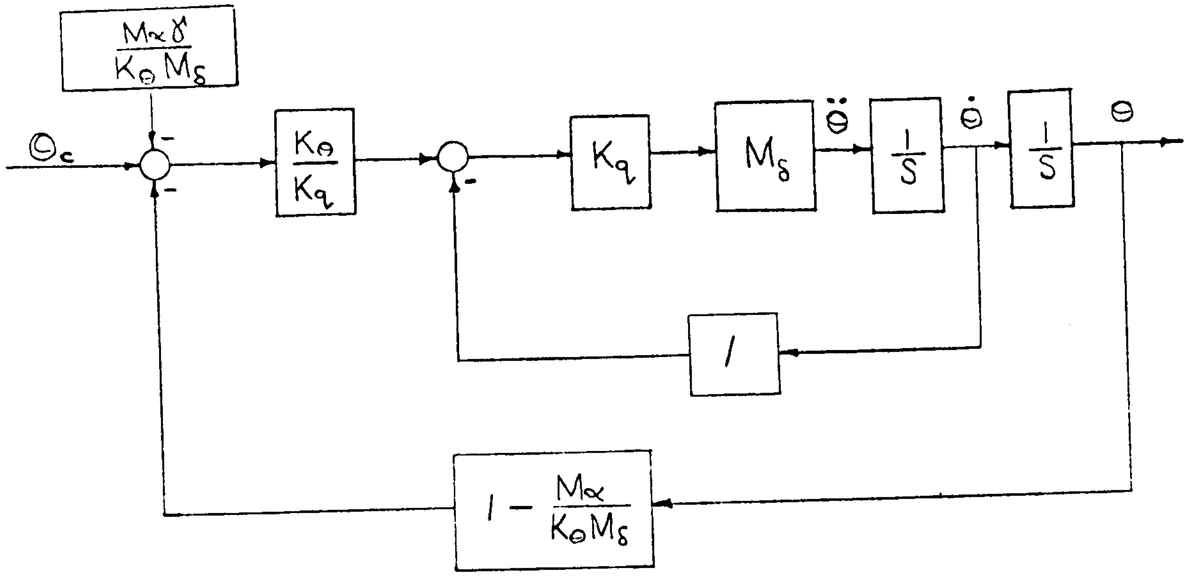
K_q = autopilot rate gain

---

*See reference, page 3-1.

$$\ddot{\theta} = K_\theta M_\delta(\theta_c - \theta) - K_q M_\delta \dot{\theta} + M_\alpha \theta - M_\alpha \gamma \qquad (7\text{-}4)$$

This equation can be represented by the following block diagram.



A full-dynamic simulation, then, would include two integrations of $\ddot{\theta}$. This is a very time-consuming computation since the integration frequency must be at least thirty times the motion frequency. On the other hand, it is known that the primary effect on the trajectory is determined by the steady-state solution to equation (7-4), rather than the transient response which requires the high integration frequency. The steady-state thrust attitude represents a moment-balance condition and can be evaluated by inspection of the above block diagram with the inner loop eliminated (or set = 1.).

Thus,

$$\dot{\theta} = \frac{K_\theta}{K_q} \left[\theta_c - \theta\left(1 - \frac{M_\alpha}{K_\theta M_\delta}\right) - \frac{M_\alpha \gamma}{K_\theta M_\delta} \right]$$

from which

$$\theta = \frac{K_\theta M_\delta \theta_c - K_q M_\delta \dot{\theta}_c - M_\alpha \gamma}{K_\theta M_\delta - M_\alpha} \qquad (7\text{-}5)$$

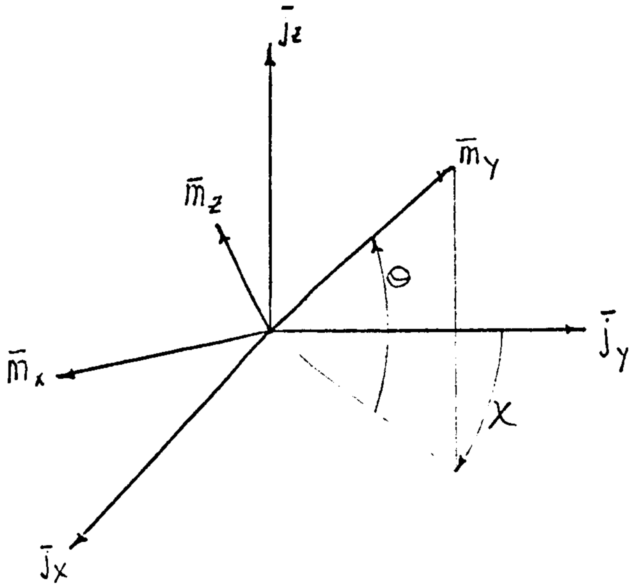
by assuming $\dot{\theta} = \dot{\theta}_c$ in steady state.

An equation of this form for θ, along with an analogous expression for the yaw angle χ, is used in computation of the nominal trajectory. Additional terms to account for thrust misalignments and winds are added for computation of dispersed trajectories.

The actual equations for θ and χ require proper evaluation of the angle γ in Eq. (7-5), as follows.

Let the inertially-fixed platform axes be the P-system of axes.

PLATFORM (P) AND MISSILE (m) COORDINATE SYSTEMS



The missile axes are the m-system. $\bar{m}_x$ is always in the $\bar{j}_x, \bar{j}_y$ plane. The thrust is directed along the $\bar{m}_y$ axis, which is the center line of the vehicle.

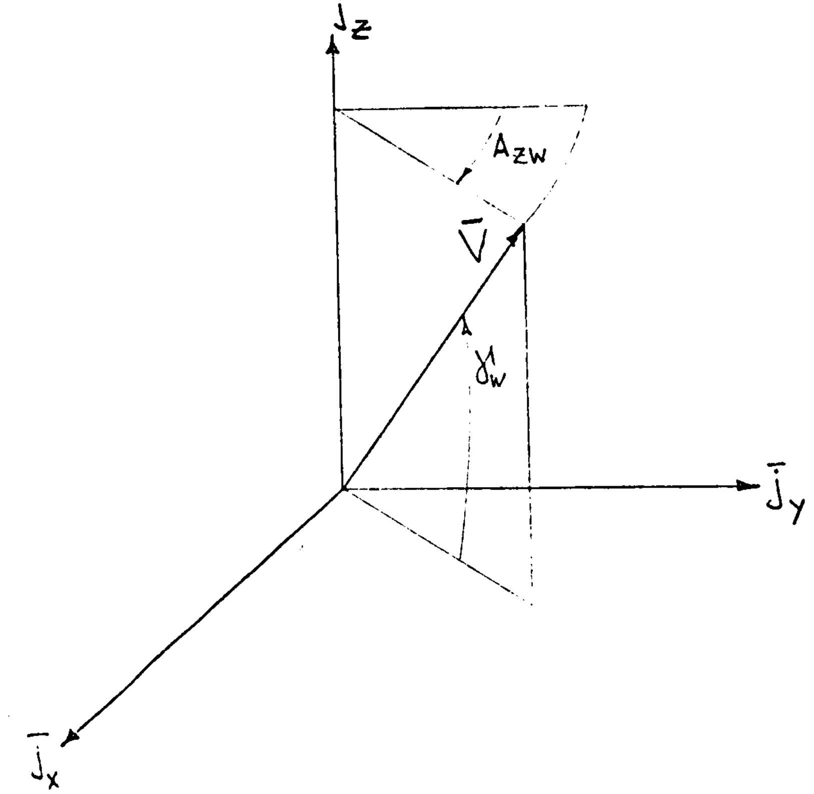
To begin we find the components of $\bar{V}$, the velocity with respect to the rotating Earth, along the P-system. The transformation from P to L is given by the C matrix defined in Section 4. To go from L to P we invert the C matrix. Thus,

$$\begin{bmatrix} V_{j_x} \\ V_{j_y} \\ V_{j_z} \end{bmatrix} = C^{-1} \begin{bmatrix} V_{i_x} \\ V_{i_y} \\ V_{i_z} \end{bmatrix}$$

V_{i_x} and V_{i_z} are 0. Therefore,

$$\begin{bmatrix} V_{j_x} \\ V_{j_y} \\ V_{j_z} \end{bmatrix} = \begin{bmatrix} C^{-1}_{12} \\ C^{-1}_{22} \\ C^{-1}_{32} \end{bmatrix} |\bar{V}|$$

We now compute the angles γ_w and A_{zw} shown below.



$$\gamma_w = \tan^{-1}\left(\frac{V_{jz}}{\sqrt{V_{jx}^2 + V_{jy}^2}}\right)$$

$$A_{zw} = \tan^{-1}\left(\frac{V_{jx}}{V_{jy}}\right)$$

The quantities θ and χ that appear in the equations of motion are then replaced by

$$\theta = \frac{K_\theta M_\delta \theta_c - K_q M_\delta \dot{\theta}_c - M_\alpha \gamma_w}{K_\theta M_\delta - M_\alpha} \qquad (7\text{-}6)$$

$$\chi = \frac{K_\theta M_\delta \chi_c - K_q M_\delta \dot{\chi}_c - M_\alpha A_{zw}}{K_\theta M_\delta - M_\alpha} \qquad (7\text{-}7)$$

where θ_c is the desired thrust attitude and $\dot{\theta}_c$ is the pitch rate. χ_c is 0 for the first three stages. $\dot{\chi}_c$ is always 0.

SECTION 8

PERFORMANCE EXCHANGE RATIOS

PERFORMANCE EXCHANGE RATIOS

One of the optional computations available in the Scout program whenever a trajectory optimization is performed is the evaluation of performance exchange ratios. These quantities are first-order sensitivities of the payoff parameter (payload, velocity, or altitude) to changes in five vehicle characteristics in each of the powered stages, evaluated under the requirement that all trajectory constraints still be satisfied. To first order, they show the increase or decrease in performance that would result from a change in one of the variables, assuming a new optimum trajectory. The five stage variables are burn rate, propellant weight, specific impulse, aerodynamic drag, and inert jettison weight. The exchange ratios are computed analytically using the solution of the adjoint equations as the basis of the calculation. The computation is performed during the backward guidance run preceding the final guidance run at completion of optimization. With this approach, up to twenty-five performance partials are obtained with less than one minute of computation time. A derivation of the relationships yielding the exchange ratios is provided at this point.

In order that the analytical technique be best understood, the derivation of the exchange ratio equations will be performed for a simplified problem. Then the corresponding relationships for the complete problem will be written in matrix notation.

Consider a single-stage vehicle operating outside of the atmosphere. Exchange ratios for specific impulse and initial weight will be evaluated since they are representative of the two types of parameters to be treated. This formulation is developed for a two-dimensional trajectory which satisfies a constraint on radius at the end of the trajectory. The equations of motion and mass flow rate can be written as

$$F = \dot{V} = \frac{T}{m} \cos \eta - g \sin \gamma \tag{8-1}$$

$$G = \dot{\gamma} = \frac{T}{mV} \sin \eta + \left(\frac{V}{r} - \frac{g}{V} \right) \cos \gamma \tag{8-2}$$

$$H = \dot{r} = V \sin \gamma \tag{8-3}$$

$$I = \dot{m} = - \frac{T}{g_o \, I_{sp}} \tag{8-4}$$

where η is the angle between the velocity and thrust vectors, and the definitions of the remaining variables are given on page 4-1.

Assume that a nominal trajectory meeting terminal conditions has been determined. One is interested in finding the influence of changes in the vehicle parameters on payload, assuming that the angle of attack is adjusted so that the terminal constraints are still satisfied.

The adjoint differential equations, with the partial derivatives evaluated along the nominal trajectory, are (analogous to Section 5.4)

$$\frac{d\lambda_V}{dt} = - \frac{\partial G}{\partial V} \lambda_\gamma - \frac{\partial H}{\partial V} \lambda_r \tag{8-5}$$

$$\frac{d\lambda_\gamma}{dt} = - \frac{\partial F}{\partial \gamma} \lambda_V - \frac{\partial G}{\partial \gamma} \lambda_\gamma - \frac{\partial H}{\partial \gamma} \lambda_r \tag{8-6}$$

$$\frac{d\lambda_r}{dt} = - \frac{\partial F}{\partial r} \lambda_V - \frac{\partial G}{\partial r} \lambda_\gamma \tag{8-7}$$

$$\frac{d\lambda_m}{dt} = - \frac{\partial F}{\partial m} \lambda_V - \frac{\partial G}{\partial m} \lambda_\gamma \tag{8-8}$$

Their solution has the following property:

$$\left[\lambda_V \; \delta V + \lambda_\gamma \; \delta_\gamma + \lambda_r \; \delta r + \lambda_m \; \delta m\right]_{t=t_f} = \left[\lambda_V \; \delta V + \lambda_\gamma \; \delta_\gamma + \lambda_r \; \delta r + \lambda_m \; \delta m\right]_{t=t_o}$$

$$- \int_{t_f}^{t_o} \left(\lambda_v \frac{\partial F}{\partial \eta} + \lambda_\gamma \frac{\partial G}{\partial \eta}\right) \delta_\eta \; dt - \int_{t_f}^{t_o} \lambda_m \frac{\partial I}{\partial I_{sp}} \; \delta I_{sp} \; dt \qquad (8\text{-}9)$$

The initial conditions for these adjoint equations are specified at the time the stopping parameter is reached, t_f. They are functions only of the terminal constraints and the stopping parameter. One separate solution of the adjoint equations is required for each terminal constraint and one for the payoff function mass.

At the initial time, t_o , the perturbations δV, $\delta\gamma$, and δr are zero. Furthermore, δI_{sp} is constant. Assume radius to be the only terminal constraint and let the second subscripts ϕ and ψ indicate that the adjoint equations are solved using the initial conditions associated with mass and radius, respectively.

Using Eq. (8-9) the expressions for terminal deviations in mass and radius are

$$\delta m_f = \lambda_{m\phi} \; \delta m_o + \int_{t_f}^{t_o} \Lambda_\phi \; \delta_\eta \; dt - R_\phi \; \delta I_{sp} \qquad (8\text{-}10)$$

$$\delta r_f = \lambda_{m\psi} \; \delta m_o + \int_{t_f}^{t_o} \Lambda_\psi \; \delta_\eta \; dt - R_\psi \; \delta I_{sp} \qquad (8\text{-}11)$$

where

$$\Lambda = - \lambda_v \frac{\partial F}{\partial \eta} - \lambda_\gamma \frac{\partial G}{\partial \eta} \qquad (8\text{-}11a)$$

and

$$R = \int_{t_f}^{t_o} \lambda_m \frac{\partial I}{\partial I_{sp}} \; dt \qquad (8\text{-}11b)$$

and $\lambda_{mf} = \lambda_{rf} = 1.0$.

We have assumed a nominal trajectory that meets terminal conditions, i.e.,

$\delta r_f = 0$. In order to maintain this condition in the presence of perturbations in initial mass and specific impulse, it will be necessary to adjust the angle of attack, η. The minimum change in η that will enable the terminal conditions to be met is found be setting $\delta\eta = C\Lambda_r$. Substituting $\delta\eta$ in Eq. (8-10) and (8-11) gives

$$\delta m_f = \lambda_{m\phi} \cdot \delta m_o - R_\phi \, \delta I_{sp} + CI_{\psi\phi} \tag{8-12}$$

$$\delta r_f = \lambda_{m\psi} \, \delta m_o - R_\psi \cdot \delta I_{sp} + CI_{\psi\psi} \tag{8-13}$$

where

$$I_{\psi\phi} = \int_{t_f}^{t_o} \Lambda_\psi \, \Lambda_\phi \, dt$$

and $I_{\phi\phi}$, correspondingly. Solving for C, with the condition that $\delta r_f = 0$, one obtains $\delta\eta$

$$\delta\eta = \frac{-\lambda_{m\psi} \, \delta m_o + R_\psi \, \delta I_{sp}}{I_{\psi\psi}} \, \Lambda_\psi \tag{8-14}$$

To determine the influence of these perturbations on final mass, substitute for $\delta\eta$ in Eq. (8-12)

$$\delta m_f = (\lambda_{m\phi} - U\lambda_{m\psi}) \, \delta m_o + (R_\phi - UR_\psi) \, \delta I_{sp} \tag{8-15}$$

where

$$U = \frac{I_{\psi\phi}}{I_{\psi\psi}} \tag{8-16}$$

The coefficients of δm_o and δI_{sp} in Eq. (8-15) give the influence of these perturbations on final mass, assuming the angle of attack is adjusted to meet terminal conditions.

As mentioned before, there are two types of parameters in the group of five to be considered. The first is one in which there is a unit change in mass over a portion (or all) of the trajectory. For this, the column of adjoint variables for mass provide the basic information. This is seen in the coefficient of δm_o in Eq.(8-15). Type-two parameters are those for which the basic sensitivity information must be generated. In the above example, the R array for specific impulse is this information. For the Scout program, the type one (mass only) stage variables are propellant weight and jettison weight. For the remaining three, the R array must be found.

The generalized form of Eq. (8-15) is

$$\delta\Phi = (L_\phi - U\, L_\psi) \begin{pmatrix} \delta m_{prop.} \\ \delta m_{jett.} \end{pmatrix} + (R_\phi - U\, R_\psi) \begin{pmatrix} \delta B.R. \\ \delta I_{sp} \\ \delta D \end{pmatrix} \qquad (8\text{-}17)$$

where (analogous to Eq. 8-16) U is found from

$$U = \left[\int_{t_o}^{t_f} \Lambda_\phi\, \Lambda_\psi^T\, dt + S_\phi\, Y^{-1}\, S_\psi^T \right] \left[\int_{t_o}^{t_f} \Lambda_\psi\, \Lambda_\psi^T\, dt + S_\psi\, Y^{-1}\, S_\psi^T \right]^{-1} \qquad (8\text{-}18)$$

It is to be understood that the elements of the arrays in Eq. (8-18) are evaluated for optimization. They have been discussed and defined in detail in Section 5.6. Evaluation of L and R is discussed on the next page.

EVALUATION OF SENSITIVITY COEFFICIENTS

Jettison Weight

The L vector for jettison weight is formed by recognizing that an increment of one pound in jettison weight implies an increase in weight of the vehicle by one pound from launch to the point at which the stage is jettisoned, after which the vehicle weight returns to the nominal value (for a given payload). Thus,

$$L_{\text{jett.}} = \left[\lambda_{im} \right]_{\text{launch}} - \left[\lambda_{im} \right]_{\text{stage burnout}} \qquad i = 1, JC \qquad (8\text{-}19)$$

Propellant Weight

The L vector for propellant weight again recognizes that an additional pound of propellant is carried from launch to the point of burning. It is assumed that the additional propellant is burned at ignition of the stage of interest and at the level of thrust and specific impulse that nominally exist there. Thus,

$$L_{\text{prop.}} = \left[\lambda_{im} \right]_{\text{launch}} + \left[\lambda \right] \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{r} \\ \dot{\psi} \\ \dot{\lambda} \\ \dot{\tau} \\ \dot{m} \end{bmatrix}_{\text{stage ignition}} \cdot \frac{g\, I_{sp}}{T_V} \qquad (8\text{-}20)$$

Burn Rate

Evaluation of the R array of sensitivities for burn rate reflects the fact that both thrust and mass flow rate are affected by a change in burn rate. The unit of change selected for this parameter is "one percent."

Generalizing from Eq. (8-11b), R is formed from

$$R_{B.R.} = \int_{t_{f_s}}^{t_{o_s}} [\lambda]\,[G]\,dt - [\lambda]\,[\dot{x}] \cdot \frac{gI_{sp}}{T_V}\Bigg|_{\text{stage ignition}} \qquad (8\text{-}21)$$

where the integral is evaluated over each stage and

$$[G] = \frac{\partial[\dot{x}]}{\partial B.R.} \quad \text{where } [\dot{x}] = \begin{bmatrix} F \\ G \\ I \\ H \\ J \\ K \\ L \end{bmatrix} = \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{r} \\ \dot{\psi} \\ \dot{\lambda} \\ \dot{\tau} \\ \dot{m} \end{bmatrix} \quad \text{so that}$$

[G] is evaluated by referring to the equations of motion in Section 4.2.

For a one percent change in burn rate,

$$\frac{\partial F}{\partial B.R.} = \frac{.01\ T_V}{m}\left[C_{21}\cos\theta\sin\chi + C_{22}\cos\theta\cos\chi + C_{23}\sin\theta\right]$$

$$\frac{\partial G}{\partial B.R.} = \frac{.01\ T_V}{mV}\left[C_{31}\cos\theta\sin\chi + C_{32}\cos\theta\cos\chi + C_{33}\sin\theta\right]$$

$$\frac{\partial I}{\partial B.R.} = 0$$

$$\frac{\partial H}{\partial B.R.} = \frac{.01\ T_V}{mV\cos\gamma}\left[C_{11}\cos\theta\sin\chi + C_{12}\cos\theta\cos\chi + C_{13}\sin\theta\right]$$

$$\frac{\partial J}{\partial B.R.} = \frac{\partial K}{\partial B.R.} = 0$$

$$\frac{\partial L}{\partial B.R.} = \frac{-.01\ T_V}{g\ I_{sp}}$$

The second term in Eq.(8-21) reflects a shortened burn time for an increased burn rate.

Specific Impulse

In this program it is assumed that an increase in specific impulse manifests itself as an increase in thrust while maintaining the nominal mass flow rate. The unit of change is again "one percent." Therefore, the R array is formed with the same expressions as for specific impulse, except that $\frac{\partial L}{\partial I_{sp}} = 0$ and there is no second term as in Eq.(8-21), since the nominal burn time is preserved.

Aerodynamic Drag

Aerodynamic drag is treated in essentially the same manner as the above two parameters. Here [G] has only one term, since drag appears only in the $\dot{V}$ equation. The unit of change was established as "square foot of reference area." Thus,

$$[G] = \begin{bmatrix} -\frac{C_D\bar{q}}{m} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

PART II

PROGRAMMING OF THE EQUATIONS

SECTION 9

OPTIMIZATION MODULE

DEFINITION OF SYMBOLS

Most of the symbols used in Section 9 are matrices, many of which are best defined in the text of previous sections. Accordingly, for many of the symbols listed below, a page or equation number is given for reference.

A	Matrix - page 9-2
B	Matrix - page 9-2
C	Matrix - page 9-2
D	Matrix - page 9-2
$d\psi$	Vector of constraint corrections - page 5-24
E	Matrix - page 9-2
F	dV/dt
G	$d\gamma/dt$
H	$d\psi/dt$
JC	Total number of trajectory constraints
K	Vector - page 9-2
P	Vector of time derivatives of trajectory variables
S	Matrix of sensitivity coefficients for adjustable parameters on constraints
Y	Weighting constants for adjustable parameters
$\delta\theta$	Change in pitch angle from previous trajectory
$\delta\tau$	Change in adjustable parameters
Λ	Vectors of sensitivity coefficients of θ and χ - page 5-24
λ	Matrix of adjoint variables - page 5-25

PROGRAMMING OF THE OPTIMIZATION EQUATIONS

The programming of Eqs. (5-29) and (5-30) of Section 5.6 will be described here. The quantities that are to be stored differ depending on whether the computation is in the closed-loop or open-loop mode. To indicate the difference, Eqs. (5-29) and (5-30) are rewritten in the following manner.

Closed Loop

$$\delta\theta = B\, d\psi - D\, dx \qquad (9\text{-}1)$$

$$\delta\tau = C\, d\psi - E\, dx \qquad (9\text{-}2)$$

where $B = \Lambda^T A$

$$A = \left[\int_{t_f}^{t_i} \Lambda \Lambda^T dt - S Y^{-1} S^T \right]^{-1}$$

$$D = B\lambda$$

$$C = -Y^{-1} S^T A$$

$$E = C\lambda$$

Open Loop

$$\delta\theta = \Lambda^T K \qquad (9\text{-}3)$$

$$\delta\tau = -Y^{-1} S^T K \qquad (9\text{-}4)$$

where $K = A\,[d\psi - \lambda\, dx]$

Note that K is evaluated only once, at the point that the computation switches from closed to open loop operation.

Terminology

A forward run goes forward in time. With the exception of the first forward run, all forward runs are either guidance or optimization runs. On a forward guidance run, one attempts only to meet terminal conditions. On a forward optimization run, one attempts to obtain payoff improvement along with meeting terminal conditions. The first forward run (the initial trajectory) uses a control program that is read in.

A backward run goes backward in time. All backward runs are either guidance or optimization runs depending on whether the following forward run is to be a guidance or optimization run. The equations for the quantities which are to be stored on the backward run are different for guidance and optimization.

A count is made of the number of integration steps. KPOINL is the number of the integration step at which the computation switches from the closed-loop to the open-loop mode of operation. KPOINL is always 20.

The number of constraints, including the payoff parameter, is given by JC. IC is one (1) for optimization runs and two (2) for guidance runs.

Computations to be made on Backward Runs

$$\Lambda_{\theta_j} = - \left[\lambda_{V_j} \frac{\partial F}{\partial \theta} + \lambda_{\gamma_j} \frac{\partial G}{\partial \theta} + \lambda_{\psi_j} \frac{\partial H}{\partial \theta} \right] \qquad (9\text{-}5)$$

$$j=1,\ JC$$

$$\Lambda_{\chi_j} = - \left[\lambda_{V_j} \frac{\partial F}{\partial \chi} + \lambda_{\gamma_j} \frac{\partial G}{\partial \chi} + \lambda_{\psi_j} \frac{\partial H}{\partial \chi} \right] \qquad (9\text{-}6)$$

The Λ_θ's are stored from the final point to KPOINL. They are placed in the DD storage area. The index ML1 is the row location in storage for the Λ matrix being stored at the current integration step. ML1 is initialized to the number of integration steps in the trajectory and is reduced by one at each integration step until KPOINL is reached. Then ML1 is set equal to 1. During the closed-loop portion of the trajectory, the current Λ matrix is stored in the first row of DD. Λ_X is used only in evaluating the S matrix.

$$I_{ij} = \int_{t_f}^{t} \left[\Lambda_{\theta_i} \Lambda_{\theta_j}\right] dt \qquad (9\text{-}7)$$

for all i, j from 1 to JC for $i \leq j$

$$J_{ij} = \sum \frac{S_{i_k} S_{jk}}{Y_K} \qquad i, j = 1....JC \text{ for } i \leq j \qquad (9\text{-}8)$$

J_{ij} is computed only for the values of k corresponding to adjustable parameters that are being optimized. Discussion of the evaluation of the S matrix is given in the following section.

I_{ij} and J_{ij} are added term by term. The matrix $(I_{ij} + J_{ij})$ is then inverted. On backward guidance runs, the first row and column of $(I_{ij} + J_{ij})$ are not included in the matrix to be inverted. On backward optimization runs, the entire matrix is inverted.

Let the inverted matrix be represented by A. Store A when KPOINC = KPOINL. It is known as A92 at that point.

$$B_{1i} = \sum_{j = IC}^{JC} \Lambda_{\theta j} A_{ij} \qquad i = IC, \ldots\ldots\ldots JC \qquad (9\text{-}9)$$

$$B_{12} = \sum_{j = 2}^{JC} B_{1j} \, d\psi_j \qquad (9\text{-}10)$$

Store B_{11} in the first column of B matrix and B_{12} in the second column.

One row of the B matrix is stored at each integration point. The index MB1 gives the row location of the matrix being stored at the current time. The B's are stored only during the closed-loop part of the trajectory.

$$C_{ki} = - \sum_{j = IC}^{JC} \frac{S_{jk} A_{ij}}{y_k} \qquad i = IC, \ldots\ldots JC \qquad (9\text{-}11)$$

The C's are computed and stored, at the launch time point only, for the launch azimuth and time of day adjustments.

$$D = B\lambda \qquad (9\text{-}12)$$

D is a 1 x 7 matrix. It is stored in the DD storage area. The index MB1 gives the row location of the D matrix being stored at the current time. D is stored only during the closed-loop part of the trajectory.

$$E = C\lambda \qquad (9\text{-}13)$$

E is not needed since dx is zero at launch.

Computation to be Made on All Forward Runs After the First

For KPOINC < KPOINL (closed-loop)

$$\delta\theta = B_{11} \, d\psi_1 + B_{12} - \sum_{\ell=1}^{7} D_{1\ell} \, \delta x_\ell \qquad (9\text{-}14)$$

$$\delta\tau_k = \sum_{j = IC}^{JC} C_{kj} \, d\psi_j \quad \text{for} \quad k = 6,7 \qquad (9\text{-}15)$$

For KPOINC = KPOINL (transition to open-loop)

Compute $K = A \, [d\psi - \lambda\delta x]$

where A and λ have been stored at this point on the last backward run.

$$\delta\tau_k = \frac{1}{y_k} \sum_{j = IC}^{JC} K_j S_{jk} \qquad k = 1,2,3,4,5 \tag{9-16}$$

For KPOINC ≥ KPOINL (open-loop)

$$\delta\theta = \sum_{j = IC}^{JC} K_j \Lambda_{\theta j} \tag{9-17}$$

EVALUATION OF S MATRIX FOR ADJUSTABLE PARAMETERS

Terms in the S_{ik} matrix are the sensitivities of the ith constraint to the kth adjustable parameter. The correspondence between k and the eleven adjustable parameters is as follows.

k	Adjustable Parameter
1	Stage 4 pitch angle, θ (when spun)
2	Stage 4 yaw angle, χ
3	Length of coast after stage 3
4	Length of coast after stage 2
5	Length of coast after stage 1
6	Launch azimuth
7	Launch time of day
8	Stage 5 pitch angle, θ (when spun)
9	Stage 5 yaw angle, χ
10	Length of coast after stage 4
11	Length of coast after stage 5

The methods of obtaining the eleven columns in the S matrix are described in this section.

Stages 4 and 5 θ and χ

When a stage is spin stabilized, θ is constant over that stage. Further, χ is always constant over each stage. Recall that Λ is the sensitivity of the terminal constraints to a change in the control angles per unit time, from which

$$d\psi = \delta\theta \int_{t_f}^{t_i} \Lambda_\theta \, dt + \delta\chi \int_{t_f}^{t_i} \Lambda_\chi \, dt \quad .$$

It is therefore apparent that

$$S_{i1} = \int_{t_{4_f}}^{t_{4i}} \Lambda_\theta \, dt, \quad S_{i2} = \int_{t_{4_f}}^{t_{4i}} \Lambda_\chi \, dt$$

and

$$S_{i8} = \int_{t_{5_f}}^{t_{5i}} \Lambda_\theta \, dt, \quad S_{i9} = \int_{t_{5_f}}^{t_{5i}} \Lambda_\chi \, dt$$

These integrals are evaluated over the respective stages on backward integrations and then stored in the S array. When stages 4 and 5 are spun (together) the integrals are taken over both stages and stored as stage-four adjustable parameters to be applied to both stages.

Coast Durations

The S terms for the coasts are sensitivities of the constraints to extensions of the coast times. This is obtained by the matrix product

$$S_{ik} = \lambda P$$

formed at the upper end of the coast, and where

$$P = \begin{bmatrix} \dot{V} \\ \dot{\gamma} \\ \dot{r} \\ \dot{\psi} \\ \dot{\lambda} \\ \dot{\tau} \\ \dot{m} \end{bmatrix}$$

and λ is the array of adjoint variables at that time.

Launch Azimuth

The launch azimuth is significant through the orientation of the inertial platform, as established at launch. The basis for evaluation of S_{16} is the fact that a change in azimuth direction of the platform axes produces the same effect on the thrust direction as does a change in yaw angle over the entire boost. Thus, the S_{16} is formed by integration of Λ_X over all stages.

Launch Time of Day

This adjustable parameter is of use only when the longitude of the ascending node is a specified constraint, since the node line is measured from the vernal equinox. The S_{17} is evaluated by

$$S_{17} = \lambda P$$

as for the coasts. Here, however, the λ's at launch are used along with the P array of derivatives before launch. At that point $\dot{\tau}$ is the Earth's rotation rate, and all the remaining derivatives are zero.

INTERMEDIATE TRAJECTORY CONSTRAINTS

Constraints on the trajectory at stage points are handled nearly the same as are terminal constraints. They are always applied at the same time (relative to stage ignition). Thus, on backward trajectories, the adjoint variables for each constraint are initialized at the constraint time by setting them equal to the partial derivative of the constraint with respect to each trajectory variable. On forward trajectories, the achieved value of each constraint is stored for comparison with the desired value.

OPTIMIZATION SUCCESS CRITERION

The method used to determine whether a forward optimization trajectory is successful takes into account the effect of errors in the trajectory constraints on the payoff parameter in judging whether or not a net increase in payoff has been accomplished.

In order to assess the relationship between payoff and errors in the constraints one starts by representing the changes in the control variable and adjustable parameters required to meet terminal conditions with no constraint on payoff. These are given by:

$$\delta\theta = \Lambda^T \left[\int_{t_f}^{t_1} \Lambda \Lambda^T dt - S Y^{-1} S^T \right]^{-1} \left[d\psi - \lambda \delta x \right] \qquad (9\text{-}18)$$

$$\delta\tau = -Y^{-1} S^T \left[\int_{t_f}^{t_1} \Lambda \Lambda^T dt - S Y^{-1} S^T \right]^{-1} \left[d\psi - \lambda \delta x \right] \qquad (9\text{-}19)$$

The equations are identical to the basic control equations with the exception that the first row and/or column of all matrices (for the payoff) are left out.

The change in payoff (δm_f) that will be produced by given changes in initial conditions, control variables and adjustable parameters is

$$\delta m_f = \lambda_1 \delta x + \int_{t_f}^{t_1} \Lambda_1 \delta\theta \, dt + S_1 \delta\tau \qquad (9\text{-}20)$$

where λ_1 is the first row of the λ matrix

Λ_1 is the first row of the Λ matrix

S_1 is the first row of the S matrix

$\delta\theta$ and $\delta\tau$, required to meet terminal conditions, are given in Eqs. (9-18) and (9-19). Substitute these values into Eq. (9-20) with zero change in initial conditions (δx).

$$\delta m_f = \left[\int_{t_f}^{t_i} \Lambda_1 \Lambda^T dt - S_1 Y^{-1} S^T \right] [A] \left[d\psi \right] \qquad (9\text{-}21)$$

where $|A|$ is the inverted matrix appearing in the brackets in Eqs. (9-18) and (9-19). Eq. (9-21) gives the change in payoff associated with adjusting $\delta\theta$ and $\delta\tau$ in order to remove errors in the trajectory constraints.

This calculation is made in the MEQ subroutine at the end of every backward trajectory and is used to correct the magnitude of the payoff parameter indicated on the previous forward trajectory. Furthermore, the elements of the row matrix

$$\left[\int_{t_f}^{t_i} \Lambda_1 \Lambda^T dt - S_1 Y^{-1} S^T \right] [A] \qquad (9\text{-}22)$$

from Eq. (9-21) are partial derivatives of payoff with respect to changes in each constraint. These are printed out on each backward trajectory. They are also stored for use on succeeding forward optimization trajectories in order to judge whether or not an iteration should be judged a success on the basis of payoff improvement. That is, only if the corrected payoff on the present iteration exceeds the corrected payoff on the previous iteration will the run be judged a success.

OPTIMIZATION FOR MAXIMUM PAYLOAD

Optimization with payload as the payoff parameter, compared to velocity or altitude, has the added complication of a varying mass history (payload) between iterations. Further, it is necessary to recognize that the final weight (payload) is determined solely by the payload chosen at the beginning of the trajectory iteration. This is true since each trajectory terminates at burnout of the final stage or the following coast. The θ program or coast durations will not affect the terminal mass; however, they and the payload together affect the terminal trajectory variables. Thus, it is natural to consider the payload as a control variable since it affects both the payoff (one to one) and the trajectory constraints. Further, since the payload assumes one value for an entire iteration, it is the equivalent of an adjustable parameter.

