General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

N69-25242
(ACCESSION NUMBER)
271
(PAGES)
CR 98415
(NASA CR OR TMX OR AD NUMBER)
(THRU)
1
(CODE)
30
(CATEGORY)
FACILITY FORM 602

SD 69-4

QUASILINEARIZATION PROGRAM FOR DETERMINING OPTIMUM FINITE-THRUST TRANSFERS BETWEEN INCLINED ORBITS

3 January 1969

by

G. A. McCue
D. F. Bender
J. M. Morford

MAR 1969
RECEIVED
NASA STI FACILITY
INPUT BRANCH

Approved by

C. R. Faulders, Manager
Flight Technology
Science & Technology

SPACE DIVISION
NORTH AMERICAN ROCKWELL CORPORATION

SD 69-4

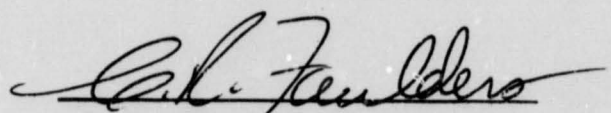
QUASILINEARIZATION PROGRAM FOR DETERMINING OPTIMUM FINITE-THRUST TRANSFERS BETWEEN INCLINED ORBITS

3 January 1969

by

G. A. McCue
D. F. Bender
J. M. Morford

Approved by



C. R. Faulders, Manager
Flight Technology
Science & Technology

SPACE DIVISION
NORTH AMERICAN ROCKWELL CORPORATION

FOREWORD

The work presented here was performed for NASA's Marshall Space Flight Center under contract NAS8-21077. The authors' are pleased to acknowledge the contributions of several individuals. Dr. Stephen A. Jurovics provided considerable assistance during the early portion of the numerical study and was responsible for the initial variational formulation. Dr. John R. Radbill provided advice and consultation throughout the study. He also supplied several computer subroutines for performing quasilinearization. Dr. John E. McIntyre contributed numerous suggestions concerning both the analytical and numerical portions of the study.

ABSTRACT

Quasilinearization was utilized to solve the discontinuous two-point boundary value problem which results from a variational formulation of the problem of minimum fuel orbital transfer. The boundary conditions were such that the transfer trajectory's end points could be assumed to be at unspecified positions upon arbitrary inclined closed orbits. The vehicle was assumed to be thrust limited and capable of controlling thrust direction and duration ("bang-bang" throttle control). Through the use of the quasi-linearization technique it was possible to determine trajectories that minimized the fuel required for orbital transfer maneuvers that were accomplished in a fixed time interval. It was found that an accurate initial iterate containing the time histories of all variables was essential to obtaining convergence. The initial iterate was derived from the corresponding optimal two-impulse orbital transfer. In many borderline cases it was found useful for indicating whether or not convergence of the process was to be expected. In cases where failure of the process was predicted it was found that, while the two-impulse transfer was indeed locally optimal, there existed neighboring three-impulse transfers that required less total impulse. An IBM 360 double-precision computer program incorporating the above techniques then was utilized to generate optimal transfers between numerous pairs of arbitrary inclined orbits. Using the resulting data, it was possible

to make a series of significant comparisons concerning the velocity changes required for corresponding optimal finite-thrust and optimal impulsive orbital transfers. Further numerical investigations demonstrated the existence of optimal transfers between "shallowly intersecting" coplanar orbits which required only one thrusting period. These maneuvers were shown to be analogous to the better known optimal one-impulse maneuver.

CONTENTS

Section		Page
I	INTRODUCTION	1
II	THE OPTIMIZATION PROBLEM	2
	The Differential Equations	2
	Boundary and Corner Conditions	8
	Multipliers Along a Keplerian Arc	12
III	QUASILINEARIZATION	20
	General Description	20
	The Jacobian	25
	Computation of the Initial Iterate	34
IV	NUMERICAL SOLUTION BY QUASILINEARIZATION	49
	Computation Techniques	49
	Initial Conditions	50
	Variable Length Storage Table	52
	Shifting Storage Table	53
	Switch Point Analysis	54
V	NUMERICAL STUDIES	57
	Introduction	57
	Convergence and Validity Tests	76

