1.5

Scientific and Technical Information
Facility (25)
P. O. Box 33
College Park, Maryland 20740
Attn: NASA Representative, S-AK/RKT

NASA TECHNICAL MEMORANDUM

NASA TM X-64636

N72-17905

APPLICATION OF THE STEEPEST ASCENT OPTIMIZATION METHOD TO A REENTRY TRAJECTORY PROBLEM

CASE FILE COPY

By Bobby G. Junkin
Computation Laboratory

December 10, 1971

NASA

George C. Marshall Space Flight Center
Marshall Space Flight Center, Alabama

MSFC - Form 3190 (Rev June 1971)

TECHNICAL REPORT STANDARD TITLE PAGE

1. REPORT NO.	2. GOVERNMENT ACCESSION NO.	3. RECIPIENT'S CATALOG NO.
NASA TM X-64636		

4. TITLE AND SUBTITLE	5. REPORT DATE
Application of the Steepest Ascent Optimization Method to a Reentry Trajectory Problem	December 10, 1971
	6. PERFORMING ORGANIZATION CODE
7. AUTHOR (S) Bobby G. Junkin	8. PERFORMING ORGANIZATION REPORT #
9. PERFORMING ORGANIZATION NAME AND ADDRESS George C. Marshall Space Flight Center Marshall Space Flight Center, Alabama 35812	10. WORK UNIT NO.
	11. CONTRACT OR GRANT NO.
	13. TYPE OF REPORT & PERIOD COVERED Technical Memorandum
12. SPONSORING AGENCY NAME AND ADDRESS National Aeronautics and Space Administration Washington, D. C. 20546	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

Prepared by Computation Laboratory, Science and Engineering

16. ABSTRACT

The direct optimization method of steepest ascent is presented in detail. Nominal values of the control variables are input parameters. Perturbations are introduced into the control variables and the resulting first order predictions of changes in the payoff, and constraint functions are then determined. Through a sequence of prescribed cycles, a trajectory is eventually obtained which is reasonably close to the optimum. The method is successfully applied to an Apollo three-dimensional reentry problem. The study of this Apollo application problem has resulted in the development of a highly flexible computer program that can be modified to consider other trajectory optimization problems.

17. KEY WORDS		18. DISTRIBUTION STATEMENT
Optimization	Adjoint equations	Unclassified-Unlimited
Apollo re-entry problem	Control variables	Bobby G. Junkin
Steepest ascent	Computer programs	
Direct method	MARVES system	
Perturbation equations	Iterative solution	

19. SECURITY CLASSIF. (of this report)	20. SECURITY CLASSIF. (of this page)	21. NO. OF PAGES	22. PRICE
Unclassified	Unclassified	108	$3.00

MSFC - Form 3292 (May 1969)

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

DEFINITION OF SYMBOLS

Symbol	Definition
A	Heading angle
C_L, C_D	Lift and drag coefficients
$d()$	Total differential
h	Altitude
m	Number of control variables
$\tilde{m}$	Spacecraft mass
n	Number of state variables
p	Number of terminal constraints
q	Extra state variable
R	Radius of earth
S	Spacecraft cross-sectional area
v	Velocity
$\bar{x}$	State variable vector
$\bar{\alpha}$	Control variable vector
β	Roll angle
β^*	Model parameter for atmosphere
γ	Angle of attack
$\delta()$	Total variation

DEFINITION OF SYMBOLS (Concluded)

Symbol	Definition
$\delta^2(\,)$	Second variation
Δ	Latitude
θ	Longitude
$\bar{\lambda}$	Adjoint variables
$\tilde{\lambda}_o$	Weight number for heating term
μ	Constant Lagrange multiplier
μ^*	Gravitational constant
ν	$p \times 1$ vector of constant Lagrange multipliers
ρ_0	Atmospheric reference density
$\bar{\phi}$	Maximizing function
$\bar{\Psi}$	Terminal constraint vector
$\bar{\Omega}$	Stopping constraint
$[\,]^*$	Partial derivative evaluation for nominal estimates

ACKNOWLEDGEMENT

The author gratefully acknowledges the cooperation and programming efforts of Mr. Bill Keenum, Computer Sciences Corporation, in connection with the development of the two program decks in the appendix.

TECHNICAL MEMORANDUM X-64636

APPLICATION OF THE STEEPEST ASCENT OPTIMIZATION TRAJECTORY PROBLEM METHOD TO A REENTRY

INTRODUCTION

Trajectory optimization problems are usually concerned with the task of controlling a dynamical system such that a particular mission trajectory is accomplished with some measure of performance being extremized (maximum or minimum). Once this measure of performance is selected, it is used with the system equations and the initial and terminal boundary conditions to formulate the optimization problem. The mathematical details of a particular formulation depends, in general, upon the complexity of the problem. The methods resulting from the various formulations can be grouped into two classes: (1) direct and (2) indirect. Most direct methods are based upon the results of Kelly [1, 2] and Bryson and Denham [3] known as the method of steepest ascent while the indirect methods stem from either (1) the calculus of variations [4], (2) Pontryagains' maximum principle [5], or (3) dynamic programming [6]. To obtain explicit solutions to the optimization problem using the indirect methods, a nonlinear two-point boundary problem must be solved. This difficulty is circumvented when the direct method of steepest ascent is used.

The purpose of this report is to present an application of the above mentioned steepest ascent method to an Apollo three-dimensional reentry optimization problem. This particular problem has been investigated by Colunga [7] using a modified sweep method (MSM). The MSM is a second-order indirect numerical optimization method whereas the steepest ascent is of first order.

THE DIRECT METHOD OF STEEPEST ASCENT

Discussion

Steepest ascent is an iterative procedure in which the nominal or beginning values of the control variables must be supplied by the analyst. Optimum values for the control variables are determined through a sequence of perturbations to the control variables; i.e., the control variables are perturbed by a certain amount and the resulting first-order predictions of changes in the payoff and constraint functions are determined. The steepest ascent method then seeks the perturbed control variable time history which results in maximizing or minimizing the payoff function while simultaneously satisfying the constraints. The steepest ascent theory can be summarized as follows: If one goes through a sequence of the prescribed cycles, which results in improved trajectories, then eventually a trajectory is obtained which is reasonably close to the optimum. The logic flow is depicted in Figure 1 which is a block diagram summary of the procedure.

Comment 1: As the optimum solution is approached, the gradient $\frac{d\phi}{dp}$ must tend to zero. If this slope is below some acceptable level, then we have obtained the solution. If this slope is not below an acceptable level, then we have to repeat the procedure.

Problem Formulation

Consider a system defined by the n state variables:

$$\underset{(n \times 1)}{\bar{x}(t)} = \begin{bmatrix} x_1(t) \\ x_2(t) \\ \cdot \\ \cdot \\ \cdot \\ x_n(t) \end{bmatrix} . \tag{1}$$

These state variables are subject to the m control variables:

$$\underset{(m \times 1)}{\bar{\alpha}(t)} = \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \cdot \\ \cdot \\ \cdot \\ \alpha_m(t) \end{bmatrix} . \tag{2}$$

The optimization problem consists of determining the control matrix $\bar{\alpha}(t)$ in the interval t_0 to T so as to maximize the function:

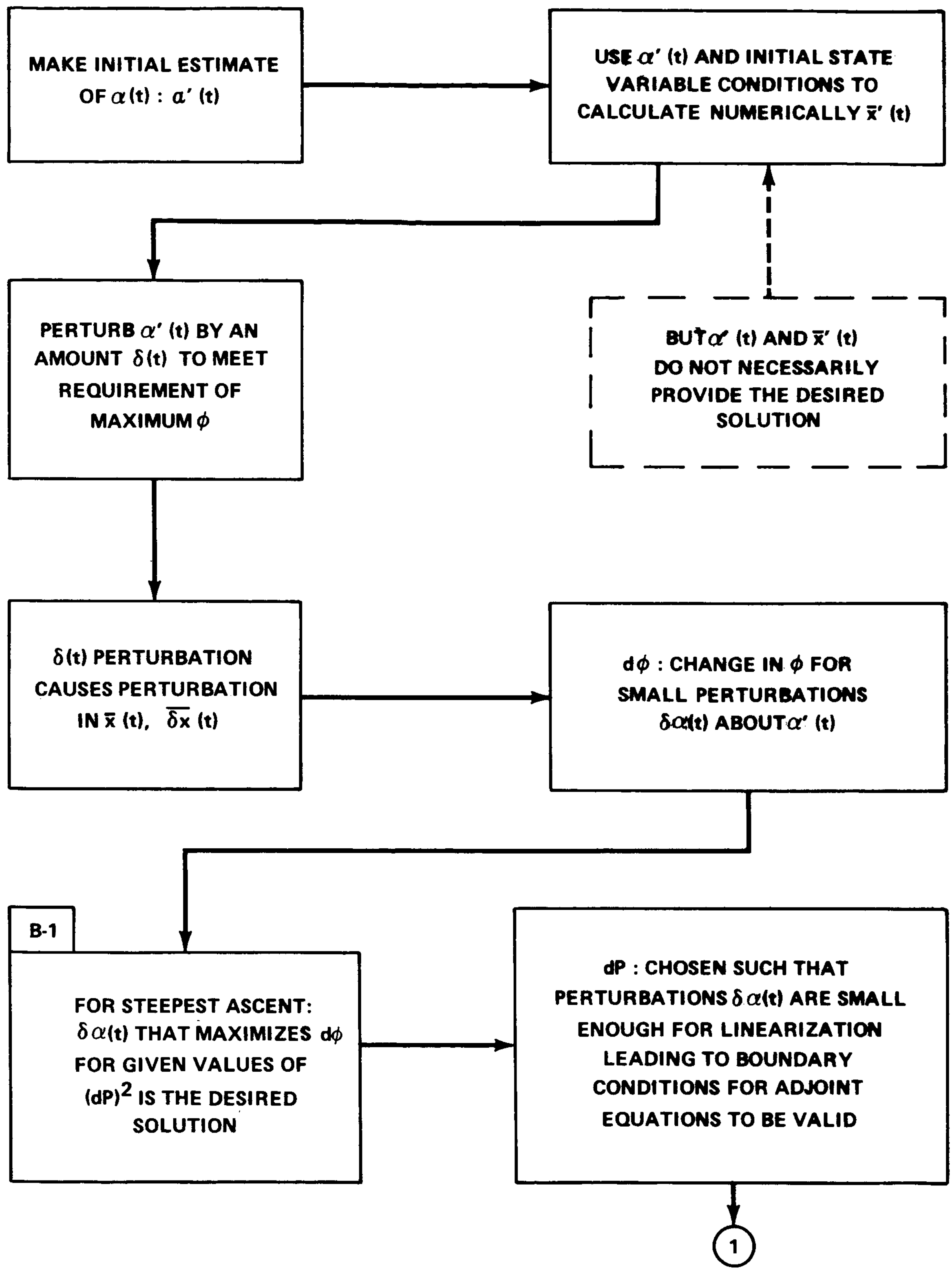

Figure 1. Summary of steepest ascent procedure.

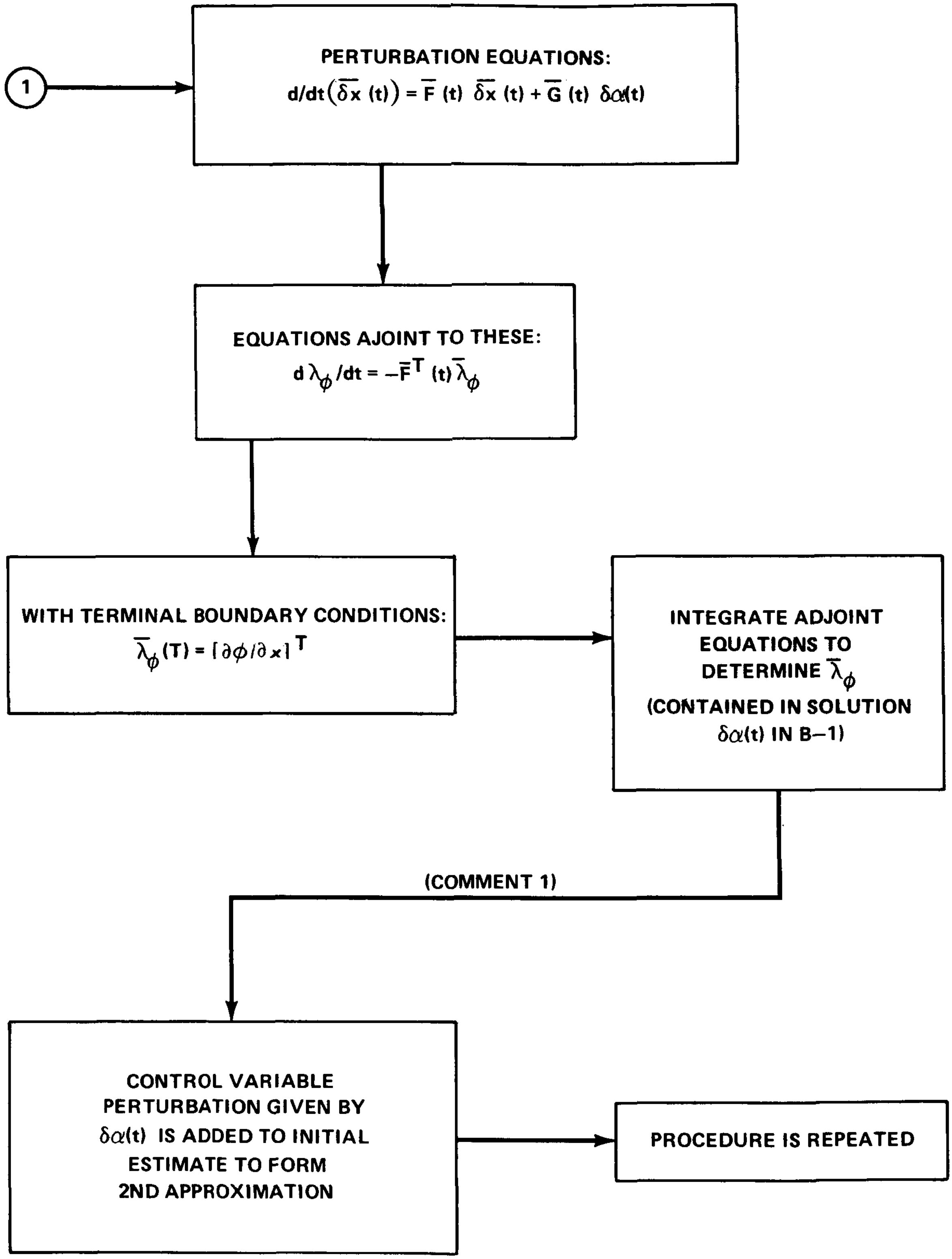

Figure 1. (Concluded).

$$\bar{\phi} = \phi\,[x_1(T), x_2(T), \ldots, x_n(T), T] \tag{3}$$

while satisfying the p terminal constraint functions:

$$\underset{(p \times 1)}{\bar{\Psi}} = \begin{bmatrix} \Psi_1[x_1(T), x_2(T), \ldots, x_n(T), T] \\ \Psi_2[x_1(T), x_2(T), \ldots, x_n(T), T] \\ \cdot \\ \cdot \\ \cdot \\ \Psi_p[x_1(T), x_2(T), \ldots, x_n(T), T] \end{bmatrix} = 0 \tag{4}$$

and the single stopping condition:

$$\bar{\Omega} = \Omega\,[x_1(T), x_2(T), \ldots, x_n(T), T] = 0 \tag{5}$$

and the n system equations:

$$\underset{(n \times 1)}{\bar{f}} = \begin{bmatrix} \frac{dx_1}{dt} \\ \frac{dx_2}{dt} \\ \cdot \\ \cdot \\ \cdot \\ \frac{dx_n}{dt} \end{bmatrix} = \begin{bmatrix} f_1[x_1(t), x_2(t), \ldots, x_n(t), \alpha_1(t), \alpha_2(t), \ldots, \alpha_m(t), t] \\ f_2[x_1(t), x_2(t), \ldots, x_n(t), \alpha_1(t), \alpha_2(t), \ldots, \alpha_m(t), t] \\ \cdot \\ \cdot \\ \cdot \\ f_n[x_1(t), x_2(t), \ldots, x_n(t), \alpha_1(t), \alpha_2(t), \ldots, \alpha_m(t), t] \end{bmatrix} . \tag{6}$$

The optimization problem thus formulated can be solved by using the steepest ascent method developed by Bryson and Denham [3]. It is assumed that the initial conditions $x(t_0)$ are specified.

To start the procedure an initial estimate must be made of the control variables in the interval between t_0 and the time at which equation (5) is satisfied. Denote this estimate by $\bar{\alpha}^{(1)}(t)$ and use it and the initial conditions and equations (6) to numerically determine $\bar{x}^{(1)}(T)$; i.e., substitute $\bar{\alpha}^{(1)}(t)$ in equations (6) and numerically integrate the resulting equations from the initial state until the stopping condition of equation (5) is reached. This yields the state variables $\bar{x}^{(1)}(T)$ and the final time T resulting from the estimate $\bar{\alpha}^{(1)}(t)$. The values of $\bar{x}^{(1)}(T)$ and T will not necessarily satisfy $\bar{\Psi}$ [equations (4)] or maximize $\bar{\phi}$ [equation (3)]. Thus, $\bar{\alpha}^{(1)}(t)$ must be changed by an amount $\overline{\delta\alpha}$ to meet these requirements.

Linear Differential Perturbation Equations

Towards this objective, consider small perturbations $\delta\alpha$ about the initial estimates of the control variables. This would lead to a second approximation given by:

$$\alpha_i^{(2)}(t) = \alpha_i^{(1)}(t) + \delta\alpha_i \quad , \tag{7}$$

where $i = 1, 2, \ldots, m$. The perturbations cause perturbations in the state variables:

$$x_j^{(2)}(t) = x_j^{(1)} + \delta x_j \tag{8}$$

where $j = 1, 2, \ldots, n$. Next, take the variations of equations (6) by first substituting equations (7) and (8) into equation (6) and then expanding the right side of equation (6) in Taylor's series about the nominal estimates $\alpha_i^{(1)}(t)$ and $x_j^{(1)}(t)$:

$$f_i\left[x_1^{(2)}(t), x_2^{(2)}(t), \ldots, x_n^{(2)}(t), \alpha_1^{(2)}(t), \alpha_2^{(2)}(t), \ldots, \alpha_m^{(2)}(t), t\right]$$

$$= f_i\left[x_1^{(1)}(t), x_2^{(1)}(t), \ldots, x_n^{(1)}(t), \alpha_1^{(1)}(t), \alpha_2^{(1)}(t), \ldots, \alpha_m^{(1)}(t), t\right]$$

$$+ \sum_{j=1}^{n}\left[\frac{\partial f_i}{\partial x_j}\left(x_j^{(2)}(t) - x_j^{(1)}(t)\right)\right] + \sum_{k=1}^{m}\left[\frac{\partial f_i}{\partial \alpha_k}\left(\alpha_k^{(2)}(t) - \alpha_k^{(1)}(t)\right)\right] \tag{9}$$

where $i = 1, 2, \ldots, n$. The system of equations (9) can be written as:

$$\delta f_i \equiv f_i\left[x_1^{(1)}(t) + \delta x_1, x_2^{(1)} + \delta x_2, \ldots, x_n^{(1)}(t) + \delta x_m, \alpha_1^{(1)}(t) + \delta\alpha_1,\right.$$

$$\left.\alpha_2^{(1)}(t) + \delta\alpha_2, \ldots, \alpha_m^{(1)}(t) + \delta\alpha_m, t\right] - f_i\left[x_1^{(1)}(t), x_2^{(1)}(t),\right.$$

$$\left.\ldots, x_n^{(1)}(t), \alpha_1^{(1)}(t), \alpha_2^{(1)}(t), \ldots, \alpha_m^{(1)}(t), t\right]$$

$$= \sum_{j=1}^{n}\left\{\frac{\partial f_i}{\partial x_j}\left[x_j^{(2)}(t) - x_j^{(1)}(t)\right]\right\} + \sum_{k=1}^{m}\left\{\frac{\partial f_i}{\partial \alpha_k}\left[\alpha_k^{(2)}(t) - \alpha_k^{(1)}(t)\right]\right\}$$

$$= \sum_{j=1}^{n}\left[\frac{\partial f_i}{\partial x_j}\delta x_j(t)\right] + \sum_{k=1}^{m}\left[\frac{\partial f_i}{\partial \alpha_k}\delta\alpha_k(t)\right] \tag{10}$$

where $i = 1, 2, \ldots, n$. Thus, the variations in $\alpha_i^{(1)}(t)$ and $x_j^{(1)}(t)$ induce a variation in f denoted as δf_i. But from the variational calculus, we can write by combining equations (6) and (10) [8]:

$$\delta f_i = \delta\left(\frac{dx_i}{dt}\right) = \frac{d}{dt}\left[\delta x_i(t)\right] \tag{11}$$

or

$$\frac{d}{dt}\left[\delta x_i(t)\right] = \sum_{j=1}^{n}\left[\frac{\partial f_i}{\partial x_j}\delta x_j(t)\right] + \sum_{k=1}^{m}\left[\frac{\partial f_i}{\partial \alpha_k}\delta\alpha_k(t)\right] \tag{12}$$

where $i = 1, 2, \ldots, n$. These equations can be written in matrix notation as:

$$\frac{d}{dt}\left[\overline{\delta x}\,(t)\right] = \overline{F}\,(t)\,\overline{\delta x}\,(t) + \overline{G}\,(t)\,\overline{\delta\alpha}\,(t) \tag{13}$$

where

$$\underset{(n \times 1)}{\frac{d}{dt}\left[\overline{\delta x}\,(t)\right]} = \begin{bmatrix} \frac{d}{dt}\left[\delta x_1\,(t)\right] \\ \frac{d}{dt}\left[\delta x_2\,(t)\right] \\ \cdot \\ \cdot \\ \cdot \\ \frac{d}{dt}\left[\delta x_n\,(t)\right] \end{bmatrix} \tag{14}$$

$$\underset{(n \times 1)}{\overline{\delta x}\,(t)} = \begin{bmatrix} \delta x_1\,(t) \\ \delta x_2\,(t) \\ \cdot \\ \cdot \\ \cdot \\ \delta x_n\,(t) \end{bmatrix} \tag{15}$$

$$\underset{(m \times 1)}{\overline{\delta\alpha}\,(t)} = \begin{bmatrix} \delta\alpha_1\,(t) \\ \delta\alpha_2\,(t) \\ \cdot \\ \cdot \\ \cdot \\ \delta\alpha_m\,(t) \end{bmatrix} \tag{16}$$

$$\underset{(n \times n)}{\overline{F}(t)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & & & \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}^* \tag{17}$$

$$\underset{(n \times m)}{\overline{G}(t)} = \begin{bmatrix} \frac{\partial f_1}{\partial \alpha_1} & \frac{\partial f_1}{\partial \alpha_2} & \cdots & \frac{\partial f_1}{\partial \alpha_m} \\ \frac{\partial f_2}{\partial \alpha_1} & \frac{\partial f_2}{\partial \alpha_2} & \cdots & \frac{\partial f_2}{\partial \alpha_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial f_n}{\partial \alpha_1} & \frac{\partial f_n}{\partial \alpha_2} & \cdots & \frac{\partial f_n}{\partial \alpha_m} \end{bmatrix} . \tag{18}$$

The $[\,]^*$ indicates that the partial derivatives are evaluated for the nominal estimates. The system of linear differential perturbation equations for $\overline{\delta x}$, as given by equations (13), play an important role in determining the changes in $\overline{\Phi}$, $\overline{\Psi}$, and $\overline{\Omega}$ as caused by control variable perturbations $\overline{\delta\alpha}$ (t). The following section ascertains this role.

Changes in $\overline{\Phi}$, $\overline{\Psi}$, and $\overline{\Omega}$ Resulting From Control Variable Perturbations

We now wish to determine the total changes $\overline{d\phi}$, $\overline{d\Psi}$, and $\overline{d\Omega}$ in $\overline{\phi}$, $\overline{\Psi}$, and $\overline{\Omega}$, respectively, for small perturbations $\overline{\delta\alpha}$ in the control variables about the nominal estimates. To do this we introduce the linear differential equations adjoint to equations (13) and defined as [9]:

$$\frac{d}{dt}\left[\bar{\lambda}_{\phi}(t)\right] = -\bar{F}^{T}(t)\,\bar{\lambda}_{\phi}(t) \tag{19}$$

$$\frac{d}{dt}\left[\bar{\lambda}_{\Psi}(t)\right] = -\bar{F}^{T}(t)\,\bar{\lambda}_{\Psi}(t) \tag{20}$$

$$\frac{d}{dt}\left[\bar{\lambda}_{\Omega}(t)\right] = -\bar{F}^{T}(t)\,\bar{\lambda}_{\Omega}(t) \tag{21}$$

where

$$\underset{(n\times 1)}{\frac{d}{dt}\left[\bar{\lambda}_{\phi}(t)\right]} = \begin{bmatrix} \frac{d\lambda_{\phi 1}}{dt} \\ \frac{d\lambda_{\phi 2}}{dt} \\ \cdot \\ \cdot \\ \cdot \\ \frac{d\lambda_{\phi n}}{dt} \end{bmatrix} \tag{22}$$

$$\underset{(n\times 1)}{\bar{\lambda}_{\phi}(t)} = \begin{bmatrix} \lambda_{\phi 1} \\ \lambda_{\phi 2} \\ \cdot \\ \cdot \\ \cdot \\ \lambda_{\phi n} \end{bmatrix} \tag{23}$$

$$\frac{d}{dt}\underset{(n\times p)}{\left[\bar{\lambda}_{\Psi}(t)\right]} = \begin{bmatrix} \frac{d\lambda_{\Psi 11}}{dt} & \frac{d\lambda_{\Psi 12}}{dt} & \cdots & \frac{d\lambda_{\Psi 1p}}{dt} \\ \frac{d\lambda_{\Psi 21}}{dt} & \frac{d\lambda_{\Psi 22}}{dt} & \cdots & \frac{d\lambda_{\Psi 2p}}{dt} \\ \vdots & \vdots & & \vdots \\ \frac{d\lambda_{\Psi n1}}{dt} & \frac{d\lambda_{\Psi n2}}{dt} & \cdots & \frac{d\lambda_{\Psi np}}{dt} \end{bmatrix} \tag{24}$$

$$\underset{(n\times p)}{\bar{\lambda}_{\Psi}(t)} = \begin{bmatrix} \lambda_{\Psi 11} & \lambda_{\Psi 12} & \cdots & \lambda_{\Psi 1p} \\ \lambda_{\Psi 21} & \lambda_{\Psi 22} & \cdots & \lambda_{\Psi 2p} \\ \vdots & \vdots & & \vdots \\ \lambda_{\Psi n1} & \lambda_{\Psi n2} & & \lambda_{\Psi np} \end{bmatrix} \tag{25}$$

$$\frac{d}{dt}\underset{(n\times 1)}{\left[\bar{\lambda}_{\Omega}(t)\right]} = \begin{bmatrix} \frac{d\lambda_{\Omega 1}}{dt} \\ \frac{d\lambda_{\Omega 2}}{dt} \\ \vdots \\ \frac{d\lambda_{\Omega n}}{dt} \end{bmatrix} \tag{26}$$

$$\underset{(n \times 1)}{\bar{\lambda}_{\Omega}(t)} = \begin{bmatrix} \bar{\lambda}_{\Omega 1} \\ \bar{\lambda}_{\Omega 2} \\ \cdot \\ \cdot \\ \cdot \\ \bar{\lambda}_{\Omega n} \end{bmatrix} . \tag{27}$$

By definition, the boundary conditions for these equations are given by the following:

$$\bar{\lambda}_{\phi}(T) = \left[\left(\overline{\frac{\partial \phi}{\partial x}} \right)^{*}_{t = T} \right]^{T} \tag{28}$$

$$\bar{\lambda}_{\Psi}(T) = \left[\left(\overline{\frac{\partial \Psi}{\partial x}} \right)^{*}_{t = T} \right]^{T} \tag{29}$$

$$\bar{\lambda}_{\Omega}(T) = \left[\left(\overline{\frac{\partial \Omega}{\partial x}} \right)^{*}_{t = T} \right]^{T} \tag{30}$$

where

$$\underset{(1 \times n)}{\overline{\frac{\partial \phi}{\partial x}}} = \begin{bmatrix} \frac{\partial \phi}{\partial x_1} & \frac{\partial \phi}{\partial x_2} & \cdots & \frac{\partial \phi}{\partial x_n} \end{bmatrix} \tag{31}$$

$$\underset{(p \times n)}{\overline{\frac{\partial \Psi}{\partial x}}} = \begin{bmatrix} \frac{\partial \Psi_1}{\partial x_1} & \frac{\partial \Psi_1}{\partial x_2} & \cdots & \frac{\partial \Psi_1}{\partial x_n} \\ \frac{\partial \Psi_2}{\partial x_1} & \frac{\partial \Psi_2}{\partial x_2} & \cdots & \frac{\partial \Psi_2}{\partial x_n} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial \Psi_p}{\partial x_1} & \frac{\partial \Psi_p}{\partial x_2} & \cdots & \frac{\partial \Psi_p}{\partial x_n} \end{bmatrix} \tag{32}$$

$$\overline{\frac{\partial \Omega}{\partial x}} = \left[\frac{\partial \Omega}{\partial x_1} \; \frac{\partial \Omega}{\partial x_2} \; \cdots \; \frac{\partial \Omega}{\partial x_n} \right] \quad . \tag{33}$$