As with other adjustable parameters, it is necessary to develop partial derivatives of the payoff and constraints with respect to a change in the parameter (payload or equivalently, launch weight). These are simply the column of adjoint variables on mass, taken at the launch point. In the TRAJ subroutine at the end of stage 1 on each backward trajectory, the λ_m are multiplied by their transpose and a weighting factor and added directly into the FJ matrix which is otherwise comprised of $SY^{-1}S^T$ terms of Eq. (9-8). By so doing we have added launch weight as a control parameter.

The implications of this move are simply that now in computing the payoff correction for constraint errors, discussed in the preceding section, the equations have these additional terms included. On guidance iterations the change in payload between iterations is set equal to the payoff correction which has been computed. Similarly, on optimization iterations the change in payload is set equal to the payoff correction plus the attempted payoff improvement.

ALLOCATION OF VARIABLES OF INTEGRATION

There is a maximum of 294 variables of integration which are treated at various times by the program. In order to minimize the computing time and core storage requirements for handling this large number of variables, the following programming techniques were employed.

First, all variables which are currently being integrated are grouped in the single-dimensioned X( ) array, with their corresponding time-derivatives in the D1( ) array. This enables handling of all variables with "DO" statements.

Secondly, the variables occur in groups whose size depends on the number of constraints selected. Also, some groups are never needed at the same time as others. Thus, by allowing each group to be relocatable in the X( ) array, variables can be stacked and can use the same positions as others do at other times. A definition of the storage allocation within the X( ) array is given here.

Let I be the subscript of X( ) and JC be the number of constraints. Then,

I = 1, 6 for the six trajectory variables

I = 7, ILAST for the λ adjoint variables

where ILAST = 7 * JC + 6 on all backward trajectories.

I = ILAST + 1, KK13 for $\Lambda_\theta * \Lambda_\theta^T$

where KK13 = 8 * JC + (JC * (JC - 1))/2 + 6 on backward guidance and optimizations during all stages except spin stabilized stages.

I = ILAST + 1, ILAST + JC for Λ_θ on backward guidance and optimizations only during spin stabilized stages.

I = ILAST + 1, ILAST + 2 + 3 * JC on backward linearization.

I = KK13 + 1, KK13 + JC for Λ_{χ} on backward guidance and optimization, over stages 4 and 5 for χ adjustments and over all time for launch azimuth adjustment.

I = 202, 201 + 3 * JC for exchange ratios only on backward guidance before final guidance.

In the program, I ranges from 1 to KK12, which is equal to the total number of variables being integrated on the present trajectory. KK12 is 6 on all forward and a maximum of 240 on backward trajectories.

SECTION 10

PROGRAMMING OF THE PITCH-PROGRAM LINEARIZATION EQUATIONS

PROGRAMMING OF THE LINEARIZATION EQUATIONS

INTRODUCTION

The implementation of the linearization of an ascent tilt program adds to the Scout program the option of converting the optimized θ history into a pitch program of up to fifteen linear segments. A satisfactory linearized pitch program is one in which the trajectory constraints and certain pitch rate limitations are also satisfied. These rate limitations include a maximum pitch rate, minimum pitch rate, only one positive rate, and a minimum change in rate of two adjacent segments. There are several features used in establishing a satisfactory linearized pitch program. For example, it is desirable to have a small difference between the linearized program and the optimized θ history over the maximum pitch rate area following vertical lift-off. It is also desirable to allow the largest change in θ over the coast segments. These two ends are aided by adjusting the (η) weighting function values (page 6-4) appropriately for these sections. The availability of fixing certain values of θ in the pitch program is used to good advantage when attempting to meet the pitch rate limitations.

GENERAL PROCESSING

The overall incorporation of the linearization processing into the optimization program is shown by the flow chart on page 20-4. After the Scout trajectory optimization is accomplished, the linearization processing begins. The first step is a backward run, evaluating the integrals needed for the linearization matrices. A linearized pitch program is then computed and used in running a forward trajectory. When terminal constraints are not satisfied, or if there are violations of pitch rate limitations in the pitch program, a satisfactory linearization has not been achieved. Further attempts at linearization are made until one is satisfactory or until a maximum number of attempts has been made. If this occurs, a transfer is made which ends all linearization processing. When a satisfactory linearization is accomplished, the option of running 6D trajectories is tested. When not indicated, a transfer is made which ends linearization processing. When the option is indicated, all succeeding attempts at linearization use the current linearized θ program as a command θ, and 6D computations are included in the trajectory. The attempts at linearization continue until a satisfactory program is again established or until the maximum number of attempts has been made.

INPUT DATA

There are several data inputs required for achieving a linearized pitch program. The basic option for linear processing is selected by inputting IP(7)=1 in data block 4. Data block 18 is solely devoted to linearization and must be input as a complete unit every time it is read in initially or changed in subsequent cases. The first word indicates the total number of linear segments to be used, allowing one segment each for coasts 5 and 6. The approximate times at which the pitch rate changes are then listed sequentially and are coded so as to indicate the powered stage number as well as the time from the beginning of the stage.

The values of the weighting functions (η from page 6-4) are initialized in the MAIN routine by setting values for all segments to one. Data block 26 allows for input of other values as discussed on page 22- 11.

The permitted deviations in terminal constraints used during linearization are input in data block 29, beginning in DA14(14). The specifications of the pitch rate limitations are input in data block 22. The maximum number of forward linearization runs permitted in attempting to satisfy terminal constraints and pitch rate limitations, including those with 6D computations added, is input as DA13(4) in data block 28.

DETAILED FLOW OF PROCESSING

Initial Call of Linear

The first step in the linearization process occurs whenever data is read into data block 18. The LINEAR routine is called immediately to establish approximate values of t_j, times from beginning of stage at which pitch rate changes. During both backward and forward linearization runs, an index is needed to indicate the current linear segment. The stage code digits in the input data are used in establishing KK14 and KK15 which serve this indexing purpose.

The trajectory optimization is then conducted through the final guidance run. When the linearization option is not chosen by input, the flow continues to statement number 10 in the MAIN program where a new case or further optional computations are begun.

Backward Linearization Trajectory

When the linearization option is chosen, the program now initializes and computes variables needed for all subsequent linearization runs. As the number of forward linearization runs to be made is limited by input, a counter of these runs is now initialized. The adjustable parameters are held fixed during linearization. Therefore, the flags indicating the use of these parameters are set to zero.

The LINEAR routine now adjusts the TLIN array (t_j) times so that, during the integration, pitch rates will change at integration step times. The option of fixing θ at any break point involves establishing the array THETAX as θ_f and setting flags in IFIXED indicating which θ_f are fixed. At this point in the processing, both of these vectors are initialized to zero. As the θ at the end of lift-off must be 90 degrees, the program uses the option for fixing θ for this purpose. If a maximum pitch rate was encountered during the optimization, t_2 is set to the stored time at the end of the maximum pitch rate period. A computation of the time duration of pitch rate segments Δt_j is made using the adjusted time points t_j of TLIN. The segments corresponding to coast stages use a Δt_j equal to the duration of coasts 5 and 6 established on the final guided run. Matrices A and B from page 6-5 are computed, and the flow returns to the MAIN routine.

The linearization runs are processed in the main body of the program basically as guidance trajectories. The MAIN routine sets all variables necessary to perform a backward linearization run accordingly. The backward trajectory is begun and processed through powered stage 4 and coast stage 6. (Linearization is performed in stages 1 - 3.)

Processing in TRAJ and DEQ

At the entry to stage 3 special processing begins in both the TRAJ and DEQ routines. In the TRAJ routine, before beginning a backward integration over a new linear segment, values of the linear segment number and the times at beginning and end of the segment are established. Call these j, t_j, and t_{j+1}. In the DEQ routine during integration over the segment, the integrals used for the linearization process are computed. These are used in establishing the matrices D, E, L and M on page 6-5. In programming these matrices, the two terms in M are treated separately, becoming

$$M_{A_{ij}} = \frac{1}{\Delta t_j} \int_{t_{j+1}}^{t_j} (t - t_j) \Lambda_{\theta_i} dt \qquad \begin{array}{l} i = 1, JC \\ j = 1, K \end{array}$$

$$M_{B_{ij}} = \frac{1}{\Delta t_j} \int_{t_{j+1}}^{t_j} (t_{j+1} - t) \Lambda_{\theta_1} dt$$

$$= \int_{t_j}^{t_j} \left(1 - \frac{t - t_j}{\Delta t_j}\right) \Lambda_{\theta_i} dt \qquad \begin{array}{l} i = 1, JC \\ j = 1, K \end{array}$$

so that

$$M_{i1} = M_{B_{i1}} \qquad i = 1, JC$$

$$M_{ij} = M_{A_{ij-1}} + M_{B_{ij}} \qquad \begin{array}{l} i = 1, JC \\ j = 2, K \end{array}$$

$$M_{iK+1} = M_{A_{iK}} \qquad i = 1, JC$$

The integrals programmed are based on the following equations.

For matrix D

$$D1(n) = \frac{2\theta}{\Delta t_j} (t_{j+1} - t)$$

For matrix E

$$D1(n) = \frac{2\theta}{\Delta t_j} (t - t_j)$$

For each constraint, column elements of L matrix

$$D1(n) = \theta \Lambda_{\theta_i} \qquad i = 1, JC$$

For each constraint, row element of M matrix

$$Dl(n) = \Lambda_{\theta_i} \frac{(t - t_j)}{\Delta t_j} \qquad i = 1, JC$$

$$Dl(n) = \Lambda_{\theta_i} - \Lambda_{\theta_i} \frac{(t - t_j)}{\Delta t_j} \qquad i = 1, JC$$

As the end of integration over each segment is reached, the TRAJ routine stores the terms in X related to matrices D, E and M into the proper linear segment elements. These X terms are then zeroed, initializing them for integration over the next segment.

Computation of Linearized θ

When the backward run is completed, MAIN routine calls the LINEAR routine, with an immediate transfer to statement 2000 in LINEAR. The region between statements 2000 and 3018 is used once per trajectory. The region beyond 3018 may be used several times, as repeated attempts are made to satisfy the various pitch rate limitations.

In the 2000 region, four operations are performed. The first is a test for constraint of $\alpha = 0$ at stage 2 ignition. If it is selected, the θ_f at that point is established from a value stored on the previous trajectory, and the η for that segment is changed from 1. to .01 in order to minimize propagation of effects from fixing θ at that point. The second operation reflects this last consideration. As various θ_f values are later established in the process of satisfying the rate limitations, the η for adjacent segments are set to .01. They are, however, returned to their nominal value for each trajectory iteration. Thus, the nominal values are stored at 2110. Next, the L and M matrices are given values. During linearization, the constraint on the payoff parameter, orbit inclination and longitude of the node are not imposed. For this, the respective elements of [L] and [M] are set to 0.

(The resultant matrix $(M \bar{P}^{-1} M^T)$ will have the diagonal element of the respective row and column set to one before inversion for the same objective.) Finally, the number of times a new linearized θ program will be computed on the basis of the previous backward linearization run is limited to five. When a pitch program violates rate limitations, a correction is attempted using an additional value of θ_f and, in some instances, changing values of η. However, when five such attempts do not produce a linearized θ program satisfying limitations, the last pitch program is used on the following forward run. Even if this run satisfies terminal constraints, further linearization processing must be done to eliminate pitch rate violations. The counter for the number of θ programs computed, KOUNTV, is initialized to 0 in this section.

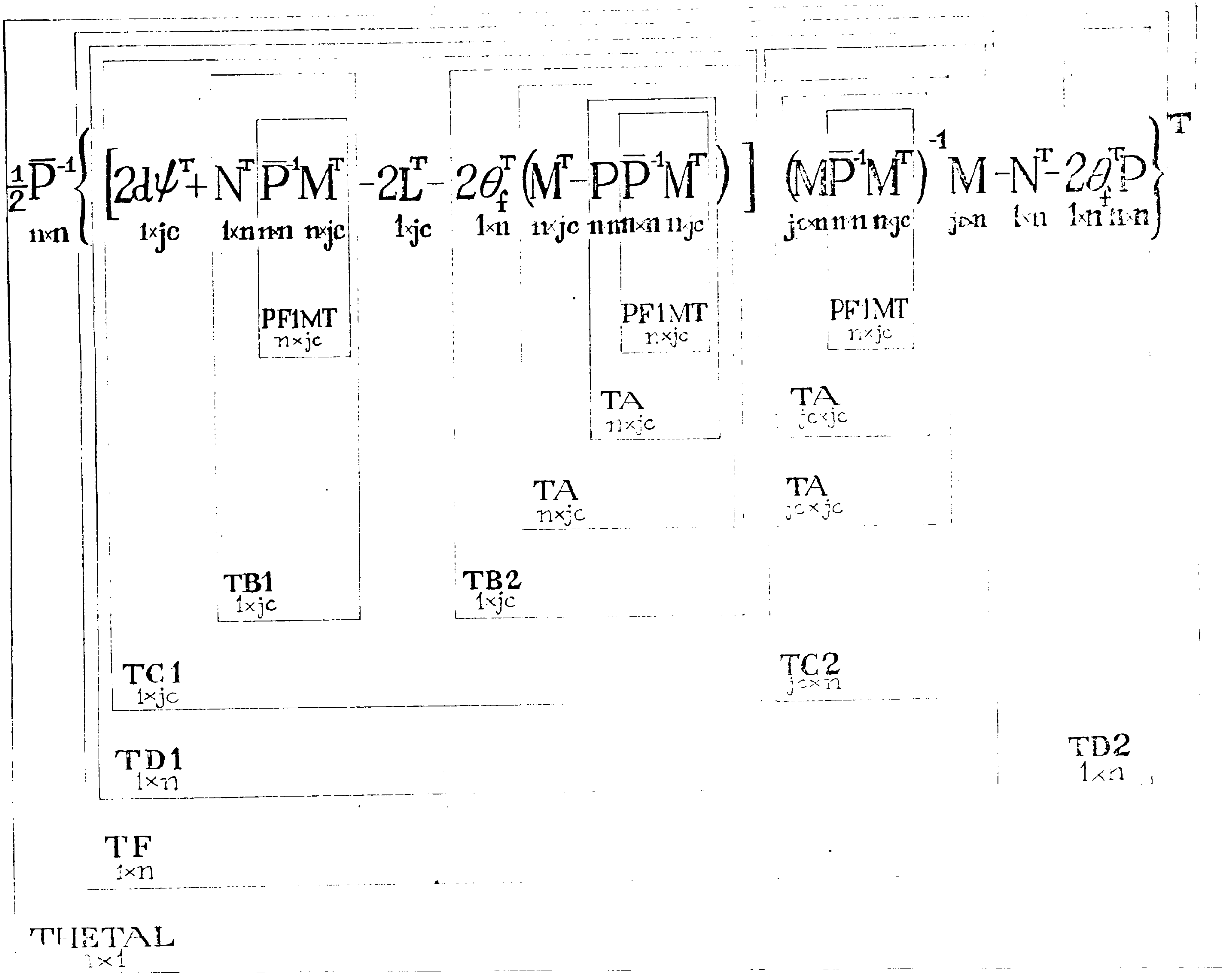
The computations following statement number 3018 involve functions which are affected by adjustments in θ_f and η. Therefore, after additional values of θ_f are made in attempting to meet pitch rate limitations, the flow returns to this point and the following computations are repeated.

The matrix P on page 6-7, $\bar{P}^{-1}$ on page 6-9, and the computations resulting in the final linearized θ program are then processed. The diagram on page 10-8 relates the programmed computational matrices to equation (6-22) on page 6-10. The final array THETAL has the fixed values of θ from θ_f array and computed values of θ from $\bar{\theta}$ computation.

The next step is determining if the linearized θ program satisfies the limitations on pitch rate. These limitations include a maximum pitch rate; minimum pitch rate; minimum change in two adjacent pitch rates; and only one positive rate over all segments, and over the coast between stages 3 and 4. It is assumed that a maximum pitch rate occurs only directly after initial lift-off. The process of matching the duration of the first linear segment to the optimum time at maximum pitch rate, discussed on page 10-4, is sufficient

PROGRAMMED COMPUTATION OF EQUATION (6-22)

$$\underset{n\times 1}{\bar{\theta}} = \frac{1}{2}\underset{n\times n}{\bar{P}^{-1}}\left\{\left[\underset{1\times jc}{2d\psi^{T}} + \underset{1\times n}{N^{T}}\underset{n\times n}{\bar{P}^{-1}}\underset{n\times jc}{M^{T}} - \underset{1\times jc}{2L^{T}} - \underset{1\times n}{2\theta_f^{T}}\left(\underset{n\times jc}{M^{T}} - \underset{n\times n}{P}\underset{n\times n}{\bar{P}^{-1}}\underset{n\times jc}{M^{T}}\right)\right]\left(\underset{jc\times n}{M}\underset{n\times n}{\bar{P}^{-1}}\underset{n\times jc}{M^{T}}\right)^{-1}\underset{jc\times n}{M} - \underset{1\times n}{N^{T}} - \underset{1\times n}{2\theta_f^{T}}\underset{n\times n}{P}\right\}^{T}$$



for treatment of the maximum rate limitation.

Positive Rate Violations

Positive pitch rate limitation is violated when more than one occurs, including the rate over the coast between stages 3 and 4. When a violation occurs, when more attempts are allowed, and when a coast segment has a positive pitch rate, then the θ at the beginning of this coast is set to a value which satisfies the minimum (-) pitch rate over that segment. The η for the preceding segment is set to .01, and the flow then transfers to statement number 3018.

When the positive rate violation occurs and when only powered stage pitch rate segments are in violation, the θ_f at the end of the second positive rate segment is set to a value satisfying the minimum (-) pitch rate over the segment. The η for the segments adjacent to the fixed θ are set to .01, and the flow then transfers to 3018.

When violation occurs and there have been five computations of a linearized θ program, none of which satisfied rate limitations, the flow transfer to 7900 in the program, ending further attempts to compile a satisfactory program.

Minimum Rate Violations

When the positive rate limit is satisfied, flow proceeds to a test for minimum pitch rate violations. The limit is violated when the absolute value of any pitch rate is smaller than the input minimum value. When a violation occurs and more attempts are permissible, the θ_f at the beginning of the first segment in violation is set to a value satisfying the limit over that segment. Flow then transfers to 3018.

Again, when a violation occurs and there have been five computations of a linearized θ program, the flow transfers to 7900.

Minimum Change in Rate

The next level of processing is for the limitation on minimum change in two adjacent pitch rates. This limit is violated when the absolute value of the change in rates of two adjacent segments is smaller than the input value. When violation occurs but there have been five computations of a linearized θ program, the flow transfers to 7900. When a violation occurs and more attempts are permissible, the three θ_f quantities over the two rate segments in violation are set to produce a constant rate over the two segments. This, in effect, eliminates a segment. Flow then transfers to 3018.

When no violation occurs, a linearized θ program has been established which satisfies all limitations on pitch rates. Before leaving the LINEAR routine, the coast stage 7 pitch rate is tested for minimum rate violation and minimum change in rate violation. A message is printed to indicate such violations. The θ_f established in attempting to meet rate limitations is eliminated and the stored weighting function is reestablished.

Forward Linearization Trajectory

On returning to MAIN routine, initialization for a forward linearization run is made. Again this run is basically a forward guidance run, but the newly computed linearized θ program is used in computing the trajectory. The special linearization steps required include, in TRAJ, setting the change to be made in adjustable parameters DTAU to zero. PCAL, on a forward linearization run, computes current θ for stages 1 through 3 on the basis of the pitch rate over the current segment. These pitch rate segment variables are established in TRAJ at the start of each linear segment.

When a forward linearization run is completed, the achieved values of the terminal constraints and their deviations from the desired values are computed in MAIN.

When terminal constraints are not satisfied and more attempts at linearization are permissible, a new attempt is made. When terminal constraints are not satisfied and no more attempts are allowed, flow transfers to statement number 10 in the MAIN routine, ending all linearization processing.

When terminal constraints are satisfied but there exists a violation of pitch rate limitations in the current linearized θ program, a new attempt at linearization is made. When no violation of pitch rate limitations exist and the option of computing a 6D trajectory after successful linearization is not indicated, flow transfers to statement number 10.

When terminal constraints are satisfied, no rate violations exist and the 6D option is indicated, all succeeding backward and forward linearization runs are performed using the linearized θ program as a command θ and adding 6D considerations to trajectory computations. This 6D linearization process is continued until terminal constraints and pitch rate limitations are satisfied or until no more attempts are allowed.

SECTION 11

PROGRAMMING OF BODY DYNAMICS FOR NOMINAL TRAJECTORY

PROGRAMMING OF BODY DYNAMIC EQUATIONS

Documentation of the programming performed for the body dynamics simulation is presented in this section. It uses the solution for the modified thrust attitude, shown in equations (7-6) and (7-7), in computation of the resultant steady-state forces acting on the point mass. All of the equations presented in this section are programmed in the subroutine SIXD.

The unit pitching moments required in equations (7-6) and (7-7) are computed from input data as functions of Mach number and time, as follows:

$$M_\alpha = \left[C_{M_{\alpha(FS)}} \cdot \ell_{ref} - C_{N_\alpha} (X_{FS} - X_{CG}) \right] \bar{q}S$$

$$M_\delta = \left[-C_{M_{\delta(FS)}} \cdot \ell_{ref} + C_{N_\delta} (X_{FS} - X_{CG}) \right] \bar{q}S$$

$$+ K_{TV_\delta} \cdot T_V (X_\delta - X_{CG})$$

where

C_{M_α} , C_{M_δ} = moment coefficients about a fixed body station

C_{N_α} , C_{N_δ} = normal force coefficients

X_{FS} = body station for moment coefficients

X_{CG} = vehicle center of gravity body station

$\bar{q}$ = dynamic pressure

ℓ_{ref}, S = reference length and area

$K_{TV\delta}$ = jet vane force coefficient

T_V = vacuum thrust

X_δ = body station of jet vanes

Now compute α and β, the in-plane and out-of-plane components of angle of attack, respectively, by calling subroutine CONTRL. When α is positive, the aerodynamic force is along $+\bar{m}_z$ axis (see diagram, page 7-3). When β is positive, the aerodynamic force is along $+\bar{m}_x$ axis.

Finally we compute the control and aerodynamic forces on the vehicle.

$$F_{CNLX} = M_\alpha \cdot \beta / (X_\delta - X_{CG})$$

$$F_{CNLZ} = M_\alpha \cdot \alpha / (X_\delta - X_{CG})$$

$$F_\alpha = \bar{q}\, SC_{N_\alpha}$$

So that the total forces along the missile axes are

$$F_{m_x} = F_{CNLX} + F_\alpha \cdot \beta$$

$$F_{m_z} = F_{CNLZ} + F_\alpha \cdot \alpha$$

The components of these two forces along the P-system axes (see diagram, page 7-3) are then found.

$$F_{jx} = F_{m_x} \cos \chi - F_{m_z} \sin \theta \sin \chi$$

$$F_{jy} = -F_{m_x} \sin \chi - F_{m_z} \sin \theta \cos \chi$$

$$F_{jz} = F_{m_z} \cos \theta$$

Knowing the forces in the P-system, we transform them to the L-system (Fig. 4-1) using the C matrix (page 4-10).

$$\begin{bmatrix} F_{i_x} \\ F_{i_y} \\ F_{i_z} \end{bmatrix} = [C] \begin{bmatrix} F_{j_x} \\ F_{j_y} \\ F_{j_z} \end{bmatrix}$$

Then, F_{i_x} is divided by $m\ V \cos \gamma$ and added to the equation for $\dot{\psi}$ in the DEQ subroutine. Similarly, F_{i_y} and F_{i_z} are divided by m and mV, and added to $\dot{V}$ and $\dot{\gamma}$, respectively.

These calculations are performed only during stage 1 and the following coast, and only during forward 6D trajectories. In all other stages, it is assumed that the vehicle follows the command pitch program.

SECTION 12

PROGRAMMING OF THE EXCHANGE RATIO EQUATIONS

PROGRAMMING OF EXCHANGE RATIO EQUATIONS

As was evident from the derivations shown in Section 8, computation of the exchange ratios makes extensive use of quantities which are already available on backward guidance and optimization trajectories. Only the calculation of matrix R and Eq. (8-17) are new operations.

The evaluation of [R] requires numerical integration of 3 * JC quantities over each stage. X(202) through X(240) are used for this integration. Thus, at the start of each stage, these locations in X( ) are zeroed, and at the end of each stage the values of X( ) are stored in [R]. During each stage [λ] [G] from Eq. (8-21) is formed in the DEQ subroutine and integrated simultaneously with all the other variables of integration.

At the beginning and end of each stage, the proper elements of the [L] matrix are stored in subroutines INSTØP and MEQ by simply using the values of the adjoint variables at those times.

Finally, at the end of the trajectory, the exchange ratios are calculated from Eq. (8-17) in subroutine MEQ. The quantity U in Eq. (8-18) is available at that time, since it is used in the process of correcting the payoff parameter to a condition of zero constraint error. This has been described in Section 9.6.

The exchange ratios are output from the MEQ subroutine after the units have been corrected, corresponding to the specified payoff parameter.

SECTION 13

AERODYNAMIC HEATING CONSTRAINT

AERODYNAMIC HEATING CONSTRAINT

By selection of data option 2, the user of the Scout program can impose a constraint on the time-integral of aerodynamic heating rate over the duration of the trajectory. The formulation is that of an inequality constraint, in which the optimization is started without the constraint being applied. If during the iterative solution, the heating integral exceeds the input allowable maximum, the heating constraint is added to the list of constraint parameters automatically and then retained for the remainder of the case.

The equation which is used for the heating rate is[*]

$$\dot{Q} = \frac{20,800}{\sqrt{R}} \sqrt{\frac{\rho}{\rho_{S.L.}}} \left(\frac{V}{10,000} \right)^{3.25}$$

where $\dot{Q}$ = heating rate, Btu/ft^2/sec

R = nose radius of curvature, ft

ρ = atmospheric density, slug/ft^3

$\rho_{S.L.}$ = sea level density

V = aerodynamic velocity

If the constraint is activated during the optimization, it is handled in the program nearly the same as the other constraints. The only difference lies in the set of adjoint equations which are solved for the heating constraint. An additional term is added to the λ_V and λ_r equations since the constraint is a time-integral function of velocity and altitude. The added terms are the partial derivatives of $\dot{Q}$ with respect to V and r, as follows.

---

[*] Feldman, S., "Hypersonic Gas Dynamic Charts of Equilibrium Air," Avco Research Laboratory, Everett, Mass., January 1957.

$$\frac{\partial \dot{Q}}{\partial V} = \frac{3.25 * 20{,}800 \sqrt{\rho}}{\sqrt{R} * (10{,}000)^{3.25} \sqrt{\rho}_{S.L.}} \quad V^{2.25}$$

$$\frac{\partial \dot{Q}}{\partial r} = \frac{.5 * 20{,}800 \; (V)^{3.25}}{\sqrt{R} * (10{,}000)^{3.25} \sqrt{\rho}_{S.L.} \sqrt{\rho}} \quad \cdot \quad \frac{\partial \rho}{\partial h}$$

Data Input

To select this option, set IP(2) = 1. The remaining data input is the same as for any constraint. In data block 8, put constraint code 14 as the last constraint listed, but do not include it in the count of constraints. In data block 17, word 7, input the nose radius. In data blocks 19 and 29, input the maximum allowable heating integral and the allowable deviations, treating heat as the last parameter listed.

SECTION 14

PROCESSING OF INPUT THRUST AND WEIGHT DATA

MODIFICATION OF INPUT THRUST TABLE

The Scout program can accept an input thrust table in which the thrust is tabulated at arbitrary time points and varies with time in any manner other than a discontinuity. In order that computation accuracy be accomplished in the use of such a thrust history when the integration package uses a fixed time step, it is necessary to modify the thrust table such that the time entries are the same as the integration times. Further, typical solid propellant thrust histories have zero thrust at time-zero and then rise sharply into an initial pulse. This representation is very difficult to use accurately in numerical solution of the equations of motion. This is especially true for stage one at lift-off. Thus, a modified thrust table is again suggested. The technique which is used to modify the input thrust data while maintaining an accurate simulation will now be discussed.

The most important similarity parameter in modification of thrust is the ideal velocity history for each stage. Therefore, the following sequence of operations is performed after the thrust is input but before the trajectory integration starts. Integrate the input thrust/weight history for each stage, using fourth order Runge-Kutta with variable time steps to match the input thrust entries. This produces an ideal velocity increment for each stage. Call it A. Next, evaluate the integration times which will be used in the trajectory computation. Eliminate the initial pulse by solving for a new thrust at time-zero which will produce the same thrust-impulse over the first integration step as in the input data. Next, from the input thrust, interpolate a new thrust table at the trajectory integration times. Integrate this new thrust/weight history, and call the integral B. It is the ideal velocity of the interpolated table. Finally, to make B=A, multiply all interpolated thrust entries by A/B to produce the final modified thrust history.

This computation is programmed in the REIN subroutine and is performed at the beginning of each case in which any data have been input which would change the ideal velocity of any stage. The ideal velocities and the final modified thrust tables are output whenever this calculation is performed.

The same calculation is made in computation of dispersed trajectories in which the ideal velocity of the vehicle is perturbed. Here, the programming is repeated in the DISPRS subroutine which serves the dispersion computations.

PROVISION FOR ARBITRARY DISTRIBUTION OF DATA POINTS AMONG STAGES

One of the contractual requirements for this computer program, as extracted from the work statement, is as follows.

> "The thrust data inputs shall be written so that the thrust for any stage can be changed without changing the other stages. In order to eliminate the necessity of extensive changes in the input caused by a change in stageweight, the weight time history shall be input as an initial stepweight, consumable stageweight remaining, and payload weight when used as an input.. . . . . Storage will be provided to insure that at least 120 tabulated values of thrust and 120 values of weight as functions of stage time may be accommodated. Provision shall be made to disperse these tabular values in different proportion among the four stages at the option of the user."

In order to be responsive to this requirement and also minimize the necessary core storage, the thrust and weight tables are first read into a temporary storage area. Columns 1-10 of the DD array, which is not used until optimization starts, serve this purpose. Then, the thrust tables for the five stages are stacked in the "THRUST" array, which then must only be dimensioned 2 * 120. The weight tables are handled similarly. Finally, if data for a subsequent case includes a change in one of the thrust tables (or weight), the five original thrust tables are first unstacked into DD, the new table read in over the old one in DD, its (new) length noted, and then all five tables again stacked in THRUST with, in general, a new distribution of data points among the stages.

This logic is all coded in the REIN subroutine.

One small deviation from the wording of the contract is that the jettison weight, rather than step weight, is input for each stage in addition to the table of consumable weight remaining. The payload is represented, for this purpose, as a part of the final stage jettison weight.

SECTION 15

DISPERSED TRAJECTORY CALCULATIONS

DISPERSIONS

The dispersion module controls the dispersion analysis of the optimum trajectory and is called as an input option. Starting from a given payload weight on a nominally performing vehicle, the dispersions in altitude, range, velocity, and flight path angle at stage two, three and four ignition and the final point on the trajectory, as well as the dispersions in the final orbital elements, are defined for each of the following parameters.

1. Variation in thrust, burn time, and consumed weight.
2. Variation in weight time history for any or all stages.
3. Variation of drag and pitching moment coefficients.
4. Thrust misalignment during first stage.
5. Control system deadbands.
6. Tipoff of spin stabilized stage(s).
7. Launch azimuth and attitude error.
8. Variation of wind velocity and direction as a function of altitude.

Output is provided so that the dispersions can be both individually examined and also grouped as three sigma variations of each condition (excluding orbital elements), both by summation and root sum square techniques. The maximum value of $\bar{q} \cdot \alpha$ encountered on the dispersed trajectory is also output. Since the SCOUT program was written originally for a four-stage vehicle and then modified to simulate up to five stages, the program contains logic in the dispersion module to squeeze five stages of data into four. This is described starting on page 15-10.

<u>Programming</u> - trajectory computation

The basic function of the dispersion module is to store the nominal vehicle/trajectory characteristics, load the dispersion data, and integrate a dispersed trajectory using this data. After computing the dispersions in the trajectory variables and orbit elements, the nominal data is restored and the process repeated for the next desired dispersion.

Due to the large volume of data that potentially must be processed, several arrays from the standard Scout-Presto package are utilized in order to minimize computer storage requirements. The FWB array is used for storage of nominal characteristics, and the FWA array is renamed GIANT, SUM, and FWA, where the GIANT matrix primarily stores the dispersions in the trajectory variables and SUM is used for the three sigma computation. The AA matrix is used for temporary storage of the staging point trajectory variables and certain quantities from the nominal trajectory, the pitch program from data input goes into D4, and the DD matrix used for thrust-weight manipulations as in the basic program. The instantaneous quantities in the new matrices are:

AA(i,j)	Staging point values of trajectory variables		
	for	j = 1	stage 2 ignition
		j = 2	stage 3 ignition
		j = 3	stage 4 ignition
		j = 4	final stage burnout
	then for	i = 1	altitude, feet
		i = 2	velocity, ft/sec
		i = 3	flight path angle, degrees
		i = 4	downrange distance, n.m.
		i = 5	crossrange distance, n.m.
	and for	i = 7	nominal azimuth, radians
		i = 8	sin (nominal downrange angle)
		i = 9	cos (nominal downrange angle)
		i = 10	sin (nominal geocentric latitude)
		i = 11	cos (nominal geocentric latitude)
		i = 12	nominal longitude, radians
		i = 13	nominal latitude, radians

but for	$j = 5$	orbital elements
then for	$i = 1$	TWOE, $(ft/sec)^2$
	$i = 2$	EH, ft^2/sec
	$i = 3$	RP, ft
	$i = 4$	EYE, degrees
	$i = 5$	BETAP, degrees
	$i = 6$	OMEGAE, degrees

D4 TTH and QTH storage when dispersions are based on a trajectory using linearized pitch program

FWA The basic FWA array contains 1280 cells. In the DISPRS subroutine, the first 1200 cells become GIANT (6, 5, 40), the next 40 become SUM (5, 4, 2), the last unused 40 remain FWA.

FWB Storage of nominal vehicle/trajectory characteristics

FWB (900, 901) = DATE(1, 2)

Storage dependent upon dispersion trajectory code

JILL Code	FWB Subscript	Value
1-8	1-260	WEIGHT
	261-580	THRUST
	581-710	SLOPEW
	711-870	SLOPET
	871-884	DA1
	885-891	STGH
	892-895	WTTOGO
9-16	1	SGx(1)
19-20	1-42	CMALPH
21-23	1-2	Data Block 49
33	1	ST5(6)
34,35	Odd numbers to 33	TTH

GIANT(i,j,JILL) Dispersions in staging point trajectory variables; cells i = 1,5, j = 1,5 analogous to AA(i,j), and JILL = current dispersion trajectory code. For GIANT(6,j,JILL)

j = 1,2 dispersion title from data block 46
j = 3 maximum $\bar{q}\cdot\alpha$ encountered
j = 4 TIMCT of $\bar{q}\cdot\alpha_{max}$
j = 5 OMEGAE

Visualize the three dimensional GIANT array as a stack of 40 horizontal planes where each plane contains the dispersion information from one dispersed trajectory, except No. 39, which is blank, and No. 40, which contains the corresponding unperturbed trajectory variables from the nominal trajectory. A dispersion is defined as (variable on dispersed trajectory) - (variable from nominal trajectory) and is computed with the equation

$$\text{GIANT}(i,j,\text{JILL}) = \text{AA}(i,j) - \text{GIANT}(i,j,40)$$

and the proper subscripts.