Section		Page
	Minimum Fuel With Final Time Open	77
	ΔV Requirements (Impulsive Thrust Versus Finite Thrust)	77
VI	CONCLUSION	83
VII	REFERENCES	84
	APPENDIXES	
	I. COMPUTER PROGRAM DESCRIPTION . . .	87
	II. EXAMPLE OF THE USE OF THE FINITE-THRUST ORBITAL TRANSFER OPTIMIZATION PROGRAM	123
	III. COMPUTER PROGRAM LISTING	175

ILLUSTRATIONS

Figure		Page
1	Coordinate Systems	4
2	Symbolic Presentation of the Time Scale	35
3	Procedure for Computation of Initial Time Histories of Multipliers, Primer, Steering Angles, Hamiltonian and Switching Function	37
4	Variable Length Storage Table (4a) and Table With Switching Points, t_s, Shifted for Next Iteration (4b) . .	55
5	The Orbits	58
6	Radius - Converged Time History (Solid Line) Compared to Initial Iterate (+)	59
7	Derivative of Radius - Converged Time History (Solid Line) Compared to Initial Iterate (+)	60
8.	Longitude - Converged Time History (Solid Line) Compared to Initial Iterate (+)	61
9	Derivative of Longitude - Converged Time History (Solid Line) Compared to Initial Iterate (+)	62
10	Latitude Angle - Converged Time History (Solid Line) Compared to Initial Iterate (+)	63

Figure		Page
11	Derivative of Latitude - Converged Time History (Solid Line) Compared to Initial Iterate (+)	64
12	Mass - Converged Time History (Solid Line) Compared to Initial Iterate (+)	65
13	Lagrange Multiplier λ_1 - Converged Time History (Solid Line) Compared to Initial Iterate (+)	66
14	Lagrange Multiplier λ_2 - Converged Time History (Solid Line) Compared to Initial Iterate (+)	67
15	Lagrange Multiplier λ_4 - Converged Time History (Solid Line) Compared to Initial Iterate (+)	68
16	Lagrange Multiplier λ_5 - Converged Time History (Solid Line) Compared to Initial Iterate (+)	69
17	Lagrange Multiplier λ_6 - Converged Time History (Solid Line) Compared to Initial Iterate (+)	70
18	Switching Function - Converged Time History (Solid Line) Compared to Initial Iterate (+)	71
19	Primer Vector Magnitude (Prime Components are λ_4, λ_5/r, $\lambda_6/r\cos\theta$) - Converged Time History (Solid Line) Compared to Initial Iterate (+)	72
20	Steering Angle ν - Converged Time History (Solid Line) Compared to Initial Iterate (+)	73

Figure		Page
21	Steering Angle ψ - Converged Time History (Solid Line) Compared to Initial Iterate (+)	74
22	Difference in ΔV Required for Finite - Thrust and Impulsive Transfers Versus Initial Thrust-to-Weight Ratio	78
23	ΔV Difference as a Function of the Relative Perigee Orientation of Two Elliptical Orbits	80
24	Velocity Change Required for Finite - Thrust and Impulsive Transfers Between "Almost Tangent" Orbits	81
25	Modular Tree Structure of Two-Impulse Orbital Transfer Optimization Program	91
26	Data Input Format and Data Description	107
27-55	Computer Output of a Sample Optimization Problem	123
56	Main Program Listing	176
57	Subroutine STPCØN Listing	182
58	Subroutine DATSET Listing	187
59	Subroutine XFIRST Listing	191
60	Subroutine HMASS Listing	194
61	Subroutine CØAST Listing	198
62	Subroutine LGRANG Listing	201
63	Subroutine LCØAST Listing	208
64	Subroutine BURN Listing	213

Figure		Page
65	Subroutine QASLN Listing	215
66	Subroutine JACØB Listing	224
67	Subroutine STØP Listing	230
68	Subroutine ØUTPUT Listing	233
69	Subroutine KHIST Listing	238
70	Subroutine QPLØT Listing	243
71	Subroutine PICTUR Listing	244
72	Function DTRUEH Listing	247
73	Subroutine RØTATD Listing	251
74	Subroutine ROTMXD Listing	252
75	Subroutine DØRBIT Listing	254
76	Subroutine XIN Listing	255
77	Subroutine ALGBC Listing	255