The $( )^*$ indicates that the partial derivatives are evaluated for the nominal estimates. If we now take the transpose of equation (19), post-multiply by $\overline{\delta x}$ and pre-multiply equation (13) by $\bar{\lambda}_\phi^T$ and add the results, the following is obtained:

$$\underset{(1\times n)}{\bar{\lambda}_\phi^T} \frac{d}{dt}\underset{(n\times 1)}{\left[\overline{\delta x}\,(t)\right]} + \frac{d}{dt}\underset{(1\times n)}{\left[\bar{\lambda}_\phi^T\,(t)\right]} \underset{(n\times 1)}{\overline{\delta x}\,(t)} = \underset{(1\times n)}{\bar{\lambda}_\phi^T (t)} \; \underset{(n\times m)}{\bar{G}\,(t)} \; \underset{(m\times 1)}{\overline{\delta\alpha}(t)} \quad . \tag{34}$$

We can also write:

$$\frac{d\left[\bar{\lambda}_\phi^T (t)\, \overline{\delta x}\,(t)\right]}{dt} = \bar{\lambda}_\phi^T (t) \frac{d}{dt}\left[\overline{\delta x}\,(t)\right] + \frac{d}{dt}\left[\bar{\lambda}_\phi^T (t)\right] \overline{\delta x}\,(t) \quad . \tag{35}$$

If we now substitute equation (35) into equation (34), the following equation is obtained:

$$\frac{d\left[\bar{\lambda}_\phi^T \,(t)\; \overline{\delta x}\,(t)\right]}{dt} = \bar{\lambda}_\phi^T (t)\; \bar{G}\,(t)\; \overline{\delta\alpha}\,(t) \quad . \tag{36}$$

By integrating equations (36) from t_0 to T, we obtain:

$$\int_{t_0}^{T} d\left[\bar{\lambda}_\phi^T (t)\; \overline{\delta x}\,(t)\right] = \int_{t_0}^{T} \bar{\lambda}_\phi^T (t)\; \bar{G}\,(t)\; \overline{\delta\alpha}\,(t)\; dt \tag{37}$$

or

$$\left[\bar{\lambda}_{\phi}^{T}(t)\,\overline{\delta x}\,(t)\right]_{t=T} - \left[\bar{\lambda}_{\phi}^{T}(t)\,\overline{\delta x}\,(t)\right]_{t=t_0}$$

$$= \int_{t_0}^{T} \bar{\lambda}_{\phi}^{T}\,(t)\;\bar{G}\,(t)\;\overline{\delta\alpha}\,(t)\;dt \quad . \tag{38}$$

In an analogous manner,

$$\left[\bar{\lambda}_{\Psi}^{T}\,(t)\,\overline{\delta x}\,(t)\right]_{t=T} - \left[\bar{\lambda}_{\Psi}^{T}\,(t)\,\overline{\delta x}\,(t)\right]_{t=t_0}$$

$$= \int_{t_0}^{T} \bar{\lambda}_{\Psi}^{T}\,(t)\;\bar{G}\,(t)\;\overline{\delta\alpha}\,(t)\;dt \quad . \tag{39}$$

$$\left[\bar{\lambda}_{\Omega}^{T}(t)\,\overline{\delta x}\,(t)\right]_{t=T} - \left[\bar{\lambda}_{\Omega}^{T}(t)\,\overline{\delta x}\,(t)\right]_{t=t_0}$$

$$= \int_{t_0}^{T} \bar{\lambda}_{\Omega}^{T}\,(t)\;\bar{G}\,(t)\;\overline{\delta\alpha}\,(t)\;dt \quad . \tag{40}$$

We now consider the functions $\bar{\phi}, \bar{\Psi}$, and $\bar{\Omega}$ that are given by equations (3), (4), and (5). The total differentials of these expressions are:

$$\left.\begin{aligned}
\underset{(1\times 1)}{\overline{d\phi}} &= \overline{\frac{\partial \phi}{\partial x}}\;\overline{dx} + \frac{\partial \phi}{\partial T}\;dT \\[2ex]
\underset{(p\times 1)}{\overline{d\Psi}} &= \overline{\frac{\partial \Psi}{\partial x}}\;\overline{dx} + \frac{\partial \Psi}{\partial T}\;dT \\[2ex]
\underset{(1\times 1)}{\overline{d\Omega}} &= \overline{\frac{\partial \Omega}{\partial x}}\;\overline{dx} + \frac{\partial \Omega}{\partial T}\;dT
\end{aligned}\right\} \tag{41}$$

where

$$\underset{(n \times 1)}{\overline{dx}} = \begin{bmatrix} dx_1 \\ dx_2 \\ \cdot \\ \cdot \\ \cdot \\ dx_n \end{bmatrix} \tag{42}$$

But equations (42) can be expressed as (see Reference 9 for a discussion of these equations):

$$\begin{bmatrix} dx_1 \\ dx_2 \\ \cdot \\ \cdot \\ \cdot \\ dx_n \end{bmatrix} = \begin{bmatrix} \delta x_1 (T) \\ \delta x_2 (T) \\ \cdot \\ \cdot \\ \cdot \\ \delta x_n (T) \end{bmatrix} + \bar{f}\, dT \tag{43}$$

or

$$\overline{dx} = \overline{\delta x} (T) + \bar{f}\, dT \quad . \tag{44}$$

By substituting equations (44) into equations (41),

$$\left.\begin{aligned} \overline{d\phi} &= \overline{\delta\phi} + \dot{\phi}\ dT \\ \overline{d\Psi} &= \overline{\delta\Psi} + \dot{\Psi}\ dT \\ \overline{d\Omega} &= \overline{\delta\Omega} + \dot{\Omega}\ dT \end{aligned}\right\} \tag{45}$$

where

$$
\left.\begin{aligned}
\underset{(1\times 1)}{\overline{\delta\phi}} &= \overline{\frac{\partial \phi}{\partial x}}\ \overline{\delta x}\ (T) \\
\underset{(p\times 1)}{\overline{\delta\Psi}} &= \overline{\frac{\partial \Psi}{\partial x}}\ \overline{\delta x}\ (T) \\
\underset{(1\times 1)}{\overline{\delta\Omega}} &= \overline{\frac{\partial \Omega}{\partial x}}\ \overline{\delta x}\ (T)
\end{aligned}\right\} \tag{46}
$$

$$
\left.\begin{aligned}
\underset{(1\times 1)}{\dot{\phi}} &= \left(\frac{\partial \phi}{\partial t} + \overline{\frac{\partial \phi}{\partial x}}\ \bar{f}\right)_{t = T} \\
\underset{(p\times 1)}{\dot{\Psi}} &= \left(\frac{\partial \Psi}{\partial t} + \overline{\frac{\partial \Psi}{\partial x}}\ \bar{f}\right)_{t = T} \\
\underset{(1\times 1)}{\dot{\Omega}} &= \left(\frac{\partial \Omega}{\partial t} + \overline{\frac{\partial \Omega}{\partial x}}\ \bar{f}\right)_{t = T}
\end{aligned}\right\} . \tag{47}
$$

If we compare equations (46) with equations (28), (29), and (30), we see that :

$$
\left.\begin{aligned}
\overline{\delta\phi} &= \bar{\lambda}_{\phi}^{T}\ (t)\ \overline{\delta x}\ (T) \\
\overline{\delta\Psi} &= \bar{\lambda}_{\Psi}^{T}\ (t)\ \overline{\delta x}\ (T) \\
\overline{\delta\Omega} &= \bar{\lambda}_{\Omega}^{T}\ (t)\ \overline{\delta x}\ (T)
\end{aligned}\right\} . \tag{48}
$$

Substituting equations (48) into equations (45) and rearranging:

$$
\left.\begin{aligned}
\bar{\lambda}_{\phi}^{T}\ (T)\ \overline{\delta x}\ (T) &= \overline{d\phi} - \dot{\phi}\ dT \\
\bar{\lambda}_{\Psi}^{T}\ (T)\ \overline{\delta x}\ (T) &= \overline{d\Psi} - \dot{\Psi}\ dT \\
\bar{\lambda}_{\Omega}^{T}\ (T)\ \overline{\delta x}\ (T) &= \overline{d\Omega} - \dot{\Omega}\ dT
\end{aligned}\right\} . \tag{49}
$$

We can now substitute equations (49) into equations (38), (39), and (40) to obtain:

$$\left.\begin{aligned}
\overline{d\phi} &= \int_{t_0}^{T} \bar{\lambda}_{\phi}^{T}(t)\, \bar{G}(t)\, \overline{\delta\alpha}(t)\, dt + \left[\bar{\lambda}_{\phi}^{T}(t)\, \overline{\delta x}(t)\right]_{t=t_0} + \dot{\phi}\, dT \\
\overline{d\Psi} &= \int_{t_0}^{T} \bar{\lambda}_{\Psi}^{T}(t)\, \bar{G}(t)\, \overline{\delta\alpha}(t)\, dt + \left[\bar{\lambda}_{\Psi}^{T}(t)\, \overline{\delta x}(t)\right]_{t=t_0} + \dot{\Psi}\, dT \\
\overline{d\Omega} &= 0 = \int_{t_0}^{T} \bar{\lambda}_{\Omega}^{T}(t)\, \bar{G}(t)\, \overline{\delta\alpha}(t)\, dt + \left[\bar{\lambda}_{\Omega}^{T}(t)\, \overline{\delta x}(t)\right]_{t=t_0} + \dot{\Omega}\, dT
\end{aligned}\right\} . \tag{50}$$

These are the desired changes in $\bar{\phi}$, $\bar{\Psi}$, and $\bar{\Omega}$ for the control variable perturbations $\overline{\delta\alpha}(t)$.

Determination of the Control Variable Perturbations

We now wish to find the $\overline{\delta\alpha}$ that maximizes $\overline{d\phi}$ in the first expression of equation (50) for given values of:[1]

1) $$(dp)^2 = \int_{t_0}^{T} \overline{\delta\alpha}^{T}(t)\, \bar{w}\, \overline{\delta\alpha}(t)\, dt \tag{51}$$

2) $\overline{d\Psi}$

3) $\overline{d\Omega}$.

First, we solve the third equation of equation (50) for dT and eliminate it from the first and second expressions of equation (50):

$$dT = \frac{1}{\dot{\Omega}}\left[\int_{t_0}^{T} \bar{\lambda}_{\Omega}^{T}(t)\, \bar{G}(t)\overline{\delta\alpha}(t)\, dt + \bar{\lambda}_{\Omega}^{T}(t_0)\, \overline{\delta x}(t_0)\right] . \tag{52}$$

1. The matrix $\bar{w}$ is an $m \times m$ symmetric weighting matrix chosen to improve convergence of the procedure.

Substituting equation (52) into the first and second expressions of equation (50),

$$\overline{d\phi} = \int_{t_0}^{T} \bar{\lambda}_{\phi}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \bar{\lambda}_{\phi}^{T}(t_0)\ \overline{\delta x}(t_0)$$

$$+\dot{\phi}\left[-\frac{1}{\dot{\Omega}}\left(\int_{t_0}^{T} \bar{\lambda}_{\Omega}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \bar{\lambda}_{\Omega}^{T}(t_0)\ \overline{\delta x}(t_0)\right)\right] \tag{53}$$

$$\overline{d\Psi} = \int_{t_0}^{T} \bar{\lambda}_{\Psi}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \bar{\lambda}_{\Psi}^{T}(t_0)\ \overline{\delta x}(t_0)$$

$$+\dot{\Psi}\left[-\frac{1}{\dot{\Omega}}\left(\int_{t_0}^{T} \bar{\lambda}_{\Omega}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \bar{\lambda}_{\Omega}^{T}(t_0)\overline{\delta x}(t_0)\right)\right] \tag{54}$$

or

$$\overline{d\phi} = \int_{t_0}^{T} \bar{\lambda}_{\phi 1}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \bar{\lambda}_{\phi 1}^{T}(t_0)\ \overline{\delta x}(t_0) \tag{55}$$

$$\overline{d\Psi} = \int_{t_0}^{T} \bar{\lambda}_{\Psi 1}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \bar{\lambda}_{\Psi 1}^{T}(t_0)\ \overline{\delta x}(t_0) \tag{56}$$

where

$$\left.\begin{aligned}
\underset{(1\times n)}{\bar{\lambda}_{\phi 1}^{T}(t)} &= \underset{(1\times n)}{\bar{\lambda}_{\phi}^{T}(t)} - \frac{\dot{\phi}}{\dot{\Omega}}\ \underset{(1\times n)}{\bar{\lambda}_{\Omega}^{T}(t)} \\
\underset{(p\times n)}{\bar{\lambda}_{\Psi 1}^{T}(t)} &= \underset{(p\times n)}{\bar{\lambda}_{\Psi}^{T}(t)} - \frac{\dot{\Psi}}{\dot{\Omega}}\ \bar{\lambda}_{\Omega}^{T}(t)
\end{aligned}\right\} . \tag{57}$$

As stated before, the problem is to find the control variable perturbation to maximize $\overline{d\phi}$ for given values of $\overline{d\Psi}$ and dp, with $\overline{d\Omega} = 0$. This is accomplished by first forming a linear combination of equation (55) with equations (51) and (56) through the use of the Lagrange multipliers ν and μ. The second variation of this linear combination is then formed. It is shown in Reference 3 that this consideration leads to the following expression for the desired control variable perturbation for $\overline{\delta\alpha}$ (t):

$$\underset{(m\times 1)}{\overline{\delta\alpha}(t)} = \frac{1}{2\mu} \underset{(m\times m)}{\overline{w}^{-1}} \underset{(m\times n)}{\overline{G}^{T}(t)} \left[\underset{(n\times 1)}{\bar{\lambda}_{\phi 1}(t)} - \underset{(n\times p)}{\bar{\lambda}_{\Psi 1}(t)} \underset{(p\times 1)}{\nu} \right] \quad . \tag{58}$$

where μ is a constant and ν is a $p \times 1$ row matrix of constants. The expression for $\overline{\delta\alpha}$ (t) can now be substituted into the equations for $\overline{d\Psi}$ and $(dp)^2$ [equations (51) and (56)] to obtain two equations that can be solved for μ and ν. If we first substitute equation (58) into equation (56), we obtain the following after some manipulation:

$$\nu = I_2^{-1} (I_1 - 2\mu \ \widetilde{d\Psi}) \tag{59}$$

where

$$\underset{(p\times 1)}{I_1} = \int_{t_0}^{T} \left[\bar{\lambda}_{\Psi 1}^{T}(t) \ \overline{G}(t) \ \overline{w}^{-1} \overline{G}^{T}(t) \ \bar{\lambda}_{\phi 1}(t) \right] dt \tag{60}$$

$$\underset{(p\times p)}{I_2} = \int_{t_0}^{T} \left[\bar{\lambda}_{\Psi 1}^{T}(t) \ \overline{G}(t) \ \overline{w}^{-1} \overline{G}^{T}(t) \ \bar{\lambda}_{\Psi 1}(t) \right] dt \tag{61}$$

$$\underset{(p\times 1)}{\widetilde{d\Psi}} = \overline{d\Psi} - \bar{\lambda}_{\Psi 1}^{T}(t_0) \ \overline{\delta x}(t_0) \quad . \tag{62}$$

From equations (58) and (59), we can write:

$$\overline{\delta\alpha}^{T}(t) = \frac{1}{2\mu}\left[\bar{\lambda}^{T}_{\phi 1}(t)\ \bar{G}(t)\ \bar{w}^{-1} - \nu^{T}\ \bar{\lambda}^{T}_{\Psi 1}(t)\ \bar{G}(t)\ \bar{w}^{-1}\right] \tag{63}$$

$$\nu^{T} = I_1^{T}\ (I_2^{-1})^{T} - 2\mu\ \widetilde{d\Psi}^{T}\ (I_2^{-1})^{T} \tag{64}$$

where

$$\widetilde{d\Psi}^{T} = \overline{d\Psi}^{T} - \overline{\delta x}^{T}(t_0)\ \bar{\lambda}_{\Psi 1}(t_0) \quad . \tag{65}$$

If we now substitute equations (58) and (63) into equation (51), the following result is obtained:

$$(dp)^{2} = \frac{1}{4\mu^{2}}\left(I_3 - I_1^{T}\ I_2^{-1}\ I_1\right) + \widetilde{d\Psi}^{T}\ I_2^{-1}\ \widetilde{d\Psi} \tag{66}$$

where

$$\underset{(1\times 1)}{I_3} = \int_{t_0}^{T}\left[\bar{\lambda}^{T}_{\phi 1}(t)\ \bar{G}(t)\ \bar{w}^{-1}\ \bar{G}^{T}(t)\ \bar{\lambda}_{\phi 1}\right]dt \quad . \tag{67}$$

Solving equation (66) for μ:

$$2\mu = \pm\sqrt{\frac{I_3 - I_1^{T}\ I_2^{-1}\ I_1}{(dp)^{2} - \widetilde{d\Psi}^{T} I_2^{-1}\ \widetilde{d\Psi}}} \quad . \tag{68}$$

Equations (59) and (68) give the solutions for μ and ν. These can be substituted into equation (58) to obtain the following:

$$\overline{\delta\alpha}(t) = \pm\ \bar{w}^{-1}\bar{G}^{T}(t)\left(\bar{\lambda}_{\phi 1} - \bar{\lambda}_{\Psi 1}\ I_2^{-1}\ I_1\right)\sqrt{\frac{(dp)^{2} - \widetilde{d\Psi}^{T} I_2^{-1}\ \widetilde{d\Psi}}{I_3 - I_1^{T}\ I_2^{-1}\ I_1}}$$
$$+\ \bar{w}^{-1}\ \bar{G}^{T}(t)\ \bar{\lambda}_{\Psi 1}\ I_2^{-1}\ \widetilde{d\Psi} \quad . \tag{69}$$

This is the desired control variable perturbation that maximizes $\overline{d\phi}$ in the first expression of equation (50) for a given value of $(dp)^2$ [equation (51)], given p values of $\overline{d\Psi}$ [the second expression of equation (50)] and $\overline{d\Omega} = 0$ in the third expression of equation (50). Thus, we substitute equation (69) in the first expression of equation (50) or equation (55) to obtain the predicted change in ϕ for the change in the control variables:

$$d\phi = \pm\sqrt{\left[(dp)^2 - \widetilde{d\Psi}^T I_2^{-1} \widetilde{d\Psi}\right] \left(I_3 - I_1^T I_2^{-1} I_1\right)}$$

$$+ I_1^T I_2^{-1} \widetilde{d\Psi} + \bar{\lambda}^T_{\phi 1} (t_0)\, \overline{\delta x}\, (t_0) \qquad . \qquad (70)$$

For $\overline{d\Psi} = 0$ and $\overline{\delta x}\,(t_0) = 0$, equation (70) becomes:

$$\overline{d\phi} = \pm\sqrt{\left(I_3 - I_1^T I_2^{-1} I_1\right) (dp)^2} \qquad (71)$$

or

$$\frac{\overline{d\phi}}{dp} = \pm\sqrt{I_3 - I_1^T I_2^{-1} I_1} \qquad . \qquad (72)$$

As the optimum solution is approached and the terminal constraints are met, this gradient must tend to zero. The + sign is used if ϕ is to be maximized and the - sign is used if ϕ is to be minimized.

The control variable perturbations as given by equations (69) are now added to the initial or previous control variable estimates to yield the new estimates:

$$\bar{\alpha}_n(t) = \bar{\alpha}_p(t) + \overline{\delta\alpha}(t) \tag{73}$$

The new estimates $\bar{\alpha}_n(t)$ are now used in equation (6) and the process is repeated until the terminal constraints are satisfied and $\frac{d\phi}{dp} \to 0$.

APPLICATION OF THE STEEPEST ASCENT METHOD TO AN APOLLO THREE-DIMENSIONAL REENTRY PROBLEM

Problem Formulation

The specific problem investigated herein has been studied extensively by Colunga using the indirect method termed the "modified sweep method" [7]. Basically, the optimization problem consists of determining the roll angle program $\beta(t)$ in the interval t_0 to t_f which can be used to control an Apollo spacecraft to minimize the following function:

$$I = \int_{t_0}^{t_f} \left[\frac{(L^2 + D^2)^{\frac{1}{2}}}{\tilde{m}} + \tilde{\lambda}_0 \rho^{\frac{1}{2}} v^3 \right] dt \quad . \tag{74}$$

This form is of the classical Langrange or Bolza problem and can be transformed to the Mayer problem [9]. This is done by introducing one additional state variable and one additional differential equation. If this state variable is denoted as q (t), then:

$$\dot{q}(t) = [\text{integrand of the function I}] \tag{75}$$

or

$$\frac{dq}{dt} = \frac{(L^2 + D^2)^{\frac{1}{2}}}{\tilde{m}} + \tilde{\lambda}_0 \rho^{\frac{1}{2}} v^3 \tag{76}$$

where the first term measures acceleration caused by aerodynamic forces and the second term measures convective heating experienced by the spacecraft. We also have:

$$q\,(t_0) = 0 \quad . \tag{77}$$

From equation (75) we can rewrite equation (74) as:

$$q\,(t) = \int_{t_0}^{t_f} \left[\frac{(L^2 + D^2)^{\frac{1}{2}}}{\tilde{m}} + \tilde{\lambda}_0\, \rho^{\frac{1}{2}}\, v^3 \right] dt \tag{78}$$

or

$$I = q\,(t) \tag{79}$$

which is of the form of the Mayer problem. Thus, the system is defined by the following state variables ($n = 7$):

$$\underset{(7 \times 1)}{\bar{x}\,(t)} = \begin{bmatrix} h\,(t) \\ \theta\,(t) \\ \Delta\,(t) \\ v\,(t) \\ \gamma\,(t) \\ A\,(t) \\ q\,(t) \end{bmatrix} \quad . \tag{80}$$

These state variables are subject to the control variable ($m = 1$):

$$\underset{(1 \times 1)}{\bar{\alpha}\,(t)} = \left[\beta\,(t)\right] \quad . \tag{81}$$

The problem is now to determine $\bar{\alpha}$ (t) in the interval $t_0 \leq t \leq t_f$ to minimize

$$\bar{\phi} = I \quad . \tag{82}$$

This minimization is to be accomplished while satisfying either of the following two cases involving the terminal constraints and the stopping conditions:

<u>Case 1 — Terminal Constraints (p = 5):</u>

$$\bar{\Psi}^{1} = \begin{bmatrix} \Psi_1^{1} \\ \Psi_2^{1} \\ \Psi_3^{1} \\ \Psi_4^{1} \\ \Psi_5^{1} \end{bmatrix} = \begin{bmatrix} h(t_f) - \bar{h}_f \\ \theta(t_f) - \bar{\theta}_f \\ \Delta(t_f) - \bar{\Delta}_f \\ \gamma(t_f) - \bar{\gamma}_f \\ A(t_f) - \bar{A}_f \end{bmatrix} \quad . \tag{83}$$

Single stopping condition is as follows:

$$\bar{\Omega}^{1} = v(t_f) - 261 \text{ m/sec } (856 \text{ ft/sec}) \quad . \tag{84}$$

<u>Case 2 — Terminal Constraints (p = 5):</u>

$$\bar{\Psi}^{11} = \begin{bmatrix} \Psi_1^{11} \\ \Psi_2^{11} \\ \Psi_3^{11} \\ \Psi_4^{11} \\ \Psi_5^{11} \end{bmatrix} = \begin{bmatrix} \theta(t_f) - \bar{\theta}_f \\ \Delta(t_f) - \bar{\Delta}_f \\ v(t_f) - \bar{v}_f \\ \gamma(t_f) - \bar{\gamma}_f \\ A(t_f) - \bar{A}_f \end{bmatrix} \quad . \tag{85}$$

Single stopping condition is as follows:

$$\bar{\Omega}^{11} = h\,(t_f) - 23\,014 \text{ m } (75\,504 \text{ ft}) = 0 \tag{86}$$

where

$$\left.\begin{aligned}
\bar{h}_f &= 23\,014 \text{ m } (75\,504 \text{ ft}) \\
\bar{\theta}_f &= 24.1 \text{ deg} \\
\bar{\Delta}_f &= -0.6 \text{ deg} \\
\bar{\gamma}_f &= -44.3 \text{ deg} \\
\bar{v}_f &= 261 \text{ m/sec } (856 \text{ ft/sec}) \\
A_f &= -29.4 \text{ deg}
\end{aligned}\right\} . \tag{87}$$

The time at which the stopping condition is satisfied is denoted as t_f. The minimization is also to be accomplished subject to the following differential equations of motion constraints:

$$\underset{(7\times 1)}{\bar{f}} \equiv \begin{bmatrix} dh/dt \\ d\theta/dt \\ d\Delta/dt \\ dv/dt \\ d\gamma/dt \\ dA/dt \\ dq/dt \end{bmatrix} = \begin{bmatrix} v \sin\gamma \\ v\cos\gamma\cos A/[(R+h)\cos\Delta] \\ v\cos\gamma\sin A/(R+h) \\ G\sin\gamma - \bar{D} \\ [(G\cos\gamma)/v] + [v\cos\gamma/(R+h)] + (\bar{L}\cos\beta/v) \\ [-v\cos\gamma\cos A\tan\Delta/(R+h)] - \{[\bar{L}\sin\beta/(v\cos\gamma)]\} \\ \dfrac{(L^2+D^2)^{\frac{1}{2}}}{\tilde{m}} + \tilde{\lambda}_0\,\rho^{\frac{1}{2}}\,v^3 \end{bmatrix} \tag{88}$$

where

$$\left.\begin{aligned}
\underline{G} &= -\mu^*/(R+h)^2 \\
\bar{D} &= \rho\, S\, v^2\, C_D/(2\,\tilde{m}) \\
\bar{L} &= \rho\, S\, v^2\, C_L/(2\,\tilde{m}) \\
L &= \frac{1}{2}\,\rho\, S\, v^2\, C_L \\
D &= \frac{1}{2}\,\rho\, S\, v^2\, C_D \\
\rho &= \rho_0\, e^{-\beta^* h}
\end{aligned}\right\} \quad . \tag{89}$$

The following initial conditions for the state variables are assumed at $t_0 = 0$:

$$\begin{bmatrix} \bar{h}\,(t_0) \\ \underline{\theta}\,(t_0) \\ \underline{\Delta}\,(t_0) \\ \underline{v}\,(t_0) \\ \underline{\gamma}\,(t_0) \\ \underline{A}\,(t_0) \\ \bar{q}\,(t_0) \end{bmatrix} = \begin{bmatrix} 121\ 920 \text{ m } (400\ 000 \text{ ft}) \\ 0 \text{ deg} \\ 0 \text{ deg} \\ 10\ 668 \text{ m/sec } (35\ 000 \text{ ft/sec}) \\ -6.5 \text{ deg} \\ 0 \text{ deg} \\ 0 \text{ deg} \end{bmatrix} \quad . \tag{90}$$

Perturbation and Adjoint Equations

The equations analogous to equations (13) governing the behavior of perturbations in the system are given by:

$$\frac{d}{dt}\,[\bar{\delta x}(t)] = \bar{F}(t)\,\bar{\delta x}(t) + \bar{G}\,(t)\,\bar{\delta\alpha}\,(t) \tag{91}$$

where

$$\frac{d}{dt}\underset{(7\times 1)}{[\bar{\delta}\, x(t)]} = \begin{bmatrix} \frac{d}{dt}[\delta h(t)] \\ \frac{d}{dt}[\delta\theta\,(t)] \\ \frac{d}{dt}[\delta\Delta(t)] \\ \frac{d}{dt}[\delta v(t)] \\ \frac{d}{dt}[\delta\gamma(t)] \\ \frac{d}{dt}[\delta A(t)] \\ \frac{d}{dt}[\delta q(t)] \end{bmatrix} \tag{92}$$

$$\underset{(7\times 1)}{\overline{\delta x}\,(t)} = \begin{bmatrix} \delta h\ (t) \\ \delta\theta\ (t) \\ \delta\Delta\ (t) \\ \delta v\ (t) \\ \delta\gamma\ (t) \\ \delta A\ (t) \\ \delta q\ (t) \end{bmatrix} \tag{93}$$

$$\underset{(1\times 1)}{\overline{\delta\alpha}\,(t)} = [\,\delta\beta(t)] \tag{94}$$

$$\underset{(7\times 7)}{\bar{F}\,(t)} = \begin{bmatrix} \frac{\partial f_1}{\partial h} & \frac{\partial f_1}{\partial\theta} & \cdots & \frac{\partial f_1}{\partial q} \\ \frac{\partial f_2}{\partial h} & \frac{\partial f_2}{\partial\theta} & \cdots & \frac{\partial f_2}{\partial q} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial f_7}{\partial h} & \frac{\partial f_7}{\partial\theta} & \cdots & \frac{\partial f_7}{\partial q} \end{bmatrix} \tag{95}$$

$$\underset{(7 \times 1)}{\bar{G}(t)} = \begin{bmatrix} \dfrac{\partial f_1}{\partial \beta} \\ \dfrac{\partial f_2}{\partial \beta} \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{\partial f_7}{\partial \beta} \end{bmatrix} \qquad (96)$$