SUM(i,j,k) Storage for the three sigma computation

i = trajectory variable index
j = staging point index
k = type of three sigma calculation
    k = 1 for Σ (dispersions)
    k = 2 for $[\Sigma\ (\text{dispersions})^2]^{\frac{1}{2}}$

Other important variables whose values are dependent upon the type of dispersion are:

CHANGE	JILL Code	Value
	9-16	stage weight increment, lb
	24-29	control deadband angle, radians
	30-32	fourth and/or fifth stage tipoff angle = effective attitude change, radians
	33	launch azimuth change, radians
	34,35	launch attitude change, radians

DATE	Dispersion type title, from data block 46
IDSP(i)	= -1 yaw plane dispersion trajectory = +1 pitch plane dispersion trajectory for i = 1 stage one thrust misalignment i = 2 stage two control deadband i = 3 stage three control deadband i = 4 stage four tipoff i = 5 winds otherwise IDSP = 0
IP(21)	= 1 dispersion module will be called = 0 no dispersed trajectories are required = -1 pitch plane dispersion is being computed = -2 yaw plane dispersion is being computed
JACK	Storage for nominal trajectory values of NSTAGE, NTHRST, and NWEIGH arrays
JILL	Code for type of dispersion trajectory currently being computed, see data block 46

Programming - three sigma variations

Three sigma variations in each of the dispersed trajectory variables are formed in two fashions, both by summing the dispersions and by the root sum square method. The dispersed trajectories are divided into two groups for the three sigma analysis, high/low trajectories and yaw trajectories, as designated by data block 46.

The three sigma variations for the yaw trajectories are computed first. Since these yaw trajectories are processed with no distinction between right and left crossrange dispersions, if the input data results in both positive and negative dispersions, the three sigma worst-on-worst summation will be less than a maximum crossrange variation (computed by summing absolute values).

The high/low trajectories represent pitch plane dispersions. The program searches the high/low group, excluding wind dispersions, and selects those trajectories whose final altitude dispersion is positive for the high three sigma computation, and those trajectories whose final altitude dispersion is negative for the low three sigma computation. If the altitude dispersion is zero, the velocity dispersion is examined (positive dispersion for high three sigma). After these three sigma arrays are output, the high/low wind dispersed trajectories are examined and added to the appropriate three sigma arrays and then output. This delay in adding the wind dispersions to the high/low three sigma computation allows the user, on one computer pass, to see the vehicle/launcher three sigma dispersions separated from the total three sigma variation that could be encountered.

Following the output of all the necessary dispersion information, the DISPRS subroutine then initiates the regeneration of the nominal trajectory, with proper storage in the standard arrays. Specifically, the nominal trajectory's TIMCT, x, y, z position history is loaded into the FWA array for RADAR and the standard trajectory variables are loaded into the FWB array for HRDOVR and IMPACT.

Programming - SIXD - Dispersion

Modifications were made to the SIXD-D subroutine to properly account for the first stage thrust misalignment and wind dispersed trajectories.

For first stage thrust misalignment, a term is added to the equation for the vehicle's resultant thrust attitude theta (chi),* computed from Eq. (7-6)-(7-7) in Section 7, Body Dynamics Simulation. The modified equation is

---

*When the JILL code indicates a high/low (yaw) trajectory, the thrust misalignment angle is assumed in the pitch (yaw) plane.

$$\theta = \frac{K_\theta M_\delta \theta_c - K_q M_\delta \dot{\theta}_c - M_\alpha \gamma_w + M_{\alpha_T} \cdot \alpha_T}{K_\theta M_\delta - M_\alpha}$$

where M_{α_T} = thrust misalignment unit pitching moment

M_{α_T} = T•(thrust application station - XCG1)

T = net thrust

α_T = thrust misalignment angle, radians,

where positive angles give positive moments

See Section 7 for additional definitions.

The additional control force required to keep the missile in rotational equilibrium and the thrust misalignment force are added to the total force F_{m_z} (F_{m_x}) in the missile axes system with the term

$$- T \cdot \alpha_T \cdot \left\{ 1 - \frac{(\text{thrust application station - XCG1})}{(x_\delta - \text{XCG1})} \right\} .$$

The wind dispersion analysis requires modification of the integrated velocity term to an aerodynamic velocity vector for angle of attack computations, recalculation of the dynamic pressure and redefinition of the aerodynamic forces.

The wind velocity and azimuth at a given altitude are obtained from the input data, and subtracted from the integrated velocity to form the air mass relative velocity of the vehicle, as

$$V_{ix} = - V_w \cdot \sin (\psi_w - \psi)$$

$$V_{iy} = V - V_w \cdot \cos (\psi_w - \psi) \cdot \cos \gamma$$

$$V_{iz} = V_w \cdot \cos (\psi_w - \psi) \cdot \sin \gamma$$

where V_w = magnitude of wind velocity

ψ_w = azimuth of wind's velocity vector

$V = X(1) =$ integrated velocity

$\gamma = X(2) =$ integrated flight path angle

$\psi = X(4) =$ integrated azimuth

The aerodynamic velocity components in the inertially fixed platform axis, $\bar{V}_j$, are found from

$$\begin{vmatrix} V_{jx} \\ V_{jy} \\ V_{jz} \end{vmatrix} = [C]^{-1} \begin{vmatrix} V_{ix} \\ V_{iy} \\ V_{iz} \end{vmatrix}$$

The dynamic pressure and drag force are next redefined using the $\bar{V}_j$ velocity.

Following the programming flow through Sections 7 and 11, the solution for the angles χ_w and A_{zw} is unchanged. However, computation of α and β, the in-plane and out-of-plane angles of attack, requires the components of $\bar{V}_j$ along the missile axes, $\bar{V}_m$.

$$\begin{vmatrix} V_{mx} \\ V_{my} \\ V_{mz} \end{vmatrix} = \begin{vmatrix} \cos \chi & -\sin \chi & 0 \\ \cos \theta \sin \chi & \cos \theta \cos \chi & \sin \theta \\ -\sin \theta \sin \chi & -\sin \theta \cos \chi & \cos \theta \end{vmatrix} \cdot \begin{vmatrix} V_{jx} \\ V_{jy} \\ V_{jz} \end{vmatrix}$$

Then $\tan \alpha = -V_{mz}/V_{my}$

and $\tan \beta = -V_{mx}/V_{my}$

Finally, the new drag force is added to the other forces in the j axes system using direction cosines from

$$F_{jx} \text{ (increment)} = - D \cdot (V_{jx} / |\bar{V}_j|)$$
$$F_{jy} \text{ (increment)} = - D \cdot (V_{jy} / |\bar{V}_j|)$$
$$F_{jz} \text{ (increment)} = - D \cdot (V_{jz} / |\bar{V}_j|)$$

then $D = 0$.

The drag term D is zeroed so the DEQ subroutine will not account for drag a second time, since $\bar{F}_j$, transformed to $\bar{F}_i$, is used as a corrective term in the equations of motion.

Although it is assumed that the dispersion analysis will always be generated including body dynamics effects, the thrust misalignment and wind dispersions are the only two that require the SIXD-D subroutine, and the other dispersion trajectories could be computed without body dynamics.

PROGRAMMING - NOMINAL DATA TRANSFORMATION

Before dispersion calculations are begun, the subroutine SQUEEZE transforms the majority of optimization module data into the form used by the dispersion package routines. SQUEEZE initially tests for conditions which are unacceptable: these include a coast after final powered stage, a linearized pitch program of more than 13 linear segments, and too many entries to THRUST or WEIGHT array when the modified histories are used on a five-stage vehicle.

For vehicles having 4 or less powered stages, the transformation is done by fairly simple data manipulations. However, 5-stage vehicles require data not previously available to the dispersion package. Two additional arrays were established, STGTW and NSTGTW, to transfer this information to DISPRS and TRAJD. These arrays contain stage 4, 5 and 9 data on times, number of integration points, jettison weights, and THRUST and WEIGHT table indices relating to coast 9 and powered stage 5. The THRUST and WEIGHT arrays are modified so that the time entries for stage 4,9, 5 are continuous from a time of 0 at stage 4 ignition. To the THRUST array is added one segment of thrust equal to zero over the coast. The WEIGHT array is modified so that each of the stage 4 weights have added to them the sum of the propellant weight at stage 5 ignition and the jettison weight of stage 4. Table indices indicating the beginning of coast 9 and stage 5 in both THRUST and WEIGHT arrays are stored in the NSTGTW array.

SQUEEZE stores the TTH table as the nominal θ history for dispersions. When dispersions are based on a linearized nominal trajectory, the linearized θ history is stored, followed by the spin stabilized stage θ, if applicable.

PROGRAMMING- COAST STAGE 9 AND POWERED STAGE 5 DISPERSION CALCULATIONS

The integration procedure has been modified so that for a five-stage vehicle, an integration step size is set at the beginning of each stage. These Δt's are determined using the current dispersed trajectory stage durations and the number of integration points specified for that stage on input to the optimization package. A critical time is also set to the value of the (dispersed) stage duration. The THRUST and WEIGHT table indices are also initialized to the correct value at the beginning of each stage. The stage is then integrated to the critical time established and whatever testing is required for control angle dispersion, etc., is done. This continues through stage 5.

Vacuum thrust variation in stages 4 and 5 require the inputs described on page 22-30. The dispersed stage durations for 4, 5, a computed value for coast 9, and the computed stage Δt's are stored. The input thrust/weight histories are first modified as described in the discussion of the SQUEEZE routine on the previous page. The modifications discussed in section 14 are now calculated. This is done by integrating over stage 4 with the stage Δt, setting the coast segment values, and then integrating over stage 5 with its stage Δt. The indices of the THRUST and WEIGHT tables relating to the first points of coast stage 9 and stage 5 are stored. They are used in the stage initializing procedures mentioned above.

Fourth and/or fifth-stage tipoff dispersions are included in the dispersion package. These are tested for and implemented at the staging points described above.

SECTION 16

FAILURE-MODE HARDOVER TURN CALCULATIONS

HARDOVER TURN MANEUVERS

Summary

As an input option, hardover turn maneuvers may be computed for the first three stages, starting from points in the optimum trajectory where a control malfunction is assumed to have occurred. These turns correspond to a hardover maneuver in pitch or yaw due to maximum deflection of the fins and jet vanes on the first stage, and control jets malfunctioning for the second and third stages.

Programming

The computational flow through HRDOVR, after setting input constants, first picks up initial conditions for the trajectory variables from a specified point along the nominal trajectory. Instantaneous forces and vehicle characteristics are computed, followed by evaluation of the over-turning moments and angular acceleration. If there is a thrust misalignment, the component of thrust along the body axis is correspondingly reduced. The equations of motion are evaluated and integrated using an Euler method technique. Checks are made for exceeding a dynamic pressure times angle of attack constraint, and for excessive tumbling or reaching the end of the stage. If these constraints are not violated, the hardover turn computation is continued until ten seconds has elapsed, after which new initial conditions are read in and the process repeated.

Equations

The equations of motion for the two dimensional problem of point mass motion over a flat Earth (non-rotating reference system) are:

$$\dot{V} = \frac{T}{m} \cos \alpha - \frac{D}{m} - g \sin \gamma$$

$$\dot{\gamma} = \frac{T}{mV} \sin \alpha + \frac{L}{mV} - \frac{g \cos \gamma}{V}$$

$$\dot{h} = \dot{r} = V \sin \gamma \qquad g = \frac{\mu}{r^2}$$

where V = velocity
γ = flight path angle of V from horizontal
h = altitude, distance above Earth's surface
r = distance from vehicle to Earth's "center"
T = net thrust
D = aerodynamic drag
L = aerodynamic lift
m = current mass
g = gravitational acceleration
α = angle of attack
μ = gravitational constant

For the special problem of hardover turn maneuvers, the maximum turning capability of the velocity vector is assumed identical in both pitch and yaw. Since this turning capability is computed only by integrating the $\dot{\gamma}$ equation, the $g \cos \gamma/V$ term is deleted as requested in Reference (a).* Also, the gravitational acceleration g and, for the upper stages, $\sin \gamma$ are assumed constant during the maneuver.

The vehicle's attitude angle, measured from horizontal to the vehicle's longitudinal axis, is defined as η, with $\alpha = \eta - \gamma$. When the over-turning forces acting normal to this axis are included, the equations of motion become:

Reference (a). COMPMR Instruction 5100.2C, Policy, Criteria and Procedures for Missile In-Flight Safety in the Pacific Missile Range PMR, 4 September 1963.

$$\dot{V} = (T \cdot \cos\alpha - D - FORCEN \cdot \sin\alpha)/m - g_o \sin\gamma_o$$

$$\dot{\gamma} = (T \cdot \sin\alpha + L + FORCEN \cdot \cos\alpha)/mV$$

$$\dot{r} = V \sin\gamma_o$$

where FORCEN for the first stage is defined as

$$FORCEN = - T \cdot \alpha_T - 2 \cdot \delta\ (CNDELT \cdot \bar{q} \cdot S + T_V \cdot K_{TV\delta})$$

α_T = thrust misalignment angle

δ = jet vanes and fins deflection angle

$K_{TV\delta}$ = jet vane effectiveness

S = aerodynamic reference area

Definitions for other terms may be found in Section 21, Coding Nomenclature.

And, for the upper stages,

$$FORCEN = - T \cdot \alpha_T \quad \text{(mode 1)} \quad \text{or}$$

$$FORCEN = - T_C \quad \text{(modes 2 and 3)}$$

where T_C = control jet thrust force

Note that positive angles and forces give positive moments (vehicle nose up).

Hardover turn maneuvers are computed assuming the following combinations of over-turning moments.

Stage 1

Control fin and jet vane deflection, with simultaneous additive thrust misalignment.

Stages 2 and 3

Mode 1. Thrust misalignment only

Mode 2. Single control jet operation

Mode 3. One pitch and one yaw control jet operation, with 45° roll angle assumed

Stage 1 simulation includes the atmospheric effects of a restoring moment due to angle of attack and pitch damping moment; all stages include jet damping effects.

The applicable moment equations are:

1. M (control deflection) $= \delta \{ \bar{q} \cdot S \cdot (-\text{CMDELT} \cdot \ell_{ref} - \text{CNDELT} \cdot (\text{XCG1} - X_{FS})) + K_{TV\delta} \cdot T_V \cdot (X_\delta - \text{XCG1}) \}$

   where ℓ_{ref} = reference length for aero moment data

   X_δ = jet vane thrust application station

2. M (thrust misalignment) $= T \cdot \alpha_T (X_{TA} - \text{XCGi})$

   where X_{TA} = ith stage effective thrust application station

   XCGi = ith stage center of gravity station

3. M (angle of attack) $= \alpha \{ \bar{q} \cdot S \cdot (\text{CMALPH} \cdot \ell_{ref} + TL_1 (\text{XCG1} - X_{FS}))\}$

4. M (pitch damping) $= \text{CMQ} \cdot \dot{\eta} \cdot \ell_{ref} \cdot \bar{q} \cdot S \cdot \ell_{ref}/2V$

   where CMQ = pitch damping coefficient

5. M (jet damping) $= \dot{\eta} \cdot (\dot{m} (\text{XCG1} - X_{TA})^2 + \dot{I}_{yy})$

   where $\dot{m}$ = instantaneous mass time derivative

   $\dot{I}_{yy}$ = pitch or yaw moment of inertia time derivative

6. M (control jets) $= T_C \cdot (X_{TAC} - \text{XCGi})$

   where X_{TAC} = control force application station

The vehicle's instantaneous attitude η is determined by integrating the angular acceleration $\ddot{\eta}$, where $\ddot{\eta} = \Sigma$ (applicable moments)$/I_{yy}$, with $\eta_o = \gamma_o$.

SECTION 17

NOMINAL IMPACT LOCUS

IMPACT

As an input option, impact points may be predicted for the booster, both assuming thrust failure during operation of the first three stages, and for the expended casings of the first three stages. The initial conditions for the impact trajectories are taken from the optimum trajectory at specified intervals, and aerodynamic drag force is included for all trajectories.

Programming

Coast stage 6 nomenclature is used for the integration of the impact trajectories, so the nominal stage 6 drag and quotient arrays are stored in the DD array. The current failure mode drag curve and corresponding quotient table are loaded into the respective stage 6 arrays. Initial conditions for the trajectory variables are taken from the FWB array, and program control is transferred to subroutine INTPIM (INsToP for IMpact). INTPIM sets the integration step size dependent upon the magnitude of the drag acceleration force. Four step sizes are used, corresponding to whether the vehicle is still in the atmosphere, has left the atmosphere, has reentered the atmosphere, or is at terminal velocity. Integration proceeds until impact occurs on a geodetic Earth model. Program control then returns to IMPACT, the impact location is output, and the process repeated for the next set of initial conditions.

SECTION 18

RADAR TRACKING COORDINATES

RADAR COMPUTATIONS

The radar subroutine computes the look angles and slant ranges from Earth-fixed locations to the optimum trajectory as an input option. The radar site locations are input in terms of geodetic latitude, longitude, and altitude, and up to twenty sites can be accommodated.

Programming

The input radar site identification and coordinates are stored in ISTATN and STCORD. The time history of the booster's x, y, z geocentric rectangular coordinates on the optimum trajectory have been stored in the FWA array. For each radar station, the geodetic location is converted to geocentric rectangular coordinates and a transformation matrix is computed. Then, starting at launch and continuing through stage-four burnout, the booster's position is taken from the FWA array at each timepoint and the look angles and slant range are computed and output. The process is then repeated for the next radar station input.

Equations

The following equations are for the computation of azimuth and elevation angles and slant range distance from a geodetic Earth-fixed location to a space vehicle:

1. Azimuth

$$a = \mathrm{Tan}^{-1} \left(\frac{Y_{gi}}{X_{gi}} \right)$$

2. Elevation

$$e = \mathrm{Tan}^{-1} \frac{-Z_{gi}}{(X^2_{gi} + Y^2_{gi})^{1/2}}$$

3. Slant Range

$$SR = (X^2_{gi} + Y^2_{gi} + Z^2_{gi})^{1/2}$$

In equations 1, 2, 3, the quantities X_{gi}, Y_{gi}, Z_{gi} are topocentric goedetic coordinates of the vehicle relative to the ith tracking station with X_{gi} directed along the local geodetic north and tangent to the spheroid; Y_{gi} direct along the local east and Z_{gi} is normal to the spheroid and positive towards the geocenter. They can be found from the following transformation:

$$\begin{vmatrix} X_{gi} \\ Y_{gi} \\ Z_{gi} \end{vmatrix} = \begin{vmatrix} A(\lambda_d, u_i) \end{vmatrix} \begin{vmatrix} x - x_i \\ y - y_i \\ z - z_i \end{vmatrix}$$

x_i = $r_i \cos\lambda \cos u_i$

y_i = $r_i \cos\lambda \sin u_i$

z_i = $r_i \sin\lambda$

u_i = Greenwich longitude, positive eastward, of the ith station

λ = Geocentric latitude

r_i = Geocentric radius

In this transformation x, y, z, and x_i, y_i, z_i are the geocentric coordinate locations of the vehicle and ith station, respectively, with the x, y coordinates lying in the equatorial plane and z along the polar axis positive north.

The transformation from the inertial geocentric to the inertial geodetic is the following.

$$A(\lambda_d, u_i) = \begin{vmatrix} -\cos u_i \sin \lambda_d & -\sin u_i \sin \lambda_d & \cos \lambda_d \\ -\sin u_i & \cos u_i & 0 \\ -\cos \lambda_d \cos u_i & -\cos \lambda_d \sin u_i & -\sin \lambda_d \end{vmatrix}$$

In the above transformation, λ_d is the geodetic latitude.

The geocentric coordinates of the ith station (x_i, y_i, z_i) are computed from the input geodetic latitude, longitude, and altitude location of the station. The oblate Earth is represented by an ellipsoid of reference having an inverse flattening of 298.3.* The equations for the geocentric rectangular coordinates are:

$$\tan RA = \frac{(f - 1)}{f} \tan \lambda_d$$

where $f = 298.3$

$$z_i = R_e \frac{(f - 1)}{f} \sin RA + ALT \sin \lambda_d$$

where R_e = equatorial radius, ALT = station's altitude

$$xy_i = R_e \cos RA + ALT \cos \lambda_d$$

$$x_i = xy_i \cos u_i$$

$$y_i = xy_i \sin u_i$$

---

[*] See reference on page 4-21.

SECTION 19

SPECIAL SUBROUTINES

SCOUT ATMOSPHERE SUBROUTINE
(1962 ARDC MODEL)

Required: Density, Pressure, Speed of Sound
(ρ) (p) (a)

Symbols

h	geometric altitude in feet $(h = r - R_e)$
H^*	geopotential altitude in geopotential feet'
ρ	density in slugs
p	pressure in lb/ft^2
a	speed of sound
T_M	molecular-scale temperature in ^oR at altitude H^*
L_M	gradient of T in terms of H^* ; i.e., $\frac{\partial T_M}{\partial H^*}$ in $^oR/ft'$

Subscripts

o	denotes property at sea level $h = H^* = 0$
b	denotes property at base of particular layer

Constants

$g_o = 32.174\ ft/sec^2$ (acceleration of gravity measured at sea level)

$R^* = 1715.4827\ ft^2/{}^oR\ sec^2$, gas constant for air

Equations

$$H^* = \frac{R_e \cdot h}{R_e + h}$$

Type I $(L_M)_b = 0$

$$T_M = (T_M)_b$$

$$\rho = \rho_b \exp\left[\frac{-g_o\,(H^* - H^*_b)}{R^*\,(T_M)_b}\right]$$

$$p = \rho\, R^*\,(T_M)_b$$

$$a = \sqrt{1.4R^*\,(T_M)_b}$$

Type II $(L_M)_b \neq 0$

$$T_M = (T_M)_b + (L_M)_b\,(H^* - H_b^*)$$

$$\rho = \rho_b\left[1 + \frac{(L_M)_b\,(H^* - H^*_b)}{(T_M)_b}\right]^{-\left(1 + \frac{g_o}{R^*(L_M)_b}\right)}$$

$$p = \rho R^*\,T_M$$

$$a = \sqrt{1.4R^*\,T_M}$$

TABLE OF CONSTANTS AT BASE ALTITUDES

H^*_b	$(T_M)_b$	$(L_M)_b$	ρ_b
0	518.69	-3.56616×10^{-3}	2.3769×10^{-3}
36,089.239	389.988	0	7.0547×10^{-4}
82,020.997	389.988	1.646592×10^{-3}	7.7615×10^{-5}
154,199.475	508.788	0	2.8829×10^{-6}
173,884.514	508.788	-2.46888×10^{-3}	1.3964×10^{-6}
259,186.352	298.188	0	4.1123×10^{-8}
295,275.591	298.188	2.19456×10^{-3}	4.2560×10^{-9}
344,488.189	406.188	1.09728×10^{-2}	2.2243×10^{-10}

DERIVATIVES OF (ρ, p, a) WITH RESPECT TO ALTITUDE

1. Assume $\frac{\partial(\ )}{\partial h} = \frac{\partial(\ )}{\partial H^*}$

2. For Type I $(L_M)_b = 0$

$$\frac{\partial \rho}{\partial h} = - \frac{\rho\ g_o}{R^*\ (T_M)_b}$$

$$\frac{\partial p}{\partial h} = R^*\ (T_M)_b\ \frac{\partial \rho}{\partial h}$$

$$\frac{\partial a}{\partial h} = 0$$

3. For Type II $(L_M)_b \neq 0$

$$\frac{\partial \rho}{\partial h} = \frac{- \rho\ \left[1 + \frac{g_o}{R^*\ (L_M)_b}\right]\ \left[\frac{(L_M)_b}{(T_M)_b}\right]}{\left[1 + \frac{(L_M)_b\ (H^* - H^*_b)}{(T_M)_b}\right]}$$

$$\frac{\partial p}{\partial h} = R^* \left(T_M\ \frac{\partial \rho}{\partial h} + \rho\ (L_M)_b\right)$$

$$\frac{\partial a}{\partial h} = \frac{0.7\ R^*\ (L_M)_b}{a}$$

THE RKAD INTEGRATION SUBROUTINE

The numerical integration subroutine in the Scout program makes use of both the Runge-Kutta and the Adams methods of integration. The Adams method is faster than the Runge-Kutta but requires stored derivatives of the variable at three points prior to the current point. It, therefore, cannot be used to start the integration. The Runge-Kutta method does not require past information and can be used to start the integration.

An integration interval constant over each stage is used in the program, both for speed and bookkeeping ease in the optimization trajectories. In powered stages an interval one to two seconds is reasonable. However, since the thrust can vary rapidly with time, the Adams method is not sufficiently accurate with that large an integration step. Therefore, the Runge-Kutta method is used during the powered stages. However, during the coasts the integration starts in Runge-Kutta and switches to the Adams method after four steps.

The Runge-Kutta method used is standard fourth order. The Adams method computes the increment in any variable in terms of the current derivative and the derivatives at the three previous points. The increment δy is given by

$$\delta y = (55\,\dot{y}_1 - 59\,\dot{y}_2 + 37\,\dot{y}_3 - 9\,\dot{y}_4)\,\frac{\delta t}{24}$$

where δt is the size of the integration step, $\dot{y}_1$ is the current derivative, and $\dot{y}_2$ is the derivative one point back, etc.

SUBROUTINE SYMVRT

Identification

F1*ML F HSIV SYMMETRIC MATRIX INVERSION

Ira C. Hanson, Lockheed
November 1962

Purpose

This subroutine calculates the inverse of a symmetric matrix.

Method

The algorithm of Cholesky is used to decompose the symmetric matrix A into a triangular matrix B such that A = BB*. The asterisk denotes transpose. If the matrix A is not positive-definite, the matrix B will contain some imaginary elements. The triangular matrix B is then inverted by direct elimination. Since $(B^{-1})^* = (B^*)^{-1}$, it is unnecessary to compute $(B^*)^{-1}$. The final inverse is then computed as follows. $A^{-1} = (B^*)^{-1} B^{-1}$. All imaginary elements drop out at this point. Only the upper triangular part of A is used in the computation.

For a complete description of the algorithm of Cholesky, see E. Rodewig, "Matrix Calculus," North Holland Publishing Company, Amsterdam, 1956, pages 110-114.

Usage

Entrance to the subroutine is made via the FORTRAN statement in the calling program.

CALL SYMVRT (A, N, ISING)

where 1) A is the label of the matrix to be inverted. Only the upper triangular part of A is required. After the inversion is complete, the inverse is stored in the lower triangular part of A. The original matrix is destroyed.

2) N is the number of rows in the matrix.

3) ISING will be set to zero if the inversion was successful. ISING will equal one if the matrix in A is singular.

The subroutine uses three temporary single subscripted arrays. These arrays must be dimensioned at least as large as the row entry of the A array. These arrays may be placed in COMMON to conserve storage if desired.

Restrictions

The Cholesky decomposition will fail if a zero appears during the computation of the diagonal elements and also if A(1,1) = 0. This does not necessarily mean the matrix is singular, but it does mean the calculation of the inverse has failed. There is no practical fool-proof method of a priorily interchanging rows and columns to avoid this trouble. Interchanging after a zero is detected is not a solution because the remaining elements may also be zero and the matrix still not be singular. Therefore, no pivot search is attempted in this subroutine and it should only be used in a physical application where it is known that this restriction is not prohibitive. Otherwise, the Crout method subroutine FL*ML F HINT should be used which has a complete pivot search.

Space Required

620 cells are required.

Timing

Running time (T in seconds) is approximately $T = .00005\ n^3$, where n is the number of rows in the matrix. A 36 x 36 case took 3.6 seconds.

Accuracy

Accuracy depends on the particular case being run. E. Bodewig says, "A feature of the method is that it yields smaller rounding errors than the method of Gauss-Doolittle or other methods." Several random cases were compared with the Crout inversion subroutine F1*ML F HINT and in all cases the $AA^{-1} = I$ check was as good as, or better, using SYMVRT.

SECTION 20

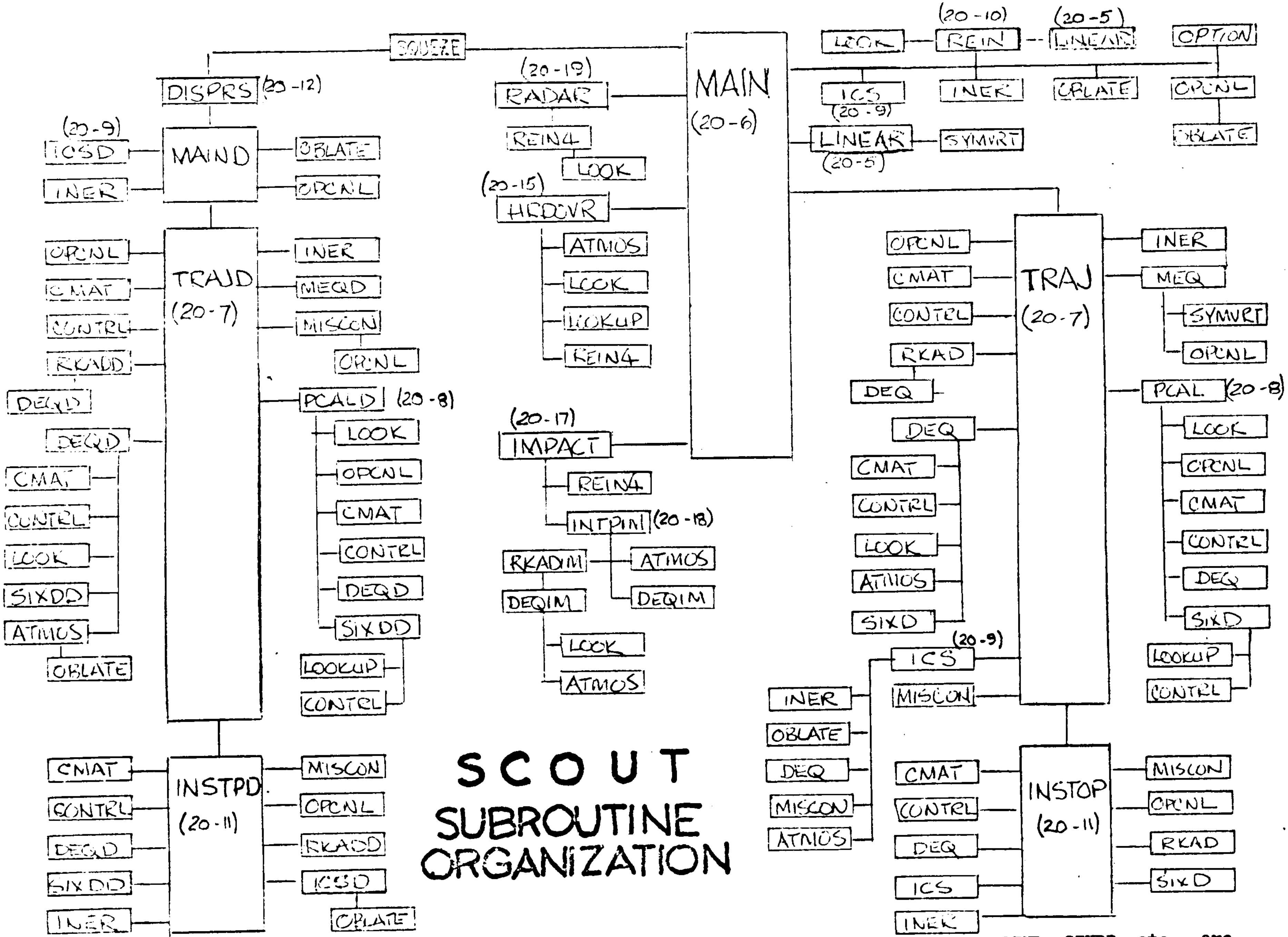
PROGRAM ORGANIZATION

PROGRAM ORGANIZATION

In this section the computational flow of the Scout program is documented. The subroutine organization and interrelationships are described, along with lists of functions performed in each subroutine. Finally, block flow diagrams of some of the individual, more complicated subroutines are provided.

Overall Computational Flow

In this program there are several different types of trajectories to be computed, dependent on the user selection of input options. The MAIN program sequences these trajectories, sets up initial values of various quantities needed for the trajectory, and then calls a subroutine to complete the calculation of the trajectory before a return to MAIN. Aside from these two levels of subroutine organization, there is a third level in which most of the program subroutines fall. These routines are called at least once per integration step and are generally concerned with evaluation of the time derivatives of the trajectory variables for numerical integration. Figure 20-1 shows the calling relationships among the routines including the grouping for the optimization and pitch program linearization on the right side, the similar group for computation of dispersed trajectories on the left, and the radar, hardover turns and impact trajectories in the center. Also indicated are the page numbers for the flow diagrams of the respective subroutines. On the following page a listing of the grouping of the subroutines in overlay links is shown. Finally, the types of operations performed in each subroutine is indicated so that the reader can receive a basic familiarity with the computational organization.



SCOUT SUBROUTINE ORGANIZATION

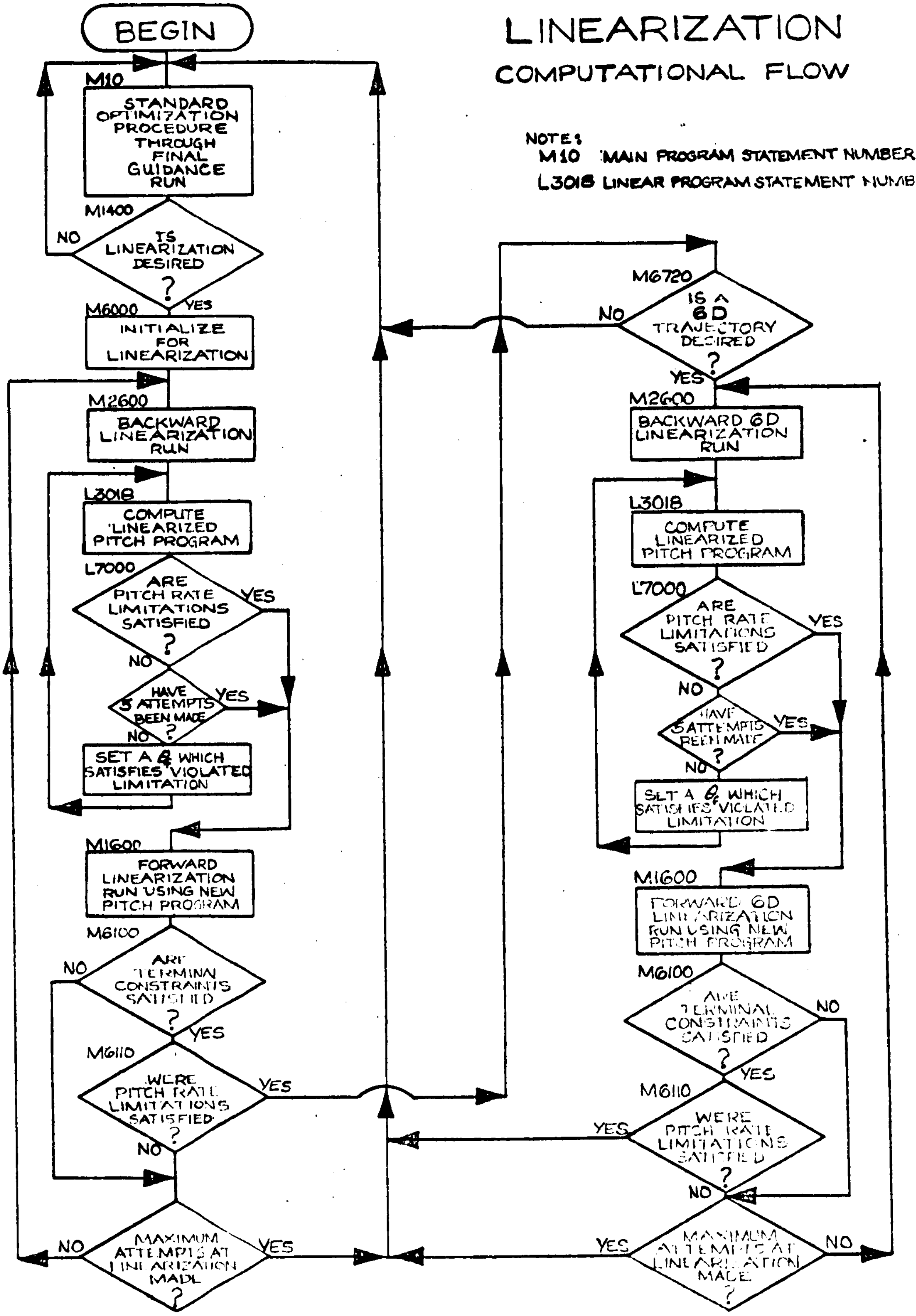
NOTE: Complete lists of subroutines called from repeated subroutines such as DEQ, DEQD, SIXD, SIXDD, etc., are shown only once.