NOMENCLATURE

A	First integral
$A_2, B_2, A_4, A_6, C_1, C_2$	Integration constants for multipliers
$\underset{\sim}{B}$	Boundary condition vector
c	Effective exhaust velocity
C_j	Combination coefficients required for quasilinearization process
D	Auxiliary variable defined by Equation 32
e	Eccentricity
E	Eccentric anomaly
f	True anomaly
F	Thrust
G	Quantity to be minimized (maximized)
$\underset{\sim}{H}_j$	Homogeneous solution vectors required for quasilinearization process
$\underset{\sim}{g}$	Time derivative vector
k	Switching function
m	Mass
N	Number of differential equations
n	Quasilinearization iteration number
$\underset{\sim}{P}$	Particular solution vector required for quasilinearization process

p	Semilatus rectum
$\underline{Q}$	Primer vector
r	Radius
R	Multiplier ratio at second impulse
t_F	Total time required for transfer maneuver
s	Arbitrary positive constant
S	Multiplier ratio at second impulse
t	Time
$\underline{U}$	Unit vector (subscript denotes axis)
$\underset{\sim}{X}$	Dependent variable vector
x_i	Components of vector $\underset{\sim}{X}$
x	Angular velocity ($\dot{\theta}$)
y	Angular velocity ($\dot{\phi}$)
β	Mass flow rate ($-\dot{m}$)
ϕ	Longitude
θ	Latitude
λ_i	Lagrange multiplier
ρ	Radial velocity ($\dot{r}$)
μ	Gravitational constant
ν	Steering angle in orbit plane
ψ	Steering angle out of orbit plane
ω	Argument of perigee

Subscripts

o	refers to initial time or to coordinate system having orbit in the reference plane
F	refers to final time
1, 2	refers to first, second impulse
C, R, B	refers to circumferential, radial, binormal

Superscripts

o	refers quantity to a coordinate system having orbit in the reference plane.
T	transpose (of a tensor)

I. INTRODUCTION

Certain realistic treatments of the optimum orbital transfer problem lead to a variational formulation wherein the differential equations have no exact closed form solution. Prior experience with the particular two point boundary value problem considered here[1] indicated that it was extremely sensitive and could not be solved by an ordinary Newton-Raphson method. Reformulation and application of "quasilinearization"[2, 3, 4] allowed the successful computation of optimal "finite-thrust" transfer trajectories between arbitrary pairs of coplanar elliptical orbits. However, successful application of quasilinearization was found to depend upon the proper use of initial conditions derived from an optimum two-impulse transfer maneuver.[5, 6, 7, 8] These initial successes were reported in References 9 and 10.

A logical continuation of the above work was the development of an IBM 360 FORTRAN H computer program for computing optimal finite-thrust transfers between inclined orbits which is reported in this document. Because of its length and complexity, the complete program documentation comprises three documents. In addition to the present report, interested readers may find it desirable to consult References 11 and 12 which describe the two-impulse transfer and quasilinearization programs.

II. THE OPTIMIZATION PROBLEM

THE DIFFERENTIAL EQUATIONS

The problem to be considered here involves transferring between a pair of arbitrary closed orbits defined by their orbital elements (p, ϵ, ω, Ω, and i). What is required is to determine that trajectory which results in an orbital transfer with minimum fuel expenditure. The formulation is essentially the three-dimensional derivation originated by Jurovics[1] and is similar to that presented by Leitmann[13]. Solution of this optimization problem involves the minimization of a functional which is a function of only the boundary values of the state variables; i.e., position, velocity, mass, and time.

The function to be minimized is the characteristic velocity:

$$G = c \ln (m_o/m_{t_F}) \tag{1}$$

where

$$G = G\,(x_{io},\ x_{it_F}) \tag{2}$$

and where the state variables are:

$$x_i = r,\ \theta,\ \phi,\ \dot{r},\ \dot{\theta},\ \dot{\phi},\ m \tag{3}$$

In the above expressions c = effective exhaust velocity, r = radius, θ = latitude, ϕ = longitude, m = mass, and the subscripts o and t_F refer to initial and final points of the trajectory.