The linear differential equations adjoint to equation (91) are:

$$\begin{bmatrix} \dot{\lambda}_{\phi 1} \\ \dot{\lambda}_{\phi 2} \\ \cdot \\ \cdot \\ \cdot \\ \dot{\lambda}_{\phi 7} \end{bmatrix} = -\bar{F}^T(t) \begin{bmatrix} \lambda_{\phi 1} \\ \lambda_{\phi 2} \\ \cdot \\ \cdot \\ \cdot \\ \lambda_{\phi 7} \end{bmatrix} \qquad (97)$$

$$\begin{bmatrix} \dot{\lambda}_{\Psi 11} & \dot{\lambda}_{\Psi 12} \cdots & \dot{\lambda}_{\Psi 15} \\ \dot{\lambda}_{\Psi 21} & \dot{\lambda}_{\Psi 22} \cdots & \dot{\lambda}_{\Psi 25} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \dot{\lambda}_{\Psi 71} & \dot{\lambda}_{\Psi 72} \cdots & \dot{\lambda}_{\Psi 75} \end{bmatrix} = -\bar{F}^T(t) \begin{bmatrix} \lambda_{\Psi 11} & \lambda_{\Psi 12} \cdots & \lambda_{\Psi 15} \\ \lambda_{\Psi 21} & \lambda_{\Psi 22} \cdots & \lambda_{\Psi 25} \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \lambda_{\Psi 71} & \lambda_{\Psi 72} \cdots & \lambda_{\Psi 75} \end{bmatrix} \qquad (98)$$

$$
\begin{bmatrix} \dot{\lambda}_{\Omega_1} \\ \dot{\lambda}_{\Omega_2} \\ \cdot \\ \cdot \\ \cdot \\ \dot{\lambda}_{\Omega_7} \end{bmatrix} = -\bar{F}^{T}(t) \begin{bmatrix} \lambda_{\Omega_1} \\ \lambda_{\Omega_2} \\ \cdot \\ \cdot \\ \cdot \\ \lambda_{\Omega_7} \end{bmatrix} . \tag{99}
$$

The boundary conditions for equations (97), (98), and (99) are given by:

$$
\underset{(7\times 1)}{\bar{\lambda}_{\phi}}(T) = \begin{bmatrix} \dfrac{\partial\bar{\phi}}{\partial h} \\ \dfrac{\partial\bar{\phi}}{\partial \theta} \\ \cdot \\ \cdot \\ \cdot \\ \dfrac{\partial\bar{\phi}}{\partial q} \end{bmatrix}_{t\,=\,T} \tag{100}
$$

$$
\underset{(7\times 5)}{\bar{\lambda}_{\Psi}}(T) = \begin{bmatrix} \dfrac{\partial\Psi_1}{\partial h} & \dfrac{\partial\Psi_2}{\partial h} & \cdots & \dfrac{\partial\Psi_5}{\partial h} \\ \dfrac{\partial\Psi_1}{\partial \theta} & \dfrac{\partial\Psi_2}{\partial \theta} & \cdots & \dfrac{\partial\Psi_5}{\partial \theta} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \dfrac{\partial\Psi_1}{\partial q} & \dfrac{\partial\Psi_2}{\partial q} & \cdots & \dfrac{\partial\Psi_5}{\partial q} \end{bmatrix}_{t\,=\,T} \tag{101}
$$

$$\bar{\lambda}_{\Omega}(T) = \begin{bmatrix} \frac{\partial \bar{\Omega}}{\partial h} \\ \frac{\partial \bar{\Omega}}{\partial \theta} \\ \cdot \\ \cdot \\ \cdot \\ \frac{\partial \bar{\Omega}}{\partial q} \end{bmatrix}_{t = T} \tag{102}$$
(7×1)

These latter three equations become for Case 1 (CI):

$$\left[\bar{\lambda}_{\phi}(T)\right]_{CI} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{103}$$
(7×1)

$$\left[\bar{\lambda}_{\Psi}(T)\right]_{CI} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{104}$$
(7×5)

$$\left[\bar{\lambda}_{\Omega}(T)\right]_{CI} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} . \tag{105}$$
(7×1)

For Case 2 (CII):

$$\underset{(7\times 1)}{\left[\bar{\lambda}_{\phi}(T)\right]}_{CII} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \tag{106}$$

$$\underset{(7\times 5)}{\left[\bar{\lambda}_{\Psi}(T)\right]}_{CII} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{107}$$

$$\underset{(7\times 1)}{\left[\bar{\lambda}_{\Omega}(T)\right]}_{CII} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{108}$$

Total Differentials of $\bar{\Phi}$, $\bar{\Psi}$, and $\bar{\Omega}$

The desired changes in $\bar{\phi}$, $\bar{\Psi}$, and $\bar{\Omega}$ for the control variable perturbations are given by (with $\overline{\delta x}\ (t_0) = 0$):

$$\overline{d\phi} = \int_{t_0}^{T} \bar{\lambda}_{\phi}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \dot{\phi}\ dt \tag{109}$$

$$d\Psi = \int_{t_0}^{T} \bar{\lambda}_{\Psi}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \dot{\Psi}\ dt \tag{110}$$

$$\overline{d\Omega} = \int_{t_0}^{T} \bar{\lambda}_{\Omega}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt + \dot{\Omega}\ dt \quad . \tag{111}$$

If we solve equation (111) for dt and substitute the result in equations (109) and (110),

$$\overline{d\phi} = \int_{t_0}^{T} \bar{\lambda}_{\phi 1}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt \tag{112}$$

$$\overline{d\Psi} = \int_{t_0}^{T} \bar{\lambda}_{\Psi 1}^{T}(t)\ \bar{G}(t)\ \overline{\delta\alpha}(t)\ dt \tag{113}$$

where

$$\left.\begin{aligned} \bar{\lambda}_{\phi 1}^{T}(t) &= \bar{\lambda}_{\phi}^{T}(t) - \frac{\dot{\phi}}{\dot{\Omega}}\ \bar{\lambda}_{\Omega}^{T}(t) \\ \bar{\lambda}_{\Psi 1}^{T}(t) &= \bar{\lambda}_{\Psi}^{T}(t) - \frac{\dot{\Psi}}{\dot{\Omega}}\ \bar{\lambda}_{\Omega}^{T}(t) \end{aligned}\right\} \tag{114}$$

$$\left.\begin{aligned} \underset{(1\times 1)}{\dot{\phi}} &= \left(\frac{\partial\phi}{\partial t} + \overline{\frac{\partial\phi}{\partial x}}\ \bar{f}\right)_{t\ =\ T} \\ \underset{(5\times 1)}{\dot{\Psi}} &= \left(\frac{\partial\Psi}{\partial t} + \overline{\frac{\partial\Psi}{\partial x}}\ \bar{f}\right)_{t\ =\ T} \\ \underset{(1\times 1)}{\dot{\Omega}} &= \left(\frac{\partial\Omega}{\partial t} + \overline{\frac{\partial\Omega}{\partial x}}\ \bar{f}\right)_{t\ =\ T} \end{aligned}\right\} \quad . \tag{115}$$

For Cases I and II, we have:

$$\left(\frac{\dot{\phi}}{\dot{\Omega}}\right)_{CI} = \frac{f_7}{f_4} \tag{116}$$

$$\left(\frac{\dot{\Psi}}{\dot{\Omega}}\right)_{CI} = \begin{bmatrix} \frac{f_1}{f_4} \\ \frac{f_2}{f_4} \\ \frac{f_3}{f_4} \\ \frac{f_5}{f_4} \\ \frac{f_6}{f_4} \end{bmatrix} \tag{117}$$

$$\left(\frac{\dot{\phi}}{\dot{\Omega}}\right)_{CII} = \left[\frac{f_7}{f_1}\right] \tag{118}$$

$$\left(\frac{\dot{\Psi}}{\dot{\Omega}}\right)_{CII} = \begin{bmatrix} \frac{f_2}{f_1} \\ \frac{f_3}{f_1} \\ \frac{f_4}{f_1} \\ \frac{f_5}{f_1} \\ \frac{f_6}{f_1} \end{bmatrix} \quad . \tag{119}$$

Control Variable Perturbations

The $\overline{\delta\alpha}$ (t) that minimizes $\overline{d\phi}$ in equation (112) subject to the terminal constraints $\overline{d\Psi}$, the single stopping condition $\overline{d\Omega} = 0$, and the constraint $(dp)^2$ on the magnitude of the control change is given by:

$$\overline{\delta\alpha}\,(t) = -\,\bar{w}^{-1}\,\bar{G}^T(t)\left(\bar{\lambda}_{\phi 1} - \bar{\lambda}_{\Psi 1}\,I_2^{-1}\,I_1\right)\sqrt{\frac{(dp)^2 - \overline{d\Psi}^T\,I_2^{-1}\,\overline{d\Psi}}{I_3 - I_1^T\,I_2^{-1}\,I_1}}$$

$$+\,\bar{w}^{-1}\,\bar{G}^T(t)\,\bar{\lambda}_{\Psi 1}\,I_2^{-1}\,\overline{d\Psi} \tag{120}$$

where

$$\left.\begin{aligned}
\underset{(5\times 1)}{I_1} &= \int_0^T \left[\bar{\lambda}^T_{\Psi 1}(t)\,\bar{G}(t)\,\bar{w}^{-1}\,\bar{G}^T(t)\,\bar{\lambda}_{\phi 1}(t)\right] dt \\
\underset{(5\times 5)}{I_2} &= \int_0^T \left[\bar{\lambda}^T_{\Psi 1}(t)\,\bar{G}(t)\,\bar{w}^{-1}\,\bar{G}^T(t)\,\bar{\lambda}_{\Psi 1}(t)\right] dt \\
\underset{(1\times 1)}{I_3} &= \int_0^T \left[\bar{\lambda}^T_{\phi 1}(t)\,\bar{G}(t)\,\bar{w}^{-1}\,\bar{G}^T(t)\,\bar{\lambda}_{\phi 1}(t)\right] dt
\end{aligned}\right\} . \tag{121}$$

We can now substitute equation (120) into equation (112) to obtain the change in $\bar{\phi}$ for the change in the control variable:

$$\overline{d\phi} = -\sqrt{\left[(dp)^2 - \overline{d\Psi}^T\,I_2^{-1}\,\overline{d\Psi}\right]\left[I_3 - I_1^T\,I_2^{-1}\,I_1\right]} + I_1^T\,I_2^{-1}\,\overline{d\Psi}\ . \tag{122}$$

For $\overline{d\Psi} = 0$, this becomes:

$$\overline{\frac{d\phi}{dp}} = - \sqrt{I_3 - I_1^T I_2^{-1} I_1} \quad . \tag{123}$$

We now add $\overline{\delta\alpha}$ (t) to the initial or previous estimate to yield:

$$\bar{\alpha}_N(t) = \bar{\alpha}_p(t) + \overline{\delta\alpha}(t) \quad . \tag{124}$$

The new estimates $\bar{\alpha}_N$ (t) are now used in equation (88) and the process is repeated until terminal constraints are met and $\overline{\frac{d\phi}{dp}} \rightarrow 0$.

STUDY RESULTS

Computer Program Development

The study of the application problem herein has resulted in the development of a highly flexible computer program that can be adapted to other trajectory optimization problems. The Marshall Vehicle Engineering Simulation System (MARVES) programming system is used extensively in this development. The use of the MARVES programming language provides the added flexibility of specifying program statements directly related to the application problem. It also provides an easy means of modifying the developed program. A complete description of the program is given in the appendix.

Application Results

In an application problem of this nature, several parameters can be studied to establish their relative importance and/or effects on the minimizing or maximizing function. The Case 1 problem involving the five terminal constraints and the velocity stopping condition was selected for various parameter perturbation runs. These various runs are summarized in Table 1.

The standard deviation of the corrections $\overline{\delta\alpha}$ (t) to the control

TABLE 1. PARAMETER PERTURBATION RUNS FOR THE CASE I APOLLO APPLICATION PROBLEM

Run Number	Cycle	k_0	dp_0	k_1	T_{end}	q	ψ 1	ψ 2	ψ 3	ψ 4	ψ 5	$\frac{d\psi}{dp}$	dp	$\sigma_{\delta\alpha}$
1	1	0.5	0.2	0.002	537.0	17.12	3.109	-0.0146	0.0008	0.532	0.6157	-0.0526	0.01948	50.72
2	1	0.5	0.2	0.002	498.17	16.80	0.9308	-0.0146	0.0009	0.3159	-0.0172	-0.1398	0.01019	8.76
	2	0.5	0.2	0.002	498.17	17.05	4.545	0.0077	0.0017	0.5375	1.93	-0.0823	0.02777	63.36
3	1	0.4	0.1	0.01	498.17	16.80	0.7447	-0.0116	0.00071	0.253	-0.0137	-0.1398	0.00477	13.99
4	1	0.1	0.01	0.01	498.17	16.80	0.1862	-0.0029	0.00018	0.063	-0.0034	-0.1398	0.0002	4.87
	2	0.1	0.01	0.01	498.17	16.84	0.3249	-0.0022	0.00022	0.083	-0.0091	-0.1311	0.0045	21.24
5	1	0.2	0.05	0.01	498.17	16.80	0.3723	-0.0058	0.00036	0.126	-0.0069	-0.1398	0.0009	6.23
	2	0.2	0.05	0.01	498.17	16.88	0.8667	-0.0027	0.00049	0.189	-0.262	-0.1208	0.0253	10.87
6	1	0.1	0.01	0.01	438.9	16.07	-0.8760	-0.0025	0.00006	-0.0650	-0.0275	-0.7226	0.00012	4.41
	2	0.1	0.01	0.01	438.9	15.94	-1.022	-0.0029	0.00002	-0.069	-0.0356	-0.7195	0.00021	5.23
7	1	0.1	0.01	0.01	437.26	16.05	-0.8876	-0.0025	0.00006	-0.068	-0.027	-0.7468	0.00014	4.23
	2	0.1	0.01	0.01	437.26	15.90	-1.0411	-0.0029	0.000005	-0.071	-0.036	-0.7395	0.00026	5.98
	3	0.1	0.01	0.01	437.26	15.62	-1.2298	-0.0037	-0.000075	-0.0731	-0.045	-1.1377	0.00082	12.26
	4	0.1	0.01	0.01	437.26	15.15	-1.3868	-0.0047	-0.00014	-0.0721	-0.052	-2.1606	0.0012	15.50
	5	0.1	0.01	0.01	437.26	16.17	-0.8015	-0.0026	-0.00012	-0.065	-0.059	-1.362	0.00004	2.64
	6	0.1	0.01	0.01	437.26	16.02	-0.929	-0.0030	-0.00012	-0.068	-0.061	-1.5015	0.00005	3.09
8	1	0.05	0.005	0.01	498.17	16.80	0.0931	-0.0015	0.00009	0.0316	-0.0017	-0.1398	0.00005	2.59
	2	0.05	0.005	0.01	498.17	16.82	0.126	-0.0013	0.00010	0.0365	-0.0061	-0.1382	0.00006	2.94
9	1	0.05	0.005	0.01	437.26	16.05	-0.44	-0.0012	0.000030	-0.0337	-0.0139	-0.7468	0.000035	2.15
	2	0.05	0.005	0.01	437.26	15.97	-0.49	-0.0014	0.000016	-0.0349	-0.0162	-0.7515	0.000049	2.58
	3	0.05	0.005	0.01	437.26	15.84	-0.54	-0.0015	-0.0000004	-0.0359	-0.0187	-0.7684	0.000077	3.38
	4	0.05	0.005	0.01	437.26	15.67	-0.60	-0.0018	-0.000022	-0.0365	-0.0212	-0.9388	0.000164	5.38
	5	0.05	0.005	0.01	437.26	15.37	-0.67	-0.0021	-0.000045	-0.0366	-0.0237	-1.659	0.000478	10.00
	6	0.05	0.005	0.01	437.26	15.23	-0.68	-0.0022	-0.000054	-0.0366	-0.0255	-1.891	0.000393	9.19
	7	0.05	0.005	0.01	437.26	15.23	-0.69	-0.0024	-0.000044	-0.0364	-0.0266	-2.262	0.000322	7.38
	8	0.05	0.005	0.01	437.26	15.48	-0.62	-0.0021	-0.000050	-0.0361	-0.0284	-1.760	0.000098	3.40
	9	0.05	0.005	0.01	437.26	15.22	-0.67	-0.0024	-0.000025	-0.0360	-0.0282	-2.10	0.00072	11.97
	10	0.05	0.005	0.01	437.26	15.10	-0.69	-0.0025	-0.000007	-0.0362	-0.0287	-2.70	0.00021	5.80
	11	0.05	0.005	0.01	437.26	15.29	-0.65	-0.0024	0.000046	-0.0360	-0.0297	-2.22	0.00034	6.44
	12	0.05	0.005	0.01	437.26	15.26	-0.67	-0.0025	0.000082	-0.0361	-0.0294	-2.91	0.00015	4.54

parameter is computed at the end of each cycle for a given run. This provides a quantitative measure of the overall magnitude change in the corrections from one cycle to the next. This quantity is denoted in Table 1 as $\sigma_{\delta\alpha}$.

It is easily seen from Table 1 that the steepest ascent method is sensitive to the selection of the parameters k_0, dp_0, and k_1. These results indicate the need for a more rigorous selection criteria for these parameters. It should be pointed out that the k_0, dp_0, and k_1 values used are not necessarily the values to use for other application problems.

APPENDIX A. COMPUTATIONAL SOLUTION PROCEDURE FOR THE STEEPEST ASCENT METHOD

This appendix lists the various computational steps in the implementation of the steepest ascent method. These are the specific steps to be taken to obtain the minimizing solution for the Apollo application problem discussed earlier. A computer program listing containing the computational details is also given. Typical output results for the application problem are included.

It should be pointed out that the program listing is for the deck operation on the IBM 7094 computer. A similar deck exists for use on the UNIVAC 1108 computer.

It is noted that both programs have double precision capability. The SC-4020 plotting procedures are also used extensively in each program.

Step 1

Integrate equation (88), given initial estimates $[\bar{\alpha}_p(t)]$ of control variable and initial conditions $[\bar{x}(t_0 = 0)]$, until stopping condition of equation (84) or equation (86) is satisfied. The time at which this occurs is denoted as T. Store the state variable values between $t = 0$ and $t = T$.

Step 2

Integrate equations (97), (98), and (99) backwards from $t = T$ to determine the adjoint variables $\bar{\lambda}_\phi$ (7×1), $\bar{\lambda}_\Psi$ (7×5), and $\bar{\lambda}_\Omega$ (7×1). The matrix $\bar{F}(t)$ is evaluated on the nominal path by reference to the stored values of the state obtained in Step 1. Thus, $\bar{F}(t)$ is a time varying array of coefficients.

Step 3

Calculate $\bar{\lambda}_{\phi 1}$, $\bar{\lambda}_{\Psi 1}$ as given by equation (114). Then form $\bar{\lambda}_{\phi 1}^T \bar{G}$ and $\bar{\lambda}_{\Psi 1}^T \bar{G}$.

Step 4

Carry out backwards integrations to obtain I_1, I_2, and I_3 [equation (121)].

Step 5

Print out the values of q, $\tilde{\Psi}_1$, $\tilde{\Psi}_2, \ldots, \tilde{\Psi}_5$ achieved by the nominal trajectory. These values result from using the initial control estimates $\bar{\alpha}_p(t)$. This is the first solution.

Step 6

Select dp_0^2 so as to obtain a reasonable value of dp_0^2/T, a mean square deviation of the control from the nominal to the next step.

Step 7

Select the changes $d\Psi_1$, $d\Psi_2, \ldots, d\Psi_5$ so that

$$d\tilde{\Psi} = -k_0 \begin{bmatrix} \tilde{\Psi}_1 \\ \tilde{\Psi}_2 \\ \cdot \\ \cdot \\ \cdot \\ \tilde{\Psi}_5 \end{bmatrix}_{t=T} \equiv -k_0 \overline{d\Psi} \quad , \quad 0 < k_0 \leq 1$$

within the limitation that:

$$dp = \left(dp_0^2 - d\tilde{\Psi}^T I_2^{-1} d\tilde{\Psi} \right) \geq 0$$

and where $\tilde{\Psi}$ refers to Case 1 [equation (83)] or Case 2 [equation (85)] for the terminal constraints. If the $\tilde{\Psi}$'s make dp negative, reduce $d\tilde{\Psi}$ by a constant factor so that dp = 0 (or nearly vanishes); i.e., the limitation on dp becomes:

$$dp = k_0^2 \left(dp_0^2 - k_1^2 \, \overline{d\Psi}^T \, I_2^{-1} \, \overline{d\Psi} \right) \geq 0$$

where $0 < k_1 \leq 1$. It is noted that two cases can occur; namely, $dp \geq 0$ or $dp < 0$. If $dp \geq 0$, we would proceed to Step 8. However, if $dp < 0$, then k_1 is selected such that $dp \geq 0$ and then control is transferred to Step 8.

Step 8

Use equation (120) to determine $\overline{\delta\alpha}$ (t) and add it to $\bar{\alpha}_p$ (t) to form $\bar{\alpha}_p$ (t) + $\overline{\delta\alpha}$ (t) for the next approximation.

Step 9

Examine the $\tilde{\Psi}_1$, $\tilde{\Psi}_2$, ..., $\tilde{\Psi}_5$ values from Step 5 to see how close to zero they are. Examine $\frac{d\phi}{dp}$ [equation (123)] to see if the solution is sufficiently close to the minimizing solution.

Step 10

For $\tilde{\Psi} \cong 0$ satisfied, if $\frac{d\phi}{dp}$ is not sufficiently close to zero, return to Step 1 and use $\bar{\alpha}_p$(t) + $\overline{\delta\alpha}$ (t) and repeat the computational cycle.

Step 11

Terminate when $\tilde{\Psi} = 0$ and $\frac{d\phi}{dp} \to 0$.

Steps 1 through 5 yield all the information for the first cycle; i.e., we have the optimum payoff function q (t) which results from using the control variable estimate $\bar{\alpha}_p(t)$. The information required in the following cycles is given in Steps 6 through 11. A value for the parameter k_0 is selected to obtain values for $d\tilde{\Psi}$ which are closer to zero. A constant k_0 is maintained in all cycles from the second value on. In checking $\frac{d\phi}{dp}$, if oscillations occur, then dp_0 is reduced by an order of magnitude; i.e., $dp = 0.01,\ 0.001,\ 0.0001$, etc.

Example of Plot Output Results for Case I Application Problem, Run Number 6 (Cycle 2) Data

H:	Altitude (Fig. A-1)
Theta:	Longitude (Fig. A-2)
Delta:	Latitude (Fig. A-3)
V:	Velocity (Fig. A-4)
Gamma:	Angle of Attack (Fig. A-5)
A:	Heading Angle (Fig. A-6)
F7A:	Acceleration Component of Q (Fig. A-7)
F7H:	Heating Component of Q (Fig. A-8)
Beta:	Roll Angle (Fig. A-9)

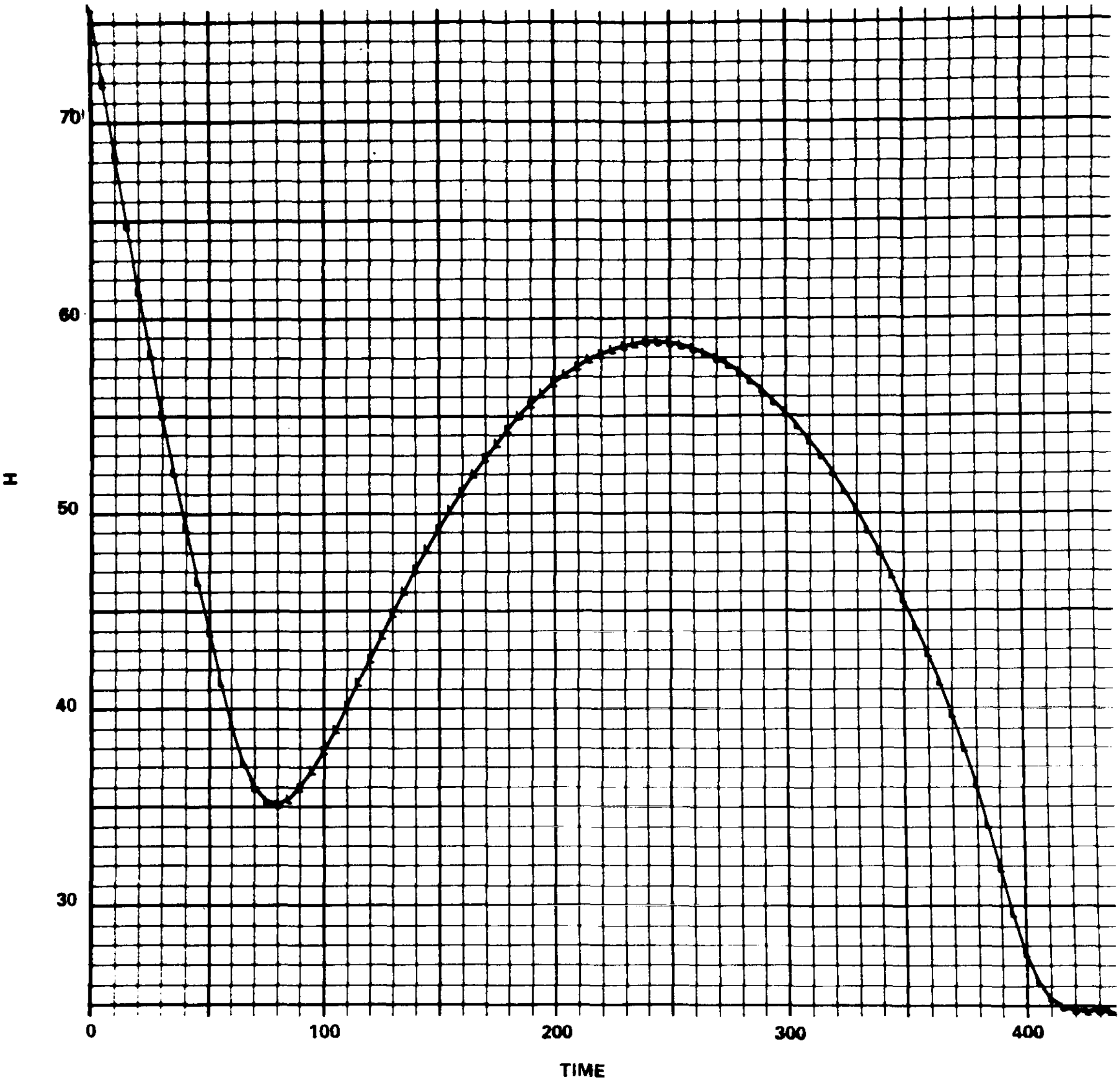

Figure A-1. Altitude versus time.