SUBROUTINE OVERLAY LINKAGE ARRANGEMENT

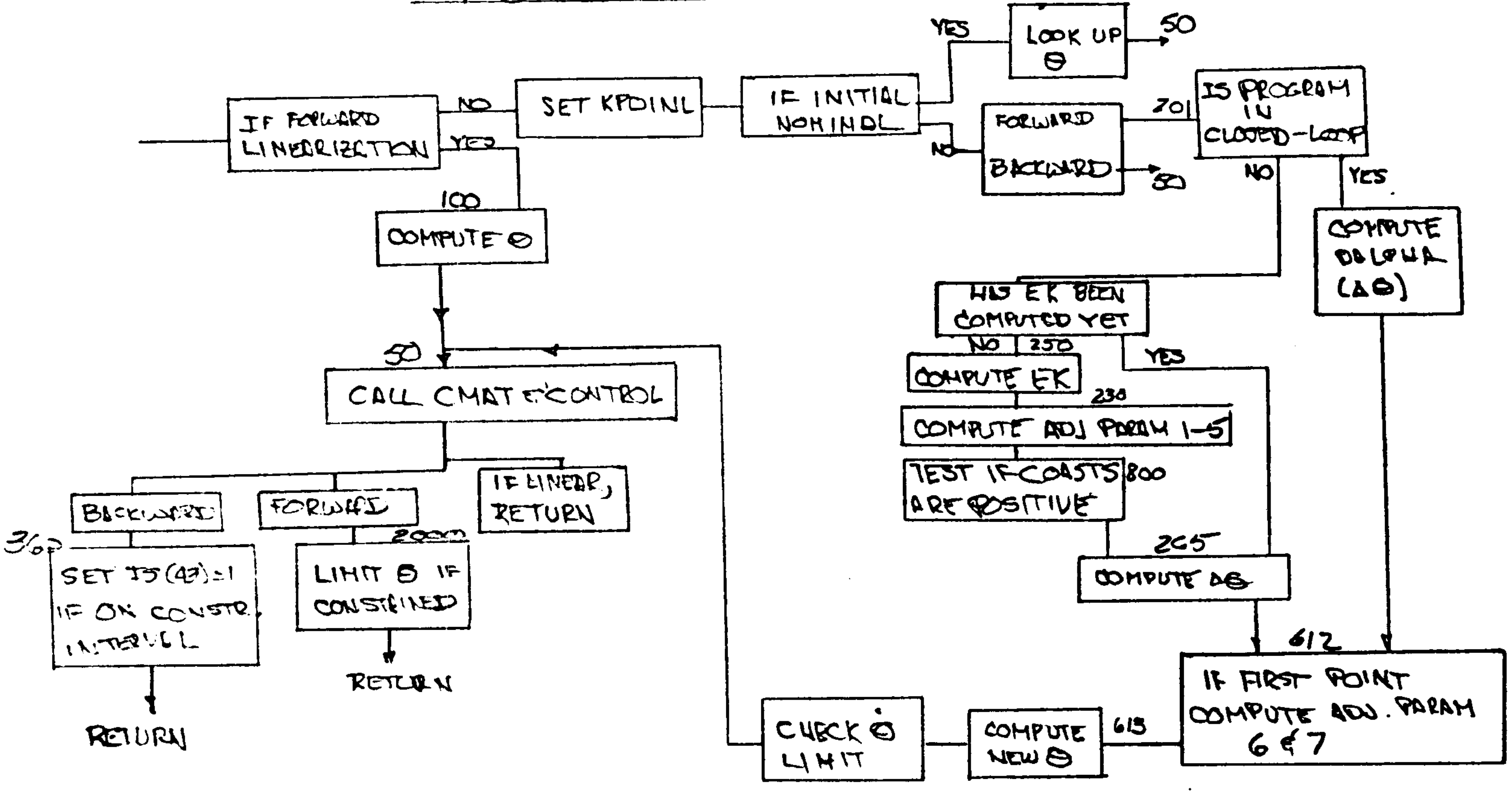
LINK ZERO	LINK ONE	LINK TWO	LINK THREE	LINK FOUR
Always in Core	Case Setup	Optimization and Exchange Ratio	Dispersion Trajectories	Hardover, Impact, Radar
ATMOS				
CMAT				
CONTRL		DEQ	DEQD	DEQIM
			DISPRS	
			SDISPR	
				HRDOVR
ICS			ICSD	
				IMPACT
INER				
		INSTOP	INSTPD	INTPIM
	RLIN	LINEAR		
LOOK				
LOOKUP				
MAIN			MAIND	
		MEQ	MEQD	
MISCON				
OBLATE				
OPCNL				
	OPTION			
		PCAL	PCALD	
				RADAR
	REIN			REIN4
		RKAD	RKADD	RKADIM
		SIXD	SIXDD	
SYMVRT				
		TRAJ	TRAJD	
	WIC			
		SQUEEZE		

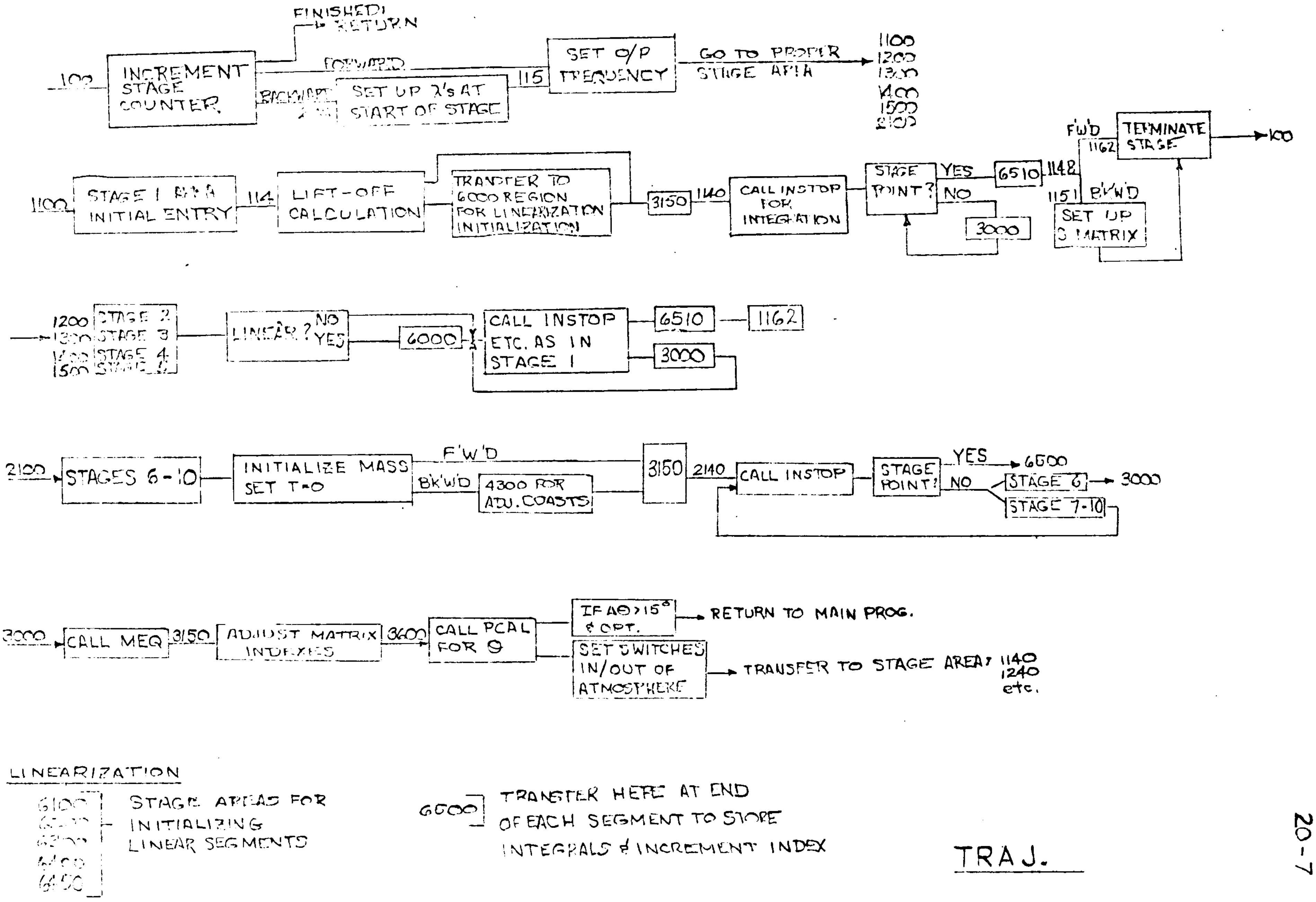
LINEARIZATION
COMPUTATIONAL FLOW

NOTES:
M10 MAIN PROGRAM STATEMENT NUMBER
L3018 LINEAR PROGRAM STATEMENT NUMBER



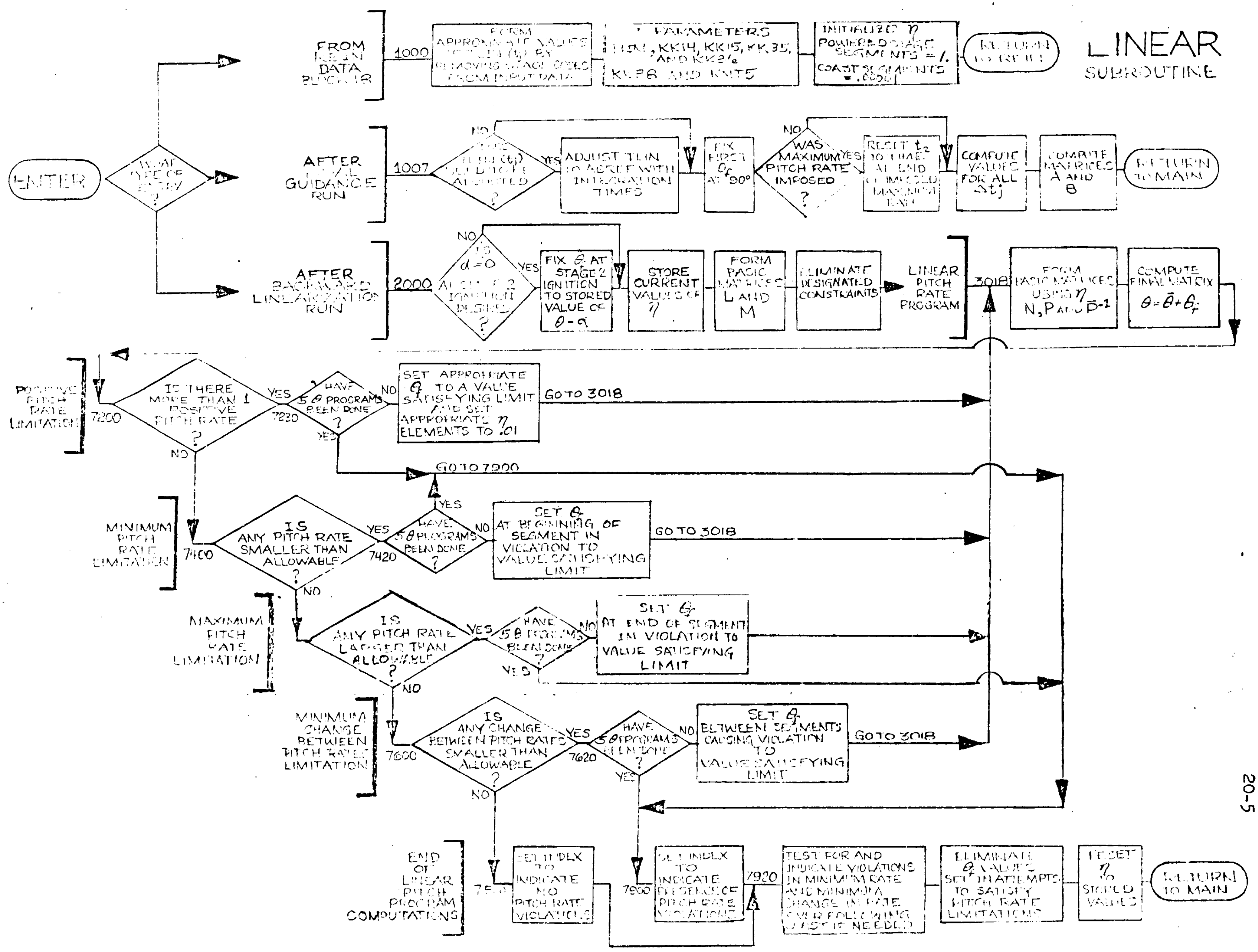
PCAL

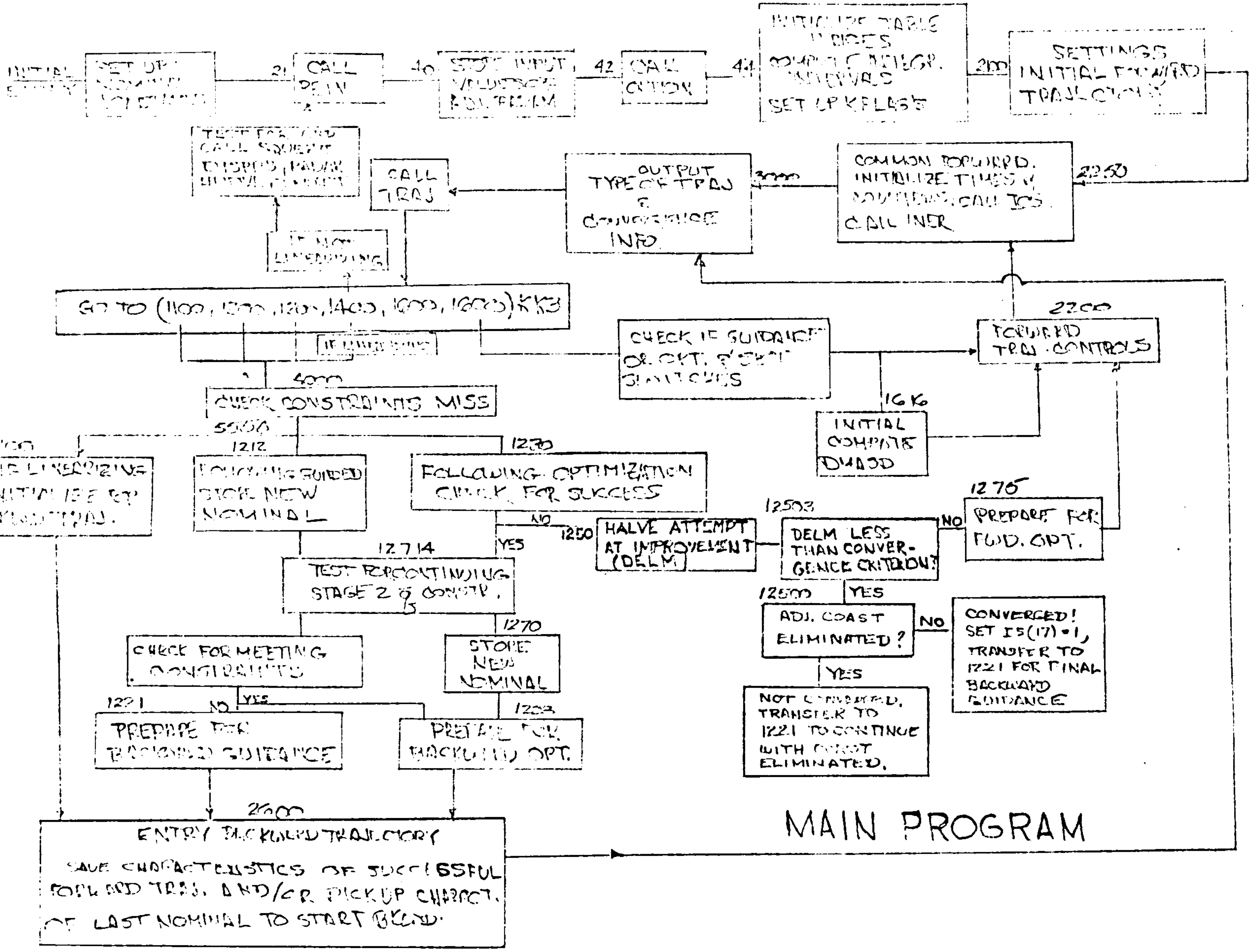

IF FORWARD LINEARIZATION
NO
YES
SET KPDINL
IF INITIAL NOMINAL
YES
LOOK UP Θ
50
NO
FORWARD
BACKWARD
50
201
IS PROGRAM IN CLOSED-LOOP
NO
YES
COMPUTE DELTA (ΔΘ)
100
COMPUTE Θ
50
CALL CMAT & CONTROL
HAS EK BEEN COMPUTED YET
NO
250
YES
COMPUTE EK
230
COMPUTE ADJ PARAM 1-5
TEST IF COASTS ARE POSITIVE
800
265
COMPUTE ΔΘ
612
IF FIRST POINT COMPUTE ADJ. PARAM 6 & 7
BACKWARD
FORWARD
IF LINEAR, RETURN
360
SET IS(43)=1 IF ON CONSTR. INTERVAL
2000
LIMIT Θ IF CONSTRAINED
RETURN
RETURN
CHECK Θ̇ LIMIT
COMPUTE NEW Θ
615


100
INCREMENT STAGE COUNTER
FINISHED, RETURN
FORWARD
BACKWARD
SET UP λ's AT START OF STAGE
115
SET O/P FREQUENCY
GO TO PROPER STAGE AREA
1100
1200
1300
1400
1500
2100
1100
STAGE 1 B & A INITIAL ENTRY
114
LIFT-OFF CALCULATION
TRANSFER TO 6000 REGION FOR LINEARIZATION INITIALIZATION
3150
1140
CALL INSTOP FOR INTEGRATION
STAGE POINT?
YES
NO
3000
6510
1148
1151
F'W'D
1162
TERMINATE STAGE
100
B'K'W'D
SET UP C MATRIX
1200
1300
1400
1500
STAGE 2
STAGE 3
STAGE 4
STAGE 5
LINEAR?
NO
YES
6000
CALL INSTOP ETC. AS IN STAGE 1
6510
1162
3000
2100
STAGES 6-10
INITIALIZE MASS SET T=0
F'W'D
BK'W'D
4300 FOR ADJ. COASTS
3150
2140
CALL INSTOP
STAGE POINT?
YES
6500
NO
STAGE 6
3000
STAGE 7-10
3000
CALL MEQ
3150
ADJUST MATRIX INDEXES
3600
CALL PCAL FOR Q
IF ΔΘ>15° & OPT.
RETURN TO MAIN PROG.
SET SWITCHES IN/OUT OF ATMOSPHERE
TRANSFER TO STAGE AREA 1140
1240
etc.
LINEARIZATION
6100
6200
6300
6400
6500
STAGE AREAS FOR INITIALIZING LINEAR SEGMENTS
6000
TRANSFER HERE AT END OF EACH SEGMENT TO STORE INTEGRALS & INCREMENT INDEX

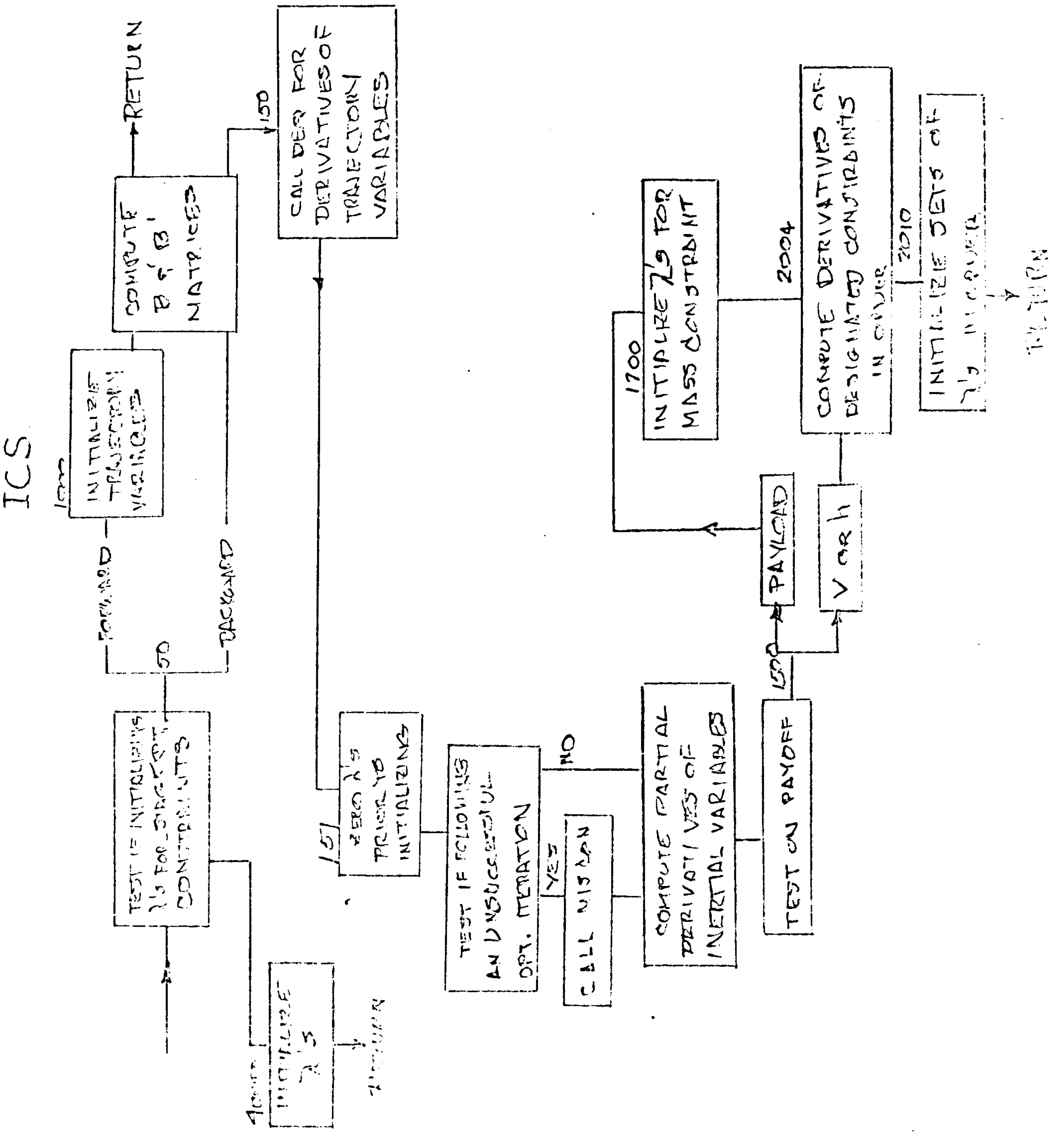
TRAJ.

LINEAR
SUBROUTINE

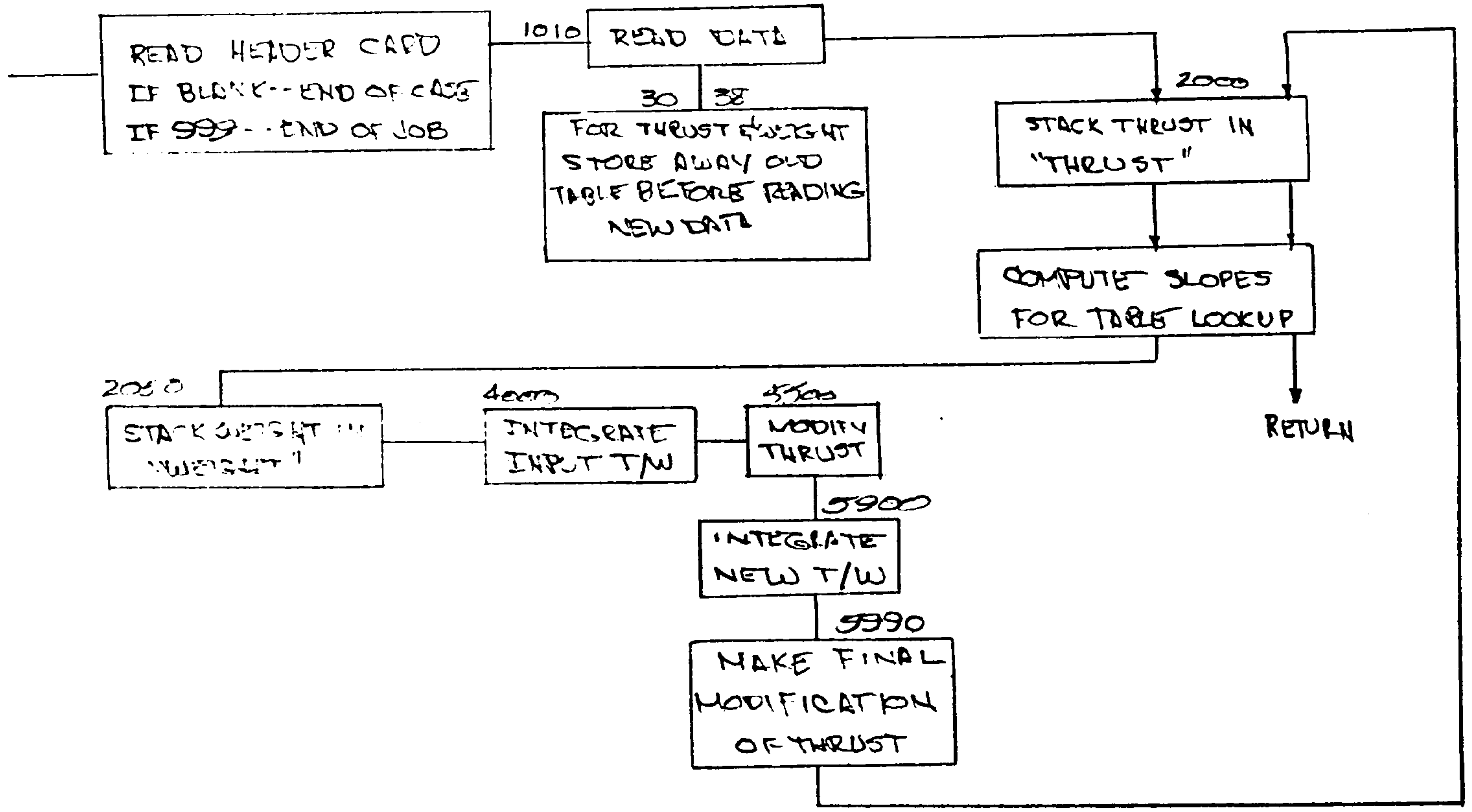




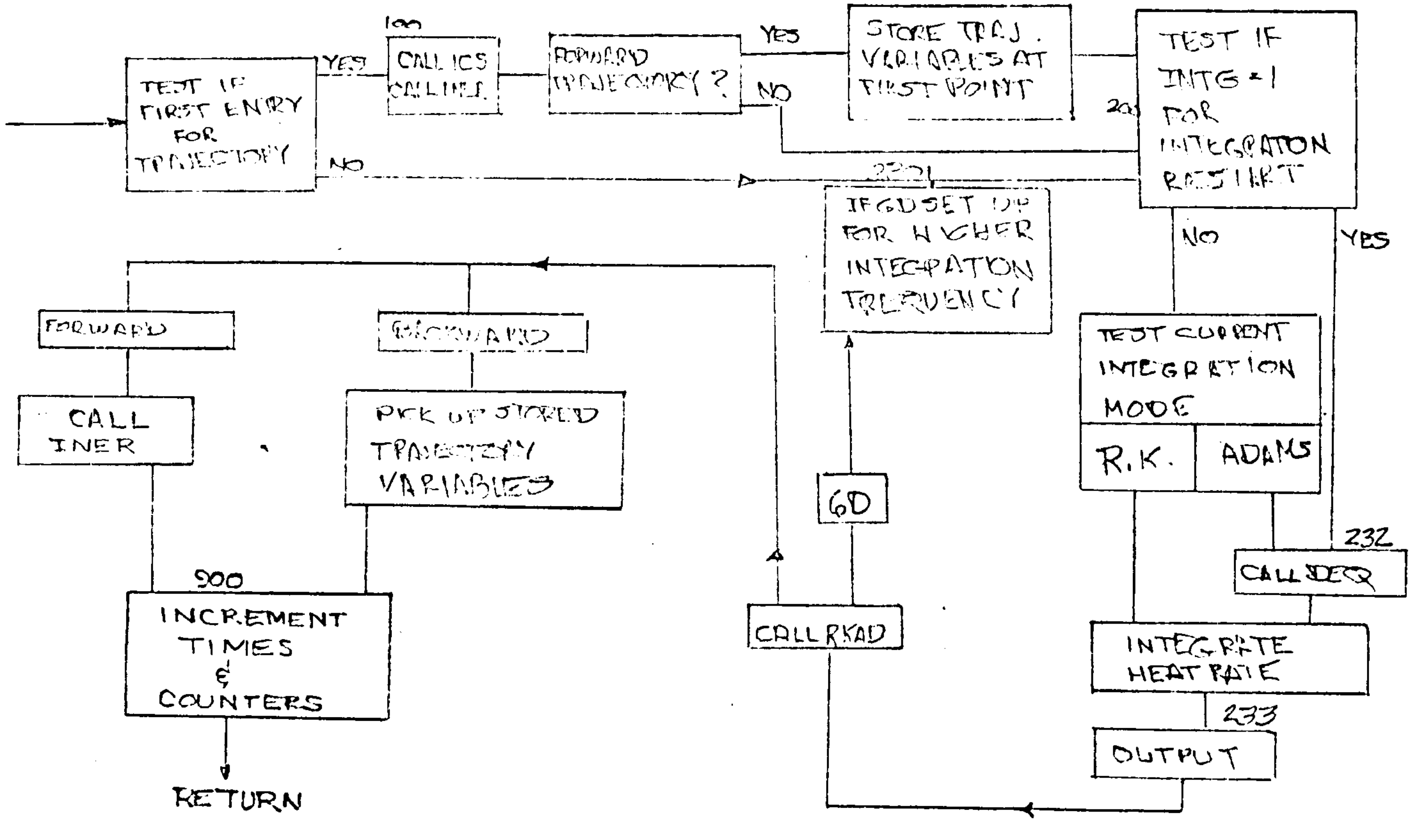
MAIN PROGRAM


ICS
1000
INITIALIZE TRAJECTORY VARIABLES
COMPUTE B & B' MATRICES
RETURN
150
CALL DER FOR DERIVATIVES OF TRAJECTORY VARIABLES
TEST IF INITIALIZING λ's FOR TARGET CONSTRAINTS
50
FORWARD
BACKWARD
4000
INITIALIZE λ's
RETURN
151
ZERO λ's PRIOR TO INITIALIZING
TEST IF FOLLOWING AN UNSUCCESSFUL OPT. ITERATION
NO
YES
CALL NISAPN
COMPUTE PARTIAL DERIVATIVES OF INERTIAL VARIABLES
TEST ON PAYOFF
1500
PAYLOAD
V or h
1700
INITIALIZE λ's FOR MASS CONSTRAINT
2004
COMPUTE DERIVATIVES OF DESIGNATED CONSTRAINTS IN ORDER
2010
INITIALIZE SETS OF λ's IN ORDER
RETURN

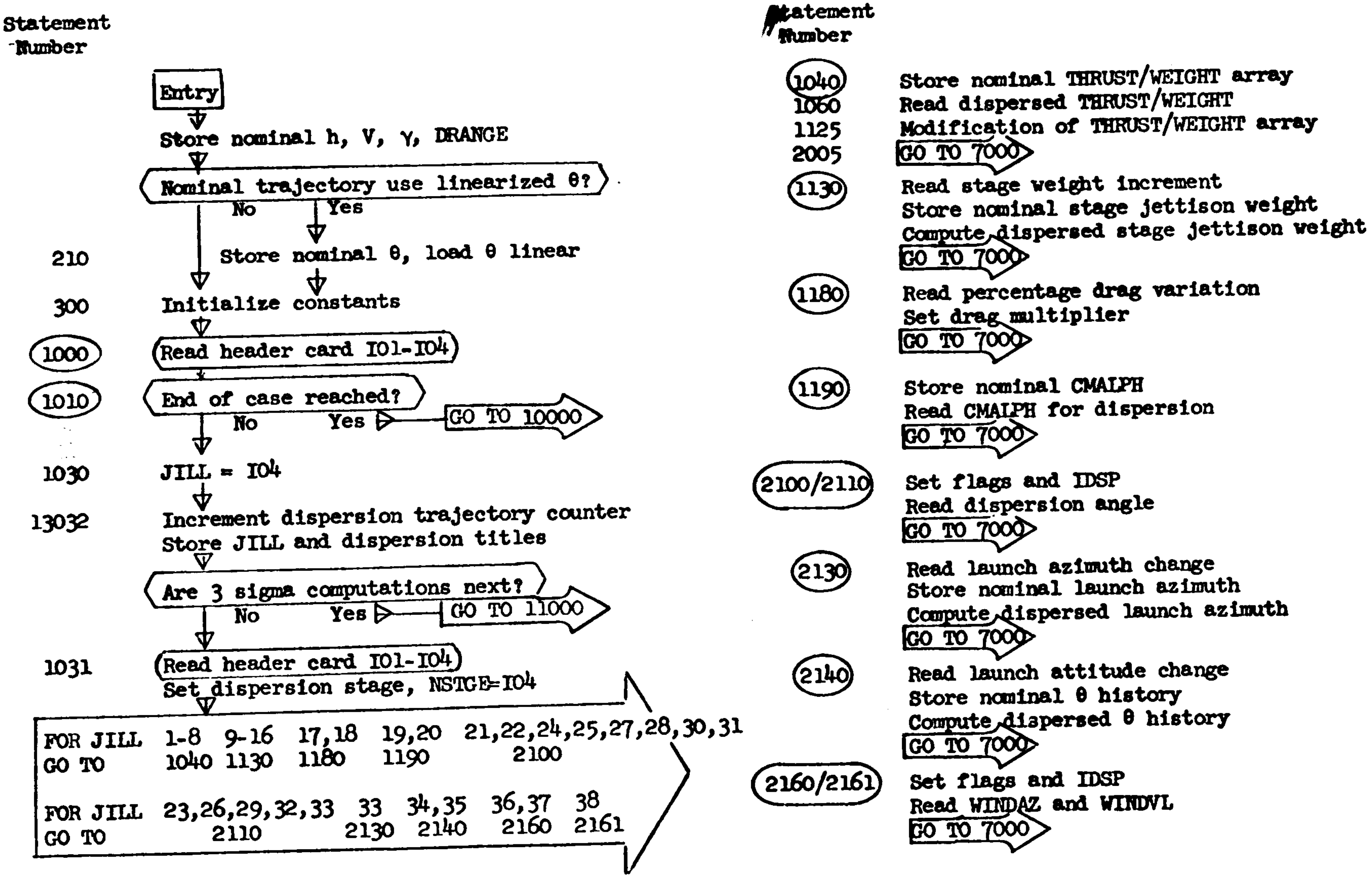
REIN


READ HEADER CARD
IF BLANK--END OF CASE
IF 999--END OF JOB
1010
READ DATA
30
38
FOR THRUST & WEIGHT STORE AWAY OLD TABLE BEFORE READING NEW DATA
2000
STACK THRUST IN "THRUST"
COMPUTE SLOPES FOR TABLE LOOKUP
RETURN
2050
STACK WEIGHT IN "WEIGHT"
4000
INTEGRATE INPUT T/W
5500
MODIFY THRUST
5900
INTEGRATE NEW T/W
5990
MAKE FINAL MODIFICATION OF THRUST

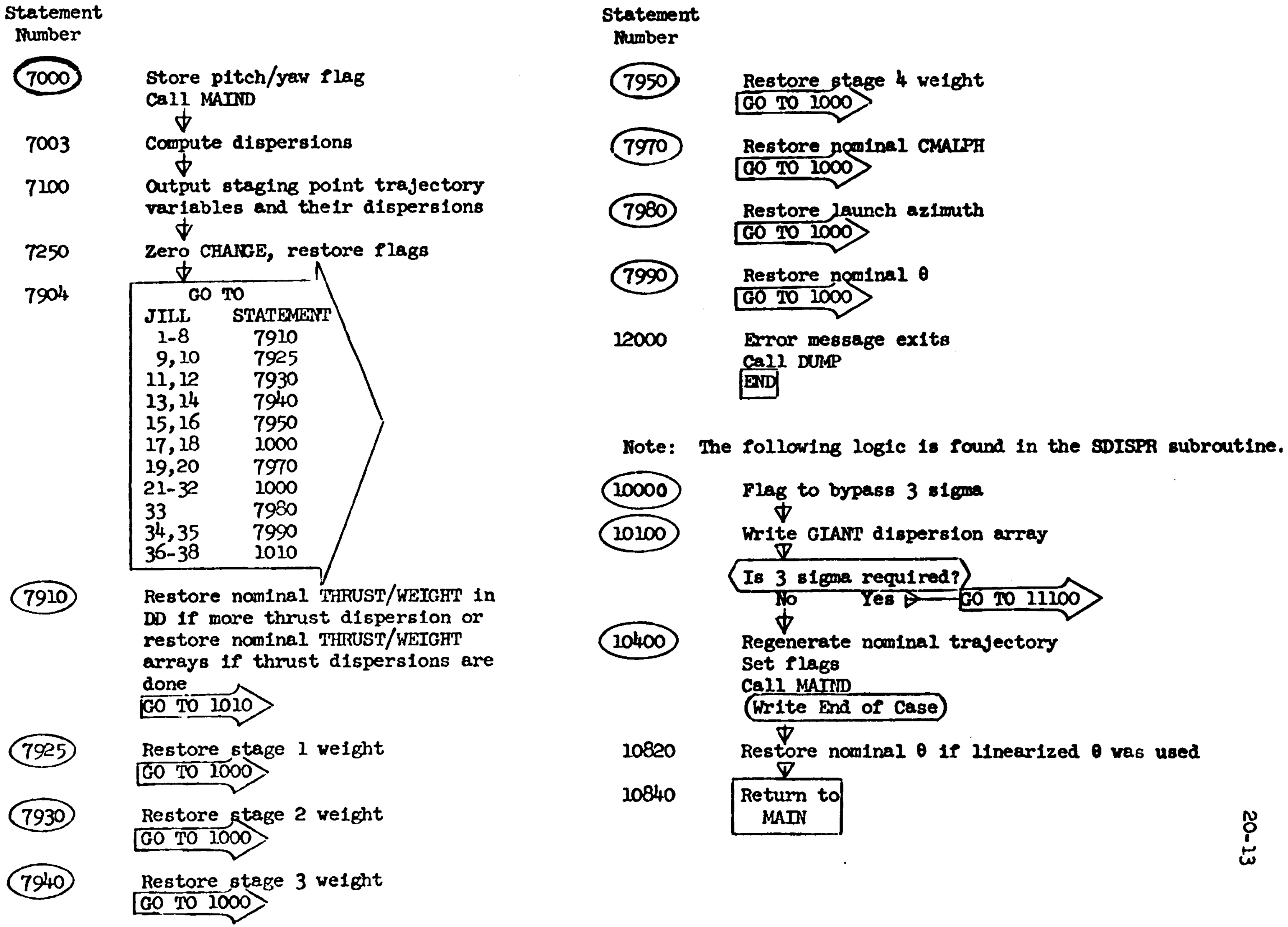
INSTOP


TEST IF FIRST ENTRY FOR TRAJECTORY
YES
NO
100
CALL ICS
CALL INEL
FORWARD TRAJECTORY ?
YES
NO
STORE TRAJ. VARIABLES AT FIRST POINT
20
TEST IF INTG = 1 FOR INTEGRATION RESTART
2321
IFGD SET UP FOR HIGHER INTEGRATION FREQUENCY
NO
YES
TEST CURRENT INTEGRATION MODE
R.K.
ADAMS
232
CALL DEQ
INTEGRATE HEAT RATE
233
OUTPUT
6D
CALL RKAD
FORWARD
BACKWARD
CALL INER
PICK UP STORED TRAJECTORY VARIABLES
900
INCREMENT TIMES & COUNTERS
RETURN

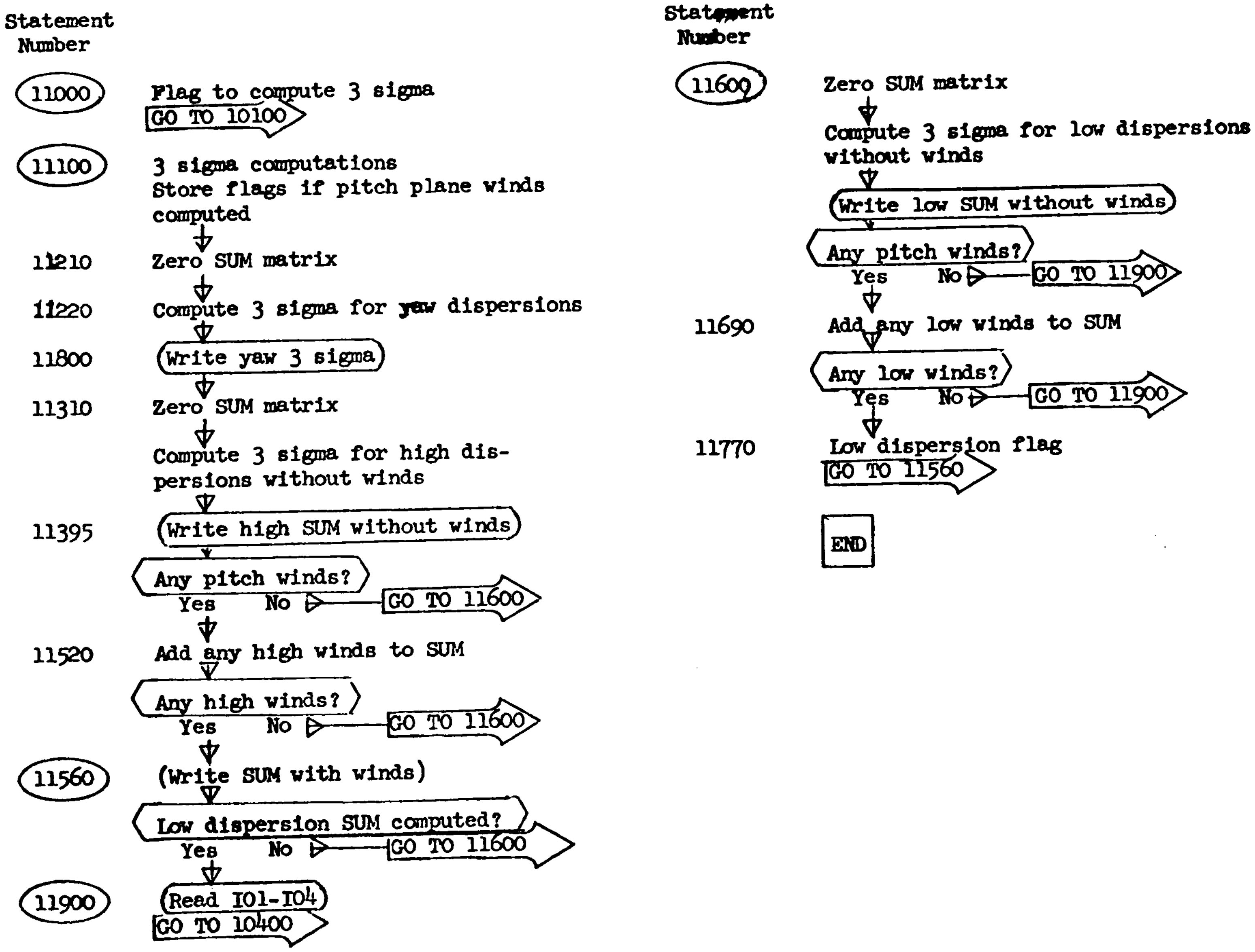
DISPRS SUBROUTINE FLOW CHART



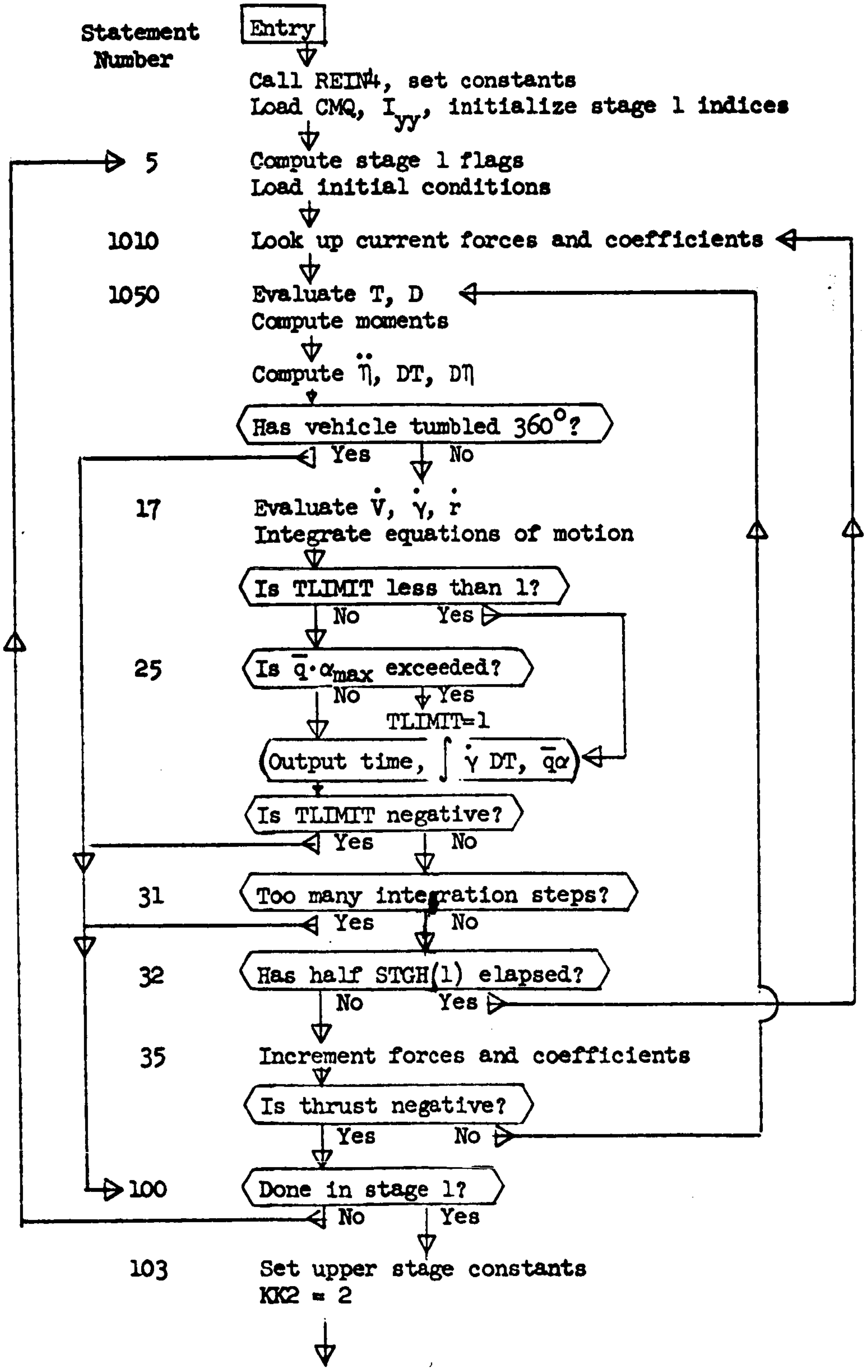
DISPRS SUBROUTINE FLOW CHART (Cont'd)
(includes SDISPR)



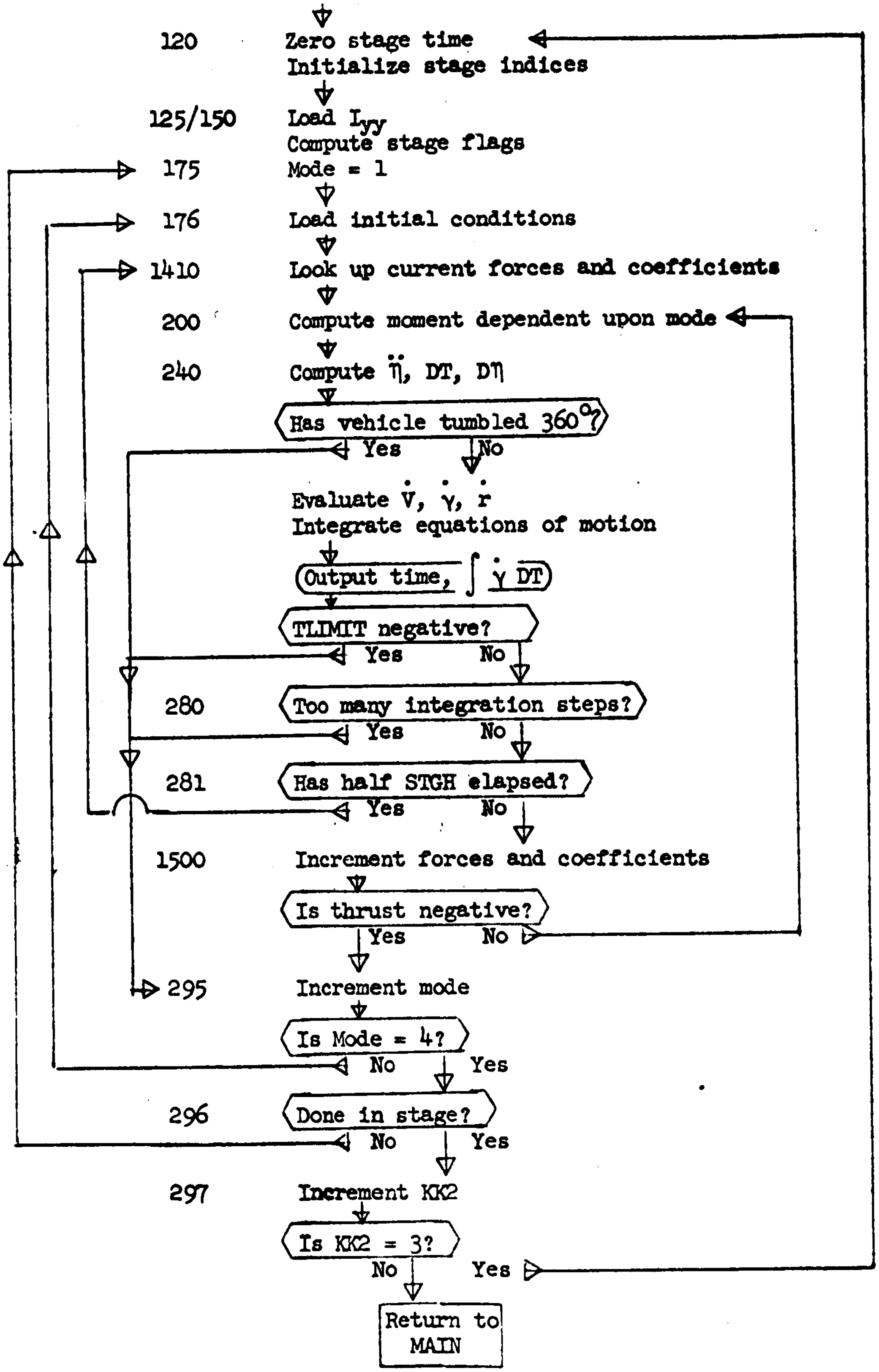
DISPRS SUBROUTINE FLOW CHART (Cont'd)
(includes SDISPR)



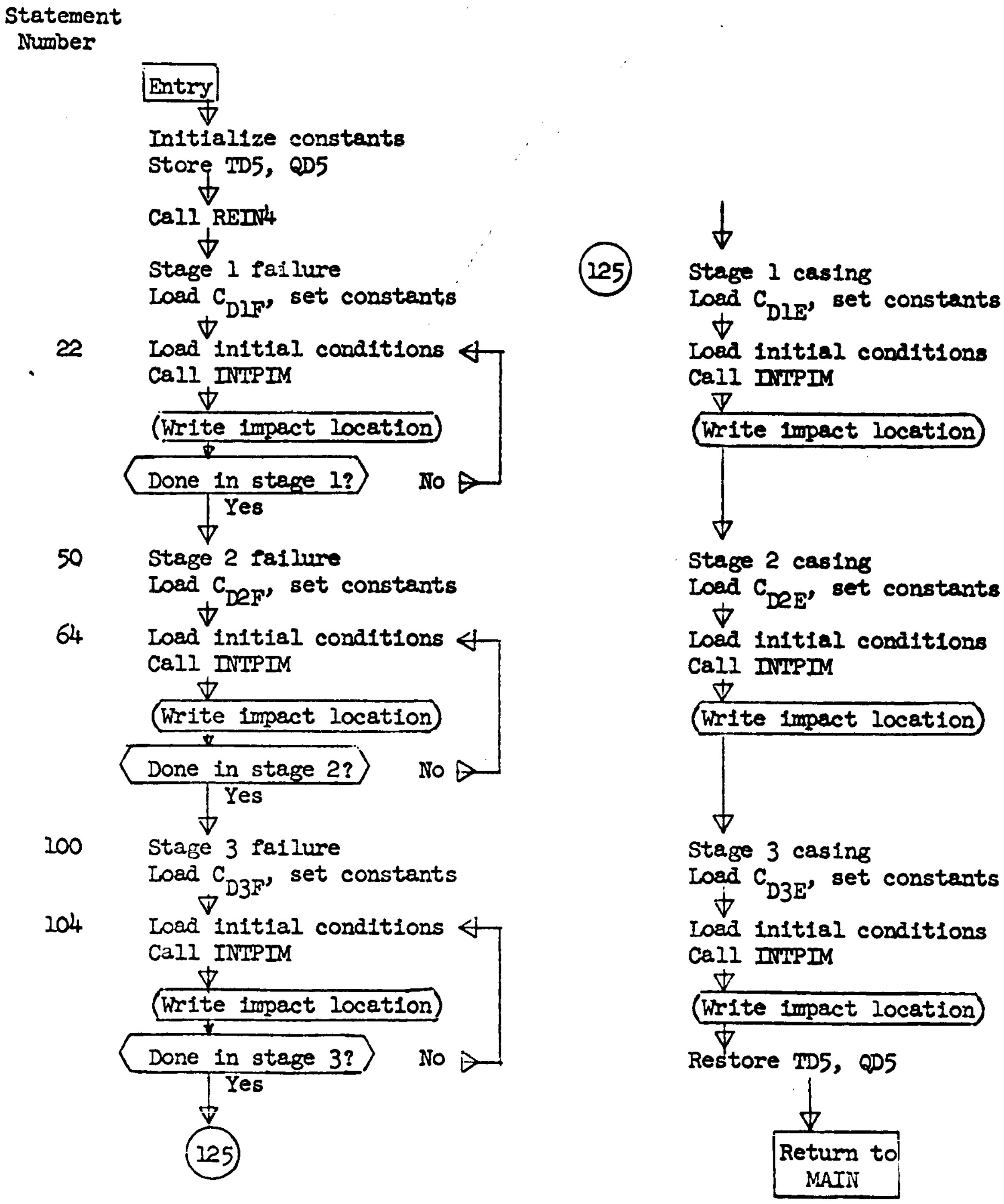
HRDOVR SUBROUTINE FLOW CHART



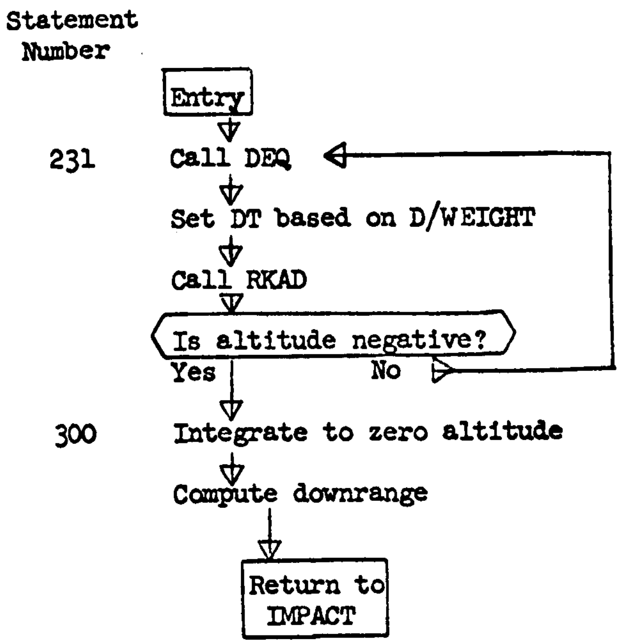
HRDOVR Subroutine Flow Chart (Cont'd)



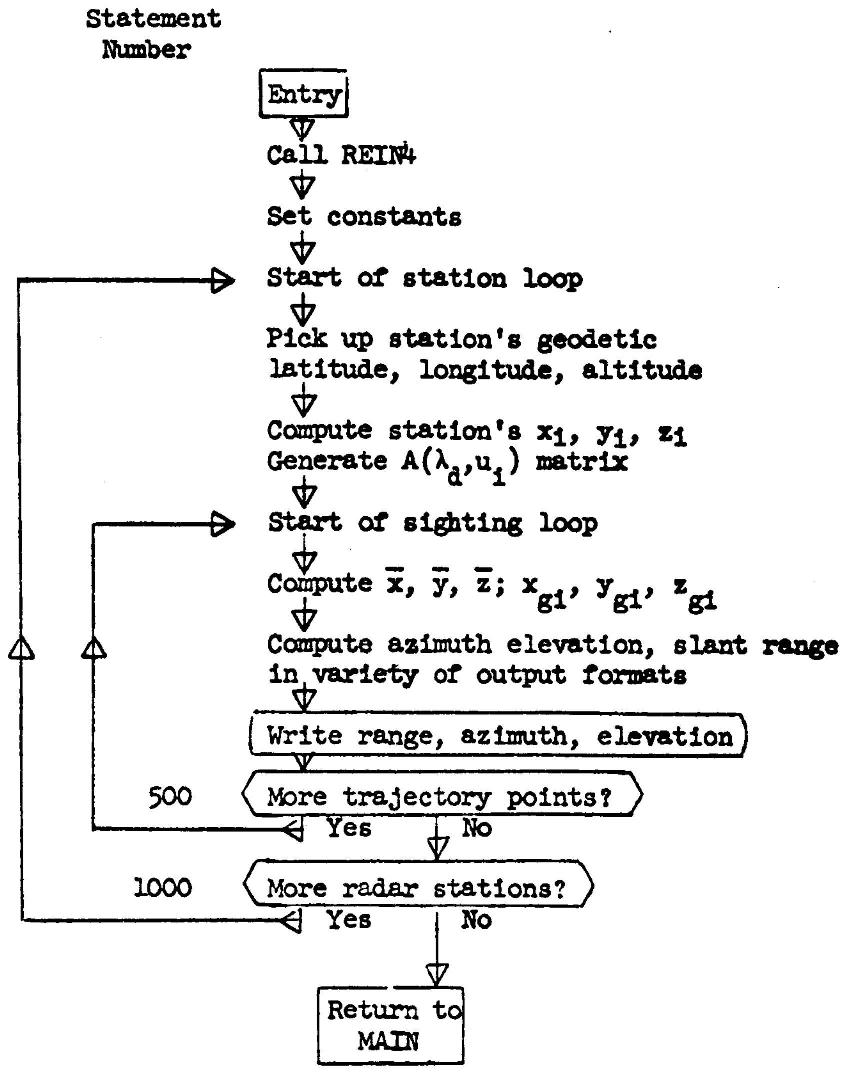
IMPACT SUBROUTINE FLOW CHART



INTPIM SUBROUTINE FLOW CHART



RADAR SUBROUTINE FLOW CHART



SUB-PROGRAM NAMES AND FUNCTIONS

1. MAIN Program

Set constants
Call REIN
Call OPTION
Compute integration intervals
Set up flags for adjustable parameters
Determine what type of trajectory is to be computed
Set up initial values for stage indices
Compute initial delt-mass desired
Increment launch and stage weights
Call ICS
Call TRAJ
Check on meeting terminal constraints
Add heating constraint
Check on success of optimization
Store characteristics of successful run
Call LINEAR
Call DISPRS
Call HRDOVR
Call IMPACT
Call RADAR

2. TRAJECTORY Subroutine

Initial entry

Set stage and bring in stage data
Compute vertical lift-off
Set output frequency (vs stage and type of trajectory)
Pick up first point on backward trajectory
Call ICS for intermediate constraints on backward trajectories
Store achieved values of intermediate constraints

Loop entry

Adjust matrices row limits for storing and picking up:
   trajectory variables
   capital lambda's or B's or D's
Call PCAL
Call INSTOP
Determine initial integration step size
Compute S matrix
Calculations at start and end of each linear segment
Test for end of stage and/or trajectory
Call MEQ
Return to loop entry or initial entry
Call MISCON at end of trajectory

3. REIN Subroutine

   Read input data
   Modify input thrust data
   Store THRUST and WEIGHT arrays

4. LINEAR Subroutine

   Compute linearized theta program

5. OPTION Subroutine

   Interrogate input options and set switches accordingly

7. MISCON (Mission Constraints) Subroutine

   Compute current value of stopping parameter
   Compute achieved values of terminal constraints

8. ICS Subroutine

   Assign initial conditions to trajectory and adjoint variables
   Call DEQ

9. PCAL (Preliminary Calculations) Subroutine

   Compute changes in pitch program
   Compute changes in adjustable parameters
   Check for switch between closed-loop and open-loop
   Compute vector EK
   Impose zero alpha constraint

10. ATMOS Subroutine

    Compute atmospheric density, pressure and speed of sound
    Compute partial derivatives re: atmosphere for adjoint equations

11. OBLATE Function

    Computes surface radius on oblate Earth

12. INSTOP Subroutine

    Call ICS
    Call INER
    Store first point when going forward
    Call OPCNL
    Call RKAD
    Pick up stored trajectory variables when integrating backward
    Stop integration when stopping parameter is reached
    Call MISCON
    Store T matrix for exchange ratios

13. RKAD Subroutine

    Logic for integration using both Runge-Kutta and Adams integration

14. DEQ Subroutine

    Calculate thrust and aerodynamic forces using ATMOS and LOOK
    Check whether trajectory equations and/or adjoint equations are required
    Compute time derivatives

15. MEQ (Matrix Equations) Subroutine

    Store trajectory variables
    Evaluate trajectory deviations
    Generate lambda and I matrices
    Store R and TT matrices for exchange ratios
    Compute A matrix
    Invert A matrix (call subroutine SYMVRT)
    Compute and output exchange ratios
    Compute B, D and C matrices
    Compute payload change for fuel error adjustment

16. SYMVRT Subroutine

    Invert symmetric matrix

17. LOOK Subroutine

    Linear interpolation table lookup using stored quotients

18. LOOKUP Subroutine

    Linear interpolation table lookup without stored quotients

19. OPCNL (Output Control) Subroutine

    Write output

20. INER Subroutine

    Compute inertial trajectory variables and downrange distance
    Store coordinates for RADAR

21. WIC Subroutine

    Write out input data

22. CMAT Subroutine

    Compute C matrix

23. CONTRL Subroutine

    Compute angles of attack

24. SIXD Subroutine

    Computes vehicles thrust attitude

SCOUT/PRESTO

Optional Computations - Subroutine Names and Functions

1. DEQD (Dispersion)

   Calculate thrust and atmospheric forces using ATMOS and LOOK.
   Call SIXDD.
   Compute time derivatives for trajectory equations.

2. DEQIM (Impact)

   Calculate drag force using ATMOS and LOOK.
   Compute time derivatives for trajectory equations.

3. DISPRS (Dispersion)

   Store staging point trajectory variables from nominal trajectory for reference.
   Read and load parameters for dispersed trajectory after storing nominal values.
   Call MAIND.
   Compute and store dispersions in staging point trajectory variables.
   Restore nominal values of dispersed parameters.
   Compute three sigma variations.

4. HRDOVER (Hardover)

   Call REIN4 and load first stage data.
   Pick up initial conditions from nominal trajectory.
   Evaluate thrust, atmospheric forces, and overturning moments.
   Compute and integrate time derivatives of trajectory variables and attitude angle.
   Write output and check for end of hardover turn.
   Repeat for all desired first stage points.

   The above logic is repeated for all three failure modes in both the second and third stages, excluding atmospheric effects.

5. ICSD (Dispersion)

   Initialize trajectory variables after lift-off.
   Calculate vertical coordinate matrices.

6. IMPACT (Impact)

Call REIN4.
Load failure mode drag curve.
Pick up initial conditions from nominal trajectory.
Call INTPIM.
Write impact output.

7. INTPIM (INSTOP for IMPACT)

Control integration step size.
Call DEQIM.
Call RCADIM.
Stop integration at zero altitude.

8. INSTPD (INSTOP for DISPRS)

Call ICS.
Call INER.
Store first trajectory point when regenerating nominal trajectory.
Call OPCNL.
Integrate forward using body dynamics; call RKAD, INER, DEQD, SIXDD.

9. MAIND (Dispersion)

Initialize indices and lift-off variables.
Call ICS.
Call INER.
Call OPCNL.
Call TRAJD.

10. MEQD (Dispersion)

Store nominal trajectory variables.

11. PCAID (Dispersion)

Control upper stage pitch control deadbands.
Call CMAT.
Call CONTRL.
Determine commanded theta.
Call DEQD.
Call SIXDD.

12. RADAR

Call REIN4.
Determine station's geocentric position and transformation matrix.
Compute look angles and slant range for nominal trajectory.
Write radar output.

13. REIN4

Read Hardover, Impact, and Radar input data.

14. RKADD (Dispersion)

Logic for integration using both Runge-Kutta and Adams integration.
Call DEQD.

15. RKADIM (Impact)

Logic for integration using both Runge-Kutta and Adams integration.
Call DEQIM.

16. SIXDD (Dispersion)

Computes vehicle attitude from commanded attitude when body dynamics are included.
Computes effects of wind dispersions.
Adds influence of first stage thrust misalignment dispersion.

17. TRAJD (Dispersion)

Initial Stage Entry

Set stage data and set integration to Runge-Kutta.
Set output frequency (use intermediate trajectory frequency).

Stage One:	lift-off calculation
Upper Stages:	store dispersed trajectory variables at stage ignition
Stage Two:	set dispersed yaw control deadband
Stage Three:	set dispersed yaw control deadband
Stage Four or Five:	set tip-off dispersions

Loop Stage Entry

Adjust matrix row limits for trajectory variable storage.
Call PCALD.
Call INSTPD.
Test for end of stage and/or trajectory.
Call MEQD.
Return to loop entry or initial entry of next stage.
Call MISCON at end of trajectory.
Store orbit elements for dispersed trajectories.

18. SQUEEZE

Output original optimization module affected values.
Test for unacceptable conditions.
For 5-stage vehicles: modify THRUST and WEIGHT arrays and indices.
store stage 4, 5 and 9 values of times, weights and thrust/weight table indices.
For all affected data, transform to 'original' dispersion module format (4 powered stages maximum)
For linearized nominal trajectory, store linearized θ program and spun stage θ's in TTH table.
Output 'squeezed' values for dispersion package.