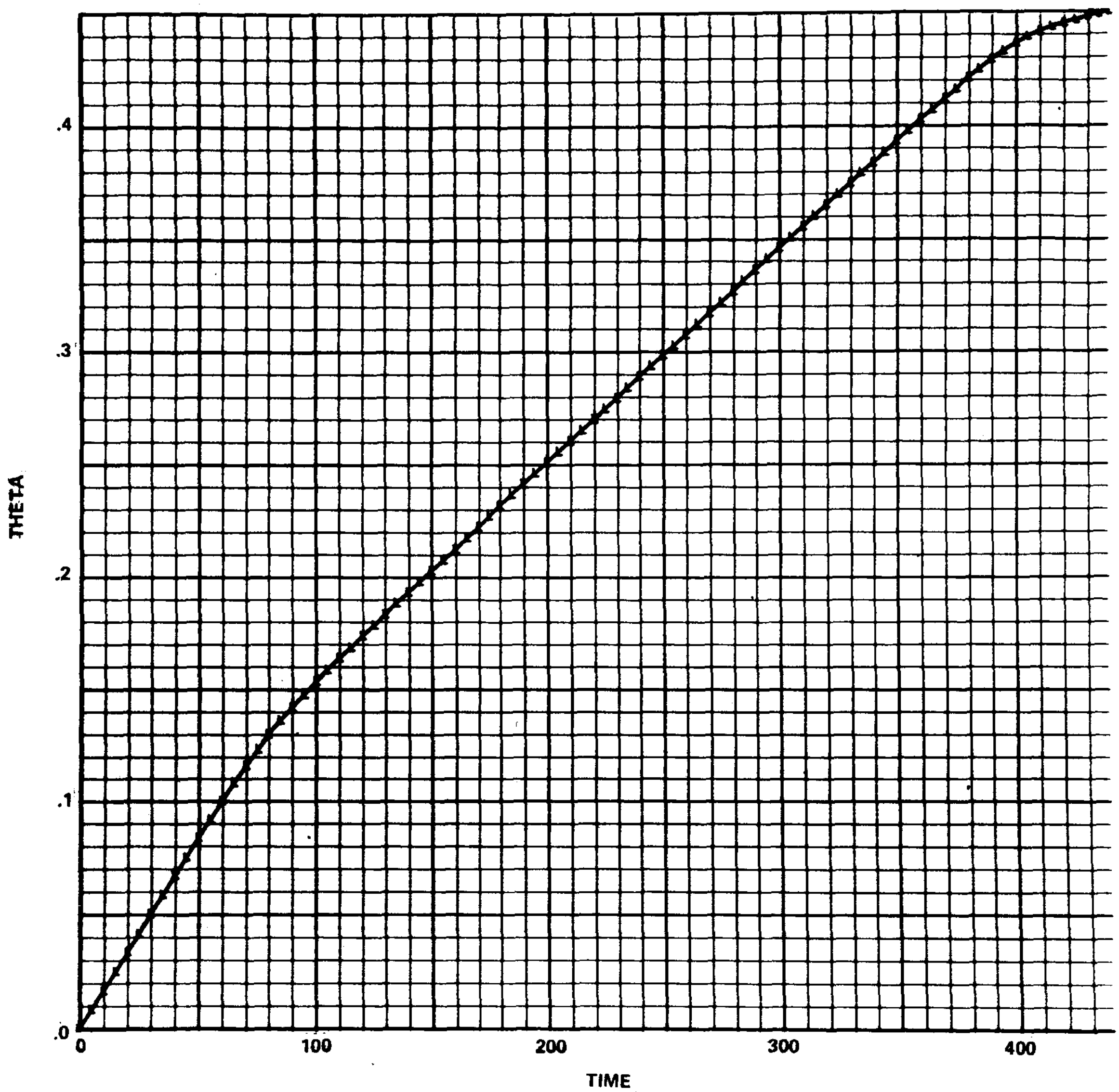

Figure A-2. Longitude versus time.

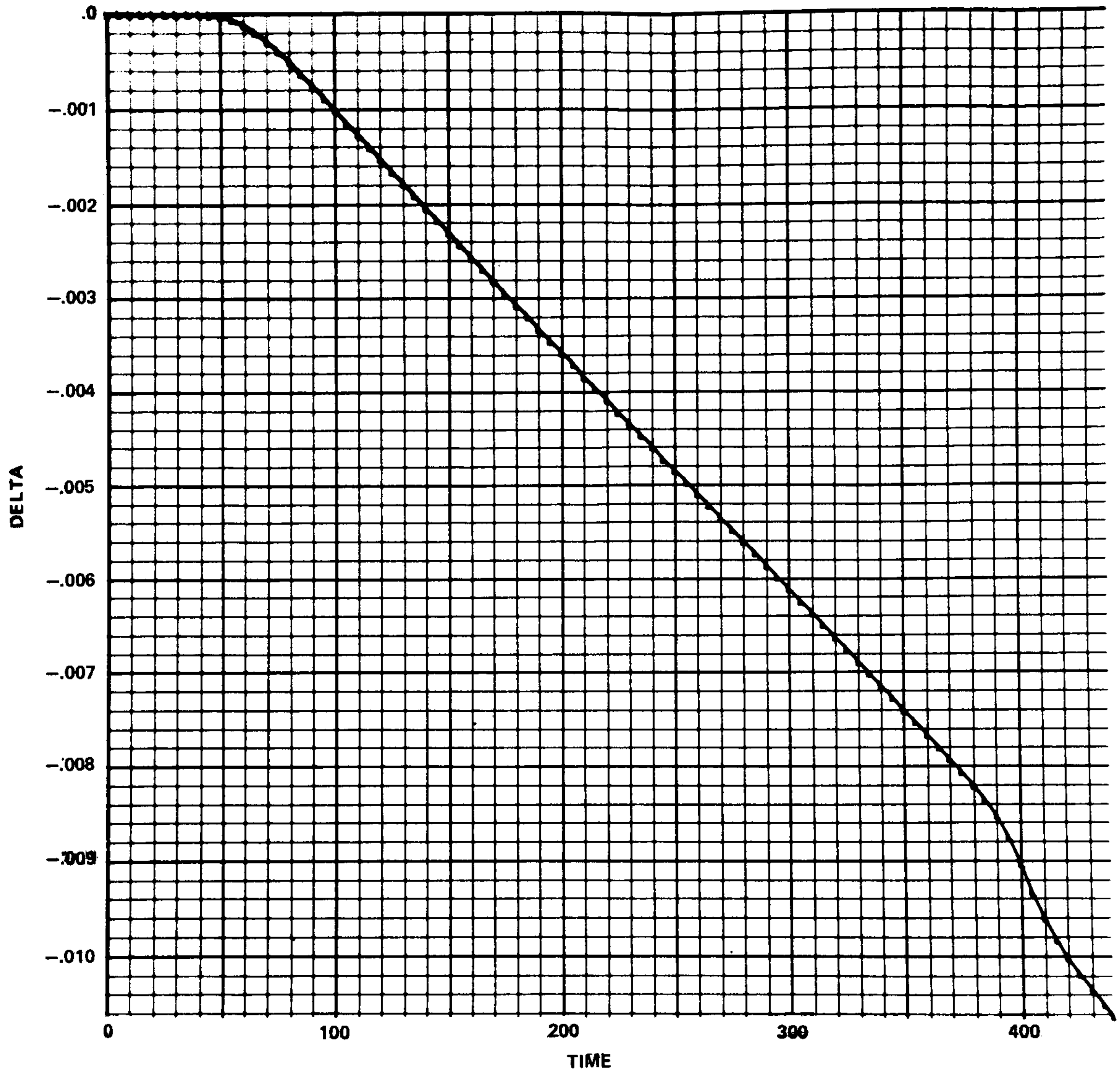

Figure A-3. Latitude versus time.

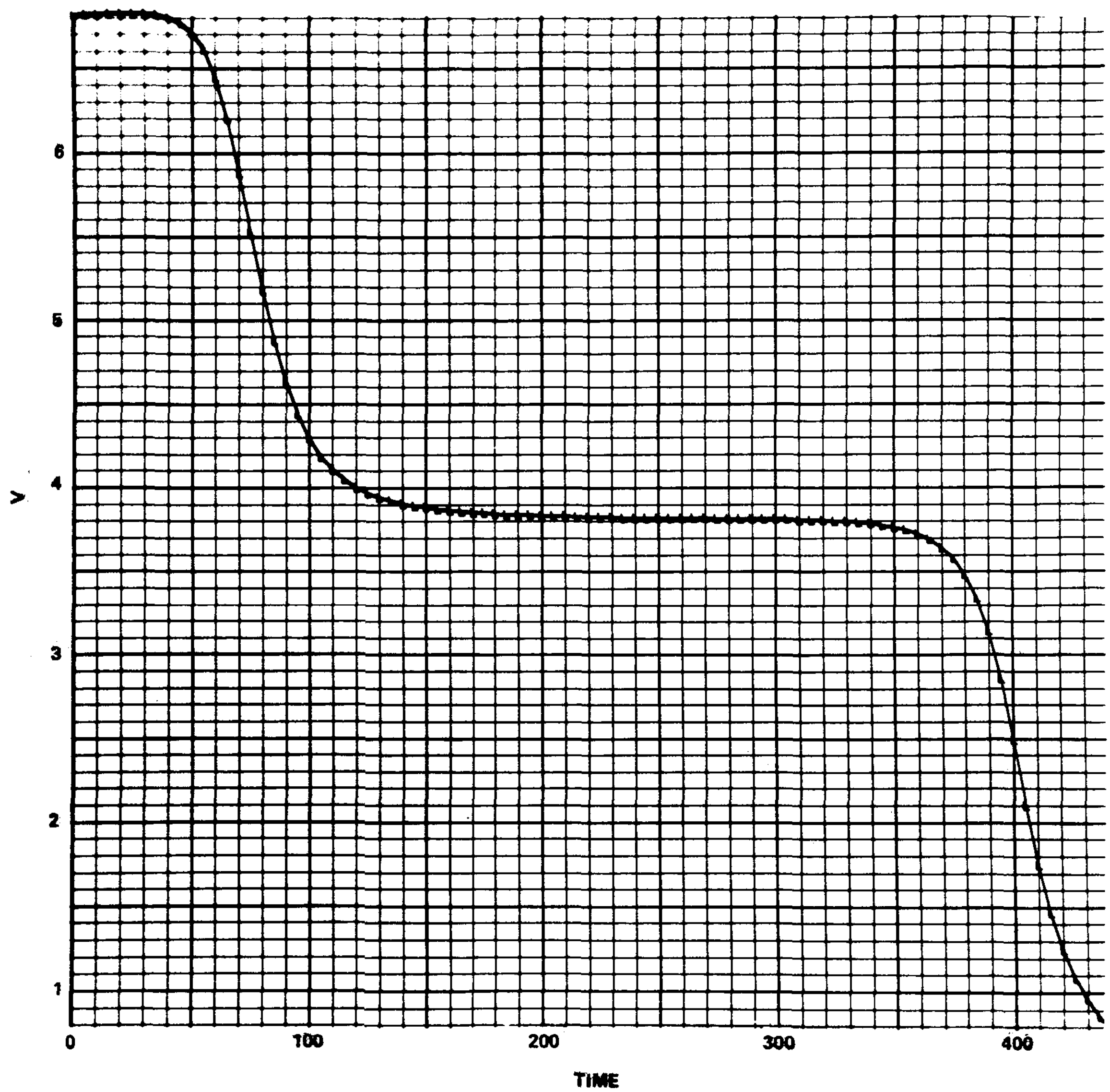

Figure A-4. Velocity versus time.

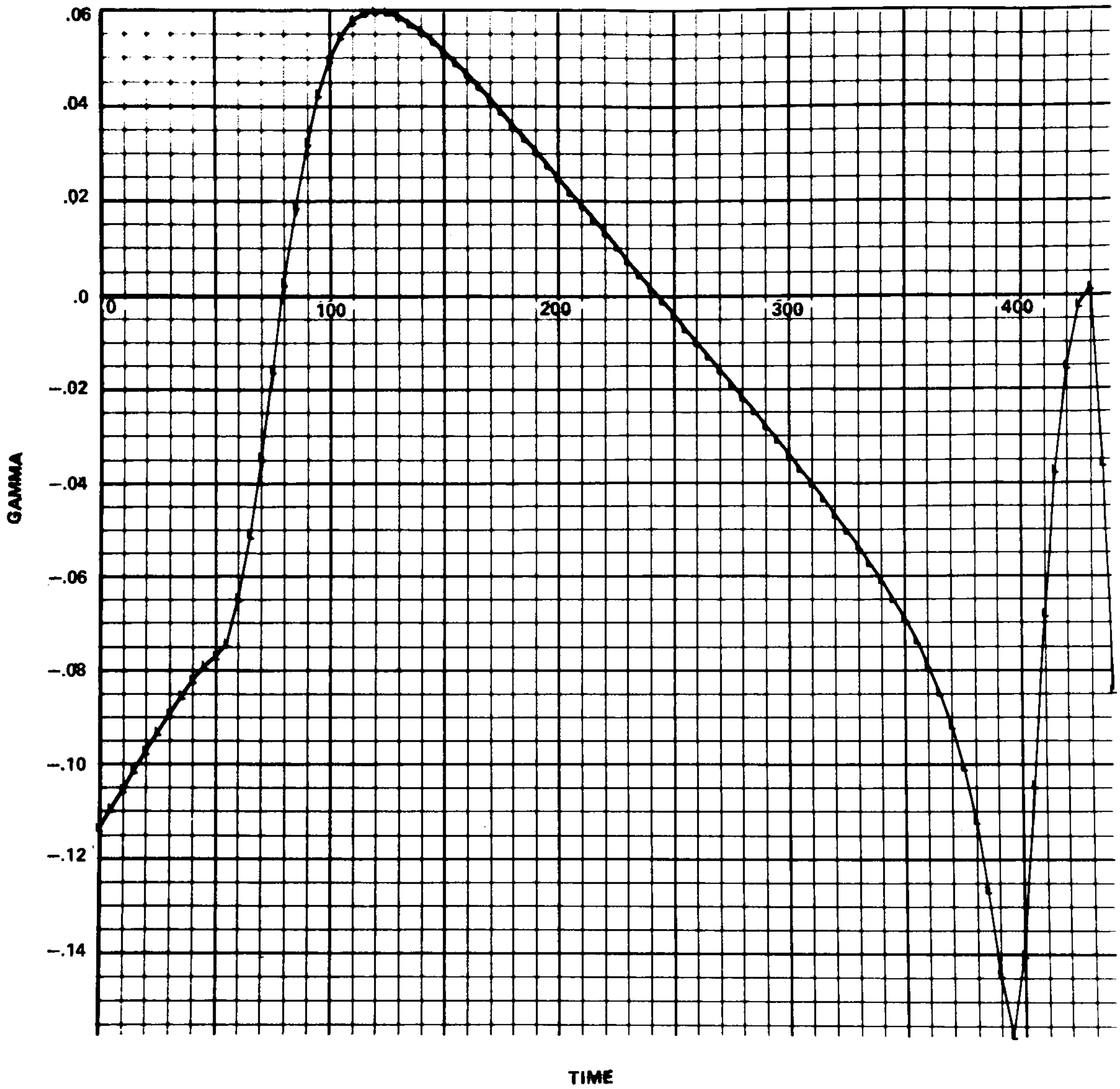

Figure A-5. Angle-of-attack versus time.

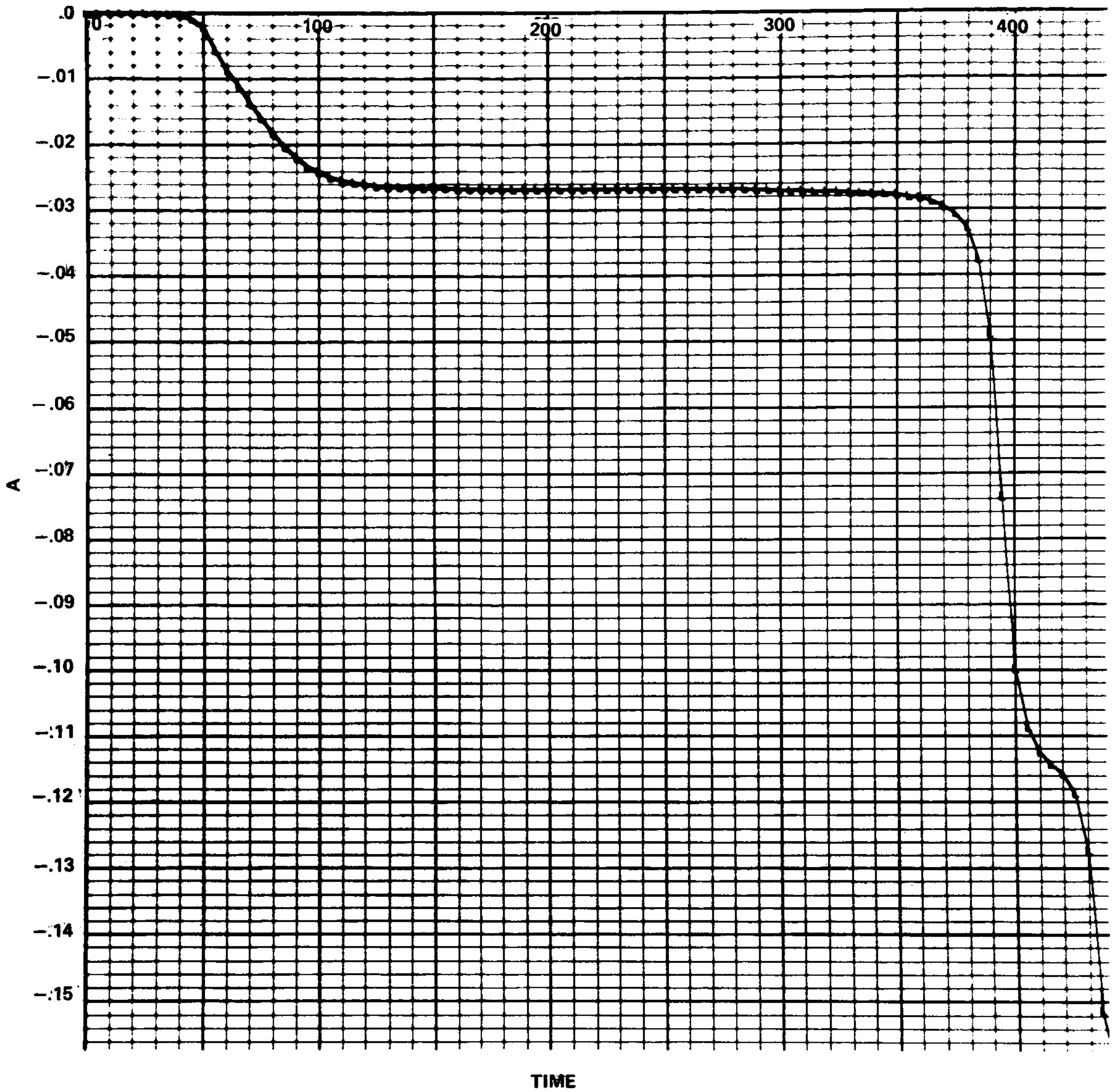

Figure A-6. Heading angle versus time.

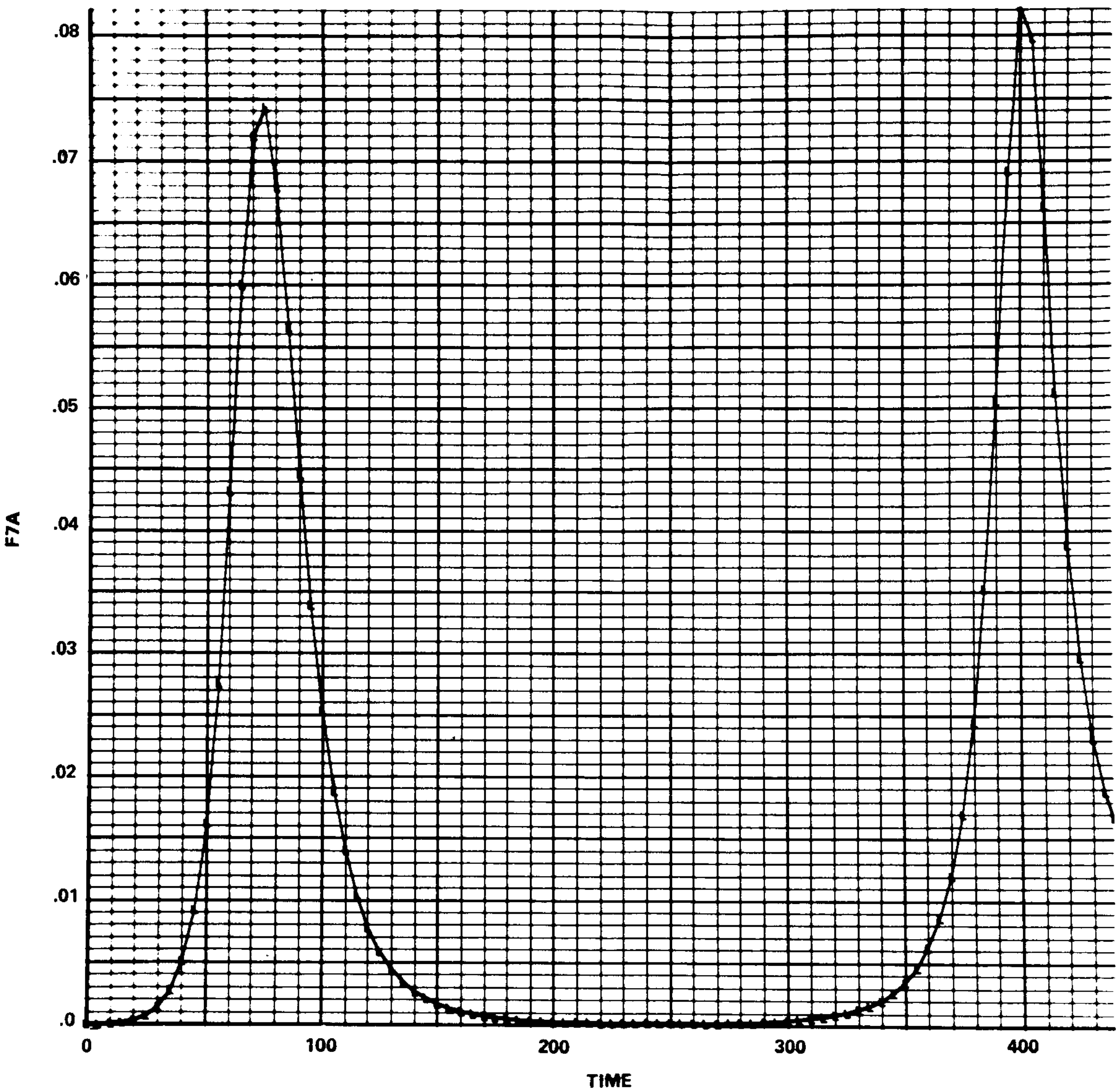

Figure A-7. Acceleration component versus time.

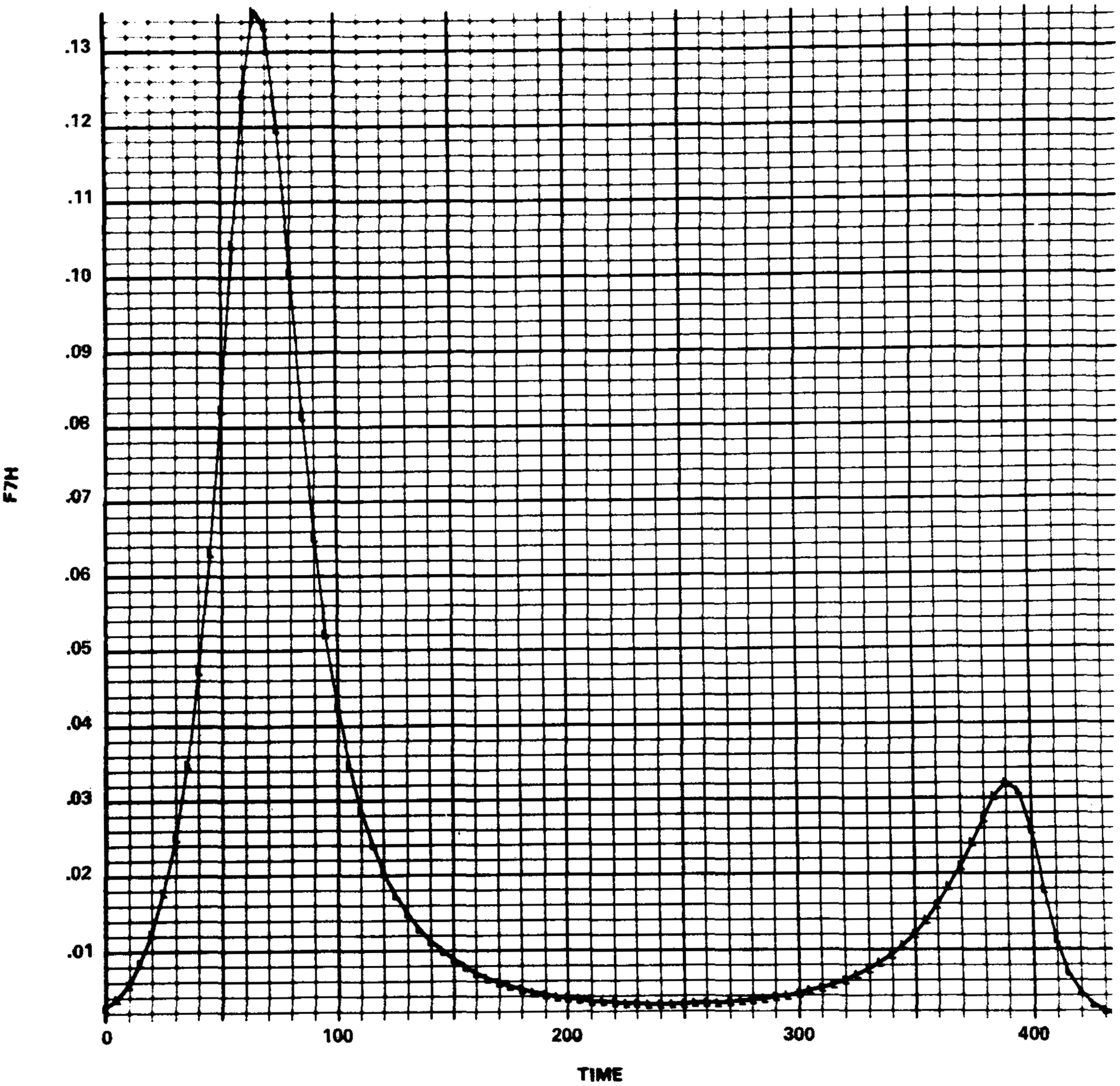

Figure A-8. Heating component versus time.

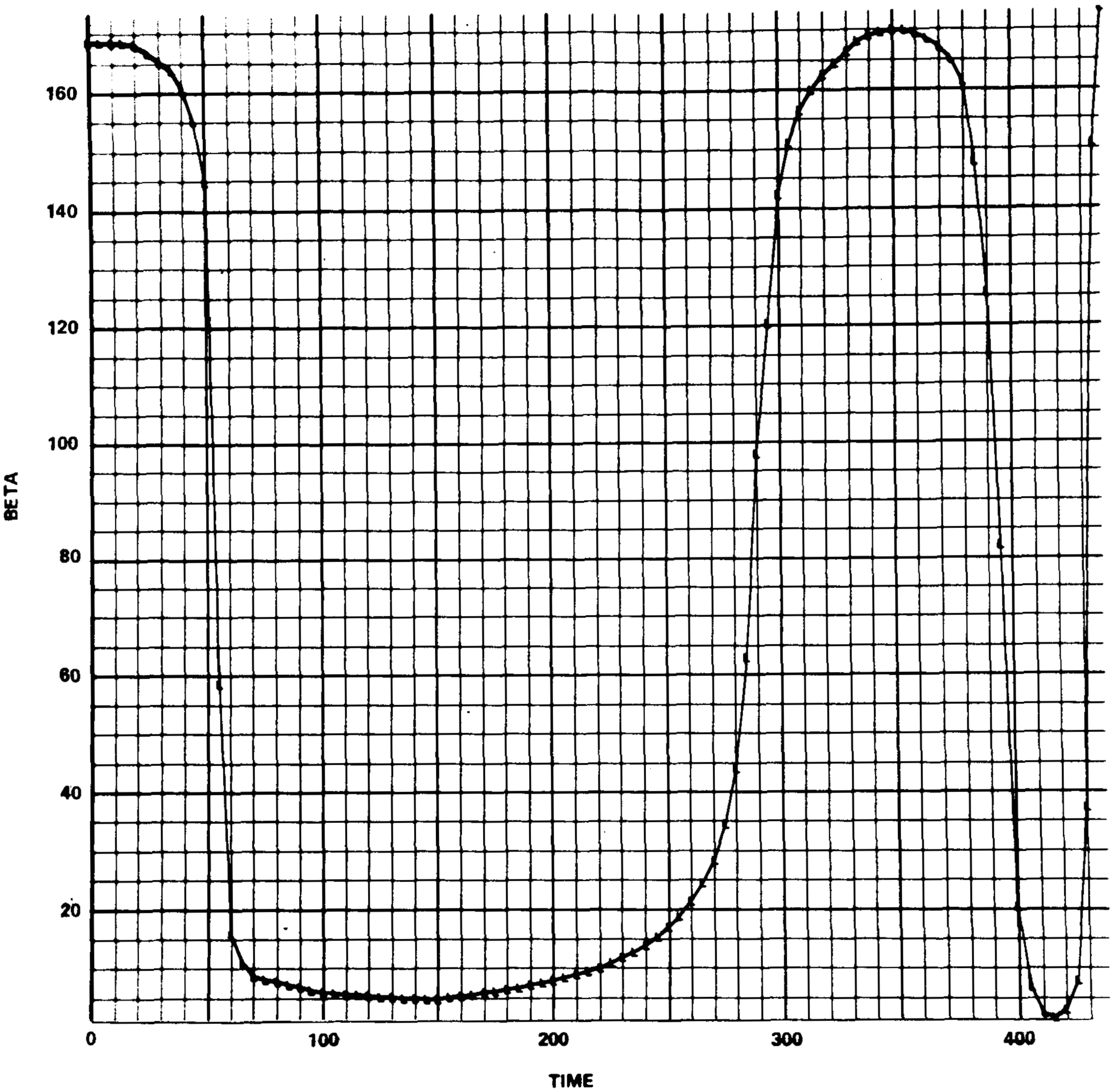

Figure A-9. Roll angle versus time.