SECTION 21

CODING NOMENCLATURE

CODING NOMENCLATURE

A3D	matrix of position and direction at present time
A92	A matrix at start of open loop computation
AA	A matrix
ACTN	answer from 4-quadrant arctan routine
ADD6D	terms added to $\dot{v}$, $\dot{\gamma}$, $\dot{\psi}$ from 6D simulation
ALIT	local speed of sound
ALPHA	resultant angle of attack from theta and chi
ALPHAE	right ascension of present position
AZW	force resolution angle in 6D
B	B matrix
B(70,2)	theta increment over integration step
BB3D	transformation of coordinates matrix BB'
BETA	in-plane range angle from ascending node
BETA6D	aerodynamic yaw angle of attack
BETAP	argument of perigee
BTU	total heating
C	C matrix
C3D	thrust resolution matrix C
CCHI	cosine of CHI
CD	total drag coefficient
CDLAM	cosine of DLAM
CGAM	cosine of flight path angle gamma X(2)
CGAMI	cosine of inertial flight path angle GAMI
CGSP	CGAM*SPSI
CHI	thrust yaw angle
CINV	transformation of coordinates matrix C^{-1}
CL	lift coefficient
CLAM	cosine of geocentric latitude X(5)
CLAMO	cosine of initial launch geocentric latitude
CLAMQ	$(\text{CLAM})^2$
CLCP	CLAM*CPSI

CLSP	CLAM*SPSI
CMALPH	data block 35
CMDELT	data block 36
CNDELT	data block 34
CPSI	cosine of azimuth X(4)
CPSII	cosine of inertial azimuth PSII
CPSIIQ	$(\text{CPSII})^2$
CRTIM	time at end of stage (or during linearization of pitch program, time at end of linear segment)
CT1	data block 10
CT2	data block 11
CT3	data block 12
CT4	data block 13
CT5	data block 14
CTAU	cosine of longitude X(6)
CTAUO	cosine of initial launch longitude TAUO
CTHETA	cosine of THETA
CTHETX	vector of function of transformation matrix C, THETA and CHI
CTCG	CTAU*CGAM
CTCX	CTHETA*CCHI
CTSG	CTAU*SGAM
CTSX	CTHETA*SCHI
D	drag
D1	first set of derivatives used in integration
D2	second set of derivatives used in integration
D3	third set of derivatives used in integration
D4	fourth set of derivatives used in integration
DA1	data block 16
DA2	data block 17
DA4	data block 19
DA5	data block 20
DA6	data block 21
DA7	data block 22
DA12	data block 27
DA13	data block 28

DA14	data block 29
DA6D	data block 33
DAH	derivative of local speed of sound/altitude
DALPHA	change to be made in control variable THETA
DATE	data block 3
DD	D matrix during optimization
DELTAY	increment in X's from integration routine
DELX	deviations from nominal trajectory
DETC	determinant of C3D matrix
DETLAM	geodetic latitude
DLAM	difference between geodetic and geocentric latitudes
DLAMO	geocentric latitude at launch
DLIN	matrix D in linearization process
DMASD	calculated initial payoff improvement
DMLIM	dump limit for floating point variables in common
DP2	estimate of integral of square of control deviations
DPDPSI	partial derivatives relating payoff to error in terminal constraints
DPH	derivative of pressure/altitude
DPSI	negative of error in terminal constraints on latest nominal trajectory
DPSIS	negative of error in terminal constraints on latest trajectory
DRANGE	downrange distance in nautical miles
DRH	derivative of density/altitude
DT	current integration interval
DTAU	changes to be made in adjustable parameters
DTOLD	integration interval from previous step
DWP	increment to be added to payload
DWPMAX	maximum change in payload at any one time

EH	angular momentum
EI	I matrix
EK	K vector
EL	L matrix in linearization of pitch program
ELIN	E matrix in linearization of pitch program
ELM92	λ matrix at start of open loop computation
ELMDA	λ matrix

EMA	matrix term of M matrix in linearization of ascent tilt program
EMB	matrix term of M matrix in linearization of ascent tilt program
EML	mass at end of closed-form lift-off
ENU	longitude position from node angle ν
ETA	ALPHA component in vertical plane
ETALIN	data block 26
EYE	orbit inclination
FJ	J matrix
FL	lift
FMU	gravity constant μ
FMUDEH	μ/EH
FWA	storage area of latest nominal trajectory
FWB	storage area of latest trajectory
G	local acceleration due to gravity
GAMI	inertial flight path angle
GAMW	force resolution angle in 6D
GEOCEN	matrix of geocentric vertical factors for transformation of coordinates
GEODET	matrix of geodetic vertical factors for transformation of coordinates
GG	exchange ratio matrix
GO	factor for converting weight to mass
H	altitude
HAERO	altitude above which aerodynamic computations are by-passed
HCAM	Mach number
HEAD1	data block 1
HEAD2	data block 2
HL	altitude at end of closed-form liftoff
HSTR	geopotential altitude
I	used as index locally
IB1	data block 6
IB3	data block 8
IB4	data block 9
IBL	vector of indexes used in REIN subroutine

IC	type of trajectory 1 = optimization 2 = guidance
INTG	RKAD integration package entry flag ... 1 = to start or restart integration using RUNGE KUTTA method A. Initial entry - set by user B. Subsequent entries - set by RKAD ... 2 = normal continuation of integration A. Set by RKAD

IS(n) SUBSCRIPTED INTEGER SIGNAL FLAGS

IS(1)	... Available
IS(2)	... Available
IS(3)	... 0 = zero density (drag = lift = 0) 1 = non-zero density
IS(4)	... 0 = zero drag 1 = non-zero drag
IS(5)	... 0 = zero lift 1 = non-zero lift
IS(6)	... Available
IS(7)	... 0 = correct flag 1 = wrong flag
IS(8)	... Available
IS(9)	... -1 = error exit from REIN 0 = proceed to case computations 1 = end of job
IS(10)	... 1 = to bypass adjoint equations 0 = to compute
IS(11)	... Available
IS(12)	... 0 = heating is not being constrained 1 = heating is being constrained
IS(13)	... 0 = do not set $\alpha = 0$ on this integration step 1 = this is the first integration step in stage 2; set $\alpha = 0$
IS(14)	... 0 = skip all exchange ratio computations 1 = end of a stage 2 = middle of a stage 3 = start of a stage
IS(15)	... 0 = unsuccessful forward trajectory 1 = successful forward trajectory

IS(16) ... 0 = DMASD has not been computed yet
1 = DMASD has been computed

IS(17) ... 0 = more forward runs are allowed
1 = no more forward runs allowed

IS(18) ... 0 = program is in closed-loop mode
1 = program is in open-loop mode

IS(19) ... 0 = time is not within $\alpha = 0$ region
1 = time is within $\alpha = 0$ region

IS(20) ... upper limit for page line count

IS(21) ... temporary value for KPOINL

IS(22) ... plus = T9 point count is greater than or equal to input value (after storage)

zero = T9 point count is less than input value

negative = T9 point count is greater than or equal to input value

IS(23) ... current output frequency

IS(24) ... current output count

IS(25) ... 0 = continue trajectory
1 = stop trajectory with non-success predicted

IS(26) ... storage for IS(10)

IS(27) ... 0 = normal outputs
1 = bypass initial outputs on initial stage

IS(28) ... 0 = bypass zero aerodynamic computations
1 = compute zero aerodynamic terms

IS(29) ... 0 = exchange ratios are not being computed
1 = exchange ratios are being computed

IS(30) ... 0 = do not store radar computations in FWA
1 = store radar computations in FWA

IS(31) ... 0 = stopping conditions not yet reached
1 = reached stopping condition

IS(32) ... 0 = failed to meet terminal conditions
1 = terminal conditions have been met

IS(33) ... -1 = call ICS from MAIN
0 = call ICS from INSTOP
1 = do not call ICS

IS(34) ... 0 = do not initialize X's for intermediate constraints in ICS
1 = call ICS to initialize X's for intermediate constraints

IS(35) ... 0 = do not store trajectory variables in FWB
1 = store trajectory variables in FWB

IS(36) ... = IS(5)

IS(37) ... storage counter for radar computations

IS(38) ... 0 = call MISCON - compute only stopping parameter
1 = call MISCON - compute all constraints

IS(39) ... 0 = linearization times not adjusted yet
1 = linearization times adjusted to match integration times

IS(40) ... -1 = backward linearization trajectory
0 = nonlinearization trajectory
1 = forward linearization trajectory

IS(41) ... -1 = constant terms in LINEAR have been computed
0 = new values of TLIN were read in for present case
1 = new values of TLIN not read in with present case

IS(42) ... 0 = do not evaluate linearization integrals
1 = evaluate linearization integrals

IS(43) ... 0 = no α constraint over this time interval
1 = α constraint applied

IS(44) ... 0 = do not apply $\dot{\theta}_{max}$ limit
1 = do apply limit

IS(45) ... 0 = no violation of pitch program limitations occurred during linearization process
1 = violation of pitch program limitations occurred during linearization process

IS(46) ... -1 = backward trajectory including body dynamics
0 = body dynamics not included
1 = forward trajectory including body dynamics

IS(47) ... 0 = do not include body dynamics in this stage
1 = include body dynamics in this stage

IS(48) ... 0 = MEQ is called from an intermediate point in a stage
1 = MEQ is called from the final point in a stage

IS(49) ... 0 = KPOINL has not yet been computed
1 = KPOINL has been computed

IS(50) ... 0 = normal integration in DEQ
1 = integration through lift-off for exchange ratios

IS(51) ... 0 = present stage uses continuous control θ
1 = present stage uses fixed θ

IS(52) ... 0 = present stage uses adjustable χ
1 = present stage uses fixed χ

IS(53) ... available

IS(54) ... 0 = stage 4 is spin stabilized
1 = stage 4 uses continuous control

IS(55) ... 0 = stage 5 is spin stabilized
1 = stage 5 uses continuous control

IS(56)	...	0 = do not ignore constraints 1, 3, 5 and 9 during linearization 1 = ignore constraints 1, 3, 5 and 9 during linearization
IS(57)	...	0 = do not consider pitch rate over coast following stages using continuous θ during linearization 1 = consider pitch rate over coast following stages using continuous θ during linearization 2 = stage 4 used continuous θ and stage 5 is spin stabilized
IS(58)	...	0 = stage 4 tipoff at stage 4 ignition on current case 1 = stage 5 tipoff at stage 6 ignition on current case
IS(59)	...	0 = present stage is not tipoff stage 1 = present stage is tipoff stage
IS(60)	...	0 = not currently in spin stabilized stage 1 = currently in spin stablized stage

ISTATN	matrix for radar station identification words from data block 41
ISTGE	data block 5
IT1	used as index or flag locally
IT2	used as index or flag locally
IT3	used as index or flag locally
IT4	used as index or flag locally
ITAP	(1) available (2) available (3) sequence number of $\bar{q}_2$ constraint
J	used as index locally
JC	number of constraints
JC1	number of constraints

TABLE INDICES

JCMALF	...	current linear segment index for tables CMALPH
JCMDLT	...	current linear segment index for table CMDELT
JCNDLT	...	current linear segment index for table CNDELT
JTD1	...	current linear segment index for table TD1
JTD2	...	current linear segment index for table TD2
JTD5	...	current linear segment index for table TD5

Table Indices, cont.

JTHRST	... current linear segment index for table THRUST
JTL1	... current linear segment index for table TL1
JTTH	... current linear segment index for table TTH
JWEIGH	... current linear segment index for table WEIGHT
JWNDAZ	... current linear segment index for table WINDAZ
JWNDVL	... current linear segment index for table WINDVL
JXCG1	... current linear segment index for table XCG1
JXCG2	... current linear segment index for table XCG2
JXCG3	... current linear segment index for table XCG3
JXCG4	... current linear segment index for table XCG4

K	used as index locally
KFLAG	flag(s) indicating use of adjustable parameter(s)
KK1	... current stage index
KK2	... current stage code 1 = powered flight stage 1 2 = powered flight stage 2 3 = powered flight stage 3 4 = powered flight stage 4 5 = powered flight stage 5 6 = coast stage 1 7 = coast stage 2 8 = coast stage 3 9 = coast stage 4 10 = coast stage 5
KK3	... trajectory type code 1 = initial forward trajectory 2 = intermediate forward trajectory 3 = intermediate forward trajectory 4 = last forward trajectory 5 = available 6 = backward trajectory
KK4	... current stage code 1 = powered flight stage 1 2 = powered flight stage 2 3 = powered flight stage 3 4 = powered flight stage 4 5 = powered flight stage 5 6 = coast stage 1 7 = coast stage 2 8 = coast stage 3 9 = coast stage 4 10 = coast stage 5

KK5 ... -1 = backward trajectory
+1 = forward trajectory

KK6 ... number of stages in the trajectory

KK7 ... forward trajectory ... KK7 = KK6
backward trajectory ... KK7 = 1

KK8 ... current index for selecting aerodynamic constants

KK9 ... storage index computed during initial forward run

KK10 ... staging constant = 1

KK11 ... type of forward trajectory

3 = first forward trajectory
2 = intermediate forward trajectory
1 = last forward trajectory

KK12 ... number of variables being interpreted

KK13 ... subscript of X for last Λ^2 being integrated

KK14 ... first time segment in stage 2 in linearized pitch program

KK15 ... first time segment in stage 3 in linearized pitch program

KK16 ... -1 = initial forward trajectory
0 = optimization run
1 = guidance run

KK17 ... 0 = maximum pitch rate not imposed
positive = point count at end of imposed maximum pitch rate

KK18 ... last time segment in present stage of a linearization run

KK19 ... return code

KK20 ... last stage

KK21 ... available

KK22 ... index for adjustable parameters

KK23 ... 0 = C matrix must be computed in DEQ
1 = C matrix is computed in PCAL before entering DEQ

KK24 ... point count at stage 2 ignition

KK25 ... KK25 = 5 - KK3

KK26 ... integration frequency for current stage when including body dynamics (equal to ST2(17))

KK27 ... counter for integration steps when including body dynamics

KK28 ... number of present case

KK29 ... number of derivatives needed for integration in linearization process

KK30	... 0 = evaluate derivatives in DEQ 1 = evaluate forces only in DEQ
KK31	... subscript of first linearization derivative
KK32	... current time segment during linearization
KK33	... last powered stage in current case
KK34	... last powered stage in previous case
KK35	... first time segment in stage 4 in linearized pitch program
KK36	... first time segment in stage 5 in linearized pitch program
KK37	... last linearized stage
KK38	... last TLIN stage code
KK39	... available

KKONE	value of KK1 at beginning of backward trajectory
KLIN	number of time segments used in linearization of pitch program
KPOINC	count of integration points from beginning of trajectory
KPOINL	integration point count to switch from closed to open loop
KPOINS	number of integration points on last successful forward run

L	used as index locally
L1	used as index locally
L2	used as index locally
L3	used as index locally
L4	used as index locally
LBD	limit on storage index for D matrix
LFWAB	limit on storage in FWA and FWB
LINE	current page output line count

M	used as index locally
MB1	upper index for storing in B matrix region
MB3	increment in MB1 per integration step
MF1	upper index for storing or picking up in FWB
MF2	lower index for storing or picking up in FWB
ML1	upper index for storing Λ matrix
ML3	increment in ML1

N	used as index locally
NB1	upper index for picking up from B matrix region
NB3	increment in NB1 per integration step
NF1	upper index for storing or picking up in FWA
NF2	lower index for storing or picking up in FWA
NF3	increment in NF1 and NF2 per integration step
NL1	upper limit for picking up Λ matrix
NL3	increment in NL1
NN	used as index locally
NPRINT	key for current type of printout in OPCNL
NSTAGE	vector with number of entries in tables of thrust and weight by stage
NTHRST	vector of subscripts of first entry for each stage in THRUST table
NWEIGH	vector of subscripts of first entry for each stage in WEIGHT table

OMEGA	earth rotation rate
OMEGA2	2 * OMEGA
OMEGAE	longitude of ascending node
OMEGAQ	(OMEGA)^2

P	local atmospheric pressure
PDA	partial derivative of drag/ALPHA
PDM	partial derivative of drag/MACH
PF	partial derivative of DV/dt
PG	partial derivative of $d\gamma/dt$
PH	partial derivative of $d\psi/dt$
PI	partial derivative of dr/dt
PJ	partial derivative of $d\lambda/dt$
PK	partial derivative of $d\tau/dt$
PLA	partial derivative of lift/ALPHA
PLM	partial derivative of lift/MACH
PO	sea level atmospheric pressure
PSII	azimuth of VI
PSMETH	flag in RKAD indicating (past) method integration for previous point

QBAR	dynamic pressure
QBARA	ALPHA * QBAR
QD1	slopes of linear segments in table TD1
QD2	slopes of linear segments in table TD2
QD5	slopes of linear segments in table TD5
QL1	slopes of linear segments in table TL1
QTH	slopes of linear segments in table TTH

R	exchange ratio matrix
RALPHA	orbit semi-major axis
RE	local radius of earth based on oblate earth computation
RHO	local atmospheric density
RP	orbit perigee radius
RSTR	gas constant for air

S	S matrix
SCHI	sine of CHI
SDLAM	sine of $\delta\lambda$
SETA	sine of ETA
SETAQ	$(\text{SETA})^2$
SG1	data block 39
SG2	data block 39
SG3	data block 39
SG4	data block 39
SG5	data block 39
SG6	data block 39
SGAM	sine of flight path angle γ
SGAMI	sine of inertial flight path angle GAMI
SLAM	sine of latitude λ
SLAMO	sine of launch latitude DLAMO
SLCP	SLAM * CPSI
SLCLCP	SLAM * CLAM * CPSI
SLOPET	slopes of linear segments in THRUST table
SLOPEW	slopes of linear segments in WEIGHT table

SLSP	SLAM * SPSI
SPHI	sine of roll angle PHI
SPSI	sine of azimuth ψ
SPSII	sine of inertial azimuth PSII
SSAM	current mass

ST1(n) CURRENT STAGE DATA

ST1(1) ... jettison weight

ST1(2) ... aerodynamic reference area

ST1(3) ... nozzle exit area

ST2(n) STORAGE

ST2(1) ... -1 = backward trajectory
1 = forward trajectory

ST2(2) ... counter for the number of forward trajectories

ST2(3) ... stored value of stopping parameter

ST2(4) ...

ST2(5) ... past DT

ST2(6) ... current DT

ST2(7) ... past time

ST2(8) ... time at upper end of current linear segment during linearization

ST2(9) ... time at end of imposed maximum pitch rate

ST2(10) ... value of θ at end of imposed maximum pitch rate (at ST2(9))

ST2(11) ... slope of θ program over current time interval (radians/sec)

ST2(12) ... time at beginning of current time segment in linearized pitch program

ST2(13) ... $\theta - \alpha$ at stage 2 ignition

ST2(14) ... pitch rate

ST2(15) ... flag for linearization with body dynamics

ST2(16) ... DWFUEL multiplicative factor

ST2(17) ... integration frequency for current stage when including body dynamics

ST2(18) ... stored payoff derivative on stage 2 $\bar{q}$

ST2(19) ... +1 constrain stage 2 $\bar{q}$; -1 eliminate $\bar{q}$ constraint

ST2(21) ... pitch rate slope on backward linearization run

ST2(22) ... DWPMAX

ST2(23) ... DA12(3)
ST2(24) ... DA12(4)
ST2(25) ... DA12(5)
ST2(26) ... payoff correction for constraint errors
ST2(27) ...
ST2(28) ...
ST2(29) ...
ST2(30) ... DA12(10)
ST2(31) ... DA12(11)
ST2(32) ... WBS(1)

ST2(33) ... WBS(2)

ST2(34) ... WBS(3)
ST2(35) ... WBS(4)
ST2(36) ... WBS(5)
ST2(37) ... linearization iteration number for output
ST2(38) ...
ST2(39) ...
ST2(40) ...

ST3(n) ... storage for terminal constraint computations containing achieved values of terminal constraints for n = 1,13

<u>ST4(n) STORAGE</u>

ST4(1) ... heating constraint term
ST4(2) ... heating constraint term
ST4(3) ... heating constraint term
ST4(4) ... heating constraint term
ST4(5) ... DA1(11)
ST4(6) ... DA1(13)
ST4(7) ... DA1(15)
ST4(8) ... DA1(17)
ST4(9) ... DA1(19)
ST4(10) ... available

ST5	storage for adjustable parameters
STAU	sine of longitude τ
STAUO	sine of launch longitude
STCG	STAU*CGAM
STGH	stage integration intervals
STHETA	sine of THETA
STSG	STAU*SGAM
T	thrust
TANAG	tangent of angle
TANN	numerator of TANAG
TAUO	launch longitude
TAUI	inertial longitude
TD1	data block 31
TD5	data block 31
TDELIN	vector of durations of time segments of linearized pitch program
TG	current time in Space Age Date
TG0	launch time in Space Age Date
THETA	thrust angle to horizontal θ
THETAC	commanded theta when including body dynamics
THETAL	linearized ascent tilt program values of θ
THRUST	table of thrusts for all stages
TIMBTU	heating rate
TIMCT	time from launch including coasts
TIMCTS	saved value of TIMCT
TIME	time from beginning of present stage
TIMEP	CT1(10)
TIMT	time from launch excluding coasts
TIMTS	saved value of TIMT
TITLE1	four word variable, page title for first line of output
TITLE2	six word variable, page title for second line of output
TITLES	21 words used for variable page titles TITLE1 and TITLE2
TL	time duration of closed form lift-off
TL1	data block 32
TLIN	data block 18

TØ	initial thrust for lift-off calculation
TT	matrix for exchange ratios
TTH	data block 40
TVL	thrust at end of lift-off calculations
TVØ	initial vacuum thrust
TWØE	twice the total energy of the orbit
VH	hyperbolic excess velocity
VI	inertial velocity
VI6	velocity components in trajectory integration coordinates
VJ6	velocity components in platform coordinates
VL	velocity at end of closed form lift-off
W	weight as measured at sea level
WBS	total weight in remaining stages (jettison weight and consumable weight)
WEIGHT	data block 38
WINDAZ	data block 54
WINDVL	data block 53
WPRØP	consumable weight for each stage
STTØGØ	total consumable weight in remaining stages

X(n) VARIABLES OF INTEGRATION

X(1)	= V aerodynamic velocity
X(2)	= γ aerodynamic flight-path angle
X(3)	= r radial distance from center of Earth to vehicle
X(4)	= ψ azimuth
X(5)	= λ geocentric latitude
X(6)	= τ longitude
X(i)	adjoint variables (i = 7, ILAST or KK12)
X(j)	integrals of functions of Λ for optimization of $\theta(t)$, optimization of stage 4 θ and χ, optimization of launch azimuth, or linearization of θ (j = ILAST + 1, $KK12_2$)
X(k)	exchange ratio R integrals (k = 202, $KK12_3$)

STORAGE ORDER FOR D1, D2, D3, D4, X, and XI ARRAYS

TRAJECTORY VARIABLES

Subscript	Value	
1	v	**velocity**
2	γ	**flight path angle**
3	r	**radial distance**
4	ψ	**azimuth**
5	λ	**geocentric latitude**
6	τ	**longitude**

ADJOINT VARIABLES

Subscript	Value	
7	λ_{v1}	
8	$\lambda_{\gamma 1}$	
9	λ_{r1}	
10	$\lambda_{\psi 1}$	
11	$\lambda_{\lambda 1}$	
12	$\lambda_{\tau 1}$	
13	λ_{n1}	
14	λ_{v2}	or for 1 constraint Λ^2_{11}
15	$\lambda_{\gamma 2}$	
16	λ_{r2}	
17	$\lambda_{\psi 2}$	
18	$\lambda_{\lambda 2}$	
19	$\lambda_{\tau 2}$	
20	λ_{m2}	
21	λ_{V3}	or for 2 constraints Λ^2_{11}
22	$\lambda_{\gamma 3}$	or " " " Λ^2_{21}
23	λ_{r3}	or " " " Λ^2_{22}
24	$\lambda_{\psi 3}$	
25	$\lambda_{\lambda 3}$	
26	$\lambda_{\tau 3}$	
27	λ_{m3}	
28	λ_{v4}	or for 3 constraints Λ^2_{11}

Subscript	Value		
29	$\lambda_{\gamma 4}$	or for 3 constraints Λ^2_{21}	
30	λ_{r4}	or for 3 constraints Λ^2_{22}	
31	$\lambda_{\psi 4}$	or for 3 constraints Λ^2_{31}	
32	$\lambda_{\lambda 4}$	or for 3 constraints Λ^2_{32}	
33	$\lambda_{\tau 4}$	or for 3 constraints Λ^2_{33}	
34	λ_{m4}		
35	λ_{v5}	or for 4 constraints Λ^2_{11}	
36	$\lambda_{\gamma 5}$	or for 4 constraints Λ^2_{21}	
37	λ_{r5}	or for 4 constraints Λ^2_{22}	
38	$\lambda_{\psi 5}$	or for 4 constraints Λ^2_{31}	
39	$\lambda_{\lambda 5}$	or for 4 constraints Λ^2_{32}	
40	$\lambda_{\tau 5}$	or for 4 constraints Λ^2_{33}	
41	λ_{m5}	or for 4 constraints Λ^2_{41}	
42	λ_{v6}	or for 4 constraints Λ^2_{42}	or for 5 constraints Λ^2_{11}
43	$\lambda_{\gamma 6}$	or for 4 constraints Λ^2_{43}	or for 5 constraints Λ^2_{21}
44	λ_{r6}	or for 4 constraints Λ^2_{44}	or for 5 constraints Λ^2_{22}
45	$\lambda_{\psi 6}$	or for 5 constraints Λ^2_{31}	
46	$\lambda_{\lambda 6}$	or for 5 constraints Λ^2_{32}	

Subscript	Value
47	λ_{T6} or for 5 constraints Λ^{2}_{33}
48	λ_{m6} or for 5 constraints Λ^{2}_{41}
49-188	Subscripts and values in same manner up to subscript 188 for all 13 possible constraints

XCG1	data block 37
XCG2	data block 37
XCG3	data block 37
XCC4	data block 37
XI	temporary storage of X(n) in RKAD
XISP	vacuum specific impulse for each stage
XR	matrix of values of exchange ratios by stage
Y	available locally for storage or temporary terms
YY	available locally for use
ZETA	true anomaly of position

DISPERSION CODING NOMENCLATURE

Variables in COMMON that have been redefined, and new variables defined for DISPRS (including SDISPR and DISOUT) and SIXDD.

AA	Temporary storage of staging point and final burnout trajectory variables, orbit elements, and certain quantities from nominal trajectory
AEROVL	Velocity relative to air mass
CHANGE	Dispersed parameter variation
DATE	Dispersion title
D4(20-74)	Nominal theta storage
FWA	Renamed GIANT, SUM, FWA
FWB	Temporary storage for nominal vehicle characteristics
GIANT	Storage for the dispersions in trajectory variables and orbit elements, trajectory titles, and nominal trajectory variables and orbit elements
IDSP(I)	I = 1 thrust misalignment dispersion I = 2 stage 2 control system deadband dispersion I = 3 stage 3 control system deadband dispersion I = 4 stage 4 tipoff dispersion I = 5 wind dispersed trajectory IDSP(I) = - yaw dispersion 0 no dispersion + pitch dispersion (high/low)
IP(21)	= -1 pitch plane dispersion = -2 yaw plane dispersion
IS40	Storage for nominal IS(40)
I4	Local variable
JACK	Storage for nominal NSTAGE, NTHRST, and NWEIGH
JILL	Code for type of dispersion trajectory being processed. See data block 46.
JILL20	JILL
JILL40	JILL-20
JWNDAZ	Wind azimuth table index
JWNDVL	Wind velocity table index
KLIN1	Number of entries in TLIN and THETAL arrays
MGOOSE	Sequential storage of JILL codes called and flag indicating pitch or yaw dispersion

MGOOSE (MGROW, 1) = JILL
MGOOSE (MGROW, 2) = 0 pitch dispersion
= -1 yaw dispersion

MGROW	Counter for total number of dispersed trajectories which have been requested at any time
MGROWK	Dispersion array output index
NPRNT	Index for output of input data
NSTGE	Stage in which dispersion parameter occurs
NSTORE	= -1 nominal parameters have not been stored in FWB for thrust dispersion trajectory = 0 nominal parameters have been stored
N1	Local index
N2	Local index
STIME	Used in loading linear theta into theta array
SUM	Storage for three sigma computation
TWOE	AEROVL during trajectory computation
WCHANG	CHANGE in input units
WINDAZ	Wind azimuth table
WINDVL	Wind velocity table
JIL785	Flag for dispersion code 7 or 8, vacuum thrust dispersions, for a 5-stage vehicle = 0 No = 1 Yes
NSTGTW(17)	Nominal first time point of coast 9 in WEIGHT table
(18)	Nominal first time point of stage 5 in WEIGHT table
(19)	Nominal first time point of coast 9 in THRUST table
(20)	Nominal first time point of stage 5 in THRUST table
(21)	Current first time point of coast 9 in WEIGHT table
(22)	Current first time point of stage 5 in WEIGHT table
(23)	Current first time point of coast 9 in THRUST table
(24)	Current first time point of stage 5 in THRUST table
STGTW(1)	Stage 4 duration nominal value
(2)	Coast 9 duration nominal value
(3)	Stage 5 duration nominal value
(4)	Number of integration points in stage 4
(5)	Number of integration points in coast 9
(6)	Number of integration points in stage 5
(7)	Δt integration step size for stage 4
(8)	Δt integration step size for coast 9
(9)	Δt integration step size for stage 5
(10)	Stage 4 jettison weight nominal value
(13)	Stage 5 jettison weight nominal value
(17)	Δt of current stage 4
(18)	Δt of current coast 9
(19)	Δt of current stage 5
(20)	Current stage 4 duration
(21)	Current coast 9 duration
(22)	Current stage 5 duration

HARDOVER CODING NOMENCLATURE

Variables in COMMON that have been redefined as well as new variables defined for HRDOVR.

ARMTV	Stage 1 thrust misalignment and pitch damping moment arm
CALPHA	Cosine of angle of attack
CON	57.295... radian to degree conversion
D1	Hardover first derivatives for integration
D2	I_{yy} = pitch axis moment of inertia
D2DOT	I_{yy} time derivative
D3	I_{yy} quotient table
D4	See data block 45
DA6D	See data block 33
DELTAY	CMQ quotient table
DEMAX	Maximum allowable attitude angle step
DETA	$\delta\eta$ = attitude angle step
DTMAX	Maximum allowable integration time step
DTMIN	Minimum allowable integration time step
EDOTDT	$\ddot{\eta}$ = angular acceleration about pitch axis
EDOT	$\dot{\eta}$ = angular velocity about pitch axis
ETA	η = attitude angle
FORCEN	Force normal to vehicle longitudinal axis
GAMOID	γ_o = flight path angle at start of hardover turn
ICKY	Access frequency for initial conditions
INK	FWB subscript for first element of current point set
INX	Relative counter for FWB points
INXLIM	INX limit
IRISH	Maximum number of integration steps allowed per turn
IRK	IRISH
IXPTR	Index for current FWB point set
JTD2	CMQ segment index
JTD5	I_{yy} segment index
LINECT	Output line counter
MODE	Stage 2 or 3 failure mode index
OPFLAG	Output time counter

QBRAMX	Integration stops one second after $\bar{q} \cdot \alpha_{max}$
SALPHA	Sine of angle of attack
SAMDOT	Mass time derivative
SGELD	Sine γ
SUMDN	Σ DETA
SUMDT	Σ DT, used as flag in evaluating forces
TIMEHD	Time since hardover turn started
TLIMIT	Time remaining in hardover turn
X(9)	Integrated change in flight path angle
XI	CMQ = pitch damping coefficient
XMACG	Moment due to angle of attack, per radian
XMCMQ	Moment due to pitch damping
XMDAMP	Total damping moment
XMDCG	Moment due to control surface deflection, per radian deflection
XMTM	Moment due to thrust misalignemtn or control jet
XTIMCT	TIMCT at stage ignition
Z	Local variable

IMPACT CODING NOMENCLATURE

Variables in COMMON that have been redefined and new variables defined for IMPACT, INTPIM, and RKADIM.

CON	57.295..., radian to degree conversion
DD	Drag curve storage
ICKY	Access frequency to FWB storage
INK	Local index for FWB trajectory variable set
INSTNT	Stage indicator
INX	Index for FWB storage
IP(23)	Impact failure mode flag = -1 thrust failure = -4 expended casings = -10 expended casings at terminal velocity
ISIZE	Index for DT branch
IVY	Number of FWB sets for current stage
IXPTR	Points to current starting values in FWB
LINECT	Output line counter
MTERMG	Terminal velocity flag
QD5	Current impact drag curve quotient table
STIMCT	Storage for TIMCT
TD5	Current impact drag curve
VCOEF	Terminal velocity coefficient
VOLD	Old velocity for reference
VTERM	Terminal velocity
XST1	Storage for STGH(5)
XST2	Temporary output storage
XST3	Temporary output storage
XST4	Temporary output storage
XTIMCT	Current stage base time

RADAR CODING NOMENCLATURE

New variables defined for RADAR in addition to redefined variables.

A11, A12, A13 A21, A22, A23 A31, A32, A33	$A(\lambda_d, u_i)$ transformation matrix
AZMILS	Azimuth angle, mills
AZMIN	Azimuth angle, minutes
CLAT	Cosine of station's geodetic latitude
CON	0.017453293, degree to radian conversion
CU	Cosine of station's longitude
DLAD	Station's geodetic latitude
DLOD	Station's longitude
ELMILS	Elevation angle, mills
ELMIN	Elevation angle, minutes
FWA	Booster's TIMCT, x, y, z history
IAZD	Integer azimuth angle
IELD	Integer elevation angle
INK	Number of stored trajectory point sets
ISLE	Number of radar stations
ISTATN	Station identification
LINECT	Output line counter
RADT	Radar output time
RA	Right ascension of station's position
SLAT	Sine of station's geodetic latitude
SRRED	Slant range reduced
SR	Slant range
STCORD	Station input coordinates
STCORD	Station input coordinates
SU	Sine of station's longitude
XALT	Station's altitude
XBAR	Geocentric coordinate difference
XGI	Topocentric geodetic coordinate
XOB	Flattening factor for oblateness
XRE	Equatorial radius
XYST	Station's XY plane projection
YBAR	Geocentric coordinate difference
YGI	Topocentric geodetic coordinate

ZBAR	Geocentric coordinate difference
ZGI	Topocentric geodetic coordinate
ZI	Station's Z geocentric coordinate
Z	Local variable

PART III

USERS' MANUAL

SECTION 22

SCOUT PROGRAM OPERATION

PROGRAM OPERATION

In the Scout program, several different types of trajectory calculations are available to the user at his option. This section provides instructions on data input and interpretation of the program output including, initially, a discussion of the sequence of trajectory computations.

Sequence of Trajectory Iterations During Optimization

Trajectory computations used in this program fall first into the categories of either "forward" or "backward." On a forward trajectory, time proceeds positively and on a backward trajectory, negatively. On a forward trajectory only the equations of motion are integrated; on a backward trajectory the adjoint differential equations are solved. The next major categories are "guidance" or "optimization." On a guidance trajectory one is concerned only with meeting terminal constraints on the trajectory variables; in optimization improvement of the payoff parameter is attempted as well.

Since solutions of the adjoint equations are used differrently for forward guidance versus optimization runs, a forward guidance run must be preceded by a backward guidance run. Similarly, a forward optimization run is preceded by a backward optimization run. Details of the differences between these are defined in Section 9.2. The remaining major categories are "successful" or "unsuccessful," as judged at the end of each forward trajectory. A successful forward guidance trajectory satisfies terminal constraints within acceptable limits, and a successful forward optimization achieves some increase in the payoff parameter as well.

The sequence of trajectory iterations starts with an initial nominal. This is simply an integration of the equations of motion using an input thrust-direction history. This nominal trajectory is then used as the basis for a backward guidance trajectory, which is followed by a forward guidance run. It is always assumed that a forward guidance trajectory represents an improvement over the previous nominal. Thus, each forward guidance trajectory becomes the new nominal and the basis for a new solution of the adjoint equations. Backward and forward guidance runs are continued until one is judged "successful." At this point a backward optimization run is made and the magnitude of the initial payoff improvement attempt is computed (see Section 5.7). A forward optimization run is then made. If it is successful, it is used as the new nominal and backward and forward optimization trajectories are computed, etc. If it is unsuccessful, that trajectory is discarded and another forward optimization is computed with half of the previous attempt at payoff improvement specified, etc. Thus, a succession of successful and unsuccessful optimization runs are computed either until the attempted payoff improvement is smaller than a specified magnitude or until the count of forward trajectories is within one of a specified limit. In either case, the sequence is then ended with a backward and a final forward guidance trajectory which represents the optimum path under the assumption of point mass motion.

Linearization of Pitch Program Option

The next step is the automatic linearization of the optimum pitch program. This is accomplished as a sequence of backward and forward guidance trajectories wherein the θ history is linearized with minimum-integral-square change in θ subject to meeting all trajectory constraints and rate limitations on the pitch program. This is done first for point mass motion and will generally require one to three iterations. When all constraints are satisfied, the linearization will be repeated, by option, with the simulated body dynamics effects added to the equations of motion as a final check on the trajectory.

Dispersed Trajectories Option

The next computation in sequence is the evaluation of trajectory dispersions due to various vehicle and environmental anomalies. This is accomplished through a series of forward trajectories with each of the dispersion sources considered in turn. After the last of these a summary statement is output of all the individual dispersions; worst-on-worst summations of high, low, and side dispersions; and RSS combinations of these numbers.

Hardover Turns Option

Failure mode turns are computed next, where failure is considered at specified intervals along the nominal trajectory.

DATA INPUT FORMAT

Data Blocks

Data required for program execution are grouped and input in data blocks which are identified by number. Each block contains a common type of information such as the case title or a stage thrust table, and utilizes a specified FORTRAN format for the entire data block. The contents of each data block are discussed and then summarized on the next several pages.

Header Cards

Input data are punched on cards which are placed behind the program binary deck. Cards for each data block are preceded and identified by a "header" card. The format of the header cards is 4I3, where the first field is the data block number and the second and third fields give the (inclusive) locations within the data block that the subsequent data words are to be placed. Thus, if the user wished to change constants 2, 3 and 4 in data block 10, the header card would be punched _ 1 0 _ _ 2 _ _ 4 _ _ _ , and the first three fields of the next card would contain the three constants. The fourth field of the header card is used to identify the vehicle stage number for the data, where necessary. The stage designation should be included only for the data blocks as specified on the following pages.

Card Sequencing

Cards within each data block must follow sequentially. Further, groups of data blocks must follow in correct order, as follows. All data for trajectory optimization and pitch program linearization (including 6D simulation in stage one) appear in data blocks 1 through 40 and must be first in sequence. A blank card then follows. Next in sequence are the groups of cards for dispersion, hardover turns, impact locus, and radar, in that order. A blank card follows each group. Data must be input for a given group only if the respective option has been selected. With the exception of the data for the dispersed trajectories, the data blocks within each group may appear in any order.

Nominal Impact Locus Option

Next, a series of ballistic trajectories are computed from specified time points along the final trajectory to impact in order to define the expected impact of the normally expended stages and the locus of possible impact in case of failure during burning or failure to ignite.

Radar Tracking Data Option

The final type of calculations performed is the history of radar tracking coordinates for up to twenty stations tracking the final trajectory.

Successive Cases

Successive cases can be run with a minimum of additional data input. For optimization and linearization, only the changes in data from the preceding case must be input. For all other computations the data must be repeated. A blank card must follow the data for each case.

999 Card

A card with 999 punched in the first three columns must end the data deck. It is to be placed behind the blank card which ends the data for the final case. When the READ routine encounters the 999 card, a normal stop is indicated to the machine operator, so that the job can be terminated.

DISCUSSION OF THE DATA BLOCKS

This discussion is a supplement to the detailed instructions beginning on page 22-18.