Computer Listing of Steepest Ascent Optimization Program

```
  154312153800 28C111 KEENUMBIN406  ,390470,04,12,140CE 5438     2      1      011Y**
          $JOB   5438-03 C111 KEENUMBIN406  ,390470,04,12,140CE
          $EXECUTE       IBJOB
IBJOB VERSION 5 HAS CONTROL.
$IBJOB          MAP
$IEDIT          SYSLB3.SCHF01
$IBLDR IBM001
$IBLDR IBM002
$IBLDR IBM003
$IBLDR IBM004
$IBLDR IBM005
$IBLDR IBM006
$IBLDR IBM007
$IBLDR IBM008
$IBLDR IBM009
$IBLDR IBM010
$IBLDR IBM011
$IBLDR IBM012
$IBLDR IBM013
$IBLDR IBM014
$IBLDR IBM015
$IBLDR IBM016
$IBLDR IBM028
$IBLDR IBM029
$IBLDR IBM030
$IEDIT
$DATA

BEGIN LOADING     154101
```

```
IBLDR

                    * MEMORY MAP *

SYSTEM                                00000 THRU  02717
FILE BLOCK ORIGIN                     02720
FILES         1.     UNIT01
              2.     UNIT02
              3.     UNIT03
              4.     UNIT04
              5.     UNIT05
              6.     UNIT06
              7.     UNIT07
              8.     UNIT08
              9.     UNIT09
             10.     UNIT10
             11.     UNIT11
             12.     UNIT12
             13.     UNIT13
             14.     UNIT14
             15.     UNIT15
FILE LIST ORIGIN                      03234
PRE-EXECUTION INITIALIZATION          03242
CALL ON OBJECT PROGRAM                03321
OBJECT PROGRAM                        03306 THRU  67226

   DECK   ORIGIN      CONTROL SECTIONS  (/NAME/=NON 0 LENGTH, (LOC)=DELETED, *=NOT REFERENCED)

 1. IBM301  03306     /NTEV   / 03310    EVEN     03307    /IINPUT/ 03312    EVEN     03311    /LIST   / 03314
                       EVEN     03313   /ISCTCH/ 03316     EVEN     03315   /IEDIT  / 03320    EVEN     03317
                      /INCARD/ 03322     EVEN     03321   /INCODE/ 03442   /IEOS   / 03446    EVEN     03445
                      /KC     / 03450    EVEN     03447   /MODE   / 03452    EVEN     03451   /INAME  / 03454
                       EVEN     03453   /ID     / 03620   /IBLANK/ 03626   /ICHAR  / 03630    EVEN     03627
                      /ISOL   / 03632    EVEN     03631   /LINPG  / 03634    EVEN     03633   /NTERP  / 03636
                      /ICOUNT/ 03640     EVEN     03637   /MSG    / 04024   /LNAME  / 04100   /ICALLS/ 04220
                      /EVTFLG/ 04222     EVEN     04227    ......   12242 *
 2. IBM302  12256     /ICHAR /(03630)   /IEDIT /(03320)   /LIST  /(03314)   /ISCTCH/(03316)   /INCARD/(03322)
                      /INCODE/(03442)   /IEOS  /(03446)   /MODE  /(03452)   /ID    /(03620)   /ISOL  /(03632)
                      /MSG   /(04024)   /KC    /(03450)   /LNAME /(04100)   /IBLANK/(03626)   /INAME /(03454)
                       EVEN     12257    METHOD   20430
 3. IBM303  20454     /IEDIT /(03320)   /LIST  /(03314)   /IEOS  /(03446)   /KC    /(03450)   /MODE  /(03452)
                      /INAME /(03454)   /ID    /(03620)   /NTERP /(03636)    EVEN     20455    ITER8    21277
 4. IBM304  21313     /LNAME /(04100)   /MSG   /(04024)   /INCODE/(03442)   /IBLANK/(03626)   /INCARD/(03322)
                      /ICHAR /(03630)   /IDEBUG/ 21314    /ID    /(03620)   /INAME /(03454)   /KODE  / 21316 *
                       EVEN     21315   /MODE  /(03452)   /KC    /(03450)   /IEOS  /(03446)   /IEDIT /(03320)
                      /LIST  /(03314)   /NTERP /(03636)   /ICALLS/(04220)   /EVTFLG/(04222)    EVEN     21317
                       SXTION   24315
 5. IBM305  24333     /NTEV  /(03310)   /INCARD/(03322)   /INCODE/(03442)   /KC    /(03450)   /ID    /(03620)
                      /MODE  /(03452)   /ICHAR /(03630)   /LIST  /(03314)   /MSG   /(04024)   /IDEBUG/(21314)
                      /IEOS  /(03446)   /IEDIT /(03320)   /LNAME /(04100)   /ICALLS/(04220)    EVENT    41752
```

```
 6. IBM006   41776    /IEDIT /(03320)   /ISCTCH/(03316)   /LIST  /(03314)   /MSG   /(04024)   /IDEBUG/(21314)
                      /KC    /(03450)   /ISOL  /(03632)   /INCARD/(03322)   /INCODE/(03442)   /NTERP /(03636)
                      /LINPG /(03634)   /MODE  /(03452)   /ID    /(03620)   /IEOS  /(03446)   /ICHAR /(03630)
                      /IBLANK/(03626)   /LNAME /(04100)    EVEN    41777     IO     46532
 7. IBM007   46546    /EVTFLG/(04222)   /INCODE/(03442)   /KC    /(03450)   /LINPG /(03634)   /INCARD/(03322)
                      /MODE  /(03452)   /NTERP /(03636)    EVEN    46547     INTERP  50453
 8. IBM008   50467     ICONPR  50523
 9. IBM009   50551    /IBLANK/(03626)   /ID    /(03620)   /KC    /(03450)   /INCARD/(03322)   /ICHAR /(03630)
                      /IEOS  /(03446)    GETHAM  50763
10. IBM010   51006    /IEOS  /(03446)   /IEDIT /(03320)   /KC    /(03450)   /INCARD/(03322)   /MODE  /(03452)
                      /LIST  /(03314)   /ISOL  /(03632)   /IINPUT/(03312)   /ISCTCH/(03316)   /INCODE/(03442)
                       EVEN    51007     GETCRD  51267
11. IBM011   51301    /INCARD/(03322)   /KC    /(03450)   /ICHAR /(03630)   /IEOS  /(03446)   /IBLANK/(03626)
                       GETCTR  51356
12. IBM012   51372     EVEN    51373     BCCBIN  51474
13. IBM013   51533     BINBCC  51637
14. IBM014   51675     NUMBER  51763
15. IBM015   52010    /KC    /(03450)   /INCARD/(03322)    EVEN    52011     ISOLST  52076
16. IBM016   52122    /MSG   /(04024)   /ICOMNT/(03640)
17. IBM028   52122    /IEDIT /(03320)   /INCARD/(03322)   /INCODE/(03442)   /KC    /(03450)   /ID    /(03620)
                      /MSG   /(04024)   /LNAME /(04100)   /ICHAR /(03630)   /IBLANK/(03626)   /ISOL  /(03632)
                      /IDEBUG/(21314)   /MODE  /(03452)   /LIST  /(03314)   /IEOS  /(03446)    EVEN    52123
                       DEFINE  53015
18. IBM029   53034    /INCODE/(03442)   /IEDIT /(03320)    EVEN    53035     PUTBUF  53171
19. IBM030   53240     EVEN    53241     TAND    53265
20. .LXCON   53301     .LXSTR  53301 *   .LXSTP  53304     .LXOUT  53352     .LXERR  53361     .LXERR2 53362 *
                       .LXCAL  53365 *   .LXRTN  53365     IBEXIT  53365 *   .DBCLS  53547 *   .LXARG  53727
                       .LO     53752 *   .CLSE   53760     .LFBL   53761 *   .LUNB   53762     .DFOUT  53763
                       .CKLNX  53765
21. .IODEF   54020     .DEFIN  54020     .ATTAC  54024 *   .CLOSE  54026     .OPEN   54030     .READ   54032
                       .WRITE  54034     .BSK    54044 *   .READR  54054     .RELES  54056 *   .LAREA  54067
                       .LFBLK  54105     .LTSX   54110 *   .AREAI  54122     .LUNBL  54130     .ENTRY  54134
                       .GOA    54167     .GO     54173     .DERR   54207     .NOPXI  54210     .COMXI  54212
                       .EX34   54234
22. .LXSL    54241     .LXSEL  54241     .LXSLI  54242     .LXTST  54245 *   .LXOVL  54305 *   .LXMOD  54347 *
                       .LXIND  54373 *   .LXDIS  54376 *   .LXFLG  54377 *   .LTCH   54400
23. .FPTRP   54406     .FFPT.  54406 *   .FPOUT  54546 *  /.COUNT/ 54552 *  /.DMQUF/ 54612 *   OVFLOW  54617 *
24. XIT      54626     EXIT    54626     .EXIT.  54626
25. FXEM     54627     .FXEM.  54627     .FXOUT  55171     .FXARG  55177    /.OPTW./ 55253 *
26. FOUT     55272     .FOUT.  55272
27. FCNV     55633     .FCON.  55633     .FCNV.  55656     .ENDFS  55671     .CNVSW  55673     .FDX1   55677
                       .FDX2   55700     .DBC    55702     .DBC10  56040     .DBC20  56066     .DDSW   56076
                       .DDFIX  56105     .FIXSW  56113     .DDBC   56170     EVEN    56415     .DDRS1  56434
                       .DDRS2  56436 *   .D1     56441     .D2     56443     .FERR2  56530 *   .ANPT   56564
                       .UNPT   56601     .LNTP   56664     .AOUT   56733     .DFLT   56751     .FLT    57105
                       .DEXPN  57175     .FXD    57176     .HOUT   57326     .INTG   57376     .LOUT   57514
                       .OOUT   57533     .XCF    57564     .TEST   60272     .KOUNT  60275     .LIST   60300
                       .DONE   60311     .OUTBF  60356     EVEN    60355     .BUF    60406     .QSTO   60407
                       .WIDTH  60410     .GAIN   60411     .GAIN1  60412     .FBDBF  60422     EVEN    60433
                       .DDDFL  60447     .DDFLG  60450     .MQD    60451     .PEX    60452     .FEXP   60453
                       .DIG    60454
```

```
28. FIOS     60472      .FIOS.  60472      .CONT.  60551 *    .SYSEF  60631 *    .FSEL.  60654      .FILR.  60660 *
                        .FRTB.  60667 *    .FRTD.  60674      .FILL.  60677      .FCLS   60701 *    .FOPN   60705 *
                        REOF    60711 *    .TOUT.  61054      .REED   61062 *    .BIN    61063 *    .FCT    61064
                        .FCKSZ  61066 *    SYSEOF  61105 *
29. FIOH     61164      .FIOH.  61164      STPNT   61246 *    .FFIL.  61751      .FRTN.  61776      MVLIST  62020 *
30. FWRD     62175      .FWRD.  62175
31. FRDD     62221      .FRDD.  62221
32. UN01     62247      .UN01.  62247
33. UN02     62250      .UN02.  62250
34. UN03     62251      .UN03.  62251
35. UN04     62252      .UN04.  62252
36. UN05     62253      .UN05.  62253
37. UN06     62254      .UN06.  62254      .BUFSZ  62255
38. UN07     62260      .UN07.  62260
39. FXP1     62261      .XP1.   62261
40. FEFT     62375      .FEFT.  62375
41. FRWT     62475      .FRWT.  62475
42. FSLDI    62602      .FSLI.  62620      .FSDI.  62626 *
43. FSLI     62637      .SLI.   62637      .SLI1.  62644      .SDI.   62652      .SDI1.  62660
44. FSLDO    62673      .FSLO.  62711      .FSDO.  62717 *
45. FSLO     62730      .SLO.   62730      .SLO2.  62736      .SDO.   62743      .SDO2.  62752
46. FTNC     62764      COTAND  62764 *    COTAN   62765 *    TAND    (53265)    TAN     62770      CRIT    63140 *
47. FVIO     63221      .FVIO.  63221
48. .IOCS    63334      .L(0)   63334      .MONSW  63354      .TEOR   63423      .DEFI.  63503      .JOINX  63547 *
                        .CLOS.  63566      .ATTC.  63601      .SH1    64013 *    .SH9    64055 *    .OPEN.  64076
                        .OP4    64124 *    .OP7    64155 *    .OP9.2  64171 *    .RLSE.  64243      .RER2.  64243
                        .READ.  64244      .RER1.  64267      .WRIT.  64271      .MNT1A  64512 *    .EOFEX  64573 *
                        .FEEIT  64643      .GTIOX  64664      .RW7    65002 *    .RE7    65425 *    .ENDTR  66066
                        .SEL59  66070 *    .BSR.   66507      .EOTOF  66634      .ETOF3  66642 *    .SWITC  66671
                        .TCHEX  67210      .BASIO  67213 *
49. .IOCSM   67216
50. UN08     67216      .UN08.  67216
51. UN09     67217      .UN09.  67217
52. UN10     67220      .UN10.  67220
53. UN11     67221      .UN11.  67221
54. UN12     67222      .UN12.  67222
55. UN13     67223      .UN13.  67223
56. UN14     67224      .UN14.  67224
57. UN15     67225      .UN15.  67225
58. KIKOFF   67226
I/O BUFFERS                          67227 THRU  77762

UNUSED CORE                          77763 THRU  77777

BEGIN EXECUTION  154120
```

```
MARVES PROCESSOR VERSION 19/051270

C
C     APOLLO 3-D REENTRY
C     OPTIMIZATION PROBLEM
C
      REAL M,MU,L,LAMO
      REAL LAMPSI,LAMPHI,LAMOMG,I1,I2,I3
      INTEGER BCDX,BCDY,BCCY
      DOUBLE PRECISION WORK
      DOUBLE PRECISION AVI2
      DOUBLE PRECISION DET
      LOGICAL LPLOT
      REAL ID
      DIMENSION ID(5,5)
      DIMENSION AVI2(25)
      DIMENSION WORK(15)
      DIMENSION X(8),F(8),PSI(5),TTABLE(100),BETAB(100),TTAB(5,7),
     1 XFT(50,17)
      DIMENSION FMAT(7,7),PSILAM(35),PSILMD(35),AI2(25),DAI2(25)
      DIMENSION PLOTAB(200,10),BCDY(9)
      DIMENSION LAMPSI(7,5),LAMPHI(7),LAMOMG(7),FMA(16),TEMP(17),GMAT(7)
      DIMENSION DLMPSI(7,5),DLMPHI(7),DLMOMG(7),DPSI(5),TLPSI1(5,7),TLPH
     *I1(1,7),DI1(5),DI2(5,5),ATEMP(5),BTEMP(5),I1(5),I2(5,5)
     * ,ATABLE(100)
      DIMENSION DEL(100)
      DIMENSION ARRLAM(2,50),TLLAM(7,5),TLPHI(7),TG(7),AL(49),AIV2(5,5),
     1 PSIW(5)
      DIMENSION BCDX(12),BCCY(12)
      EQUIVALENCE (AL(1),TLLAM(1,1)),(AL(36),TLPHI(1)),(AL(43),TG(1))
      EQUIVALENCE (LAMPSI(1,1),PSILAM(1)), (DLMPSI(1,1),PSILMD(1)),
     1 (I2(1,1),AI2(1)),(DI2(1,1),DAI2(1))
      DATA (BCDX(I),I=1,12)/6HTIME  ,11*6H
      DATA (BCDY(I),I=1,9)/1HH,5HTHETA,5HDELTA,1HV,5HGAMMA,1HA,3HF7A,
```

```
     * 3HF7H,4HBETA/
      DATA (BCCY(I),I=1,12)/12*6H
C
C     LPLOT- LOGICAL PLOT FLAG SET TRUE FOR PLOTS
C     AKO- KO USED TO CALCULATE THE CHANGE IN BETA TABLE
C     AK1- K1 USED TO CALCULATE THE CHANGE IN BETA TABLE
C     DPO- INITIAL VALUE FOR CONSTANT USED TO CALCULATE BETA TABLE CHANGE
C     PROGRAM HAS TWO WORKING TAPES (LAMDA,STATE AND DERIVIATIVE VECTOR(SDV))
C
      INITIALIZE
      INTEGRATION ACCURACY=11
      IBK=0
      ICYC=0
      NPTS=61
      DO 300 I=1,100
      ATABLE(I)=0.
 300  TTABLE(I)=-100.
      INPUT NAMELIST(INPUT)R,CD,LAMO,M,S,CL,RHOO,BSTAR,MU,TO,DTSTEP,X,
     * TTABLE,BETAB,DTPRNT,DTPLOT
     * , LPLOT,AKO,AKI,DPO,W
      WRITE(6,INPUT)
      IF(LPLOT)GO TO 30
      SET STATUS(PTOT)OFF
 30   CONTINUE
 230  SKIP PAGE
      AK1=AKI
      OUTPUT TABLES TTABLE(61),BETAB
      IF(LPLOT)CALL CAMRAV(935)
      ISECT=1
      DTSTEP=ABS(DTSTEP)
      DTPRNT=ABS(DTPRNT)
      REWIND 13
      REWIND 9
```

```
   25 CONTINUE
      SKIP PAGE
      IDUM=0
      SECTION 1
C
C     INITIALIZE FOR TRAJECTORY PORTION
C
      TIME=TO
      PRINTT=TIME
      JJ=0
      KL=0
      PLOTT=0.
      N=100
      ICAS=ICAS+1
      SECTION 2
C
C     INITIALIZE FOR BACKWARD INTEGRATION OF TRAJECTORY
C
      DTSTEP=-DTSTEP
      DTPRNT=-DTPRNT
      TI=TIME
      SKIP PAGE
      OUTPUT TABLES (TABLE(61),,BETAB
      SECTION 3
C
C     INITIALIZE INTEGRATION OF LAMDA  MATRICES
C
      IDUM=0
      TIME=TI
      DO 4 I=1,7
      LAMPHI(I)=0.
      LAMOMG(I)=0.
      DO 4 K=1,5
    4 LAMPSI(I,K)=0.
      LAMPHI(7)=1.0
      LAMOMG(4)=1.
      LAMPSI(1,1)=1.
      LAMPSI(2,2)=1.
      LAMPSI(3,3)=1.
      LAMPSI(5,4)=1.
      LAMPSI(6,5)=1.
      DO 5 I=1,7
      DO 5 K=1,7
    5 FMAT(I,K)=0.
      DO 12 K=1,7
   12 GMAT(K)=0.
      DO 250 I=1,5
      I1(I)=0.
      DO 250 K=1,5
  250 I2(I,K)=0.
      I3=0.
C
C     READ FIRST RECORD  FROM SOV TAPE
C
      REWIND 9
      READ(9)XFT
      K=25
      DO 26 I=1,K
      KK=50-I+1
      DO 27 IA=1,17
   27 TEMP(IA)=XFT(KK,IA)
      DO 28 IA=1,17
   28 XFT(KK,IA)=XFT(I,IA)
      DO 29 IA=1,17
   29 XFT(I,IA)=TEMP(IA)
   26 CONTINUE
      DIFFERENTIAL EQUATIONS
      SECTION 1,2
      T=R+X(1)
      G=-MU/(T*T)
      RHO=RHOO*EXP(-BSTAR*X(1))
      TI=RHO*S*X(4)*X(4)
      BD=TI*CD/(2.*M)
      BL=TI*CL/(2.*M)
      L=.5*TI*CL
      D=.5*TI*CD
C
C     TABLE LOOKUP FOR BETA
C
      IDOM=0
      CALL LATLUM(IERR,IDUM,1,N,2,TIME,TTABLE,BETAB,BETA)
      BETA=BETA/57.2957795
C
C     DERIVIATIVES OF STATE VECTOR (EQ. 13)
C
      F(1)=X(4)* SIN(X(5))
      F(2)=X(4)* COS(X(5))* COS(X(6))/(T* COS(X(3)))
      F(3)=X(4)* COS(X(5))* SIN(X(6))/T
      F(4)= G *  SIN(X(5)) - BD
      F(5)= G *  COS(X(5)) /X(4) + X(4) *  COS(X(5)) / T + BL *
     1  COS(BETA) / X(4)
      F(6)= - X(4) *  COS(X(5)) *  COS(X(6)) *  TAN(X(3)) /T - BL *
     1  SIN(BETA) /(X(4) *  COS(X(5)) )
      F(7)=SQRT( L*L + D*D ) / M
      F(8)=LAMD * SQRT(RHO) * X(4) * X(4) * X(4)
      SECTION 3
    7 IF(TIME.GT.XFT(1,1))GO TO 6
      READ(9)XFT
      K=25
      DO 46 I=1,K
      KK=50-I+1
      DO 47 IA=1,17
```

```
 47   TEMP(IA)=XFT(KK,IA)
      DO 48 IA=1,17
 48   XFT(KK,IA)=XFT(I,IA)
      DO 49 IA=1,17
 49   XFT(I,IA)=TEMP(IA)
 46   CONTINUE
      GO TO 7
  6   CONTINUE
C
C     TABLE LOOKUP FOR STATE VARIABLES AND THEIR DERIVIATIVES
C
      IDUM=0
      CALL LATLUM(IERR,IDUM,16,50,2,TIME,XFT(1,1),XFT(1,2),FMA(1))
      IDOM=0
      CALL LATLUM(IERR,IDOM,1,N,2,TIME,TTABLE,BETAB,BETA)
      BETA=BETA/57.2957795
      SDEL=SIN(FMA(3))
      CDEL=COS(FMA(3))
      SGAM=SIN(FMA(5))
      CGAM=COS(FMA(5))
      SA=SIN(FMA(6))
      CA=COS(FMA(6))
      FMAT(1,4)=-SGAM
      T=R+FMA(1)
      G=-MU/(T*T)
      RHO=RHOO*EXP(-BSTAR*FMA(1))
      T1=RHO*S*FMA(4)*FMA(4)
      BD=T1*CD/(2.*M)
      BL=T1*CL/(2.*M)
      L=.5*T1*CL
      D=.5*T1*CD
      T1=FMA(4)*CGAM
      T2= CDEL * T
C
C     CALCULATE F(T) (EQ. 26)
C
      FMAT(1,5)=-T1
      FMAT(2,1)=T1 * CA  /(T2 * T )
      FMAT(2,3)=-FMAT(2,1)* SDEL * T  / CDEL
      FMAT(2,4)=- CGAM * CA / T2
      FMAT(2,5) = FMA(4) * CA * SGAM / T2
      FMAT(2,6)= T1 * SA / T2
      FMAT(3,1)= FMAT(2,6) * CDEL / T
      FMAT(3,4)= - CGAM * SA / T
      FMAT(3,5)=   FMA(4) * SA * SGAM / T
      FMAT(3,6)= - T1 * CA / T
      FMAT(4,1)= 2.* G * SGAM / T - BSTAR * D / M
      FMAT(4,4)= RHO * S * CD * FMA(4) / M
      FMAT(4,5)= - G * CGAM
      FMAT(5,1)= 2. * G * CGAM / (FMA(4)*T) + T1/ (T*T) + BSTAR*BL*
     1 COS(BETA) / FMA(4)
      FMAT(5,4)=G*CGAM/(FMA(4)*FMA(4)) - CGAM/T - RHO*S*CL* COS(BETA)/
     1 (2.*M)
      FMAT(5,5)= G*SGAM/FMA(4) + FMA(4)*SGAM/ T
      T2= CGAM * CA
      T3=T*CDEL
      FMAT(6,1)=-FMA(4)*T2*SDEL/T3
      FMAT(6,3)= FMA(4) * T2 /(CDEL*T3)
      FMAT(6,4)= T2*SDEL/ T3     + RHO*S*CL* SIN(BETA)/(2.*M*CGAM)
      FMAT(6,5)=-FMA(4)*CA*SDEL*SGAM / T3 - RHO*S*FMA(4)* SIN(BETA)*SGAM
     1 *(-CL)/(2.*M*CGAM*CGAM)
      FMAT(6,6)= - FMA(4)*CGAM*SDEL*SA / T3
      FMAT(7,1)= BSTAR*( SQRT(L*L + D*D ) / M  + LAHO*FMA(4)*FMA(4)*
     1 FMA(4)*SQRT(RHO) / 2. )
      FMAT(7,4)=-RHO*S*FMA(4)*(L*CL+D*CD) /(M*SQRT(L*L+D*D) )
      GMAT(5)= - BL *  SIN(BETA) / FMA(4)
      GMAT(6)= - BL *  COS(BETA) / ( FMA(4) * CGAM )
      CALL MATMUL(DLMPSI,FMAT,LAMPSI,-7,7,5)
      CALL MATMUL(DLMPHI,FMAT,LAMPHI,-7,7,1)
      CALL MATMUL(DLMOMG,FMAT,LAMOMG,-7,7,1)
      DPSI(1)=FMA(9)/FMA(12)
      DPSI(2)=FMA(10)/FMA(12)
      DPSI(3)=FMA(11)/FMA(12)
      DPSI(4)=FMA(13)/FMA(12)
      DPSI(5)=FMA(14)/FMA(12)
C
C     CALCULATE LAMDA PSI ONE (EQ. 40)
C     CALCULATE LAMDA PHI ONE (EQ. 40)
C
      DO 303 I=1,7
```

```
      DO 303 K=1,5
  303 TLPSI1(K,I)=DPSI(K)*LAMOMG(I)
      DO 13 I=1,5
      DO 13 K=1,7
      TTAB(I,K)=LAMPSI(K,I)
   13 TLPSI1(I,K)=TTAB(I,K)-TLPSI1(I,K)
      DO 14 K=1,7
   14 TLPHI1(1,K)=LAMPHI(K)-((FMA(15)+FMA(16))/FMA(12))*LAMOMG(K)
C
C     ATEMP= SUMMATION OF G * LAMDA PHI 1
C
      ATEMP(1)=0.
      DO 301 I=1,7
  301 ATEMP(1)=GMAT(I)*TLPHI1(1,I)+ATEMP(1)
      CALL MATMUL(BTEMP,TLPSI1,GMAT,5,7,1)
C
C     DI1 CALCULATION (EQ. 48)
C
      DO 15 K=1,5
   15 DI1(K)= BTEMP(K) * 8 * ATEMP(1)
      CALL MATMUL(ATEMP,GMAT,TLPSI1,-1,-7,5)
C
C     DI2 CALCULATION (EQ. 45)
C
      DO 16 K=1,5
   16 ATEMP(K) = ATEMP(K) * 4
      CALL MATMUL(DI2,BTEMP,ATEMP,5,1,5)
C
C     DI3 CALCULATION (EQ. 48)
C
      ATEMP(1)=0.
      DO 304 I=1,7
  304 ATEMP(1)=GMAT(I)*TLPHI1(1,I)+ATEMP(1)
      BTEMP(1)=0.
      DO 302 I=1,7
  302 BTEMP(1)=TLPHI1(1,I)*GMAT(I)+BTEMP(1)
      DI3=BTEMP(1)   * 4 * ATEMP(1)
      INTEGRATION
      SECTION 1,2
      METHOD  SHANKS 8,10  ORDER=1, X(1)8,F(1)8
      SECTION 3
      METHOD SHANKS 8,10
     1                    ORDER=1,PSILAM(1)35,PSILMO(1)35,LAMPHI(1)7,
     1 DLMPHI(1)7,LAMOMG(1)7,DLMOMG(1)7,I1(1)5,DI1(1)5,AI2(1)25,
     2 DAI2(1)25,I3,DI3
      EVENTS
      EVENT VARIABLES
      SECTION 2
      JJ=JJ+1
      XFT(JJ,1)=TIME
      DO 1 K=1,8
      XFT(JJ,K+1)=X(K)
    1 XFT(JJ,K+9)=F(K)
      IF(JJ.LT.50)GO TO 2
C
C     WRITE SOV TAPE
C
      WRITE(9)XFT
      JJ=1
      XFT(1,1)=XFT(50,1)
      DO 3 K=1,8
      XFT(1,K+1)=XFT(50,K+1)
    3 XFT(1,K+9)=XFT(50,K+9)
    2 CONTINUE
      SECTION 3
      IF(IBK.EQ.2)GO TO 231
      DO 232 I=1,59
  232 ARRLAM(2,I)= ARRLAM(1,I)
  231 ARRLAM(1,1)=TIME
      DO 233 I=1,7
      ARRLAM(1,I+36)=TLPHI1(1,I)
  233 ARRLAM(1,I+43)=GMAT(I)
      IJKL=2
      DO 234 I=1,7
      DO 234 J=1,5
      ARRLAM(1,IJKL)=TLPSI1(J,I)
  234 IJKL=IJKL+1
      IF(IBK.EQ.3)GO TO 235
C
C     WRITE LAMDA TAPE
C
      WRITE(13) ARRLAM
  235 IBK=100
      SECTION 1,2
      EVENT(PRINT)TIME=PRINTT,POST
      PRINTT=PRINTT+DTPRNT
      SECTION 1
      EVENT(PTOT)TIME=PLOTT
      PLOTT=PLOTT+DTPLOT
C
C     STORE VARIABLES TO PLOT
C
      KL=KL+1
```

```
      PLOTAB(KL,1)=TIME
      DO 20 I=1,6
20    PLOTAB(KL,I+1)=X(I)
      PLOTAB(KL,8)=F(7)
      PLOTAB(KL,9)=F(8)
      PLOTAB(KL,10)=BETA*57.2957795
      SECTION 2,3
      EVENT(EVUP)TIME=TI,DISC,POST
      EVENUP DTSTEP
      PRINTT=AINT(TI)+DTPRNT
      SET STATUS(EVUP)OFF
      SECTION 2,3
      EVENT(STOP)TIME=0.,POST
      IF(ISECT.EQ.3)GO TO 878
      JJ=JJ+1
      DO 879 K=JJ,50
      DO 879 I=1,17
879   XFT(K,I)=-K
C
C     FINISH SDV TAPE
C
      WRITE(9)XFT
      END FILE 9
878   CONTINUE
      EXIT TO TERMINAL CONDITIONS
      SET STATUS(EVUP)ON
      SECTION 1
      EVENT(EXIT)TIME=498.17852,POST
      PSI(1)= X(1) -75509. / 5280.
      PSI(2)= X(2) - 24.1 / 57.2957795
      PSI(3)= X(3) + .6/ 57.2957795
      PSI(4)= X(5) + 44.3 / 57.2957795
      PSI(5)= X(6) + 29.4 /57.2957795
      IF(.NOT.LPLOT)GO TO 9
C
C     STORE END CONDITIONS OF PLOT VARIABLES
C
      KL=KL+1
      PLOTAB(KL,1)=TIME
      DO 21 I=1,6
21    PLOTAB(KL,I+1)=X(I)
      PLOTAB(KL,8)=F(7)
      PLOTAB(KL,9)=F(8)
      PLOTAB(KL,10)=BETA*57.2957795
9     CONTINUE
      SECONDARY EVENT ACTION
      SECTION 1,2
      OUTPUT LIST TIME,DTSTEP/X(1),X(2),X(3),X(4)/X(5),X(6),X(7),X(8)/
     * F(1),F(2),F(3),F(4)/F(5),F(6),F(7),F(8)/BETA///
      TERMINAL COMPUTATIONS
      SECTION 1
      Q=X(7)+X(8)
      IF(.NOT.LPLOT)GO TO 10
C
C     CALL PLOT ROUTINES
C
      DO 22 I=1,9
      BCCY(I)=BCDY(I)
      CALL QUIK3V(-1,35,BCDX,BCCY,    -KL,PLOTAB(1,1),PLOTAB(1,I+1))
22    CONTINUE
      CALL CLEAN(KL)
10    CONTINUE
      SECTIONS ALL
      IF(ISECT.EQ.3)GO TO 969
      ISECT=ISECT+1
      GO TO 25
969   CONTINUE
      OUTPUT LIST Q/PSI(1),PSI(2),PSI(3),PSI(4),PSI(5)
C
C     FINISH LAMDA TAPE
C
      WRITE(13) ARRLAM
      END FILE 13
      REWIND 13
C
C     COMPUTE STEEPEST ASCENT OPTIMIZATION PARAMETERS
C
      DO 200 I=1,5
200   PSIW(I)= -AKO * PSI(I)
      K=0
      DO 201 J=1,5
      DO 201 I=1,5
      K=K+1
201   AVI2(K)=I2(I,J)
      WRITE(6,220)(PSIW(I),AVI2(I),AVI2(I+5),AVI2(I+10),AVI2(I+15),
     1  AVI2(I+20),I=1,5)
      CALL INVRT(AVI2,5,0,DET)
      IF(DET.NE.0.0)GO TO 202
      WRITE(6,203)
203   FORMAT(1H1,16H SINGULAR MATRIX / 15H RUN TERMINATED )
      CALL KILOFF
202   CONTINUE
      K=0
      DO 368 J=1,5
      DO 368 I=1,5
      K=K+1
```

```
 368  AIV2(I,J)=AVI2(K)
      CALL MATMUL(ATEMP,AIV2,PSI,5,5,1)
      CALL MATMUL(BTEMP,PSI,ATEMP,-1,5,1)
      DP = AKO*AKO * ( DPO*DPO - AKI*AKI*BTEMP(1) )
      IF(DP.GE.0.0) GO TO 204
C
C     PARAMETER TOO LARGE , REDUCE IT
C
      AKI=AKI-.2*AKI
      DO 205 I=1,5
 205  PSI(I) = AKI * PSI(I)
      IF(AKI.GE.0.)GO TO 202
      WRITE(6,206)
 206  FORMAT(1H1,18H KI LESS THAN ZERO/15H RUN TERMINATED  )
      CALL KIKOFF
 204  CALL MATMUL(ATEMP,AIV2,I1,5,5,1)
      CALL MATMUL(ID,I2,AIV2,5,5,5)
      WRITE(6,220) (PSI(I),AIV2(I,1),AIV2(2,I),AIV2(3,I),AIV2(4,I),
     1 AIV2(5,I) ,I=1,5)
      WRITE(6,330)(ID(1,I),ID(2,I),ID(3,I),ID(4,I),ID(5,I),I=1,5)
 330  FORMAT(30X,15HIDENTITY MATRIX,/ (5E18.8) )
      CALL MATMUL(BTEMP,I1,ATEMP,-1,5,1)
      OUTPUT TABLES ATEMP(5),BTEMP,I1
      TEMP(1)= I3 - BTEMP(1)
      OUTPUT LIST DP,TEMP(1)
      DPC= SQRT( DP / ABS( TEMP(1) )  )
C     DPSIDP  CALCULATION
      DPSIDP= - SQRT( ABS( TEMP(1) )  )
      DO 207 I=1,100
      JJ=101-I
      IF( TTABLE(JJ) .LE. 0.0 ) GO TO 207
 210  READ(13) ARRLAM
      IF( TTABLE(JJ) .GT. ARRLAM(2,1) ) GO TO 207
      IF( TTABLE(JJ) .LT. ARRLAM(1,1) ) GO TO 210
      IDM=0
      CALL LATLOM(IERR,IDM,49,2,2,TTABLE(JJ),ARRLAM(1,1),ARRLAM(1,2),
     1 AL(1))
      CALL MATMUL(TEMP,TLLAM,ATEMP,-7,5,1)
      DO 213 IJKL=1,7
 213  TEMP(IJKL+7)=TLPHI(IJKL)-TEMP(IJKL)
      CALL MATMUL(BTEMP,TG,TEMP(8),-1,7,1)
      ATABLE(JJ)=W * BTEMP(1) * DPC
      CALL MATMUL(ATEMP,AIV2,PSI,5,5,1)
      CALL MATMUL(TEMP,TS,TLLAM,-1,-7,5)
      CALL MATMUL(BTEMP,TEMP,ATEMP,1,5,1)
C
C     THE CHANGE (A)  TO THE BETA TABLE CALCULATED
C
      ATABLE(JJ)= ATABLE(JJ) * W * BTEMP(1)
 207  CONTINUE
C
C     CONVERT A TO DEGREES
C
      DO 701 I=1,100
 701  ATABLE(I)=ATABLE(I)*57.2957795
C
C     ADD A TO BETA TABLE
C
      DO 215 I=1,100
 215  BETAB(I)=BETAB(I) + ATABLE(I)
      ISOL=0
      DO  216  I=1,5
      IF( ABS(PSI(I)) .GT. .001 ) GO TO 217
 216  CONTINUE
C
C     SOLUTION  FOUND
C
      ISOL=1
 217  WRITE(6,218)( I1(I),I2(1,I),I2(2,I),I2(3,I),I2(4,I),I2(5,I),
     1  I=1,5 )
 218  FORMAT(1H1,8X, 2HI1,36X,2HI2,/ (E18.8,5X,5E18.8) )
      WRITE(6,219) AKO,I3,DPO, DPSIDP, DP, AKI
 219  FORMAT(1H0,4H KO ,E16.8, 4H I3 ,E16.8, 5H DPO ,E16.8, 9H DPSI/DP
     1E16.8,4H DP ,E16.8,4H KI ,E16.8)
 220  FORMAT(1H0,3X,4HPSI,,40X,10HI2 INVERSE,/ (E18.8,5X,5E18.8) )
      ICYC=ICYC+1
      WRITE(6,221) ICYC
 221  FORMAT(1H0,14H END OF CYCLE ,I5)
      OUTPUT TABLES ATABLE(61)
C
C     CALCULATE STANDARD DEVIATION OF A TABLE
C
      ADEL=0.
      DO 702 I=1,NPTS
 702  ADEL=ADEL+ATABLE(I)
      FNPTS=NPTS
      ADEL=ADEL / FNPTS
      SD=0.
      DO 704 I=1,NPTS
      DEL(I)=(ATABLE(I)-ADEL)**2
 704  SD=SD+DEL(I)
      FNPTS=NPTS-1
      SD=SQRT(SD/FNPTS)
      WRITE(6,222)SD
 222  FORMAT(1H0,19H STANDARD DEVIATION,2X,E15.8)
      IF(ISOL .EQ. 0) GO TO 230
      IF(LPLOT)CALL CLEAN
      STOP
      END
*END
```

```
        740 LINES OUTPUT.
            $IBSYS
            $REWIND          SYSCK1
            $EXECUTE         IBJOB
IBJOB VERSION 5 HAS CONTROL.
$IBJOB          MAP
$IEDIT          SYSCK1,SCHF01
$IBFTC APOLLO

BEGIN COMPILING  154416

          APOLLO      -  EFN   SOURCE STATEMENT  -  IFN(S)  -

C
C     APOLLO 3-D REENTRY
C     OPTIMIZATION PROBLEM
C
      REAL M,MU,L,LAMO
      REAL LAMPSI,LAMPHI,LAMOMG,I1,I2,I3
      INTEGER BCDX,BCDY,BCCY
      DOUBLE PRECISION WORK
      DOUBLE PRECISION AVI2
      DOUBLE PRECISION DET
      LOGICAL LPLOT
      REAL ID
      DIMENSION ID(5,5)
      DIMENSION AVI2(25)
      DIMENSION WORK(15)
      DIMENSION X(8),F(8),PSI(5),TTABLE(100),BETAB(100),TTAB(5,7),
     1 XFT(50,17)
      DIMENSION FMAT(7,7),PSILAM(35),PSILMD(35),AI2(25),DAI2(25)
      DIMENSION PLOTAB(200,10),BCDY(9)
      DIMENSION LAMPSI(7,5),LAMPHI(7),LAMOMG(7),FMA(16),TEMP(17),GMAT(7)
```

```
      DIMENSION DLMPSI(7,5),DLMPHI(7),DLMUMG(7),DPSI(5),TLPSI1(5,7),TLPH
     *I1(1,7),DI1(5),DI2(5,5),ATEMP(5),BTEMP(5),I1(5),I2(5,5)
     * ,ATABLE(100)
      DIMENSION DEL(100)
      DIMENSION ARRLAM(2,50),TLLAM(7,5),TLPHI(7),TG(7),AL(49),AIV2(5,5),
     1 PSIW(5)
      DIMENSION BCDX(12),BCCY(12)
      EQUIVALENCE (AL(1),TLLAM(1,1)),(AL(36),TLPHI(1)),(AL(43),TG(1))
      EQUIVALENCE (LAMPSI(1,1),PSILAM(1)), (DLMPSI(1,1),PSILMD(1)),
     1 (I2(1,1),AI2(1)),(DI2(1,1),DAI2(1))
      DATA (BCDX(I),I=1,12)/6HTIME   ,11*6H
      DATA (BCDY(I),I=1,9)/1HH,5HTHETA,5HDELTA,1HV,5HGAMMA,1HA,3HF7A,
     * 3HF7H,4HBETA/
      DATA (BCCY(I),I=1,12)/12*6H
C
C     LPLOT- LOGICAL PLOT FLAG SET TRUE FOR PLOTS
C     AKO- KO USED TO CALCULATE THE CHANGE IN BETA TABLE
C     AK1- K1 USED TO CALCULATE THE CHANGE IN BETA TABLE
C     DPO- INITIAL VALUE FOR CONSTANT USED TO CALCULATE BETA TABLE CHANGE
C     PROGRAM HAS TWO WORKING TAPES (LAMDA,STATE AND DERIVIATIVE VECTOR(SDV))
C
      COMMON/ISECT/ISECT
      COMMON/SR0001/SR0001
      NAMELIST/INPUT / R,CD,LAMO,M,S,CL,RHOO,BSTAR,MU,TO,DTSTEP,X,TTABLE
     *,BETAB,DTPRNT,DTPLOT,LPLOT,AKO,AK1,DPO,W
31300 FORMAT(1H1)
31301 FORMAT((1H1 8X6HTTABLE12X6HBETAB /52(1X2E18.8 /)))
31302 FORMAT(1H1)
31303 FORMAT(1H1)
31304 FORMAT((1H1 8X6HTTABLE12X6HBETAB /52(1X2E18.8 /)))
      DOUBLE PRECISION SD0000,SD0001,SD0002
      COMMON/SD0000/SD0000 /SD0001/SD0001 /SD0002/SD0002
      COMMON/I00009/I00009/I00008/I00008
      DOUBLE PRECISION SD0040,SD0003,SDP003
      COMMON/SD0040/SD0040/SD0003/SD0003,SDP003/I00012/I00012
```

```
            APOLLO      -  EFN    SOURCE STATEMENT   -   IFN(S)   -

       DOUBLE PRECISION SD0201
       DIMENSION SD0201(  176),MV0201(     3)
       DATA MV0201/
      *  2,  8,  8/
       DOUBLE PRECISION SD0203
       DIMENSION SD0203( 1760),MV0203(    13)
       DATA MV0203/
      * 12, 35, 35,  7,  7,  7,  7,  5,  5, 25, 25,  1,  1/
 31305 FORMAT(4X6HTIME  E16.8 ,4X6HDTSTEPE16.8 /4X6HX(1)  E16.8 ,4X6HX(2)
      *  E16.8 ,4X6HX(3)  E16.8 ,4X6HX(4)  E16.8 /4X6HX(5)  E16.8 ,4X6HX(
      *6)  E16.8 ,4X6HX(7)  E16.8 ,4X6HX(8)  E16.8 /4X6HF(1)  E16.8 ,4X6H
      *F(2)  E16.8 ,4X6HF(3)  E16.8 ,4X6HF(4)  E16.8 /4X6HF(5)  E16.8 ,4X
      *6HF(6)  E16.8 ,4X6HF(7)  E16.8 ,4X6HF(8)  E16.8 /4X6HBETA  E16.8 /
      *// 1X)
 31306 FORMAT(4X6HQ     E16.8 /4X6HPSI(1)E16.8 ,4X6HPSI(2)E16.8 ,4X6HPSI(
      *3)E16.8 ,4X6HPSI(4)E16.8 ,4X6HPSI(5)E16.8 ,1X)
 31307 FORMAT((1H1 8X6HATEMP 12X6HBTEMP 12X6HI1    /52(1X3E18.8 /)))
 31308 FORMAT(4X6HDP    E16.8 ,4X6HTEMP(1E16.8 ,1X)
 31309 FORMAT((1H1 8X6HATABLE/52(1X1E18.8 /)))
       DOUBLE PRECISION SD0030
       COMMON /MV0009/MV0009( 5,4)/MV0006/MV0006( 5)
      */MV0007/MV0007( 5)/SD0030/SD0030( 5)
       COMMON /SR0010/SR0010( 5)
       COMMON /MV0010/MV0010( 1)
       DOUBLE PRECISION SD0004
       COMMON /SD0004/SD0004( 5)

       DIMENSION MV0030( 5)
       DATA (MV0030(I),I=1, 5)/  1101, 1100, 1001, 1101, 1101/
C
C********** INITIALIZATION **********
C
       I00002 =  5
       I00003 =  0
```

```
      I00004=0
      I00021=0
      I00015=0
      I00007=0
      I00005 =  5
C
C********** SET STATUS ALL EVENTS **********
C
      DO 30005 I00000 = 1, 5
      I00025 = MV0030(I00000)
      DO 30005 I00001 = 1,4
      I00026 = 10**(4-I00001)                                    16
      MV0009(I00000,I00001) = I00025/I00026
30005 I00025 = I00025 - (I00025/I00026)*I00026
C
C**********END SET STATUS OF ALL EVENTS**********
C**********START USERS INITIALIZATION**********
C
      SR0001=1.E-11
      IBK=0
      ICYC=0
      NPTS=61
      DO 300 I=1,100
      ATABLE(I)=0.
  300 TTABLE(I)=-100.
      READ (5     ,INPUT )                                       34
      WRITE(6,INPUT)                                             35
      IF(LPLOT)GO TO 30
C
C     SET STATUS(PTOT)OFF
C
      MV0009( 2,1) = 0
   30 CONTINUE
  230 WRITE(6     ,31300)                                        41
      AKI=AKI
```

```
      WRITE(6      ,31301)(TTABLE(I00000),BETAB(I00000),I00000=1,61)       42
      IF(LPLOT)CALL CAMRAV(935)                                             51
      ISECT=1
      DTSTEP=ABS(DTSTEP)
      DTPRNT=ABS(DTPRNT)
      REWIND 13                                                             54
      REWIND 9                                                              55
   25 CONTINUE
      WRITE(6      ,31302)                                                  56
      IDUM=0
C
C     INITIALIZE FOR TRAJECTORY PORTION
C
      GO TO(31100,31101,31101),ISECT
31100 CONTINUE
      TIME=TO
      PRINTT=TIME
      JJ=0
      KL=0
      PLOTT=0.
      N=100
      ICAS=ICAS+1
C
C     INITIALIZE FOR BACKWARD INTEGRATION OF TRAJECTORY
C
31101 CONTINUE
      GO TO(31103,31102,31103),ISECT
31102 CONTINUE
      DTSTEP=-DTSTEP
      DTPRNT=-DTPRNT
      TI=TIME
      WRITE(6      ,31303)                                                  65
      WRITE(6      ,31304)(TTABLE(I00000),BETAB(I00000),I00000=1,61)
C
C     INITIALIZE INTEGRATION OF LAMDA  MATRICES
C                                                                           66
```

```
31103 CONTINUE
      GO TO(31105,31105,31104),ISECT
31104 CONTINUE
      IDUM=0
      TIME=T1
      DO 4 I=1,7
      LAMPHI(I)=0.
      LAMOMG(I)=0.
      DO 4 K=1,5
 4    LAMPSI(I,K)=0.
      LAMPHI(7)=1.0
      LAMOMG(4)=1.
      LAMPSI(1,1)=1.
      LAMPSI(2,2)=1.
      LAMPSI(3,3)=1.
      LAMPSI(5,4)=1.
      LAMPSI(6,5)=1.
      DO 5 I=1,7
      DO 5 K=1,7
 5    FMAT(I,K)=0.
      DO 12 K=1,7
 12   GMAT(K)=0.
      DO 250 I=1,5
      I1(I)=0.
      DO 250 K=1,5
 250  I2(I,K)=0.
      I3=0.
C
C     READ FIRST RECORD  FROM SDV TAPE
C
      REWIND 9
      READ(9)XFT
      K=25
```

```
      DO 26 I=1,K
      KK=50-I+1
      DO 27 IA=1,17
 27   TEMP(IA)=XFT(KK,IA)
      DO 28 IA=1,17
 28   XFT(KK,IA)=XFT(I,IA)
      DO 29 IA=1,17
 29   XFT(I,IA)=TEMP(IA)
 26   CONTINUE
31105 CONTINUE
      I00021=0
      I00012=0
      DO 30008 I00000=1,I00005
30008 SD0004(I00000)=1.D30
      I00008=0
C
C**********EVENTS AND NUMERICAL INTEGRATION INITIALIZATION.**********
C
30010 CONTINUE
C             DT=DTSTEP
      SD0001=DTSTEP
      SD0002=DTSTEP
C             TURN INITIALIZATION FLAGS ON
      I00009=1
      I00006=0
      I00010=1
      DO 30015 I00000=1,I00003
30015 MV0010(I00000)=0
C
C**********TURN TEMPORARY STATUS FLAGS ON.**********
C
      DO 30020 I00000=1,I00002
30020 MV0007(I00000)=1
      IF(I00004.EQ.0)GO TO 30021
      DO 30022 I00000=1,I00002
      IF(ABS(TIME-SR0010(I00000)).LT.1.E-14)MV0007(I00000)=0
```

```
30022 CONTINUE
      IF(IDD012.EQ.0)GO TO 30021
      SD3002=IDINT((SD3303-SD0033)/DMIN1(SD0001,SD0040))+1
      SD3002=SD3303+SD3002*DMIN1(SD3301,SD3340)-SD0000
      IF(SD0002.EQ.0.)SD3302=SD3331
30021 CONTINUE
C           MAKE TIME DOUBLE PRECISION FOR THE EVS AND NUM. INTEG. SUBROUTINES

      SD3000=TIME
C
C***********END ALL INITIALIZATION**********
C
C
C***********DIFFERENTIAL EQUATIONS**********
C
30040 CONTINUE
      GO TO(31106,31106,31107),ISECT
31106 CONTINUE
      T=R+X(1)
      G=-MU/(T*T)
      RHO=RHO0*EXP(-BSTAR*X(1))                                         197
      T1=RHO*S*X(4)*X(4)
      BD=T1*CD/(2.*M)
      BL=T1*CL/(2.*M)
      L=.5*T1*CL
      D=.5*T1*CD
C
C     TABLE LOOKUP FOR BETA
C
      IDOM=0
      CALL LATLUM(IERR,IDOM,1,N,2,TIME,TTABLE,BETAB,BETA)               199
      BETA=BETA/57.2957795
C
C     DERIVIATIVES OF STATE VECTOR (EQ. 13)
C
      F(1)=X(4)* SIN(X(5))                                              201
      F(2)=X(4)* COS(X(5))* COS(X(6))/(T* COS(X(3)))                    202   203   204
      F(3)=X(4)* COS(X(5))* SIN(X(6))/T                                 205   206
      F(4)= G *  SIN(X(5)) - BD                                         207
      F(5)= G *  COS(X(5)) /X(4) + X(4) *  COS(X(5)) / T + BL *
     1  COS(BETA) / X(4)                                                208   209
      F(6)= - X(4) *  COS(X(5)) *  COS(X(6)) *  TAN(X(3)) /T - BL *
     1  SIN(BETA) /(X(4) *  COS(X(5)) )                                 210   211   212   213
      F(7)=SQRT( L*L + D*D ) / M                                        214
      F(8)=LAMD * SQRT(RHO) * X(4) * X(4) * X(4)                        215
31107 CONTINUE
      GO TO(31109,31109,31108),ISECT
31108 CONTINUE
 7    IF(TIME.GT.XFT(1,1))GO TO 6
```

```
      READ(9)XFT                                                        221
      K=25
      DO 46 I=1,K
      KK=50-I+1
      DO 47 IA=1,17
 47   TEMP(IA)=XFT(KK,IA)
      DO 48 IA=1,17
 48   XFT(KK,IA)=XFT(I,IA)
      DO 49 IA=1,17
 49   XFT(I,IA)=TEMP(IA)
 46   CONTINUE
      GO TO 7
 6    CONTINUE
C
C     TABLE LOOKUP FOR STATE VARIABLES AND THEIR DERIVIATIVES
C
      IDUM=0
      CALL LATLUM(IERR,IDUM,16,50,2,TIME,XFT(1,1),XFT(1,2),FMA(1))      253
      IDOM=0
      CALL LATLUM(IERR,IDOM,1,N,2,TIME,TTABLE,BETAB,BETA)               256
      BETA=BETA/57.2957795
      SDEL=SIN(FMA(3))                                                  258
      CDEL=COS(FMA(3))                                                  259
      SGAM=SIN(FMA(5))                                                  260
      CGAM=COS(FMA(5))                                                  261
      SA=SIN(FMA(6))                                                    262
      CA=COS(FMA(6))                                                    263
      FMAT(1,4)=-SGAM
      T=R+FMA(1)
      G=-MU/(T*T)
      RHO=RHOO*EXP(-BSTAR*FMA(1))                                       264
      T1=RHO*S*FMA(4)*FMA(4)
      BD=T1*CD/(2.*M)
      BL=T1*CL/(2.*M)
      L=.5*T1*CL
      D=.5*T1*CD
```

```
      T1=FMA(4)*CGAM
      T2= CDEL * T
C
C     CALCULATE F(T) (EQ. 26)
C
      FMAT(1,5)=-T1
      FMAT(2,1)=T1 * CA  /(T2 * T )
      FMAT(2,3)=-FMAT(2,1)* SDEL * T  / CDEL
      FMAT(2,4)=- CGAM * CA / T2
      FMAT(2,5) = FMA(4) * CA * SGAM / T2
      FMAT(2,6)= T1 * SA / T2
      FMAT(3,1)= FMAT(2,6) * CDEL / T
      FMAT(3,4)= - CGAM * SA / T
      FMAT(3,5)=   FMA(4) * SA * SGAM / T
      FMAT(3,6)= - T1 * CA / T
      FMAT(4,1)= 2.* G * SGAM / T - BSTAR * D / M
      FMAT(4,4)= RHO * S * CD * FMA(4) / M
      FMAT(4,5)= - G * CGAM
      FMAT(5,1)= 2. * G * CGAM / (FMA(4)*T) + T1/ (T*T) + BSTAR*BL*
     1 COS(BETA) / FMA(4)                                               265
      FMAT(5,4)=G*CGAM/(FMA(4)*FMA(4)) - CGAM/T - RHO*S*CL* COS(BETA)/
     1 (2.*M)                                                           266
      FMAT(5,5)= G*SGAM/FMA(4) + FMA(4)*SGAM/ T
      T2= CGAM * CA
      T3=T*CDEL
      FMAT(6,1)=-FMA(4)*T2*SDEL/T3
      FMAT(6,3)= FMA(4) * T2 /(CDEL*T3)
      FMAT(6,4)= T2*SDEL/ T3    + RHO*S*CL* SIN(BETA)/(2.*M*CGAM)       267
      FMAT(6,5)=-FMA(4)*CA*SDEL*SGAM / T3 - RHO*S*FMA(4)* SIN(BETA)*SGAM
     1 *(-CL)/(2.*M*CGAM*CGAM)                                          268
      FMAT(6,6)= - FMA(4)*CGAM*SDEL*SA / T3
      FMAT(7,1)= BSTAR*( SQRT(L*L + D*D ) / M  + LAMO*FMA(4)*FMA(4)*
```

```
  1 FMA(4)*SQRT(RHO) / 2. )                                    269    270
      FMAT(7,4)=-RHO*S*FMA(4)*(L*CL+D*CD) /(M*SQRT(L*L+D*D) )  271
      GMAT(5)= - BL *  SIN(BETA) / FMA(4)                      272
      GMAT(6)= - BL *  COS(BETA) / ( FMA(4) * CGAM )           273
      CALL MATMUL(DLMPSI,FMAT,LAMPSI,-7,7,5)                   274
      CALL MATMUL(DLMPHI,FMAT,LAMPHI,-7,7,1)                   276
      CALL MATMUL(DLMOMG,FMAT,LAMOMG,-7,7,1)                   278
      DPSI(1)=FMA(9)/FMA(12)
      DPSI(2)=FMA(10)/FMA(12)
      DPSI(3)=FMA(11)/FMA(12)
      DPSI(4)=FMA(13)/FMA(12)
      DPSI(5)=FMA(14)/FMA(12)
C
C     CALCULATE LAMDA PSI ONE (EQ. 40)
C     CALCULATE LAMDA PHI ONE (EQ. 40)
C
      DO 303 I=1,7
      DO 303 K=1,5
  303 TLPSI1(K,I)=DPSI(K)*LAMOMG(I)
      DO 13 I=1,5
      DO 13 K=1,7
      TTAB(I,K)=LAMPSI(K,I)
   13 TLPSI1(I,K)=TTAB(I,K)-TLPSI1(I,K)
      DO 14 K=1,7
   14 TLPHI1(1,K)=LAMPHI(K)-((FMA(15)+FMA(16))/FMA(12))*LAMOMG(K)
C
C     ATEMP= SUMMATION OF G * LAMDA PHI 1
C
      ATEMP(1)=0.
      DO 301 I=1,7
  301 ATEMP(1)=GMAT(I)*TLPHI1(1,I)+ATEMP(1)
      CALL MATMUL(BTEMP,TLPSI1,GMAT,5,7,1)
C
C     DI1 CALCULATION (EQ. 48)
C                                                              323
      DO 15 K=1,5
```

```
 15   DI1(K)= BTEMP(K) * W * ATEMP(1)
      CALL MATMUL(ATEMP,GMAT,TLPSI1,-1,-7,5)
C
C     DI2 CALCULATION (EQ. 48)
C                                                        333
      DO 16 K=1,5
 16   ATEMP(K) = ATEMP(K) * W
      CALL MATMUL(DI2,BTEMP,ATEMP,5,1,5)
C
C     DI3 CALCULATION (EQ. 48)
C                                                        343
      ATEMP(1)=0.
      DO 304 I=1,7
 304  ATEMP(1)=GMAT(I)*TLPHI1(1,I)+ATEMP(1)
      BTEMP(1)=0.
      DO 302 I=1,7
 302  BTEMP(1)=TLPHI1(1,I)*GMAT(I)+BTEMP(1)
      DI3=BTEMP(1)   * W * ATEMP(1)
31109 CONTINUE
C
C**********END DIFFERENTIAL EQUATIONS**********
C
C
C**********TEST FOR END OF STEP**********
C
      IF(I00010.EQ.1)GO TO 30250
C
C**********START NUMERICAL INTEGRATION**********
C
30050 CONTINUE
      GO TO(31110,31110,31111),ISECT
31110 CONTINUE
      DO 32000 I00003=1,    8
      SD0201(I00000+  8)=F      (I00009    )
32000 CONTINUE
```

```
      IF(100006.NE.0)GO TO 30200
      DO 32001 100000=1,     8
      SD0201(100000     )=X      (100000     )
32001 CONTINUE
      100006=     1
30200 CALL SYSKS(SD0201,MV0201,    16.    11.     6,100010)                388
30600 TIME=SD0000
      DO 32002 100000=1,     8
      X      (100000     )=SD0201(100000     )
32002 CONTINUE
      GO TO 30040
31111 CONTINUE
      GO TO(31113,31113,31112),ISECT
31112 CONTINUE
      DO 32003 100000=1,    35
      SD0203(100000+ 35)=PSILMD(100000     )
32003 CONTINUE
      DO 32004 100000=1,     7
      SD0203(100000+ 77)=DLMPHI(100000     )
      SD0203(100000+ 91)=DLMOMG(100000     )
32004 CONTINUE
      DO 32005 100000=1,     5
      SD0203(100000+103)=DI1    (100000     )
32005 CONTINUE
      DO 32006 100000=1,    25
      SD0203(100000+133)=DAI2   (100000     )
32006 CONTINUE
      SD0203(   160)=DI3
      IF(100006.NE.0)GO TO 30201
      DO 32007 100000=1,    35
      SD0203(100000     )=PSILAM(100000     )
32007 CONTINUE
      DO 32008 100000=1,     7
      SD0203(100000+ 70)=LAMPHI(100000     )
      SD0203(100000+ 84)=LAMOMG(100000     )
```

```
32008 CONTINUE
      DO 32009 I00000=1,    5
      SD0203(I00000+ 98)=I1     (I00000     )
32009 CONTINUE
      DO 32010 I00000=1,   25
      SD0203(I00000+108)=AI2    (I00000     )
32010 CONTINUE
      SD0203(  159)=I3
      I00006=    1
30201 CALL SYSKS(SD0203,MV0203,  160,   11,    6,I00010)                              474
30601 TIME=SD0000
      DO 32011 I00000=1,   35
      PSILAM(I00000     )=SD0203(I00000     )
32011 CONTINUE
      DO 32012 I00000=1,    7
      LAMPHI(I00000     )=SD0203(I00000+ 70)
      LAMOMG(I00000     )=SD0203(I00000+ 84)
32012 CONTINUE
      DO 32013 I00000=1,    5
      I1    (I00000     )=SD0203(I00000+ 98)
32013 CONTINUE
      DO 32014 I00000=1,   25
      AI2   (I00000     )=SD0203(I00000+108)
32014 CONTINUE
      I3           =SD0203(  159)
      GO TO 30040
31113 CONTINUE
30250 DTSTEP=SD0001
C
C**********EVENT VARIABLES.**********
C
C**********VARIABLES WHICH MUST BE UPDATED DURING EVENT ITERATIONS
C          ARE COMPUTED HERE.**********

30260 CONTINUE
      GO TO(31115,31114,31115),ISECT
```

```
31114 CONTINUE
      JJ=JJ+1
      XFT(JJ,1)=TIME
      DO 1 K=1,8
      XFT(JJ,K+1)=X(K)
 1    XFT(JJ,K+9)=F(K)
      IF(JJ.LT.50)GO TO 2
C
C     WRITE SDV TAPE
C
      WRITE(9)XFT                                                        530
      JJ=1
      XFT(1,1)=XFT(50,1)
      DO 3 K=1,8
      XFT(1,K+1)=XFT(50,K+1)
 3    XFT(1,K+9)=XFT(50,K+9)
 2    CONTINUE
31115 CONTINUE
      GO TO(31117,31117,31116),ISECT
31116 CONTINUE
      IF(IBK.EQ.0)GO TO 231
      DO 232 I=1,50
 232  ARRLAM(2,I)= ARRLAM(1,I)
 231  ARRLAM(1,1)=TIME
      DO 233 I=1,7
      ARRLAM(1,I+36)=TLPHI1(1,I)
 233  ARRLAM(1,I+43)=GMAT(I)
      IJKL=2
      DO 234 I=1,7
      DO 234 J=1,5
      ARRLAM(1,IJKL)=TLPSI1(J,I)
 234  IJKL=IJKL+1
      IF(IBK.EQ.0)GO TO 235
C
C     WRITE LAMDA TAPE
```

```
C
      WRITE(13) ARRLAM
  235 IBK=100
31117 CONTINUE
      DO 30270 I00000=1,100002
30270 MV0006(I00000)=0
      GO TO(32103,32103,32104),ISECT
32103 CONTINUE
      SD0030( 1) = PRINTT
      MV0006( 1) = 1
32104 CONTINUE
      GO TO(32105,32106,32106),ISECT
32105 CONTINUE
      SD0030( 2) = PLOTT
      MV0006( 2) = 1
32106 CONTINUE
      GO TO(32108,32107,32107),ISECT
32107 CONTINUE
      SD0030( 3) = TI
      MV0006( 3) = 1
32108 CONTINUE
      GO TO(32110,32109,32109),ISECT
32109 CONTINUE
      SD0030( 4) = 0.
      MV0006( 4) = 1
32110 CONTINUE
      GO TO(32111,32112,32112),ISECT
32111 CONTINUE
      SD0030( 5) = 498.17852
      MV0006( 5) = 1
32112 CONTINUE
C
C**********END EVENT VARIABLES**********
C
C
```

```
C**********TEST FOR EVENTS**********
C
C             DID THE PREVIOUS EVENT HAVE BOTH DISC AND POST OPTIONS
      IF(I00007.EQ.1)GO TO 30851
30300 CONTINUE
C             DID THE PREVIOUS EVENT HAVE AN EXIT LOGIC STATEMENT.
      IF(I00021.EQ.1)GO TO 30860
      CALL EVS(SR0008,I00003,I00004,I00006,I00002)
C
C             I00009 IS THE EVENT STATUS FLAG
C             -1=TIME IS EQUAL TO THE TIME OF AN EVENT
C              0=PROGRAM IS ITERATING TO FIND THE TIME OF AN EVENT
C             1=THEREWERE NO EVENTS DURING THE PREVIOUS DTSTEP.
C
      IF(I00009)30400,30050,30330
30330 IF(I00015.EQ.0)GO TO 30050
C             RESET TEMPORARY STATUS FLAGS
      DO 30340 I00000=1,I00002

30340 MV0007(I00000)=1
      I00015=0
      GO TO 30050
C
C
C**********EVENT TIME FOUND**********
C
30400 I00015=1
C             IS THE PRE OPTION ON
      IF(MV0009(I00004,3).EQ.1)GO TO 30852
C             NO PRE GO TO PRIMARY ACTION

30420 GO TO ( 30501,30502,30503,30504,30505),I00004
C
C
C**********EVENTS PRIMARY ACTION**********
```

617

```
C
C
C                                                       EVENT NO.  1
C       EVENT(PRINT)TIME=PRINTT,POST
C
30501 CONTINUE
      PRINTT=PRINTT+DTPRNT
      GO TO 30800

C
C                                                       EVENT NO.  2
C       EVENT(PTOT)TIME=PLOTT
C
30502 CONTINUE
      PLOTT=PLOTT+DTPLOT
C
C       STORE VARIABLES TO PLOT
C
      KL=KL+1
      PLOTAB(KL,1)=TIME
      DO 20 I=1,6
 20   PLOTAB(KL,I+1)=X(I)
      PLOTAB(KL,8)=F(7)
      PLOTAB(KL,9)=F(8)
      PLOTAB(KL,10)=BETA*57.2957795
      GO TO 30800

C
C                                                       EVENT NO.  3
C       EVENT(EVUP)TIME=TI,DISC,POST
C
30503 CONTINUE
      I00012=1
      PRINTT=AINT(TI)+DTPRNT
C
```

```
C       SET STATUS(EVUP)OFF
C
        MV0009( 3,1) = 0
        GO TO 30830

C
C                                                EVENT NO.  4
C       EVENT(STOP)TIME=0.,POST
C
30504   CONTINUE
        IF(ISECT.EQ.3)GO TO 878
        JJ=JJ+1
        DO 879 K=JJ,50
        DO 879 I=1,17
 879    XFT(K,I)=-K
C
C       FINISH SDV TAPE
C
        WRITE(9)XFT                                                                   672
        END FILE 9                                                                    674
 878    CONTINUE
C             TURN EXIT         FLAG ON
        I00021=1
C
C       SET STATUS(EVUP)ON
C
        MV0009( 3,1) = 1
        GO TO 30830

C
C                                                EVENT NO.  5
C       EVENT(EXIT)TIME=498.17852,POST
C
30505   CONTINUE
        I00021=1
```

```
      PSI(1)= X(1) -75504. / 5280.
      PSI(2)= X(2) - 24.1 / 57.2957795
      PSI(3)= X(3) + .6/ 57.2957795
      PSI(4)= X(5) + 44.3 / 57.2957795
      PSI(5)= X(6) + 29.4 /57.2957795
      IF(.NOT.LPLOT)GO TO 9
C
C     STORE END CONDITIONS OF PLOT VARIABLES
C
      KL=KL+1
      PLOTAB(KL,1)=TIME
      DO 21 I=1,6
 21   PLOTAB(KL,I+1)=X(I)
      PLOTAB(KL,8)=F(7)
      PLOTAB(KL,9)=F(8)
      PLOTAB(KL,10)=BETA*57.2957795
 9    CONTINUE
      GO TO 30800
C
C
C**********END PRIMARY ACTION ON EVENTS**********
C
C           CONT-DISC OPTION
30800 SD0001=DTSTEP
      IF(MV0009(I00004,2).EQ.1)GO TO 30810
      SR0010(I00004)=TIME
C           DISC-IF THE POST OPTION IS ON SET THE POST FLAG=1
      IF(MV0009(I00004,4).EQ.1)I00007=1
      GO TO 30010
C           CONTINUOUS,IS THE POST OPTION ON
30810 IF(MV0009(I00004,4).EQ.0)GO TO 30260
C           YES
C
C
C**********START SECONDARY ACTION**********
```

```
C
C          THE CONT AND POST OPTIONS ARE ON.
30850 I00011=1
      GO TO 30853
C          THE DISC AND POST OPTIONS ARE ON.
30851 I00011=-1
      GO TO 30853
C          THE PRE OPTION IS ON
30852 I00011=0
30853 CONTINUE
      GO TO(31118,31118,31119),ISECT
31118 CONTINUE
      WRITE(6      ,31305)TIME,DTSTEP,X(1),X(2),X(3),X(4),X(5),X(6),X(7),
     *X(8),F(1),F(2),F(3),F(4),F(5),F(6),F(7),F(8),BETA                    719
31119 CONTINUE
C          RETURN FROM SECONDARY ACTION
      IF(I00011)30854,30420,30262
30854 I00007=0
      GO TO 30260
C          START THE TERMINAL COMPUTATIONS
30860 CONTINUE
      GO TO(31120,31121,31121),ISECT
31120 CONTINUE
      Q=X(7)+X(8)
      IF(.NOT.LPLOT)GO TO 10
C
C     CALL PLOT ROUTINES
C
      DO 22 I=1,9
      BCCY(I)=BCDY(I)
      CALL QUIK3V(-1,35,BCDX,BCCY,   -KL,PLOTAB(1,1),PLOTAB(1,I+1))        735
 22   CONTINUE
      CALL CLEAN(KL)                                                      739
 10   CONTINUE
31121 CONTINUE
```

```
      IF(ISECT.EQ.3)GO TO 989
      ISECT=ISECT+1
      GO TO 25
  989 CONTINUE
C
C     FINISH LAMDA TAPE
C
      WRITE(6      ,31306)Q,PSI(1),PSI(2),PSI(3),PSI(4),PSI(5)              746
      WRITE(13) ARRLAM                                                     747
      END FILE 13                                                          749
      REWIND 13
C
C     COMPUTE STEEPEST ASCENT OPTIMIZATION PARAMETERS
C                                                                          750
      DO 200 I=1,5
  200 PSIW(I)= -AKO * PSI(I)
      K=0
      DO 201 J=1,5
      DO 201 I=1,5
      K=K+1
  201 AVI2(K)=I2(I,J)
      WRITE(6,220)(PSIW(I),AVI2(I),AVI2(I+5),AVI2(I+10),AVI2(I+15),
     1  AVI2(I+20),I=1,5)                                                  770
      CALL INVRT(AVI2,5,0,DET)                                             782
      IF(DET.NE.0.0)GO TO 202
      WRITE(6,203)                                                         786
  203 FORMAT(1H1,16H SINGULAR MATRIX / 15H RUN TERMINATED )
      CALL KIKOFF                                                          787
  202 CONTINUE
      K=0
      DO 368 J=1,5
      DO 368 I=1,5
      K=K+1
  368 AIV2(I,J)=AVI2(K)
      CALL MATMUL(ATEMP,AIV2,PSI,5,5,1)                                    801
      CALL MATMUL(BTEMP,PSI,ATEMP,-1,5,1)                                  803
```

```
      DP = AKO*AKO * ( DPO*DPO - AKI*AKI*BTEMP(1) )
      IF(DP.GE.0.0) GO TO 204
C
C     PARAMETER TOO LARGE , REDUCE IT
C
      AKI=AKI-.2*AKI
      DO 205 I=1,5
 205  PSI(I) = AKI * PSI(I)
      IF(AKI.GE.0.)GO TO 202
      WRITE(6,206)                                                    818
 206  FORMAT(1H1,18H KI LESS THAN ZERO/15H RUN TERMINATED  )
      CALL KIKOFF                                                     819
 204  CALL MATMUL(ATEMP,AIV2,I1,5,5,1)                                821
      CALL MATMUL(ID,I2,AIV2,5,5,5)                                   823
      WRITE(6,220) (PSIW(I),AIV2(1,I),AIV2(2,I),AIV2(3,I),AIV2(4,I),
     1 AIV2(5,I) ,I=1,5)                                              824
      WRITE(6,330)(ID(1,I),ID(2,I),ID(3,I),ID(4,I),ID(5,I),I=1,5)     836
 330  FORMAT(30X,15HIDENTITY MATRIX,/ (5E18.8) )
      CALL MATMUL(BTEMP,I1,ATEMP,-1,5,1)                              847
      WRITE(6     ,31307)(ATEMP(I00000),BTEMP(I00000),I1(I00000),I00000=
     *1,5)                                                            848
      TEMP(1)= I3 - BTEMP(1)
      WRITE(6     ,31308)DP,TEMP(1)                                   857
      DPC= SQRT( DP / ABS( TEMP(1) )  )
C     DPSIDP  CALCULATION                                             858
      DPSIDP= - SQRT( ABS( TEMP(1) )  )                               859
      DO 207 I=1,100
      JJ=101-I
      IF( TTABLE(JJ) .LE. 0.0 ) GO TO 207
 210  READ(13) ARRLAM                                                 868
      IF( TTABLE(JJ) .GT. ARRLAM(2,1) ) GO TO 207
      IF( TTABLE(JJ) .LT. ARRLAM(1,1) ) GO TO 210
      IDM=0
      CALL LATLUM(IERR,IDM,49,2,2,TTABLE(JJ),ARRLAM(1,1),ARRLAM(1,2),
     1 AL(1))                                                         879
```

```
      CALL MATMUL(TEMP,TLLAM,ATEMP,-7,5,1)                              882
      DO 213 IJKL=1,7
  213 TEMP(IJKL+7)=TLPHI(IJKL)-TEMP(IJKL)
      CALL MATMUL(BTEMP,TG,TEMP(8),-1,7,1)                              891
      ATABLE(JJ)=W * BTEMP(1) * DPC
      CALL MATMUL(ATEMP,AIV2,PSIW,5,5,1)                                894
      CALL MATMUL(TEMP,TG,TLLAM,-1,-7,5)                                896
      CALL MATMUL(BTEMP,TEMP,ATEMP,1,5,1)
C
C     THE CHANGE (A)  TO THE BETA TABLE CALCULATED
C                                                                       898
      ATABLF(JJ)= ATABLE(JJ) + W * BTEMP(1)
  207 CONTINUE
C
C     CONVERT A TO DEGREES
C
      DO 701 I=1,100
  701 ATABLE(I)=ATABLE(I)*57.2957795
C
C     ADD A TO BETA TABLE
C
      DO 215 I=1,100
  215 BETAB(I)=BETAB(I) + ATABLE(I)
      ISOL=0
      DO  216  I=1,5
      IF( ABS(PSI(I)) .GT. .001 ) GO TO 217
  216 CONTINUE
C
C     SOLUTION  FOUND
C
      ISOL=1
  217 WRITE(6,218)( I1(I),I2(1,I),I2(2,I),I2(3,I),I2(4,I),I2(5,I),
     1  I=1,5 )                                                         933
  218 FORMAT(1H0,8X, 2HI1,36X,2HI2,/ (E18.8,5X,5E18.8) )
```

```
      WRITE(6,219) AKO,I3,DPO, DPSIDP, DP, AK1                                945
219   FORMAT(1H0,4H KO ,E16.8, 4H I3 ,E16.8, 5H DPO ,E16.8, 9H DPSI/DP ,
     1E16.8,4H DP ,E16.8,4H K1 ,E16.8)
220   FORMAT(1H0,3X,4HPSIW,40X,10HI2 INVERSE,/ (E18.8,5X,5E18.8) )
      ICYC=ICYC+1
      WRITE(6,221) ICYC                                                       946
221   FORMAT(1H0,14H END OF CYCLE ,I5)
C
C     CALCULATE STANDARD DEVIATION OF A TABLE
C
      WRITE(6     ,31309)(ATABLE(I00000),I00000=1,61)                         947
      ADEL=0.
      DO 702 I=1,NPTS
702   ADEL=ADEL+ATABLE(I)
      FNPTS=NPTS
      ADEL=ADEL / FNPTS
      SD=0.
      DO 704 I=1,NPTS
      DEL(I)=(ATABLE(I)-ADEL)**2
704   SD=SD+DEL(I)
      FNPTS=NPTS-1
      SD=SQRT(SD/FNPTS)                                                       970
      WRITE(6,222)SD                                                          971
222   FORMAT(1H0,19H STANDARD DEVIATION,2X,E15.8)
      IF(ISOL .EQ. 0) GO TO 230
      IF(LPLOT)CALL CLEAN                                                     976
      STOP
      END
```

```
            APOLLO                                   STORAGE  MAP
                                                      MAIN PROGRAM

                                                   COMMON   VARIABLES

                 COMMON BLOCK      ISECT          ORIGIN      00001          LENGTH     00001
SYMBOL     LOCATION     TYPE            SYMBOL   LOCATION     TYPE          SYMBOL     LOCATION      TYPE
ISECT        00000        I
                 COMMON BLOCK      SR0001         ORIGIN      00032          LENGTH     00001
SR0001       00000        R
                 COMMON BLOCK      SD0000         ORIGIN      00003          LENGTH     00002
SD0000       00000        D
                 COMMON BLOCK      SD0001         ORIGIN      00005          LENGTH     00002
SD0001       00000        D
                 COMMON BLOCK      SD0002         ORIGIN      00007          LENGTH     00002
SD0002       00000        D
                 COMMON BLOCK      I00009         ORIGIN      00011          LENGTH     00001
I00009       00000        I
                 COMMON BLOCK      I00008         ORIGIN      00012          LENGTH     00001
I00008       00000        I
                 COMMON BLOCK      SD0040         ORIGIN      00013          LENGTH     00002
SD0040       00000        D
                 COMMON BLOCK      SD0003         ORIGIN      00015          LENGTH     00004
SD0003       00000        D        SDP003    00002        D
                 COMMON BLOCK      I00012         ORIGIN      00021          LENGTH     00001
I00012       00000        I
                 COMMON BLOCK      MV0009         ORIGIN      00022          LENGTH     00024
MV0009       00000        I
                 COMMON BLOCK      MV0006         ORIGIN      00046          LENGTH     00005
MV0006       00000        I
                 COMMON BLOCK      MV0007         ORIGIN      00053          LENGTH     00005
MV0007       00000        I
                 COMMON BLOCK      SD0030         ORIGIN      00060          LENGTH     00012
SD0030       00000        D
                 COMMON BLOCK      SR0010         ORIGIN      00072          LENGTH     00005
SR0010       00000        R
                 COMMON BLOCK      MV0010         ORIGIN      00077          LENGTH     00001
MV0010       00000        I
                 COMMON BLOCK      SD0004         ORIGIN      00100          LENGTH     00012
```