Data Blocks 1, 2, 3

Alpha-numeric comment cards that are output at the top of each page. Data block 1 is the first line; data blocks 2 and 3 appear on the second line.

Data Block 4

Basic computation options are as follows:

(1) Selection of payoff function influences the required sequence of constraint parameters (data block 8) and the units of the performance exchange ratios (see page 22-31).

(2) If heating constraint is to be imposed, see Section 13.

(3) If selected, performance exchange ratios are computed at end of optimization. See Section 8.

(4) Not used.

(5) Not used.

(6) Select 0 if running only to obtain an optimum trajectory with no requirement for exact definition of trajectory. Normal sequence of trajectory iterations would call for inclusion of body dynamics effects following optimization and point mass linearization, for which set = 1. This option can be set = 2 if only one forward trajectory is specified in Data Block 28.

(7) Linearization of the θ history at completion of optimization will be done if set = 1. Requires input of times of beginning of each linear segment in Data Block 18. If IP(6) = 1, then IP(7) must = 1.

(8) Not used.

(9) Specification of spin/no-spin in stages 4 and 5. Set = 3 only when there are five stages. Otherwise, 1 or 2.

(10) Yaw angle in stages 4 and 5 coupled or independent. When 4 and 5 are spun, set = 1.

(11) In order to minimize the $\bar{q} \cdot \alpha$ history over the final design trajec-

tory, the user can impose a zero-alpha constraint over any portion of stage 1 by this option. The time interval is input in Data Block 22. In general, the beginning time should allow for the initial pitch over maneuver. Suggest start and stop times of approximately 15. and 60. seconds. This constraint is applied only during optimization.

(12) By selection of this option, the pitch angle θ will be adjusted to produce a zero angle of attack at ignition of stage 2. This will be accomplished during linearization as well as optimization. No further input is required.

(13) Select type of $\overline{q}$ constraint at stage 2 ignition.

(14) Not used.

(15) For convenience, primarily in checkout, the floating-point numbers in common have been grouped and can be dumped in the floating-point format when requesting a dump at the end of the job.

(16) Not used.

(17) Selection of this option provides a three-page output of all the input data for each case. The data are identified by data block number and a few descriptive words.

(18) Not used.

(19) Not used.

(20) Not used.

(21) Data for running dispersed trajectories includes that normally required for running one forward trajectory with 6D, plus data for stages 2 and 3 in Data Blocks 33 and 37, plus that described in the discussion of Data Blocks 46 through 54.

(22) See discussion for input to Data Blocks 44 and 45.

(23) See discussion for input to Data Blocks 42 and 43.

(24) See discussion for input to Data Block 41.

Data Block 5

The input stage sequence must always be _ _ 1 _ _ 6 _ _ 2 _ _ 7 _ _ 3 _ _ 8 _ _ 4 _ _ 9 _ _ 5 _ 1 0 _ _ 0 except that the sequence can be terminated with the final zero after any powered or coast stage. That is, the sequence must always begin with 1, end with 0, and have the stages appear in the above order.

Data Block 6

For each adjustable parameter selected here, a weighting constant must be input in Data Block 27. If coasts are driven to zero during optimization, they will automatically be eliminated as adjustable parameters. When the sensitivity coefficients reveal a possible payoff improvement with any or all increased coasts, the coasts will automatically be adjusted.

Data Block 8

Specification of trajectory constraint parameters. In listing the parameters, the sequence is important. The first word is the total number of trajectory constraints, including the payoff parameter but not including either the heating or α constraints in the count. The second word must be the payoff parameter. Next are listed the terminal constraint parameters, starting with altitude if it is to be constrained but is not the payoff. When mass is not being optimized it is not included as a constraint. Then, the constraints to be imposed at stage points are listed in reverse chronological sequence (starting from the end of the trajectory). Finally, the heating constraint code (14) is listed if the constraint is to be imposed. The constraints at stage points are coded by the number of the stage at the end of which the constraint is to be applied. Understand that a constraint can be imposed at the beginning of a powered stage by coding it for the end of the preceding coast stage.

Data Block 9

Frequence of output points. There is generally no need to output every computed point. In fact, from an economy standpoint, there is every need to minimize the output, since that operation is relatively slow. Output of nearly every computed point is often desirable on the final (optimum) trajectory, but can be almost entirely eliminated on the previous iterations. The user also has the ability to vary the output frequency between stages if, for example, a more frequent output is desired during the atmospheric phase of boost than is required for the upper stages. Regardless of the output frequency, the initial and final points of each stage are always output. There is no output of the trajectory variables on the backward runs. Linearizations are treated as final trajectories and dispersion runs as intermediate trajectories.

Data Block 10

Nominal constants. Of particular note is the quantity HAERO, the altitude above which all aerodynamic computations are bypassed regardless of the aerodynamics option selection. This logic is in the interest of computation speed. The nominal altitude of 250,000 feet for this cutoff generally corresponds to a dynamic pressure of less than 1.0 lb/ft^2. All the constants in this data block are a part of the program binary deck and must be input as data only if a change is desired.

Data Blocks 11, 12, 13, 14

Constants for ATMOSphere subroutine. The 1962 ARDC model constitutes the nominal atmosphere as described in detail in Section 19. The various constants used in each exponential segment can be changed here with data input.

Data Block 16

Control of stage times and number of integration steps. In order to gain maximum speed in integration, a constant time increment is used in each stage. The user specifies this integration frequency by inputing the stage duration and the number of integration steps desired for the stage. This information must be input for stages 1 through 10 if they have been specified in the stage sequence. The total number of integration steps in stages 1, 2, 3, 4 and 5 must not exceed 168. The total number of integration steps in stages 1, 2, 3, 4, 5 and 6 should not exceed 177. There must be at least 25 steps in stage 1. Care should be taken to limit an integration step to cover no greater than about 400 ft/sec for the powered stages and no greater than 10 to 15 seconds during the coasts on the final trajectory.

Data Block 17

Various input data. Words 5, 6 and 9,10 are the nominal attitudes in stages 4 and 5. Word 7 is used in the heating computation. Word 8 is the time before lift-off at which the control system gyros are uncaged. Word 15 is a weighting factor which should be set = 1.0 for SCOUT payload maximization cases. When the total vehicle launch weight is much greater than SCOUT, at about 39,000 pounds, the weighting constant should be decreased when optimizing mass. Word 16 is a limit on the adjustment in coast time on any one iteration and must be input if any coast is used as an adjustable parameter. This number should not exceed 100 seconds.

Data Block 18

Times at which the pitch rate is changed. The linearization routine requires times at the end of the vertical lift-off period, at stage 1 burnout, and at ignition and burnout of all non-spun powered stages. Additional times may be specified in the middle of these stages. The first word input is the number of linear segments. This count must not exceed 15, does not include the vertical lift-off nor any maneuver after burnout of the final non-spun stage, but does include one segment each for intermediate coast stages. When proceeding into dispersion calculations the number of linear segments must not exceed 13. Next follows the list of times, coded by powered stage number and seconds from stage ignition. The first digit is the stage number and the remaining decimal number the time. Thus, the list of times for stages 1, 6, 2 would be 103.0, 182.6, 200.0, 239.2 for one segment in each stage, a lift-off time of 3 seconds, and stage 1 and 2 burn times of 82.6 and 39.2, respectively. The complete list of times must be re-input if any change is desired.

Data Block 19

Desired magnitudes of the trajectory constraint parameters that were specified in Data Block 8. Note that since the list of constraints can appear in any order the magnitudes must be in pure units; hence, all angles in radians. Range has units of nautical miles. Heating is in Btu/ft^2.

Data Block 20

The second word in this data block is the launch date, expressed in decimal fractions of days since 0.0 hours, January 1960. Expressed another way, it is the Julian date minus 2,436,934.5 days. It must be input whenever the terminal constraints involve an inertial longitude reference.

Data Block 21

Initial conditions on the trajectory variables are input in the rotating frame. Velocity must be zero, longitude in degrees ± from the prime meridian, latitude in geocentric degrees ± from the equator, azimuth degrees east of north, flight path angle must be 90 degrees.

Data Block 22

Magnitudes for control variable constraints. Times to initiate and to release $\alpha = 0.$ constraint are measured in seconds from launch.

Data Block 26

These weighting constants affect the amount of change in θ over each segment during linearization. Nominal values are automatically set into the program. However, the user should input 5.0 on segment 1 and 2.5 on segment 2 to retain the optimum initial pitchover.

Data Block 27

Weighting constants for optimization parameters. For numerical reasons, it is necessary to use non-unity weighting constants for some of the adjustable parameters. The recommended weighting constant for spun stage θ and χ is 10. For coast stages use 10^{-4}, and for launch azimuth and date use 2.0 and 10^{-6}. Smaller numbers will produce larger adjustments on any one iteration. The weighting constants must be input for all specified adjustable parameters.

Data Block 28

Convergence data. The first word is the initial payoff improvement to be used if the automatic computation of this number fails. Five percent of the expected final value would be a reasonable number. The second word is the "epsilon" or allowable proximity to optimum for ceasing the optimization. When the attempted payoff improvement becomes less than this number, the final guidance runs are made. Due to the halving process that is used and the usual nonlinearities that are encountered, the user should input a number that is no bigger than one-third of the actual desired condition. For the third word, thirty iterations are generally more than sufficient. Four linearization attempts are usually adequate.

Data Block 29

Allowable deviations in trajectory constraint parameters. Since each forward run is judged successful or not partially on the basis of satisfying the terminal constraints, an acceptable "deadband" must be specified for each constraint parameter. This "deadband" is important primarily on the intermediate iterations and hardly affects the constraints miss on the final (optimized) trajectory. If too tight limits are imposed, the payoff improvement will proceed with unnecessarily small steps. If the limits are too loose, some iterations will unwisely be judged "successful." Recommended limits are 20,000 feet on distances, one degree on angles, 20 feet per second on velocity, and 10 pounds per ft^2 on $\bar{q}$. Starting in DA14(14), corresponding numbers are needed for the linearization. Here, the user should put in his required tolerances.

Data Block 30

Vacuum thrust tables are input as functions of stage time for each powered stage specified. The total number of data points cannot exceed 120. If a change in data is necessary, the entire table for the given stage must be re-input.

Data Block 31

Aerodynamic drag coefficients are input as functions of Mach number for stages 1, 2, 6.

Data Block 32

One table of C_{L_α} versus Mach number is required. It is used for stages 1 and 6.

Data Blocks 33, 34, 35, 36, 37

Assorted data for 6D simulation. All body stations are in inches. Into DA6D(12) to (15) must be read integration frequency multipliers. They are normally 1.0, but if a more frequent integration is desired during 6D forward trajectories, the user should input 2. or 3. or whatever integer multiplier is desired. Center of gravity histories are expressed as body station in inches. Hardover turns and impact computations for stages 1, 2 and 3 also utilize these data blocks.

Data Blocks 38 and 39

Stage weight and area data. The total number of weight data points cannot exceed 120. Weight is input as consumable pounds remaining and jettison weight for each stage. If a change in Data Block 38 is necessary, the entire table for the given stage must be re-input.

Data Block 40

Command pitch program for initial nominal trajectory only. When running dispersion analysis following initial nominal, the input theta table should be carried to the ignition time and theta of the spun stage(s) and the table then ended with 1.E10.

DISCUSSION OF THE DATA BLOCKS

OPTIONAL COMPUTATIONS

The general organization when optional computations (dispersion analysis, hardover turns, impact, or radar) are desired will be to group together all the data necessary for that module and add it behind the blank card of the optimization data for which the computations are desired. Data must not appear for modules that will not be called. The sequence must be:

- Basic Scout optimization data, including setting proper options
- Blank card
- Dispersion analysis data

  #46 Header card
  title

  #XX Header for required data
  Data

  #46 Header card
  Title

  #XX Header for required data
  Data

- Blank card
- Hardover turn data

  #44 Header cards
  Data

  #45 Header card
  Data

    #33 Header card
    Words 7, 10, 11, 16

    #37 Header card
    All of data

- Blank card
- Impact prediction data

  #42 Header cards
  Data

  #43 Header cards
  Data

    #33 Header card
    Words 17, 18, 19

- Blank card

- Radar angles and ranges

  #41 Header card
  Data (one card per station)

- Blank card
- 999 card if end of job

Unlike the basic optimization data input capabilities, the optional computations data package does not have the ability to change an individual data word by manipulation of the second and third fields of the data header card. This means the data package must be correct and complete with no "patches." Further, successive cases are not allowed.

Data Block 41: RADAR

Header field 4 indicates the number of cards to be read, one radar station identification and location per card.

Data Block 42: IMPACT

Drag coefficient tables for failure mode impact predictions. Initial conditions are taken from the optimum trajectory at each integration point of the first three powered stages, and the equations of motion are integrated to impact assuming zero thrust using these drag tables which must start at Mach number zero.

Data Block 43: IMPACT

Drag coefficient tables for expended stage casings; must start at Mach number zero.

Data Block 44: HARDOVER

Loads both the pitch moment of inertia of the first three stages and the first stage pitch damping coefficient.

Data Block 45: HARDOVER

Assortment of various constants, test flags, and necessary data to generate hardover turns.

Data Block 46: DISPERSION

Header field 4 indicates the type of dispersed trajectory desired from a list of 38 possibilities which must be specified in increasing order. Block 46 data is a title card that must include the dispersion code as listed on page 22-26. Data necessary to run each particular dispersed trajectory, preceded by the appropriate header card, must immediately follow its block 46 cards. Available dispersion trajectories and necessary data are listed in the data input section. Dispersion data must include the correct signs. Only trajectories for which dispersion information is provided will be computed. When the fourth or fifth stage of a five-stage vehicle is investigated for effect of thrust on dispersion, both stages must be input along with data block 55. This special input is described on page 22-30.

Three Sigma Variation are computed by loading the last data block 46 card as trajectory type 40 with the title "nominal traj."

Data Block 47: DISPERSION

Constant jettison weight increment to be added to stage specified on header card.*

Data Block 48: DISPERSION

Percentage drag variation to be used for all stages.*

Data Block 49: DISPERSION

Thrust misalignment angle for first stage, positive angle gives nose up/yaw right moment.*

Data Block 50: DISPERSION

Magnitude of control system deadbands.*

Data Block 51: DISPERSION

Magnitude of effective angular'change in thrust attitude.* This effective angular change is a precomputed input, since an analytical solution of the vehicle's instantaneous space attitude is beyond the scope of this program. See NASA TR R-110, "Analytical Method of Approximating the Motion of a Spinning Vehicle with Variable Mass and Inertia Properties Acted Upon by Several Disturbing Parameters," by J. J. Buglia, G. R. Young, J. D. Timmons, and H. S. Brinkworth, for a complete discussion.

---

*Data must carry correct sign. Data for all yaw dispersions should cause resultant crossrange dispersions to be in the same direction to realize maximum worst-on-worst three sigma summation.

Here it is assumed that if the spin stabilized fourth stage receives a tipoff rate at separation, as the stage spins about the total angular momentum vector, the out-of-plane motion averages to zero so that the effective result of the tipoff is an attitude angle offset in the plane of the tipoff. For example, if the total angular momentum vector $\overline{H} = \begin{vmatrix} I_{xx} & 0 \\ 0 & I_{yy} \end{vmatrix} \cdot \begin{vmatrix} p \\ q \end{vmatrix}$

where

I_{xx} = roll moment of inertia
I_{yy} = pitch moment of inertia
p = spin stabilization rate
q = spin rate resulting from tipoff

then the effective thrust attitude change ϕ is

$$\tan \phi = (I_{yy}\, q) \,/\, (I_{xx}\, p)$$

Data Block 52: DISPERSION

Incremental change in launch azimuth or attitude.*

Data Block 53: DISPERSIONS

Table of wind velocity as a function of altitude.

Data Block 54: DISPERSIONS

Table of azimuth of wind velocity vector (measured clockwise from north) as a function of altitude. Each wind dispersed trajectory requires both wind tables. Yaw winds must incur crossrange dispersions in the same direction as other yaw dispersions.

Data Block 55: DISPERSIONS

When simulating five-stage SCOUT in dispersions of thrust in stage 4 or 5 (Dispersion Code 7 or 8) it is necessary to input the time duration of the stages as used in the dispersion analysis. These must be the same as final entries in data blocks 30 and 38. Special input to data blocks 30 and 38 is described on page 22-30. Only when both fourth and fifth stages are considered nominal can this input be ignored.

---

*Data must carry correct sign. Data for all yaw dispersions should cause resultant crossrange dispersions to be in the same direction to realize maximum worst-on-worst three sigma summation.

DATA INPUT

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
1	Main Heading	12A6	
2	Case Heading	12A6	
3	Date	2A6	
4	Options	24I1	IP(n)--See list of available options, page 22-24.
5	Stage Sequence	11I3	ISTGE(n) Powered flight stages 1 through 5 Coast stages 6 through 10 *End sequence with zero
6	Adjustable Parameters	12I3	IB1(n) IB1(1)=The number of adjustable parameters to be optimized IB1(2)=The adjustable parameter codes : IB1(12)

** Adjustable Parameter Codes **

1 Stage 4 pitch angle, θ
2 Stage 4 yaw angle, χ
3 Length of coast after stage 3
4 Length of coast after stage 2
5 Length of coast after stage 1
6 Launch azimuth
7 Launch time of day
8 Stage 5 pitch angle, θ
9 Stage 5 yaw angle, χ
10 Length of coast after Stage 4
11 Length of coast after Stage 5

8	Trajectory Constraint Parameters	14I3	IB3(n) IB3(1)=Number of trajectory constraints IB3(2)=List of constraint codes : IB3(14) *Payoff parameter must be first constraint. Payload can only be a payoff in this data block.

** Terminal Constraint Parameter Codes **

0 Payload
1 Inertial velocity
2 Inertial path angle
3 (V_I - Local circular orbit velocity)

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
			4 Radius 5 Aerodynamic velocity 6 Aerodynamic path angle 7 Altitude 8 Downrange distance 9 $2E = V_I^2 - 2\mu/r$ 10 Orbit inclination 11 Perigee radius 12 Longitude of ascending node 13 Argument of perigee 14 Total heating * * Constraints at Stage Points ** X5 Aerodynamic velocity X6 Aerodynamic path angle X7 Altitude X8 Downrange distance X9 Dynamic pressure X = The number of the stage at the end of which the constraint is to be applied. X = 1,9
9	Output Frequency	40I3	IB4(n) Frequency Defined As: $\frac{\text{(number of computed points)}}{\text{(number of output points)}}$ IB4(1) - Initial trajectory IB4(2) - Intermediate IB4(3) - Final IB4(4) - Stage code IB4(5) - Initial trajectory IB4(6) - Intermediate IB4(7) - Final IB4(8) - Stage code ⋮ IB4(40)- etc., for all 10 stages
10	Nominal Constants	15E12.8	CT1(n) Nominal values shown Input only if changes are desired CT1(1) - 7.29211E-5 rad/sec Omega CT1(2) - 1.407735E16 ft^3/sec^2 FMU CT1(3) - 20,925,696. ft RE at Equator CT1(4) - 1716.4827 gas constant for atmosphere subroutine CT1(5) - 32.174 ft/sec^2 measured sea level gravity; weight-to-mass conversion factor (GO)

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
			CT1(6) - 2116.2 lb/ft^2 PO Sea level ambient pressure
			CT1(7) - 250,000 ft HAERO Altitude above which all aerodynamic computations are by-passed
			CT1(8) - 54. Upper limit of lines per page
			CT1(9) - .017453292 degrees to radians conversion
			CT1(10)- .02 sec TIMEP Time-epsilon increment to insure hitting critical time
			CT1(11)- 1.0827 x 10^{-3}=J_2 oblate Earth constant
			CT1(12)- 298.3 Earth flattening factor
			CT1(13)- 1.0
11	Atmosphere	8E12.8	CT2(n) Nominal values tabulated
12	Subroutine	8E12.8	CT3(n) in report section 19.1
13	Constants	8E12.8	CT4(n)
14	Constants	8E12.8	CT5(n) *Input only if changes from nominal values are desired.
16	Time Duration and Number of Integration Steps Within Each Stage	20E12.8	DA1(n) Time in Seconds
			DA1(1) Stage 1 time duration
			DA1(2) Stage 1 points
			DA1(3) Stage 2 time
			DA1(4) Stage 2 points
			DA1(5) Stage 3 time
			DA1(6) Stage 3 points
			DA1(7) Stage 4 time duration
			DA1(8) Stage 4 points
			DA1(9) Stage 5 time
			DA1(10) Stage 5 points
			DA1(11) Stage 6 time
			DA1(12) Stage 6 points
			DA1(13) Stage 7 time
			DA1(14) Stage 7 points
			⋮
			DA1(20) Stage 10 points

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
17	Input Data	20E12.8	DA2(n) DA2(5) Stage 4 θ (deg) DA2(6) Stage 4 χ (deg) DA2(7) Nose radius (ft) DA2(8) Seconds before launch to uncage gyros DA2(9) Stage 5 θ (deg) DA2(10) Stage 5 χ (deg) ⋮ DA2(15) Payload weighting factor DA2(16) Limit on coast time change each iteration DA2(17) Minimum coast duration, sec.
18	Pitch Program Linearization Times	17E12.8	TLIN(n) TLIN(1) Number of linear segments TLIN(2) Coded stage times at start of segments ⋮ TLIN(17) Form: 100n+t, where n is stage number and t is seconds from stage ignition. Input complete data block.
19	Trajectory Constraint Magnitudes	13 E12.8	DA4(n) DA4(1) are the desired values of ⋮ the constraints listed in the same order as in Data DA4(13) Block Number 8 *Units are feet, radians, seconds, nautical miles for range.
20	Initial Times	2E12.8	DA5(n) DA5(1) Time duration for vertical lift-off calculation (sec) DA5(2) Launch time in Space Age Date (days from 0.0 hours, Jan. 1 1960)
21	Initial Conditions	6E12.8	DA6(n) DA6(1) Altitude (feet) DA6(2) Longitude (degrees ± from prime meridian) DA6(3) Geocentric latitude (deg) DA6(4) Velocity (ft/sec) DA6(5) Azimuth (deg) DA6(6) Flight path angle (deg)
22	Control Variable Constraints	6E12.8	DA7(n) DA7(1) Time to start $\alpha = 0$ DA7(2) Time to end $\alpha = 0$ DA7(3) Max. pitch rate (deg/sec) DA7(4) Min. pitch rate DA7(5) Min. change in pitch rate
26	Weighting Constants for Linearization Segments	15E12.8	ETALIN(1) For linear segments ⋮ identified in Data ETALIN(15) Block 18

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
27	Weighting Constants for Optimization Parameters	11E12.8	DA12(1) : DA12(11) For adjustable parameters identified by code number as listed for Data Block 6
28	Convergence Data	6E12.8	DA13(n)
			DA13(1) Initial payoff improvement to be used if (automatic) internal computation fails
			DA13(2) Magnitude for stopping attempted payoff improvement
			DA13(3) Maximum number of forward trajectories per case during optimization
			DA13(4) Maximum number of linear trajectory iterations
29	Permitted Values of Trajectory Constraint Deviations During Optimization and During Linearization	26E12.8	DA14(n) List in same order as in Data Block 8, excluding payoff parameter
			DA14(1) : DA14(12) During optimization
			DA14(14) : DA14(25) During linearization
30	Vacuum Thrust Tables	250E12.8	THRUST(n) *Stage code required on header cards Sequence: time, thrust, time, thrust ....
31	Drag Coefficient Tables (ALPHA = 0)	42/22/22E12.8	TD1, TD2, TD6 *Stage code required on header cards Sequence: blank, mach, C_D, mach, C_D, mach 1.0E10
32	Lift Coefficient per ALPHA Table	42E12.8	TL1 Sequence: blank, mach vs C_{L_α}, 1.0E10 Units: C_L per radian
33	Data for 6D Simulation	20E12.8	DA6D(1) K_θ Position Gain
			(4) K_q Rate Gain
			(7) X_δ Body station of jet vanes and fins in stage 1
			(10) K_{TV_δ} jet vane effectiveness (lb/lbTV/deg/vane)
			(11) X_{FS} body station reference for moment coefficients
			(12) : (15) 6D integration frequency ratio for stages 1-4

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
			DA6D(16) Reference length for aero moment data (ft) (17) (18) (19) n: where impact locations are computed from every nth integration step on nominal for stages 1, 2, 3.
34	Fin Normal Force Coefficient per Delta	42E12.8	CNDELT(n) Sequence: blank, mach vs C_{N_δ} , 1.0E10 Units: C_N/radian/fin
35	Pitching Moment Coefficient per ALPHA	42E12.8	CMALPH(n) Sequence: Same Units: C_M/radian
36	Fin Pitching Moment Coefficient per Delta	42E12.8	CMDELT(n) Sequence: Same Units: C_M/radian/fin
37	Stage Center of Gravity Histories	22E12.8	X_{CG1}, X_{CG2}, X_{CG3}, X_{CG4} each dimensioned 22 *Stage code required on header cards Sequence: blank, stage time vs c.g. body station, 1.0E10
38	Stage Consumable-Weight-Remaining Histories	250E12.8	WEIGHT(n) *Stage code required on header cards Sequence: time, weight, time, weight ....
39	Stage Data	3E12.8	SG1, SG2, SG3, SG4, SG5 (each dimensioned 3), SG6 (2) *Stage code required on header cards SGX(1) Jettison weight (lb) SGX(2) Aerodynamic Reference Area (ft^2) SGX(3) Total nozzle exit area (ft^2)
40	Theta History for Nominal Trajectory	32E12.8	TTH(n) Sequence: blank, time vs theta, 1.0E10 Units: degrees, seconds from launch

DATA INPUT

IP(1)	Payoff Function	0	Payload
		1	Aerodynamic or Inertial velocity
		2	Altitude
IP(2)	Heating Constraint	0	No constraint
		1	Impose inequality constraint
IP(3)	Exchange Ratios	0	Do not compute exchange ratios
		1	Do compute exchange ratios
IP(4)		0	
IP(5)		0	
IP(6)	6D	0	No 6D at all
		1	6D after successful linearization
		2	6D on initial nominal
IP(7)	Pitch Program Linearization	0	No linearization
		1	Linearization after optimization
IP(8)		0	
IP(9)	Spin Stabilized or Continuous Control in Stages 4 & 5	1	Stages 4 & 5 Spun: $\theta_5 = \theta_4$; $\chi_5 = \chi_4$
		2	Stage 4 continuous and Stage 5 spun
		3	Stages 4 and 5 continuous
IP(10)	Stage 5 χ stages	0	Fewer than 5 stages
		1	$\chi_5 = \chi_4$
		2	χ_5 independent of χ_4
IP(11)	Zero-Alpha Constraint in Stage 1	0	None
		1	ALPHA = 0. over specified time
IP(12)	Constraint on Alpha at Stage 2 Ignition	0	None
		1	Do constrain $\alpha = 0$
IP(13)	Constraint on $\bar{q}$ at Stage 2 Ignition	0	None or equality constraint
		1	Inequality ($\bar{q} \leq x$) constraint
IP(14)		0	
IP(15)	Memory Dump	0	Do not dump memory
		1	Dump memory at end of job (includes floating-point dump of floating-point numbers in common)
IP(16)		0	
IP(17)	Output of Input Data	0	Do not output data
		1	Output data

IP(18)		0	
IP(19)		0	
IP(20)		0	
IP(21)	Nominal Dispersion	0	None
		1	Compute specified dispersions
IP(22)	Failure-Mode Hardover Turns	0	None
		1	Compute turns
IP(23)	Impact from Nominal Trajectory	0	None
		1	Compute impact locus
IP(24)	Radar Tracking Coordinates	0	None
		1	Compute for specified stations

ADDITIONAL DATA INPUT FOR OPTIONAL COMPUTATIONS
RADAR, IMPACT, HARDOVER

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
41	Radar Station Locations	2A6,3E12.8	4th field of header = total number of stations. station identification, geodetic latitude, degrees, longitude, degrees, altitude, feet
42	Impact Failure Mode Drag Coefficient Tables	22E12.8	CD1F, CD2F, CD3F *Stage code required on header cards Sequence: blank, Mach, C_D, Mach, C_D... 1.0E10
43	Impact Expended Casings Drag Coefficient Tables	22E12.8	CD1E, CD2E, CD3E *Stage code required on header cards Sequence: blank, Mach, C_D, Mach, C_D... 1.0E10
44	Hardover Moment of Inertia about Pitch Axis, I_{yy}	22E12.8	IYY1, IYY2, IYY3 *Stage code required on header cards Sequence: blank, time, I_{yy}, time, I_{yy} ... 1.0E10 Units: $slug \cdot ft^2$
44	Pitch Damping Coefficient for Hardover - Stage 1	22E12.8	CMQ(n) *Stage code 4 required Sequence: blank, Mach, CMQ, Mach, CMQ ... 1.0E10 Units: CMQ per radian about instantaneous CG
45	Hardover Data	22E12.8	1. n: Stage 1 hardover turns are computed every nth integration step. 2. $(\bar{q} \cdot \alpha)_{max}$, lb deg/ft^2, hardover integration stops one second after $(\bar{q} \cdot \alpha)_{max}$. 3. $(DETA)_{max}$, degrees, maximum allowable change in pitch attitude per integration step. 4. $(DT)_{max}$, seconds, maximum integration step size. 5. $(DT)_{min}$, seconds, minimum integration step size. 6. Maximum number of integration steps per turn.

ADDITIONAL DATA INPUT FOR OPTIONAL COMPUTATIONS HARDOVER, DISPERSED TRAJECTORIES

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
45 (Cont'd)	Hardover Data	26E12.8	7. 1st stage thrust misalignment angle, degrees. 8. 2nd stage thrust misalignment angle, degrees. 9. 3rd stage thrust misalignment angle, degrees. 10. Stage 1 thrust application station, inches. 11. 1st stage control fin and vane deflection, degrees. 12. Mode B stage 2 control jet force, lb. 13. Mode B stage 3 control jet force, lb. 14. Mode C stage 2 control jet force, lb. 15. Mode C stage 3 control jet force, lb. 16. Stage 2 thrust application station, inches. 17. Stage 3 thrust application station, inches. 18. Stage 2 control jet application station, inches. 19. Stage 3 control jet application station, inches. 20. Output frequency, seconds. 21. n: stage 2 hardover turns are computed every nth integration step. 22. n: stage 3 hardover turns are computed every nth integration step. 23. $(\text{DETA})_{max}$, degrees, maximum allowable change in pitch attitude per integration step, Upper Stages. 24. $(\text{DT})_{max}$, seconds, maximum integration step size, Upper Stages. 25. $(\text{DT})_{min}$, seconds, minimum integration step size, Upper Stages. 26. Maximum number of integration steps per turn, Upper stages.
46	Dispersion Trajectory Code and Title	2A6	Type of trajectory to be computed is indicated by the code in field 4 of header card 46. Codes of available dispersions are tabulated on the following page, along with the type of data required for each dispersed trajectory.

ADDITIONAL DATA INPUT FOR OPTIONAL COMPUTATIONS
DISPERSION TRAJECTORY CODES

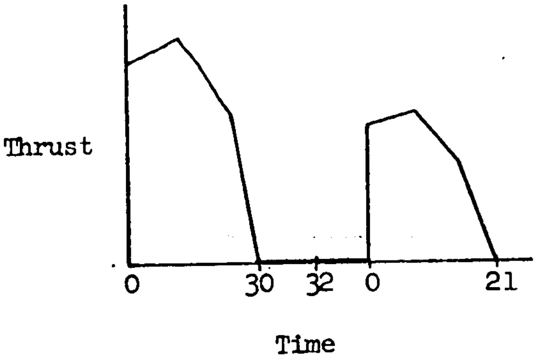
	Dispersion Trajectory			Data Input
Code	**Type**	**Disperse**	**Stage**	
1	Vacuum Thrust Variation	High	1	#30 thrust tables
2	" " "	Low	1	#38 weight tables
3	" " "	High	2	#55 stage 4 and 5 durations for
4	" " "	Low	2	codes 7 and 8 on five-stage
5	" " "	High	3	vehicle. Also, see page 22-30.
6	" " "	Low	3	
7	" " "	High	4	*Stage code required on header
8	" " "	Low	4	cards
9	Weight Increment	High	1	#47 stage weight increment
10	" "	Low	1	
11	" "	High	2	
12	" "	Low	2	*Stage code required on header
13	" "	High	3	cards
14	" "	Low	3	
15	" "	High	4	
16	" "	Low	4	
17	Drag Variation	High	1,6,2	#48 percentage change in drag
18	" "	Low	1,6,2	
19	C_{M_α} Variation	Forward	1	#35 C_{M_α} table (complete)
20	"	Aft	1	
21	Thrust Misalignment	High	1	#49 misalignment angle in degrees,
22	" "	Low	1	thrust application station, inches
23	" "	Yaw	1	*Stage code required
24	Control System Deadband	High	2	#50 deadband angle in degrees
25	" " "	Low	2	
26	" " "	Yaw	2	*Stage code required
27	" " "	High	3	
28	" " "	Low	3	
29	" " "	Yaw	3	
30	4th and/or 5th Stage Tipoff	High	4 or 5	#51 effective angular change
31	" " "	Low	4 or 5	in degrees
32	" " "	Yaw	4 or 5	*Stage code required
33	Launch Azimuth	Yaw	1	#52 angular change in degrees
34	Launch Attitude	High	1	
35	" "	Low	1	
36	Winds	High		#53 WINDVL, alt. vs velocity table
37	"	Low		#54 WINDAZ, alt. vs azimuth table
38	"	Yaw		
40	Compute three sigma variations. Load "Nominal Traj" as title. No additional data required.			

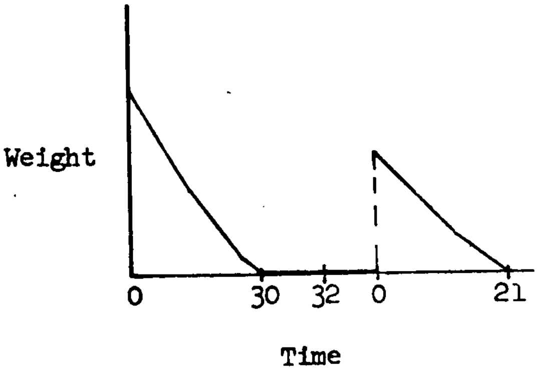
ADDITIONAL DATA INPUT FOR OPTIONAL COMPUTATIONS
DISPERSED TRAJECTORIES

DATA BLOCK NUMBER	TITLE	FORMAT	COMMENTS
47	Stage Weight Increment	E12.8	Increment in pounds *Stage code required on header card
48	Percentage Drag Variation	E12.8	Input as 0.xxx
49	Stage 1 Thrust Misalignment	2E12.8	Misalignment angle in degrees, thrust application station, inches *Stage code required
50	Control System Deadband Angle	E12.8	Angle in degrees *Stage code required
51	Fourth Stage Tipoff	E12.8	Effective angular change in attitude, degrees *Stage code required
52	Launch Dispersions	E12.8	Launch azimuth or attitude change, degrees
53	Wind Velocity	32E12.8	WINDVL(n) = wind velocity Sequence: blank, altitude, velocity, altitude, velocity ... 1.0E10 Units: feet, ft/sec
54	Wind Azimuth	32E12.8	WINDAZ(n) = azimuth of wind vector measured from North Sequence: blank, altitude, azimuth, altitude, azimuth ... 1.0E10 Units: feet, degrees
55	Stage 4 and 5 Durations	2E12.8	(1) = Stage 4 duration (2) = Stage 5 duration

SPECIAL INPUT FOR THRUST DISPERSION IN FIVE-STAGE SCOUT

When simulating a five-stage Scout, thrust dispersions in stage 4 and/or 5 require input to data blocks 30 and 38 in a special manner. This is necessitated by the change in program format, during development, from four-stage to five-stage. Input of the thrust and weight histories of both stages 4 and 5 must be done into the stage 4 tables. In order to facilitate input the user shall form the tables in the following manner.