APOLLO STORAGE MAP

SD0004 00000 D

DIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
LAMPSI	00173	R	LAMPHI	00363	R	LAMOMG	00372
I1	00401	R	I2	00301	R	BCDX	00406
BCDY	00422	I	BCCY	00433	I	WORK	00450
AVI2	00506	D	ID	00570	R	X	00621
F	00631	R	PSI	00641	R	TTABLE	00646
BETAB	01012	R	TTAB	01156	R	XFT	01221
FMAT	02743	R	PSILAM	00173	R	PSILMD	00236
AI2	00301	R	DAI2	00332	R	PLOTAB	03024
FMA	06744	R	TEMP	06764	R	GMAT	07005
DLMPSI	00236	R	DLMPHI	07014	R	DLMOMG	07023
DPSI	07032	R	ILPSI1	07037	R	TLPHI1	07102
DI1	07111	R	DI2	00332	R	ATEMP	07116
BTEMP	07123	R	ATABLE	07130	R	DEL	07274
ARRLAM	07440	R	TLLAM	00112	R	TLPHI	00155
TG	00164	R	AL	00112	R	AIV2	07604
PSIW	07635	R	SD0201	07642	D	MV0201	10402
SD0203	10406	D	MVC203	17306	I	MV0030	17323

UNDIMENSIONED PROGRAM VARIABLES

SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION	TYPE	SYMBOL	LOCATION
M	17330	R	MU	17331	R	L	17332
LAMO	17333	R	I3	17334	R	DET	17336
LPLOT	17340	L	R	17341	R	CD	17342
S	17343	R	CL	17344	R	RHOO	17345
BSTAR	17346	R	IO	17347	R	DTSTEP	17350
DTPRNT	17351	R	DTPLOT	17352	R	AKD	17353
AKI	17354	R	DPC	17355	R	W	17356
I00002	17357	I	I00003	17360	I	I00004	17361
I00021	17362	I	I00015	17363	I	I00007	17364
I00005	17365	I	I00000	17366	I	I00025	17367
I00001	17370	I	I00026	17371	I	IBK	17372
ICYC	17373	I	NPTS	17374	I	I	17375
AKI	17376	R	IDUM	17377	I	TIME	17400
PRINTT	17401	R	JJ	17402	I	KL	17403
PLOTT	17404	R	N	17405	I	ICAS	17406
T1	17407	R	K	17410	I	KK	17411
I00006	17412	I	I00010	17413	I	T	17414
G	17415	R	RHO	17416	R	T1	17417
BD	17420	R	BL	17421	R	D	17422
IDUM	17423	I	IERR	17424	I	BETA	17425
SDEL	17426	R	CDEL	17427	R	SGAM	17430
CGAM	17431	R	SA	17432	R	CA	17433
T2	17434	R	T3	17435	R	DI3	17436
IJKL	17437	I	J	17440	I	SR0008	17441
I00011	17442	I	W	17443	R	DP	17444
DPC	17445	R	DPSILP	17446	R	IDM	17447
ISOL	17450	I	ADEL	17451	R	FNPTS	17452
SD	17453	R					

APOLLO

STORAGE MAP

ENTRY POINTS

......	SECTION	37

SUBROUTINES CALLED

.XPI.	SECTION	38	.FRDU.	SECTION	39	.FWRU.	SECTION	40
.FWRD.	SECTION	41	CAMRAV	SECTION	42	.FRWT.	SECTION	43
.FRDB.	SECTION	44	.FBLI.	SECTION	45	EXP	SECTION	46
LATLON	SECTION	47	SIN	SECTION	48	COS	SECTION	49
TAN	SECTION	50	SQRT	SECTION	51	MATMUL	SECTION	52
SYSKS	SECTION	53	.FWRB.	SECTION	54	.FBLO.	SECTION	55
EVS	SECTION	56	.FCFT.	SECTION	57	QUIK3V	SECTION	58
CLEAN	SECTION	59	INVRT	SECTION	60	KIKOFF	SECTION	61
.EXIT.	SECTION	62	.FXEM.	SECTION	63	.UNC5.	SECTION	64
.UN06.	SECTION	65	.FFIL.	SECTION	66	.FCNV.	SECTION	67
.UN13.	SECTION	68	.UN09.	SECTION	69	.FRLR.	SECTION	70
.FBLT.	SECTION	71	.FBDT.	SECTION	72	.FWLR.	SECTION	73
E.1	SECTION	74	E.2	SECTION	75	E.3	SECTION	76
E.4	SECTION	77	CC.1	SECTION	78	CC.2	SECTION	79
CC.3	SECTION	80	CC.4	SECTION	81	SYSLOC	SECTION	82

EFN IFN CORRESPONDENCE

EFN	IFN	LOCATION	EFN	IFN	LOCATION	EFN	IFN	LOCATION
31300	FORMAT	17550	31301	FORMAT	17551	31302	FORMAT	17561
31303	FORMAT	17562	31304	FORMAT	17563	31305	FORMAT	17573
31306	FORMAT	17662	31307	FORMAT	17734	31308	FORMAT	17716
31309	FORMAT	17725	30005	19A	20155	300	31A	20203
30	40A	20225	230	41A	20225	25	56A	20300
31100	58A	20317	31101	62A	20333	31103	74A	20374
31102	64A	20343	31105	147A	20604	31104	76A	20404
4	85A	20421	5	95A	20455	12	103A	20464
250	113A	20500	26	144A	20602	27	128A	20546
28	134A	20563	29	140A	20575	30008	153A	20613
30010	157A	20621	30015	164A	20640	30020	171A	20647
30021	193A	20772	30022	185A	20674	30040	194A	20775
31106	196A	21005	31107	216A	21324	31109	361A	22567
31108	218A	21334	7	219A	21334	6	251A	21433
46	247A	21430	47	230A	21374	48	236A	21411
49	243A	21423	303	285A	22400	13	299A	22433
14	307A	22444	301	316A	22457	15	326A	22477
16	338A	22524	304	348A	22544	302	356A	22554
30250	512A	23062	30050	365A	22573	31110	367A	22603
31111	399A	22654	32000	373A	22610	30200	387A	22631
32001	364A	22624	30600	389A	22642	32002	395A	22650
31113	511A	23062	31112	401A	22664	32003	407A	22671
32004	417A	22704	32005	425A	22714	32006	433A	22724
30201	473A	23036	32007	444A	22743	32008	454A	22756
32009	462A	22766	32010	470A	22776	30601	475A	23017
32011	481A	23025	32012	491A	23036	32013	499A	23045
32014	507A	23054	30260	513A	23064	31115	544A	23154
31114	515A	23074	1	523A	23113	2	543A	23154
3	538A	23147	31117	587A	23263	31116	546A	23164

APOLLO STORAGE MAP

231	558A	23177	232	553A	23172	233	564A	23205
234	577A	23234	235	586A	23261	30270	591A	23267
32103	596A	23302	32104	597A	23307	32105	599A	23317
32106	600A	23324	32108	603A	23341	32107	602A	23334
32110	606A	23355	32109	605A	23351	32111	608A	23365
32112	609A	23371	30851	714A	23725	30300	613A	23375
30860	724A	24022	30400	631A	23432	30330	620A	23414
30340	626A	23424	30852	716A	23730	30420	637A	23441
30501	638A	23453	30502	640A	23457	30503	655A	23513
30504	658A	23524	30505	678A	23603	30800	698A	23674
20	646A	23475	878	675A	23576	879	668A	23543
9	696A	23673	21	688A	23656	30810	708A	23715
30850	712A	23722	30853	717A	23731	31118	719A	23741
31119	720A	24015	30854	722A	24020	31120	726A	24032
31121	741A	24074	10	740A	24074	22	736A	24065
989	746A	24104	200	754A	24151	201	765A	24174
220	FORMAT	20011	202	788A	24265	203	FORMAT	17733
368	795A	24302	204	820A	24415	205	811A	24372
206	FORMAT	17743	330	FORMAT	17753	207	902A	25007
210	868A	24624	213	886A	24712	701	908A	25012
215	916A	25020	216	930A	25041	217	933A	25045
218	FORMAT	17761	219	FORMAT	17771	221	FORMAT	20022
702	957A	25151	704	966A	25204	222	FORMAT	20027