That is, the thrust and weight are tabulated in stage time as during optimization but are stacked for the two stages in the stage 4 area. Note that the coast duration is input arbitrarily as two seconds in the above tables; the program automatically changes this to the optimized value.

OUTPUT FORMAT

OPTIMIZATION AND LINEARIZATION COMPUTATIONS

The program output consists primarily of the tabulated history of the trajectory variables for each forward iteration. There is also an output of all data which has been input (selected by option 17), output of the corrections to be made in the terminal constraints on each iteration, output of several mass improvement parameters, performance exchange ratios, and linearized command pitch program. A definition of all output quantities follows:

Column Headings for Trajectory Listings

TIME	Total time from launch
VEL	Velocity in rotating frame
GAMMA	Vertical-plane path angle of VEL from geocentric horizontal
PSI	Azimuth of VEL, degrees east of north
ALTITUDE	Distance above local Earth surface
GEOC LAT	Geocentric latitude, degrees north of equator
TAU	Longitude in rotating frame, degrees east of prime meridian
QBAR	Dynamic pressure
THRUST	Net thrust
WEIGHT	Sea level weight
GEOD LAT	Geodetic latitude
RADIUS	Distance from vehicle to Earth center
HEAT RATE	Stagnation point heating rate
THETA	Vehicle pitch attitude referenced to launch geodetic horizontal
DTHETA	Change in theta from current nominal trajectory
VI	Velocity in inertial frame
GAMI	Vertical-plane path angle of VI
PSII	Azimuth of VI
MACH	Mach number, VEL/local speed of sound
ALPHA	Vertical angle between THRUST and VEL
DRAG	Aerodynamic drag
LIFT	Aerodynamic lift
QALPHA	Product of QBAR and ALPHA
RANGE	Great circle arc between launch site and vehicle, converted to nautical miles by 1 n.mi. = 1 minute of arc
BETA6D	Sideslip angle of attack
CHI	Yaw angle between thrust and platform pitch plane
THETAC	Commanded vehicle pitch attitude
THETADOT	Commanded pitch rate

Output of Terminal Conditions Orbit Elements

TWOE	$2E = V_I^2 - 2\mu/r$
EH	Angular momentum, $V_I \cdot r \cdot \cos \gamma_I$
RP	Perigee radius
EYE	Inclination, degrees
BETAP	Argument of perigee, degrees
ECCENTRICITY	Orbital eccentricity
RALPHA	Semi-major axis, feet
PERIOD	Orbital period, minutes
OMEGAE	Longitude of ascending node, degrees

Output at Start of Each Iteration

Constraint Corrections	These quantities, called $d\psi_i$ in the equations, are the negative of the errors in constraints on the current nominal. Units are the same as in data block 19.
Nominal Payoff Increment	Present attempted improvement in payoff function.
Initial Payoff Increment	Automatically computed initial payoff.
Payload Increment	When payload is payoff only.
Corrected Payoff	Payoff function corrected to condition of zero error in all constraints.

Performance Exchange Ratios

When option 3 is set = 1, performance exchange ratios are computed and printed out during the backward guidance run preceding the "final guidance" trajectory at completion of trajectory optimization. These quantities are first order approximations to reoptimized trajectory solutions for the sensitivities of the payoff function to unit changes in each of five vehicle parameters. The units depend on the payoff and the vehicle parameters but are of the form X units of payoff change per unit of vehicle parameter change. Units are as follows:

Payoff

Payload - - pounds
Velocity - - feet/sec
Altitude - - feet

Vehicle Parameter

Burn rate - - 1%
Specific impulse - - 1%

OUTPUT FORMAT

DISPERSION ANALYSIS

The dispersion analysis of the optimum trajectory from error sources listed on page 22-28 provides three different forms of output: (1) staging trajectory variables and their dispersions for each individual dispersion trajectory, (2) a summary of corresponding individual dispersions, and (3) three sigma variations, if selected.*

The trajectory variables that are considered are:

ALTITUDE	Distance above the Earth's surface
VELOCITY	Velocity in rotating frame
GAMMA	Vertical path angle of VELOCITY from horizontal
DOWNRANGE	Great circle distance between vehicle and launch site, assuming 60 n.m. per degree of arc
CROSSRANGE	Distance between vehicle and instantaneous plane of motion existing at nominal vehicle staging position, measured in great circle normal to this plane, assuming 60 n.m. per degree of arc

Note all definitions are identical to these for basic program, except CROSSRANGE, which appears only for yaw dispersed trajectories.

In addition, the orbit elements achieved on the dispersed trajectory, and their dispersions from those of the nominal trajectory, are output. The standard definitions are repeated here for completeness.

TWOE	$2E = V_I^2 - 2\mu/r$, twice the energy per unit mass
EH	$V_I \cdot R \cdot \cos \gamma_I$, angular momentum
RP	Radius of perigee
EYE	Inclination
BETAP	Argument of perigee
OMEGAE	Longitude of ascending node

Dispersions are defined as (trajectory variable on the "dispersed" trajectory) - (trajectory variable from nominal boost history).

---

*For wind dispersed trajectories, the vehicle's velocity relative to the air mass is output as AEROVL with the standard trajectory output.

Aerodynamic drag - - 1 ft^2 of drag area
Jettison weight - - 1 pound
Propellant weight - - 1 pound

Linearized Pitch Program

At the start of each forward linearization trajectory, the linearized pitch program is output as a series of points of pitch angle θ versus stage time. Then, during the trajectory, the pitch rates are output at the start of each linear segment.

Three sigma variations are computed using two methods: (1) a summation of the dispersed trajectory variables, and (2) root sum square. Trajectories are grouped according to the stage 4 burnout altitude dispersion, high or low. If there is no altitude dispersion, the stage 4 burnout velocity dispersion is used. Yaw dispersion trajectories, as specified by the input codes, are handled as a distinct and separate set for the three sigma variations.

IMPACT

Impact points on the geodetic Earth are predicted for the optimum trajectory assuming stage failure during boost of first three stages, and also for the expended stage casings. Output consists of the failure time from launch, predicted impact time from launch, and the impact location in geodetic latitude, longitude, downrange distance, and geocentric latitude, all standard definitions. Drag effects are included for all trajectories.

HARDOVER TURNS

Assuming a control system failure, hardover turn maneuvers are computed for up to ten seconds duration for the first three stages at a frequency specified by the user. The first stage failure mode corresponds to a hardover maneuver in pitch or yaw due to maximum deflection of the fins and jet vanes on the first stage in conjunction with a thrust misalignment that will add to the angular rate produced. The second and third stage control malfunctions are:

Mode 1:	Thrust misalignment with no jet control
Mode 2:	Single jet control operation
Mode 3:	One pitch and yaw jet operations, assuming the vehicle rolled 45 degrees

Output for each hardover turn indicates the time in the nominal trajectory that the failure occurred, and:

TIME	From start of hardover turn
CHG GAMMA	Change in direction of the nominal relative velocity vector
QBAR-A	Dynamic pressure times angles of attack (first stage only)

RADAR

In addition to radar station identification and location, the slant range and look angles computed from the optimum trajectory are output as a function of time from launch for as many as twenty different tracking station locations that are input in terms of latitude, longitude and altitude.

RANGE	Slant range from vehicle to tracking station
YDS	Range in yards, decimal system
YDS(OCT)	Range in yards, IBM Octal system
YDS-1165000	Range in yards - 1165000. yards, IBM Octal system

AZIMUTH	Azimuth of the vehicle with respect to the tracking station measured clockwise from north to the local geodetic horizontal plane
DEG	Azimuth in degrees, decimal system
DEG(OCT)	Azimuth in degrees, IBM Octal system
DEG MIN	Azimuth in degrees and minutes, decimal system
MILS	Azimuth in mils, decimal system

ELEVATION	Elevation of the vehicle with respect to the tracking station geodetic horizontal plane
DEG	Elevation in degrees, decimal system
DEG(OCT)	Elevation in degrees, IBM Octal system
DEG MIN	Elevation in degrees and minutes, decimal system
MILS	Elevation in mils, decimal system

National Aeronautics and Space Administ
Washington, D. C. 20546
Attention: Library, Code USS-10

N68-18185

7

ERRATA

NASA Contractor Report 66515

TOLIP - TRAJECTORY OPTIMIZATION AND LINEARIZED PITCH COMPUTER PROGRAM

By Robert E. Willwerth, Jr. and Richard C. Rosenbaum

LOCKHEED MISSILES AND SPACE COMPANY
Palo Alto, California

NASA Contract NAS1-5106

Issue date: 2-20-69

H4.1 DEFINITION OF SYMBOLS FOR HYPERBOLIC ASYMPTOTE CONTRAINTS

B	Booster burnout
H	Direction of Outward Radial
H_1	Equatorial projection of H
i = EYE	Inclination of geocentric orbit plane

OUTWARD RADIAL = the vector direction of hyperbolic excess velocity vector translated to Earth center

P	Perigee of orbit
P_1	Equatorial projection of P
V_H	Magnitude of hyperbolic excess velocity
W	Ascending node of the orbit plane
α_{e_H}	Right ascension of Outward Radial
β_H	In-plane angle from ascending node
β_p	Argument of perigee
δ_{e_H}	Declination of Outward Radial
ζ	True anomaly of booster burnout
ζ_H	True anomaly of hyperbolic asymptote
ν_H	Inertial longitude angle from ascending node
Ω_e	Longitude of W
Υ	Vernal equinox

H4.6 DIAGRAM AND EQUATIONS FOR HYPERBOLIC EXCESS ASYMPTOTE ORBIT ELEMENTS

The nature of the trajectory optimization process in SCOUT is virtually the same for all missions. The differences lie only in the form of the constraints imposed on the trajectory. For near Earth missions terminal constraints can be imposed on the trajectory variables explicitly, or they can be specified in terms of conventional orbit elements which are functions of the trajectory variables.

For the addition of the Earth escape missions, the terminal constraints have been formulated as parameters which are functions of Earth-referenced orbit elements. Thus, multiple use of the coding is made possible and the form of the constraints is conceptually similiar for all missions.

The unit sphere diagram and orbit equations for the hyperbolic excess velocity asymptote are documented on the following pages. The three terminal constraints that have been added to the program are (1) the magnitude of the hyperbolic excess velocity vector, (2) its right ascension angle (in radians), and (3) its declination angle (in radians). The direction of the vector is that of the asymptote to the departure hyperbola, and the desired values for these parameters are input with the other constraint information.

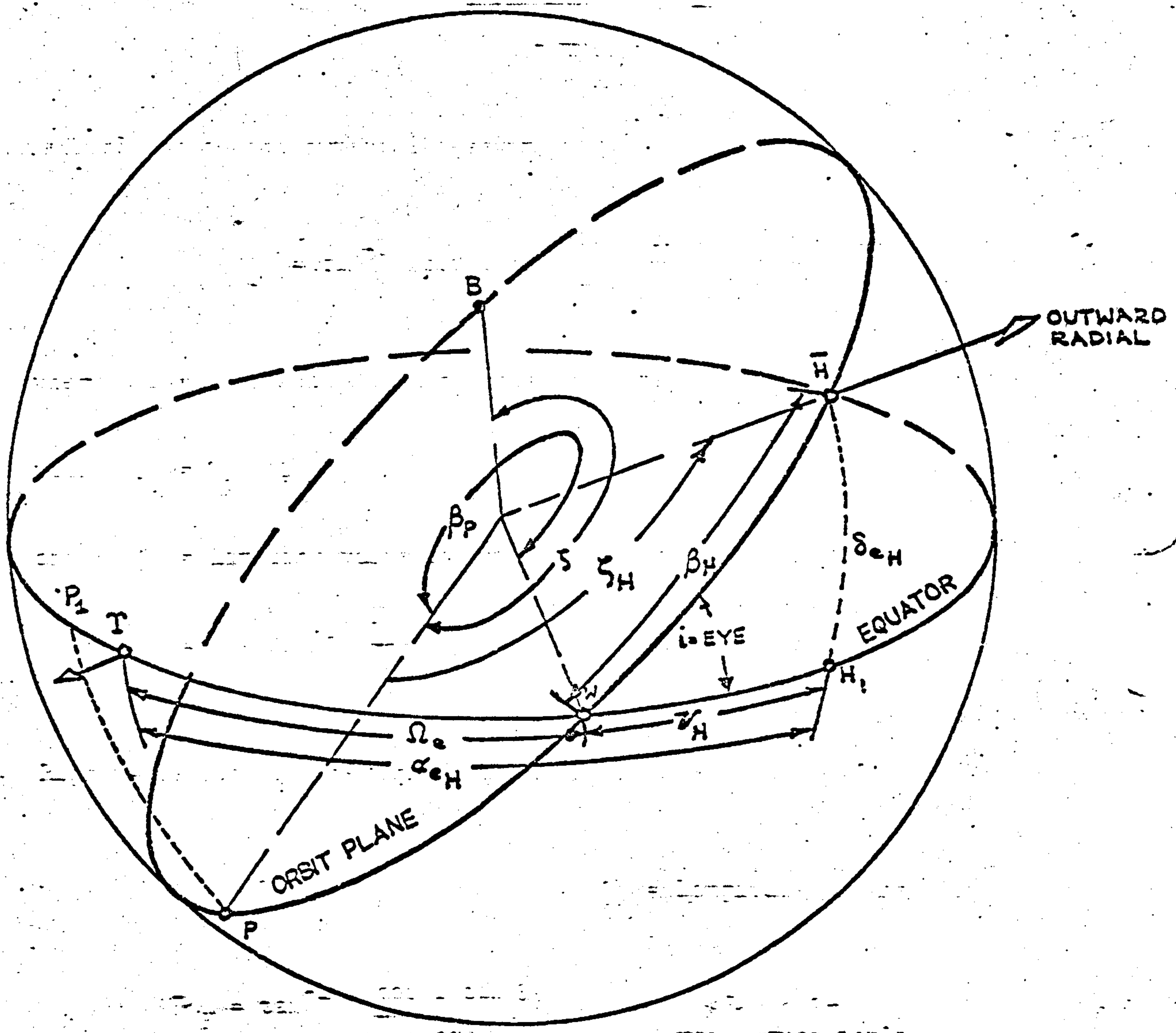
The analysis incorporates the following approximations, which allow a closed form solution of the Earth departure ballistic path. Since the direction of the asymptote is stated in terms of its right ascension and declination, an

Earth-centered origin of the asymptote is implied. The first approximation employed is to accept any hyperbola of the required energy whose asymptote is parallel to the required direction. Since typically the asymptote passes within one or two Earth-radii of Earth center, this approximation is relatively slight.

The second approximation is that of using the booster burnout conditions to define the hyperbolic ballistic path. Since the SCOUT program incorporates an oblate Earth model, the gravity model will perturb the ballistic coast path from that of a simple hyperbola after booster burnout. Although these perturbations should be small, their effect can be included by introducing an integrated coast following the final stage. In this way, the terminal conditions of the coast stage should provide a more realistic representation for injection onto the hyperbolic path when higher accuracy is required.

Again, it should be realized that with the above formulation the optimum injection conditions are automatically found as an implicit part of the trajectory optimization process.

GEOMETRY USED TO DEFINE THE HYPERBOLIC EXCESS VELOCITY VECTOR ORBIT ELEMENTS



HYPERBOLIC EXCESS VELOCITY VECTOR ORBIT ELEMENTS

1. Hyperbolic excess velocity (V_H).

$$V_H = \sqrt{V_I^2 - \frac{2\mu}{r}} = \sqrt{2E}$$

2. Declination of the hyperbolic asymptote (δ_{e_H})

$$\delta_{e_H} = \sin^{-1} (\sin i \sin \beta_H) \quad -\frac{\pi}{2} \le \delta_{e_H} \le \frac{\pi}{2}$$

where: i = inclination, β_p = argument of perigee

with $\beta_H = \beta_p + \zeta_H$

and ζ_H = limiting value of true anomaly of hyperbola

$$\zeta_H = \pi - \tan^{-1} \frac{HV_H}{\mu}, \qquad H = \text{angular momentum}$$

3. Right Ascension of the hyperbolic asymptote (α_{e_H})

$$\alpha_{e_H} = \Omega_e + \nu_H \qquad \Omega_e = \text{Longitude of Ascending Node}$$

$$\nu_H = \tan^{-1} \left(\frac{\cos i \sin \beta_H}{\cos \beta_H} \right) \qquad \begin{array}{l} 0 \le \nu < 2\pi \\ \text{use 4 quad } \tan^{-1} \end{array}$$

H5.5 INITIAL CONDITIONS FOR INTEGRATION OF THE ADJOINT EQUATIONS

The adjoint differential equations, shown on page 5-9 of Section 5.2, are integrated backward along the trajectory to form the λ (JC, 7) matrix, which is a time function relating perturbations in the state variables to the perturbation in each constraint at the final time. Recall that the adjoint differential equations themselves do not depend upon the constraint parameter involved. Only the initial conditions for the adjoint equations depend upon the form of the constraint parameter, as discussed in Section 5.2.

Since the constraints of hyperbolic excess velocity, asymptotic right ascension, and asymptotic declination are applied at burnout of the last stage, the initial conditions for these terminal constraints are simply the partial derivatives of these constraints with respect to the trajectory variables evaluated at the final time. They are evaluated using a chain rule procedure of the form

$$\frac{\partial\,(\text{constraint})}{\partial\,(\text{state variables})} = \frac{\partial\,(\text{constraint})}{\partial\,(\text{inertial variables})} \cdot \frac{\partial\,(\text{inertial variables})}{\partial\,(\text{state variables})}$$

In Section H5.5-1, the complete matrix of partial derivatives for initializing the adjoint variables is given for these three constraints. The partial derivatives of the inertial trajectory variables used in these equations have already been coded in SCOUT, and are defined in Section 5.5. The new, additional terms necessary to complete the adjoint variable initialization are the partial derivatives of certain orbit elements as well as the partial derivatives of the three constraints with respect to the inertial trajectory variables, and these are defined in Sections H5.5-2. and H5.5-3. The nomenclature and equations describing and defining the constraint parameters themselves are given in Section H4.1 and H4.6.

H5.5-1a

INITIALIZATION OF ADJOINT VARIABLES
FOR HYPERBOLIC EXCESS VELOCITY CONSTRAINT

$$\lambda_{X@t\,=\,t_f} = \frac{\partial\, VH}{\partial\, X} = \frac{\partial\, VH}{\partial\, XI} \cdot \frac{\partial\, XI}{\partial\, X}$$

$$\lambda_{V@t} = t_f = \frac{\partial\, VH}{\partial\, V} = \frac{\partial\, VH}{\partial\, VI}\,\frac{\partial\, VI}{\partial\, V} + 0 + 0 + 0 + 0 + 0$$

$$\lambda_{\gamma@t} = t_f = \frac{\partial\, VH}{\partial\, \gamma} = \frac{\partial\, VH}{\partial\, VI}\,\frac{\partial\, VI}{\partial\, \gamma} + 0 + 0 + 0 + 0 + 0$$

$$\lambda_{r@t} = t_f = \frac{\partial\, VH}{\partial\, r} = \frac{\partial\, VH}{\partial\, VI}\,\frac{\partial\, VI}{\partial\, r} + 0 + \frac{\partial\, VH}{\partial\, r} + 0 + 0 + 0$$

$$\lambda_{\psi@t} = t_f = \frac{\partial\, VH}{\partial\, \psi} = \frac{\partial\, VH}{\partial\, VI}\,\frac{\partial\, VI}{\partial\, \psi} + 0 + 0 + 0 + 0 + 0$$

$$\lambda_{\lambda@t} = t_f = \frac{\partial\, VH}{\partial\, \lambda} = \frac{\partial\, VH}{\partial\, VI}\,\frac{\partial\, VI}{\partial\, \lambda} + 0 + 0 + 0 + 0 + 0$$

$$\lambda_{\tau@t} = t_f = \frac{\partial\, VH}{\partial\, \tau_I} = 0 + 0 + 0 + 0 + 0 + 0$$

H5.5-1b

INITIALIZATION OF ADJOINT VARIABLES
FOR RIGHT ASCENSION CONSTRAINT

$$\lambda_{X@t = t_f} = \frac{\partial RA}{\partial X} = \frac{\partial RA}{\partial X_I} \frac{\partial X_I}{\partial X}$$

$$\lambda_{V@t = t_f} = \frac{\partial RA}{\partial V} = \frac{\partial RA}{\partial V_I} \frac{\partial V_I}{\partial V} + \frac{\partial RA}{\partial \gamma_I} \frac{\partial \gamma_I}{\partial V} + 0 + \frac{\partial RA}{\partial \psi_I} \frac{\partial \psi_I}{\partial V} + 0 + 0$$

$$\lambda_{\gamma@t = t_f} = \frac{\partial RA}{\partial \gamma} = \frac{\partial RA}{\partial V_I} \frac{\partial V_I}{\partial \gamma} + \frac{\partial RA}{\partial \gamma_I} \frac{\partial \gamma_I}{\partial \gamma} + 0 + \frac{\partial RA}{\partial \psi_I} \frac{\partial \psi_I}{\partial \gamma} + 0 + 0$$

$$\lambda_{r@t = t_f} = \frac{\partial RA}{\partial r} = \frac{\partial RA}{\partial V_I} \frac{\partial V_I}{\partial r} + \frac{\partial RA}{\partial \gamma_I} \frac{\partial \gamma_I}{\partial r} + \frac{\partial RA}{\partial r} + \frac{\partial RA}{\partial \psi_I} \frac{\partial \psi_I}{\partial r} + 0 + 0$$

$$\lambda_{\psi@t = t_f} = \frac{\partial RA}{\partial \psi} = \frac{\partial RA}{\partial V_I} \frac{\partial V_I}{\partial \psi} + \frac{\partial RA}{\partial \gamma_I} \frac{\partial \gamma_I}{\partial \psi} + 0 + \frac{\partial RA}{\partial \psi_I} \frac{\partial \psi_I}{\partial \psi} + \frac{\partial RA}{\partial \lambda} + 0$$

$$\lambda_{\lambda@t = t_f} = \frac{\partial RA}{\partial \lambda} = \frac{\partial RA}{\partial V_I} \frac{\partial V_I}{\partial \lambda} + \frac{\partial RA}{\partial \gamma_I} \frac{\partial \gamma_I}{\partial \lambda} + 0 + \frac{\partial RA}{\partial \psi_I} \frac{\partial \psi_I}{\partial \lambda} + 0 + 0$$

$$\lambda_{\tau@t = t_f} = \frac{\partial RA}{\partial \tau_I} = 0 + 0 + 0 + 0 + 0 + 1.$$

H5.5-1c

INITIALIZATION OF ADJOINT VARIABLES
FOR DECLINATION CONSTRAINT

$$\lambda_{X@t} = t_f = \frac{\partial\,(\text{DECLIN})}{\partial X} = \frac{\partial\,(\text{DECLIN})}{\partial X_I}\frac{\partial X_I}{\partial X}$$

$$\lambda_{V@t} = t_f = \frac{\partial D}{\partial V} = \frac{\partial D}{\partial V_I}\frac{\partial V_I}{\partial V} + \frac{\partial D}{\partial \gamma_I}\frac{\partial \gamma_I}{\partial V} + 0 + \frac{\partial D}{\partial \psi_I}\frac{\partial \psi_I}{\partial V} + 0 + 0$$

$$\lambda_{\gamma@t} = t_f = \frac{\partial D}{\partial \gamma} = \frac{\partial D}{\partial V_I}\frac{\partial V_I}{\partial \gamma} + \frac{\partial D}{\partial \gamma_I}\frac{\partial \gamma_I}{\partial \gamma} + 0 + \frac{\partial D}{\partial \psi_I}\frac{\partial \psi_I}{\partial \gamma} + 0 + 0$$

$$\lambda_{r@t} = t_f = \frac{\partial D}{\partial r} = \frac{\partial D}{\partial V_I}\frac{\partial V_I}{\partial r} + \frac{\partial D}{\partial \gamma_I}\frac{\partial \gamma_I}{\partial r} + \frac{\partial D}{\partial r} + \frac{\partial D}{\partial \psi_I}\frac{\partial \psi_I}{\partial r} + 0 + 0$$

$$\lambda_{\psi@t} = t_f = \frac{\partial D}{\partial \psi} = \frac{\partial D}{\partial V_I}\frac{\partial V_I}{\partial \psi} + \frac{\partial D}{\partial \gamma_I}\frac{\partial \gamma_I}{\partial \psi} + 0 + \frac{\partial D}{\partial \psi_I}\frac{\partial \psi_I}{\partial \psi} + 0 + 0$$

$$\lambda_{\lambda@t} = t_f = \frac{\partial D}{\partial \lambda} = \frac{\partial D}{\partial V_I}\frac{\partial V_I}{\partial \lambda} + \frac{\partial D}{\partial \gamma_I}\frac{\partial \gamma_I}{\partial \lambda} + 0 + \frac{\partial D}{\partial \psi_I}\frac{\partial \psi_I}{\partial \lambda} + \frac{\partial D}{\partial \lambda} + 0$$

$$\lambda_{\tau@t} = t_f = \frac{\partial D}{\partial \tau_I} = 0 + 0 + 0 + 0 + 0 + 0$$

H5.5-2 Partial Derivatives of Orbital Elements

H5.5-2a Inclination, i

$$\frac{\partial i}{\partial \lambda} = \frac{\sin \lambda \sin \psi_I}{\sin i}$$

$$\frac{\partial i}{\partial \psi_I} = - \frac{\cos \lambda \cos \psi_I}{\sin i}$$

H5.5-2b Longitude of Ascending Node, Ω_e

$$\frac{\partial \Omega_e}{\partial \lambda} = - \frac{\cos i \cos \psi_I}{\sin^2 i}$$

$$\frac{\partial \Omega_e}{\partial \psi_I} = - \frac{\sin \lambda}{\sin^2 i}$$

$$\frac{\partial \Omega_e}{\partial \tau} = 1$$

H5.5-2c Argument of Perigee, β_p

$$\frac{\partial \beta_p}{\beta V_I} = \frac{2 (\sin^2 \zeta) \frac{\mu}{H} \cdot \frac{r}{H}}{V_I \tan \gamma_I}$$

$$\frac{\partial \beta_p}{\partial r} = \frac{\sin^2 \zeta \frac{\mu}{H}}{H \tan \gamma_I}$$

$$\frac{\partial \beta_p}{\partial \gamma} = -(\sin^2 \zeta)\left[\frac{1 - \frac{\mu}{H}\cdot\frac{r}{H}}{\sin^2 \gamma_I} + 2\cdot\frac{\mu}{H}\cdot\frac{r}{H}\right]$$

$$\frac{\partial \beta_p}{\partial \lambda} = \frac{\sin \beta \cos \beta}{\sin \lambda \cos \lambda}$$

$$\frac{\partial \beta_p}{\partial \psi_I} = \frac{\sin \beta \cos \beta \sin \psi_I}{\cos \psi_I}$$

H5.5-2d Hyperbola's Limiting True Anomaly

$$\frac{\partial \zeta_H}{\partial V_I} = -\frac{\frac{H\cdot V_H}{\mu}\left[1 + \left(\frac{V_I}{V_H}\right)^2\right]}{V_I\left[1 + \left(\frac{H V_H}{\mu}\right)^2\right]}$$

$$\frac{\partial \zeta_H}{\partial \gamma_I} = \frac{\left(\frac{H V_H}{\mu}\right)\tan \gamma_I}{1 + \left(\frac{H V_H}{\mu}\right)^2}$$

$$\frac{\partial \zeta_H}{\partial r} = -\frac{\frac{H V_H}{\mu}\left[1 + \frac{\mu}{r V_H^2}\right]}{r\left[1 + \left(\frac{H V_H}{\mu}\right)^2\right]}$$

H5.5-2e Hyperbola's Limiting In-plane Angle, β_H

$$\frac{\partial \beta_H}{\partial V_I} = \frac{\partial \beta_p}{\partial V_I} + \frac{\partial \zeta_H}{\partial V_I}$$

$$\frac{\partial \beta_H}{\partial \gamma_I} = \frac{\partial \beta_p}{\partial \gamma_I} + \frac{\partial \zeta_H}{\partial \gamma_I}$$

$$\frac{\partial \beta_H}{\partial r} = \frac{\partial \beta_p}{\partial r} + \frac{\partial \zeta_H}{\partial r}$$

$$\frac{\partial \beta_H}{\partial \psi_I} = \frac{\partial \beta_p}{\partial \psi_I}$$

$$\frac{\partial \beta_H}{\partial \lambda} = \frac{\partial \beta_p}{\partial \lambda}$$

H5.5-3 Partial Derivatives of Hyperbolic Asymptote

H5.5-3a Hyperbolic Excess Velocity

$$\frac{\partial V_H}{\partial V_I} = \frac{2\ V_I}{2\ V_H} = \frac{V_I}{V_H}$$

$$\frac{\partial V_H}{\partial r} = \frac{2\ \frac{\mu}{r^2}}{2\ V_H} = \frac{\mu/r^2}{V_H}$$

H5.5-3b Declination of Hyperbolic Asymptote

$$\frac{\partial \delta_{e_H}}{\partial V_I} = \frac{\sin i \cos \beta_H}{\sqrt{1 - \sin^2 \delta_{e_H}}} \frac{\partial \beta_H}{\partial V_I}$$

$$\frac{\partial \delta_{e_H}}{\partial \gamma_I} = \frac{\sin i \cos \beta_H}{\sqrt{1 - \sin^2 \delta_{e_H}}} \frac{\partial \beta_H}{\partial \gamma_I}$$

$$\frac{\partial \delta_{e_H}}{\partial r} = \frac{\sin i \cos \beta_H}{\sqrt{1 - \sin^2 \delta_{e_H}}} \frac{\partial \beta_H}{\partial r}$$

$$\frac{\partial \delta_{e_H}}{\partial \psi_I} = \frac{1}{\sqrt{1 - \sin^2 \delta_{e_H}}} \left(\sin i \cos \beta_H \frac{\partial \beta_H}{\partial \psi_I} + \sin \beta_H \cos i \frac{\partial i}{\partial \psi_I} \right)$$

$$\frac{\partial \delta_{e_H}}{\partial \lambda} = \frac{1}{\sqrt{1 - \sin^2 \delta_{e_H}}} \left(\sin i \cos \beta_H \frac{\partial \beta_H}{\partial \lambda} + \sin \beta_H \cos i \frac{\partial i}{\partial \lambda} \right)$$

H5.5-3c Right Ascension of Hyperbolic Asymptote

$$\frac{\partial \alpha_{e_H}}{\partial V_I} = \frac{\cos i / \cos^2 \beta_H}{\left(1 + \tan^2 \nu_H \right)} \frac{\partial \beta_H}{\partial V_I}$$

$$\frac{\partial \alpha_{e_H}}{\partial r} = \left(\frac{\cos i/\cos^2 \beta_H}{1 + \tan^2 \nu_H} \right) \frac{\partial \beta_H}{\partial r} + \frac{\partial \Omega_e}{\partial \psi_I} \frac{\partial \psi_I}{\partial r}$$

$$\frac{\partial \alpha_{e_H}}{\partial \gamma_I} = \left(\frac{\cos i/\cos^2 \beta_H}{1 + \tan^2 \nu_H} \right) \frac{\partial \beta_H}{\partial \gamma_I}$$

$$\frac{\partial \alpha_{e_H}}{\partial \lambda} = \frac{\partial \Omega_e}{\partial \lambda} + \left(\frac{\cos i/\cos^2 \beta_H}{1 + \tan^2 \nu_H} \right) \frac{\partial \beta_H}{\partial \lambda} - \frac{\tan \beta_H}{1 + \tan^2 \nu_H} \sin i \frac{\partial i}{\partial \lambda}$$

$$\frac{\partial \alpha_{e_H}}{\partial \psi_I} = \frac{\partial \Omega_e}{\partial \psi_I} + \left(\frac{\cos i/\cos^2 \beta_H}{1 + \tan^2 \nu_H} \right) \frac{\partial \beta_H}{\partial \psi_I} - \frac{\tan \beta_H}{1 + \tan^2 \nu_H} \sin i \frac{\partial i}{\partial \psi_I}$$

$$\frac{\partial \alpha_{e_H}}{\partial \tau} = \frac{\partial \Omega_e}{\partial \tau} = 1 \qquad \frac{\partial \alpha_{e_H}}{\partial T_G} = \frac{\partial \Omega_e}{\partial T_G} = \omega$$

H21.1 HYBERBOLIC ASYMPTOTE CODING NOMENCLATURE

Definitions for new variables appearing in ICS, LINEAR, MAIN and MISCON subroutines for the hyperbolic asymptote constraints.

BH	in-plane angle between ascending node and hyperbolic asymptote
CBEH	Cosine (BEta H) = cosine (BH)
CENUH	Cosine (ENUH)
CI	Cosine (Inclination)
DECLIN	DECLINation of hyperbolic asymptote
ENUH	ν_H = inertial longitude angle of hyperbolic asymptote from ascending node
IL101	flag for right ascension constraint

P AEH GI	partial derivatives of α_{e_H} (right ascension of hyperbolic asymptote) with respect to inertial state variables	inertial gamma
P AEH L		Latitude
P AEH R		Radius
P AEH SI		inertial azimuth
P AEH VI		inertial velocity

P BH GI	partial derivatives of BH with respect to inertial state variables	γ_I
P BH L		λ
P BH R		r
P BH SI		ψ_I
P BH VI		V_I

P BP GI	partial derivatives of BP (argument of perigee) with respect to inertial state variables	γ_I
P BP L		λ
P BP R		r
P BP SI		ψ_I
P BP VI		V_I

P DEH GI	partial derivatives of DEclination of the Hyperbolic asymptote with respect to inertial state variables	γ_I
P DEH L		λ
P DEH R		r
P DEH SI		ψ_I
P DEH VI		V_I
P I L	partial derivative of Inclination with respect to:	latitude
P I SI		inertial azimuth
P OME L	partial derivatives of Ω_e (ascending node) with respect to:	latitude
P OME SI		inertial azimuth
P VH R	partial derivatives of VH (hyperbolic excess velocity) with respect to:	radius
P VH VI		inertial velocity
P ZH GI	partial derivatives of ZH (true anomaly of hyperbolic asymptote) with respect to:	γ_I
P ZH R		r
P ZH VI		V_I
RTAS	RighT AScension of hyperbolic asymptote	
SBEH	Sine (BEta H) = sine (BH)	
SENUH	Sine (ENUH)	
SI	Sine (Inclination)	
SIQ	SI sQuared = $(SI)^2$	
SZETA	Sine (ZETA = true anomaly of position)	
SZETAQ	SZETA sQuared = $(SZETA)^2$	
TENUH	Tangent (ENUH)	
VH	magnitude of hyperbolic excess velocity	
ZETAH	ZETA Hyperbola = limiting true anomaly of hyperbolic asymptote	

H22.3 DISCUSSION OF THE DATA BLOCKS

Data Block 8

Three new parameters have been added to the list of terminal constraints; namely, hyperbolic excess velocity, right ascension of the hyperbolic asymptote, and declination of the hyperbolic asymptote. These are terminal constraints which must be specified before listing of intermediate stage points. In general, the Earth-departure missions utilizing these three new constraints will also require imposing the perigee radius constraint for consistent optimization.

H22.4 DEFINITION OF DATA INPUT

Data Block No.	Title	Format
8	Trajectory Constraint Parameters	14I3
**	Terminal Constraint Parameter Codes	**
100	Hyperbolic excess velocity (VH), ft/sec	
101	Right ascension of asymptote, radians	
102	Declination of asymptote, radians	

H22.5 OUTPUT FORMAT

The three new constraint parameters have been added to the list of terminal condition orbit elements that are output, defined as:

HYPERBOLIC EXCESS VELOCITY	ft/sec
ASYMPTOTE DECLINATION	degrees
ASYMPTOTE RIGHT ASCENSION	degrees