THE FIRST LOCATION NOT USED BY THIS PROGRAM IS 25307.

```
$IEDIT          SYSLB3,SCHF01
$IBLDR IBM017
$IBLDR IBM019
$IBLDR IBM025
$IEDIT          SYSLB3,SCHF02
$IBLDR LATLUM
$IBLDR MATMUL
$IEDIT
$IBFTC KIKOFF

BEGIN COMPILING  154512

      KIKOFF     -  EFN   SOURCE STATEMENT  -  IFN(S)  -

SUBROUTINE KIKOFF
CALL DUMP                                                      2
RETURN
END

$IBFTC INVRT
```

```
            INVRT       -  EFN   SOURCE STATEMENT  -  IFN(S)  -

      SUBROUTINE INVRT(A,N,M,DETER)
C     MATRIX INVERSION AND SIMULTANEOUS EQUATIONS SOLVER
C     A=INPUT MATRIX FOR INVERISON OR AUGUMENTED MATRIX FOR SIME. EQS.  IG500030
C     N=ORDER OF COEFFICIENT MATRIX                                     IG500040
C     M=0 FOR INVERISON ONLY
C     M=NUMBER OF CONSTANT VECTORS
C     DETER=DETERMINANT OF COEFFICIENT MATRIX
      DOUBLE PRECISION A(1),DETER,SIGN,AMAX
      DIMENSION IPIV(20), INDEX(20,2)
      DETER=1.0D0
      SIGN=1.0D0
      DO 20 J=1,N
   20 IPIV(J)=0
      NN=N+M
      DO 182 K=1,N
      AMAX=0.0D0
   40 DO 76 I=1,N
      IF (IPIV(I)-1)50,76,50
   50 DO 75 J=1,N
      IF (IPIV(J)-1)55,75,250
   55 IND=(J-1)*N+I
      IF(AMAX-DABS(A(IND))) 60,75,75
   60 IR=I
      IC=J
      AMAX=DABS(A(IND))
   75 CONTINUE
   76 CONTINUE
      IPIV(IC)=IPIV(IC)+1
      IF (IR-IC)90,115,90
   90 SIGN=-SIGN
      DO 110 L=1,NN
      IND=(L-1)*N+IR
      IND2=(L-1)*N+IC
```

```
      AMAX=A(IND)
      A(IND)=A(IND2)
  110 A(IND2)=AMAX
  115 INDEX(K,1)=IR
      INDEX(K,2)=IC
      IND=(IC-1)*N+IC
      AMAX=A(IND)
      DETER=DETER*AMAX
      IF (DETER)140,255,140
  140 A(IND)=1.0D0
      DO 150 L=1,NN
      IND=(L-1)*N+IC
  150 A(IND)=A(IND)/AMAX
      DO 181 L=1,N
      IF (L-IC)165,181,165
  165 IND=(IC-1)*N+L
      AMAX=A(IND)
      A(IND)=0.0D0
      DO 180 I=1,NN
      IND=(I-1)*N+L
      IND2=(I-1)*N+IC
      A(IND)=A(IND)-A(IND2)*AMAX
  180 CONTINUE
  181 CONTINUE
  182 CONTINUE
      DO 235 I=1,N
      L=N+1-I
      IR=INDEX(L,1)
      IC=INDEX(L,2)
      IF (IR-IC)210,235,210
  210 DO 230 K=1,N
      IND=(IR-1)*N-K
      IND2=(IC-1)*N-K
      AMAX=A(IND)
      A(IND)=A(IND2)
  230 A(IND2)=AMAX
  235 CONTINUE
      DETER=SIGN*DETER
      RETURN
  250 M=-1
  255 RETURN
       END
```

Typical Output Results for Case I Application Problem, Run Number 8 (Cycle 1)

```
$INPUT
R      =      0.39600000E 04,
CD     =      0.13000000E 01,
LAMO   =      0.17809700E-05,
M      =      0.20400000E 03,
S      =      0.46379973E-05,
CL     =      0.35000000E 00,
RHOO   =      0.39743447E 09,
BSTAR  =      0.22176000E 00,
MU     =      0.95629856E 05,
TO     =      0.00000000E-38,
DTSTEP =      0.10000000E 01,
X      =      0.75757576E 02,   0.00000000E-38,   0.00000000E-38,   0.68181818E 01,  -0.11344640E 00,
              0.00000000E-38,   0.00000000E-38,   0.00000000E-38,
TTABLE =      0.00000000E-38,   0.19000000E 02,   0.37000000E 02,   0.44000000E 02,   0.46000000E 02,
              0.50000000E 02,   0.51700000E 02,   0.53800000E 02,   0.55600000E 02,   0.57500000E 02,
              0.61200000E 02,   0.69600000E 02,   0.77700000E 02,   0.85100000E 02,   0.98100000E 02,
              0.14810000E 03,   0.19510000E 03,   0.20190000E 03,   0.22040000E 03,   0.23890000E 03,
              0.25180000E 03,   0.25740000E 03,   0.26860000E 03,   0.27600000E 03,   0.28160000E 03,
              0.28530000E 03,   0.29720000E 03,   0.29910000E 03,   0.29280000E 03,   0.29630000E 03,
              0.29810000E 03,   0.30190000E 03,   0.30560000E 03,   0.30930000E 03,   0.31670000E 03,
              0.32590000E 03,   0.33700000E 03,   0.35180000E 03,   0.36290000E 03,   0.37400000E 03,
              0.38140000E 03,   0.38510000E 03,   0.38880000E 03,   0.39750000E 03,   0.39440000E 03,
              0.39630000E 03,   0.39320000E 03,   0.40000000E 03,   0.40190000E 03,   0.40930000E 03,
              0.41300000E 03,   0.41670000E 03,   0.42230000E 03,   0.42790000E 03,   0.42980000E 03,
              0.43150000E 03,   0.43340000E 03,   0.43530000E 03,   0.43700000E 03,   0.43890000E 03,
              0.10000000E 04,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
             -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
             -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
             -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
             -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
             -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
             -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
             -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,  -0.10000000E 03,
BETAR  =      0.16865000E 03,   0.14865000E 03,   0.16400000E 03,   0.15800000E 03,   0.15400000E 03,
              0.14467000E 03,   0.13130000E 03,   0.78670000E 02,   0.52000000E 02,   0.26000000E 02,
              0.18000000E 02,   0.10690000E 02,   0.66500000E 01,   0.53200000E 01,   0.46700000E 01,
              0.46700000E 01,   0.66500000E 01,   0.79800000E 01,   0.99800000E 01,   0.13300000E 02,
              0.17300000E 02,   0.19330000E 02,   0.26000000E 02,   0.34650000E 02,   0.46000000E 02,
              0.62660000E 02,   0.78670000E 02,   0.92000000E 02,   0.10865000E 03,   0.12467000E 03,
              0.13800000E 03,   0.14937000E 03,   0.15131000E 03,   0.15531000E 03,   0.16067000E 03,
              0.16467000E 03,   0.16865000E 03,   0.16990000E 03,   0.16865000E 03,   0.16467000E 03,
              0.15800000E 03,   0.14467000E 03,   0.13131000E 03,   0.10467000E 03,   0.91350000E 02,
              0.65320000E 02,   0.46000000E 02,   0.29980000E 02,   0.20670000E 02,   0.14000000E 02,
              0.12670000E 02,   0.11340000E 02,   0.12670000E 02,   0.20000000E 02,   0.39330000E 02,
              0.45320000E 02,   0.13131000E 03,   0.15131000E 03,   0.16200000E 03,   0.16865000E 03,
              0.16865000E 03,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
             -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,  -0.00000000E-19,
DTPRNT =      0.20000000E 03,
DTPLOT =      0.50000000E 01,
LPLOT  =      T,
```

```
AKO     =     0.50000000E-01,
AKI     =     0.10000000E-01,
DPO     =     0.50000000E-02,
W       =     0.10000000E 01,

$ END
```

TTABLE	BETAB
0.00000000E-38	0.16865000E 03
0.19000000E 02	0.16865000E 03
0.37000000E 02	0.16400000E 03
0.44000000E 02	0.15800000E 03
0.46000000E 02	0.15400000E 03
0.50000000E 02	0.14467000E 03
0.51900000E 02	0.13130000E 03
0.53800000E 02	0.78670000E 02
0.55600000E 02	0.52000000E 02
0.57500000E 02	0.26000000E 02
0.61200000E 02	0.18000000E 02
0.68600000E 02	0.10690000E 02
0.77700000E 02	0.66500000E 01
0.85100000E 02	0.53200000E 01
0.98100000E 02	0.46700000E 01
0.14810000E 03	0.46700000E 01
0.18510000E 03	0.66500000E 01
0.20190000E 03	0.79800000E 01
0.22040000E 03	0.99800000E 01
0.23890000E 03	0.13300000E 02
0.25180000E 03	0.17300000E 02
0.25740000E 03	0.19330000E 02
0.26860000E 03	0.26000000E 02
0.27600000E 03	0.34650000E 02
0.28160000E 03	0.46000000E 02
0.28530000E 03	0.62660000E 02
0.28720000E 03	0.78670000E 02
0.28910000E 03	0.92000000E 02
0.29280000E 03	0.10865000E 03
0.29630000E 03	0.12467000E 03
0.29810000E 03	0.13800000E 03
0.30190000E 03	0.14437000E 03
0.30560000E 03	0.15131000E 03
0.30930000E 03	0.15531000E 03
0.31670000E 03	0.16047000E 03
0.32590000E 03	0.16467000E 03
0.33700000E 03	0.16865000E 03
0.35180000E 03	0.16998000E 03
0.36290000E 03	0.16865000E 03
0.37400000E 03	0.16467000E 03
0.38140000E 03	0.15800000E 03
0.38510000E 03	0.14467000E 03
0.38880000E 03	0.13131000E 03
0.39250000E 03	0.10467000E 03
0.39440000E 03	0.91350000E 02
0.39630000E 03	0.65320000E 02
0.39820000E 03	0.46000000E 02
0.40000000E 03	0.29980000E 02
0.40190000E 03	0.20670000E 02
0.40930000E 03	0.14000000E 02
0.41300000E 03	0.12670000E 02
0.41670000E 03	0.11340000E 02
0.42230000E 03	0.12670000E 02
0.42790000E 03	0.20000000E 02
0.42980000E 03	0.39330000E 02
0.43150000E 03	0.85320000E 02
0.43340000E 03	0.13131000E 03
0.43530000E 03	0.15131000E 03
0.43700000E 03	0.16200000E 03
0.43890000E 03	0.16865000E 03
0.10000000E 04	0.16865000E 03

TIME	0.00000000E-38	DTSTEP	0.10000000E 01				
X(1)	0.75757576E 02	X(2)	0.00000000E-38	X(3)	0.00000000E-38	X(4)	0.68181818E 01
X(5)	-0.11344640E 00	X(6)	0.00000000E-38	X(7)	0.00000000E-38	X(8)	0.00000000E-38
F(1)	-0.77184007E 00	F(2)	0.15785829E-02	F(3)	0.00000000E-38	F(4)	0.65085787E-03
F(5)	0.82244189E-03	F(6)	-0.10798297E-06	F(7)	0.14297659E-04	F(8)	0.25305892E-02
BETA	0.29434977E 01						

TIME	0.20000000E 03	DTSTEP	0.10000000E 01				
X(1)	0.56618257E 02	X(2)	0.25093708E 00	X(3)	-0.36435340E-02	X(4)	0.38217102E 01
X(5)	0.24596127E-01	X(6)	-0.26733249E-01	X(7)	0.31071792E 01	X(8)	0.74803582E 01
F(1)	0.93989793E-01	F(2)	0.95085323E-03	F(3)	-0.25425285E-04	F(4)	-0.44814938E-03
F(5)	-0.57825146E-03	F(6)	0.56176674E-06	F(7)	0.31313725E-03	F(8)	0.37207879E-02
BETA	0.13665200E 00						

TIME	0.40000000E 03	DTSTEP	0.10000000E 01				
X(1)	0.26353662E 02	X(2)	0.43439025E 00	X(3)	-0.91795361E-02	X(4)	0.22526721E 01
X(5)	-0.15353361E 00	X(6)	-0.12288045E 00	X(7)	0.47866613E 01	X(8)	0.93760489E 01
F(1)	-0.34450368E 00	F(2)	0.55424107E-03	F(3)	-0.68449850E-04	F(4)	-0.85412660E-01
F(5)	0.68560556E-02	F(6)	-0.52122590E-02	F(7)	0.89407175E-01	F(8)	0.21844101E-01
BETA	0.52324971E 00						

TIME	0.49817852E 03	DTSTEP	0.10000000E 01				
X(1)	0.12438351E 02	X(2)	0.44981932E 00	X(3)	-0.12250474E-01	X(4)	0.16210001E 00
X(5)	-0.14050027E 01	X(6)	-0.47874320E 00	X(7)	0.71907792E 01	X(8)	0.96095773E 01
F(1)	-0.15987724E 00	F(2)	0.59777755E-05	F(3)	-0.31023190E-05	F(4)	-0.38070528E-02
F(5)	-0.22095537E-01	F(6)	-0.19379119E-01	F(7)	0.10132443E-01	F(8)	0.38078093E-04
BETA	0.29434977E 01						

TTABLE	BETAP
0.00000000E-38	0.16865000E 03
0.19000000E 02	0.16865000E 03
0.37000000E 02	0.16400000E 03
0.44000000E 02	0.15800000E 03
0.46000000E 02	0.15400000E 03
0.50000000E 02	0.14467000E 03
0.51900000E 02	0.13130000E 03
0.53800000E 02	0.78670000E 02
0.55600000E 02	0.52000000E 02
0.57500000E 02	0.26000000E 02
0.61200000E 02	0.18000000E 02
0.68600000E 02	0.10690000E 02
0.77700000E 02	0.66500000E 01
0.85100000E 02	0.53200000E 01
0.98100000E 02	0.46700000E 01
0.14810000E 03	0.46700000E 01
0.18510000E 03	0.66500000E 01
0.20190000E 03	0.79800000E 01
0.22040000E 03	0.99800000E 01
0.23890000E 03	0.13300000E 02
0.25180000E 03	0.17300000E 02
0.25740000E 03	0.19330000E 02
0.26860000E 03	0.26000000E 02
0.27600000E 03	0.34650000E 02
0.28160000E 03	0.46000000E 02
0.28530000E 03	0.62660000E 02
0.28720000E 03	0.78670000E 02
0.28910000E 03	0.92000000E 02
0.29280000E 03	0.10865000E 03
0.29630000E 03	0.12467000E 03
0.29810000E 03	0.13800000E 03
0.30190000E 03	0.14437000E 03
0.30560000E 03	0.15131000E 03
0.30930000E 03	0.15531000E 03
0.31670000E 03	0.16067000E 03
0.32590000E 03	0.16467000E 03
0.33700000E 03	0.16865000E 03
0.35180000E 03	0.16998000E 03
0.36290000E 03	0.16865000E 03
0.37400000E 03	0.16467000E 03
0.38140000E 03	0.15800000E 03
0.38510000E 03	0.14467000E 03
0.38880000E 03	0.13131000E 03
0.39250000E 03	0.10467000E 03
0.39440000E 03	0.91350000E 02
0.39630000E 03	0.65320000E 02
0.39820000E 03	0.46000000E 02
0.40000000E 03	0.29980000E 02
0.40190000E 03	0.20670000E 02
0.40930000E 03	0.14000000E 02
0.41300000E 03	0.12670000E 02
0.41670000E 03	0.11340000E 02
0.42230000E 03	0.12670000E 02
0.42790000E 03	0.20000000E 02
0.42980000E 03	0.39330000E 02
0.43150000E 03	0.85320000E 02
0.43340000E 03	0.13131000E 03
0.43530000E 03	0.15131000E 03
0.43700000E 03	0.16200000E 03
0.43890000E 03	0.16865000E 03
0.10000000E 04	0.16865000E 03

TIME	0.49817852E 03	DTSTEP	-0.10000000E 01				
X(1)	0.12438351E 02	X(2)	0.44981932E 00	X(3)	-0.12250474E-01	X(4)	0.16210001E 00
X(5)	-0.14050027E 01	X(6)	-0.47874320E 00	X(7)	0.71907792E 01	X(8)	0.96095773E 01
F(1)	-0.15987724E 00	F(2)	0.59777755E-05	F(3)	-0.31023190E-05	F(4)	-0.38070528E-02
F(5)	-0.22095537E-01	F(6)	-0.19373119E-01	F(7)	0.10132443E-01	F(8)	0.38078093E-04
BETA	0.29434977E 01						

TIME	0.29800000E 03	DTSTEP	-0.10000000E 01				
X(1)	0.55018610E 02	X(2)	0.34380004E 00	X(3)	-0.61264472E-02	X(4)	0.38001524E 01
X(5)	-0.33729868E-01	X(6)	-0.27024682E-01	X(7)	0.31320986E 01	X(8)	0.78048232E 01
F(1)	-0.12815434E 00	F(2)	0.94561835E-03	F(3)	-0.25560778E-04	F(4)	-0.22621585E-03
F(5)	-0.63639813E-03	F(6)	-0.14714685E-04	F(7)	0.44145037E-03	F(8)	0.43681244E-02
BETA	0.23956293E 01						

TIME	0.98000000E 02	DTSTEP	-0.10000000E 01				
X(1)	0.37397589E 02	X(2)	0.15123415E 00	X(3)	-0.10191123E-02	X(4)	0.43367283E 01
X(5)	0.47590312E-01	X(6)	-0.24303393E-01	X(7)	0.26039146E 01	X(8)	0.61222635E 01
F(1)	0.20630834E 00	F(2)	0.10833401E-02	F(3)	-0.26334011E-04	F(4)	-0.27920634E-01
F(5)	0.14152031E-02	F(6)	-0.13888870E-03	F(7)	0.28620008E-01	F(8)	0.45804837E-01
BETA	0.81594140E-01						

TIME	-0.00000000E-38	DTSTEP	-0.10000000E 01				
X(1)	0.75757825E 02	X(2)	-0.47835859E-07	X(3)	0.37095236E-07	X(4)	0.68181810E 01
X(5)	-0.11344680E 00	X(6)	-0.22600954E-06	X(7)	0.59596339E-06	X(8)	0.70378204E-05
F(1)	-0.77184325E 00	F(2)	0.16785825E-02	F(3)	-0.37937565E-09	F(4)	0.65086139E-03
F(5)	0.82244154E-03	F(6)	-0.10803929E-06	F(7)	0.14296868E-04	F(8)	0.25305186E-02
BETA	0.29434977E 01						

```
C          0.14800354E 02
PSI(1) -0.18616487E 01     PSI(2)  0.29194977E-01     PSI(3) -0.17784986E-02     PSI(4) -0.63182189E 00     PSI(5)  0.34383595E-01

PSIW                                              I2
 0.93082434E-01            -0.19672479E 04    0.50019438E 01    0.10962726E-01   -0.34923039E 00   -0.15093973E 00
-0.14597488E-02             0.50019438E 01   -0.16064550E-01    0.30610459E-03   -0.49780922E-01   -0.14277439E-02
 0.88924931E-04             0.10962726E-01    0.30610459E-03   -0.55642540E-04    0.50392982E-02    0.63045279E-04
 0.31591094E-01            -0.34923039E 00   -0.49780922E-01    0.50392982E-02   -0.77230516E 00   -0.26969404E-01
-0.17191797E-02            -0.15093973E 00   -0.14277439E-02    0.63045279E-04   -0.26969404E-01   -0.50340316E-01

PSIW                                              I2 INVERSE
 0.93082434E-01            -0.30503369E 00   -0.11918542E 03   -0.19122107E 02    0.76902941E 01    0.15096752E 00
-0.14597488E-02            -0.11918542E 03   -0.46647065E 05   -0.74043064E 04    0.30103011E 04    0.58343740E 02
 0.83924931E-04            -0.19122107E 02   -0.74043064E 04   -0.45695504E 05    0.18384943E 03    0.11161121E 03
 0.31591094E-01             0.76902941E 01    0.30103011E 04    0.18384943E 03   -0.19752622E 03   -0.23828764E 01
-0.17191797E-02             0.15096752E 00    0.58343740E 02    0.11161121E 03   -0.23828764E 01   -0.20555802E 02
                                     IDENTITY MATRIX
 0.99999989E 00    -0.16092332E-08     0.19780569E-09    -0.28609415E-07    -0.99147875E-09
 0.30258382E-03     0.99999946E 00    -0.25587232E-07     0.37711204E-05     0.15409542E-06
 0.33188109E-04     0.25076495E-07     0.99999998E 00     0.16640727E-05     0.56499837E-07
-0.11621001E-03     0.34866411E-06    -0.46341425E-08     0.10000008E 01     0.19715197E-07
 0.96331852E-07    -0.18865882E-09    -0.34401379E-11     0.88197161E-09     0.99999999E 00

    ATEMP               BTEMP             I1
-0.54479777E-01    -0.38191532E 01     0.56119533E 02
-0.10232786E 02     0.34410979E-04    -0.56969150E-01
-0.25393389E 02     0.11559114E-05    -0.82846596E-02
-0.11848249E 01    -0.66646532E-04     0.13143647E 01
 0.42681481E-01    -0.18029329E-05     0.51037776E-01

DP        0.49713496E-04     TEMP(1 -0.19546629E-01

     I1                                           I2
 0.56119533E 02            -0.19672479E 04    0.50019438E 01    0.10962726E-01   -0.34923039E 00   -0.15093973E 00
-0.56969150E-01             0.50019439E 01   -0.16064550E-01    0.30610459E-03   -0.49780922E-01   -0.14277439E-02
-0.82846596E-02             0.10962726E-01    0.30610459E-03   -0.55642540E-04    0.50392982E-02    0.63045279E-04
 0.13143647E 01            -0.34923039E 00   -0.49780922E-01    0.50392982E-02   -0.77230516E 00   -0.26969404E-01
 0.51037776E-01            -0.15093973E 00   -0.14277439E-02    0.63045279E-04   -0.26969404E-01   -0.50340316E-01

KO   0.50000000E-01 I3  -0.38386999E 01 DPO   0.50000000E-02 DPSI/DP  -0.13980926E 00 DP   0.49713496E-04 K1   0.10000000E-01

END OF CYCLE      1
```

ATABLE
0.00000000E-38
-0.37859737E 00
-0.11297009E 01
-0.12573565E 01
-0.12642827E 01
-0.83496822E 00
-0.35713923E 00
0.10268370E 01
0.18158104E 01
0.15891993E 01
0.15653574E 01
0.14710601E 01
0.73687708E 00
0.14376408E 00
-0.58606559E 00
-0.10354708E 01
-0.92674252E 00
-0.93224668E 00
-0.10559184E 01
-0.14326078E 01
-0.19899937E 01
-0.22975588E 01
-0.31691423E 01
-0.39929586E 01
-0.48167652E 01
-0.55881380E 01
-0.59020315E 01
-0.57880715E 01
-0.50101223E 01
-0.39054490E 01
-0.30616606E 01
-0.23216937E 01
-0.16810111E 01
-0.12599427E 01
-0.72448914E 00
-0.36277309E 00
-0.12347461E 00
0.27527372E-01
0.14267810E 00
0.37095640E 00
0.79208457E 00
0.14455883E 01
0.24483271E 01
0.42146569E 01
0.51263537E 01
0.54859768E 01
0.49668197E 01
0.37740816E 01
0.28723058E 01
0.20131207E 01
0.16483863E 01
0.12977961E 01
0.11442774E 01
0.13920739E 01
0.23562284E 01
0.29239127E 01
0.19506116E 01
0.10952833E 01
0.64305334E 00
0.41724765E 00
0.00000000E-38

STANDARD DEVIATION 0.25913357E 01

REFERENCES

1. Kelley, H. J.: Gradient Theory of Optimal Flight Paths. ARS Journal, vol. 30, 1960, pp. 947-954.

2. Kelley, H. J.: Method of Gradients. G. Leitman, ed., Optimization Techniques, Chapter 6. Academic Press, New York, 1962.

3. Bryson, A. E., and Denham, W. F.: A Steepest Ascent Method for Solving Optimum Programming Problems. Journal of Applied Mechanics, vol. 29, 1962, pp. 247-257.

4. Fox, C.: An Introduction to the Calculus of Variations. Oxford University Press, London, 1954.

5. Pontryagin, L. S., Boltyanskii, V. G., Gamkvelidze, R. V., and Mishchenko, E. F.: The Mathematical Theory of Optimal Processes, Interscience Division of John Wiley and Sons, Inc., New York, 1962.

6. Bellman, R. E., and Dreyfus, S. E.: Applied Dynamic Programming, Princeton University Press, Princeton, New Jersey, 1962.

7. Colunga, D.: A Modified Sweep Method for Control Optimization. Ph.D. Dissertation, The University of Texas, Austin, NASA TM-X-58047, 1970.

8. Hildebrand, F. B.: Methods of Applied Mathematics. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1952, p. 132.

9. Citron, S. J.: Elements of Optimal Control. Holt, Rinehart, and Winston, Inc., New York, 1969, pp. 62, 64, 77, 188, and 196.

APPROVAL

APPLICATION OF THE STEEPEST ASCENT OPTIMIZATION METHOD TO A REENTRY TRAJECTORY PROBLEM

By Bobby G. Junkin

The information in this report has been reviewed for security classification. Review of any information concerning Department of Defense or Atomic Energy Commission programs has been made by the MSFC Security Classification Officer. This report, in its entirety, has been determined to be unclassified.

This document has also been reviewed and approved for technical accuracy.

H. Hoelzer

H. HOELZER
Director, Computation Laboratory

☆ U. S. GOVERNMENT PRINTING OFFICE: 1972—745381/4042

DISTRIBUTION

INTERNAL

DIR
Dr. Rees

DEP-T
Dr. W. R. Lucas
Mr. E. W. Neubert

AD-S
Dr. E. Stuhlinger

S&E-DIR
Mr. H. Weidner

S&E-COMP-DIR
Dr. H. Hoelzer
Mr. C. Prince

S&E-COMP-R
Mr. R. J. Cochran

S&E-COMP-RD
Mr. D. G. Aichele

S&E-COMP-RDA
Mr. B. G. Junkin (30)

S&E-COMP-RR
Mr. Paul O. Hurst

S&E-COMP-RRV
Mr. J. A. Jones

S&E-COMP-RRP
Mr. P. R. Harness

S&E-COMP-DS
Mr. J. C. Lynn

S&E-AERO-DIR
Dr. E. Geissler

S&E-CSE-DIR
Dr. W. Haeussermann

S&E-R-DIR
Mr. W. G. Johnson

S&E-ASTR-DIR
Mr. F. B. Moore

S&E-ASTN-DIR
Mr. K. L. Heimburg

S&E-SSL-DIR
Mr. G. B. Heller

S&E-SSL-TR
Mr. M. C. Davidson

S&E-PT-DIR
Dr. M. Siebel

S&E-QUAL-DIR
Mr. D. Grau

PD-DIR
Mr. J. T. Murphy
Mr. J. A. Downey
Dr. W. A. Mrazek

PD-SA-DIR
Mr. W. G. Huber

A&TS-TU (6)

PD-DO-DIR
Mr. E. Goerner
Mr. Bill Goldsby
Mr. Tom French

PD-PS-DIR
Mr. J. C. Goodrum

PD-MP-DIR
Mr. H. P. Gierow

PM-DIR
Mr. J. T. Shepherd

PM-SL-MGR
Mr. L. F. Belew

PM-MO-MGR
Dr. F. A. Speer

PM-SAT-MGR
Mr. R. G. Smith

PM-MO-O
Mr. F. Kurtz
Mr. M. Horst

PM-PR-M

A&TS-MS-IL (8)

A&TS-MS-IP (2)

A&TS-MS-H

A&TS-PAT
Mr. L. D. Wofford, Jr.

EXTERNAL

Scientific and Technical Information
Facility (25)
P. O. Box 33
College Park, Maryland 20740
Attn: NASA Representative, S-AK/RKT

Computer Sciences Corporation
Huntsville, Alabama 35802
Attn: Mr. Don Stout (2)

Manned Spacecraft Center
National Aeronautics and Space Administration
Houston, Texas 77058
Attn: Dr. J. M. Lewallen
Chief, Theory and Analysis Office
Mr. Dennis A. Johnston, Code ED 13

Dr. E. B. Walters
Head, Mathematics Department
Delta State College
Cleveland, Mississippi