The rocket and its environment are defined in accordance with the following assumptions:

1. The rocket is a variable mass particle.
2. The thrust magnitude (F) is a linear function of the mass flow rate (β):

$$F = c\beta \tag{4}$$

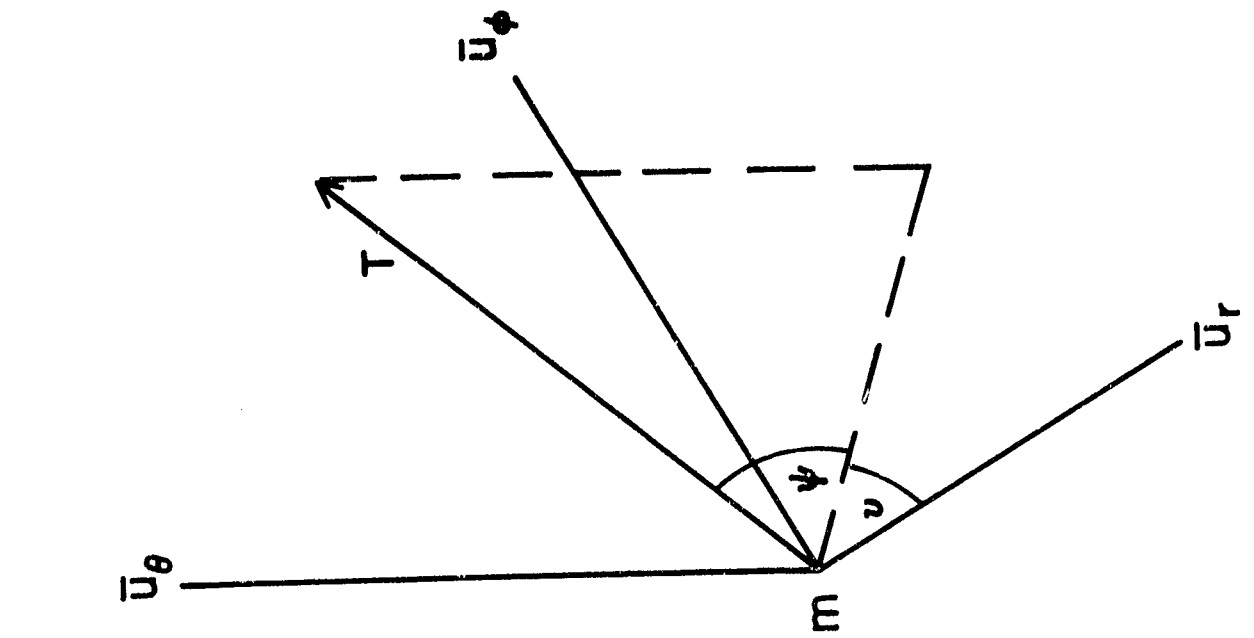
3. The vehicle is capable of thrust direction and throttle control, and the control is instantaneous.
4. Further, the transfer maneuver is between two orbits about a single planet with a spherically symmetrical central gravitational field.

Equations in Motion

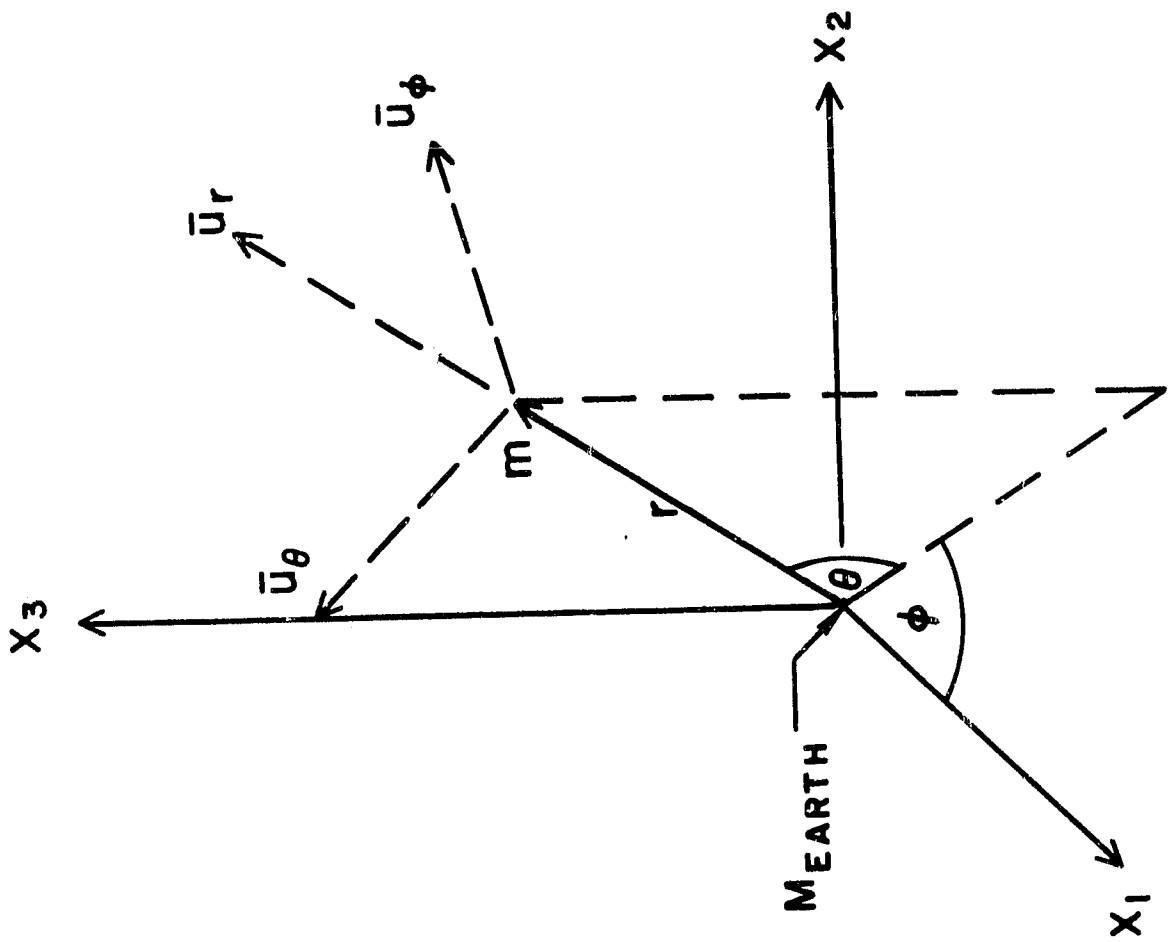
In spherical coordinates (Figure 1) the three second-order equations of motion are:

$$\ddot{r} - r\dot{\theta}^2 - r\cos^2\theta\,\dot{\phi}^2 = F\cos\psi\sin\nu/m \tag{5}$$

$$\frac{d}{dt}\left(r^2\dot{\theta}\right) + r^2\cos\theta\sin\theta\,\dot{\phi}^2 = Fr\sin\psi/m \tag{6}$$



1b



1a

Figure 1. Coordinate Systems

$$\frac{d}{dt}\left(r^2 \cos^2 \theta \, \dot{\phi}\right) = Fr \cos \psi \sin \nu \cos \theta / m \tag{7}$$

where μ = gravitational constant

ν = steering angle measured from local vertical in the orbit plane (Figure 1b)

ψ = out-of-plane steering angle (Figure 1b)

These equations may be reduced to first-order form, where the new variables ρ, x, and y are defined by the first three equations.

$$\dot{r} = \rho \tag{8}$$

$$\dot{\theta} = x \tag{9}$$

$$\dot{\phi} = y \tag{10}$$

$$\dot{\rho} = r\,x^2 + r\,y^2 \cos^2 \theta - \mu/r^2 + c\beta \cos \psi \sin \nu / m \tag{11}$$

$$\dot{x} = -2\rho x/r - y^2 \cos \theta \sin \theta + c\beta \sin \psi/(mr) \tag{12}$$

$$\dot{y} = 2xy \tan \theta - 2\rho y/r + c\beta \cos \psi \sin \nu \sec \theta/(mr) \tag{13}$$

$$\dot{m} = -\beta \tag{14}$$

The optimum path must satisfy the above equations of motion. In addition, for conventional rockets, the solution is subject to the following constraints:

$$c = \text{constant} \tag{15}$$

$$\beta_{min} \leq \beta \leq \beta_{max} \tag{16}$$

where normally $\beta_{min} = 0$, and β_{max} is specified. In the problem considered here we utilize the following additional constraint to impose "bang-bang" control of mass flow rate:

$$\beta(\beta - \beta_{max}) = 0 \tag{17}$$

If G is to possess an extremum subject to the constraints imposed by Eqs. 7 to 15 and 17, one must require the first variation of the constrained functional to vanish. The following Euler-Lagrange differential equations for the Lagrange multipliers result:

$$\dot{\lambda}_1 = -\lambda_4 (x^2 + y^2 \cos^2\theta + 2\mu/r^3) - \lambda_5 (2\rho x - c\beta \sin\psi/m)/r^2$$
$$-\lambda_6 (2\rho y - c\beta \cos\psi \sin\nu \sec\theta/m)/r^2 \tag{18}$$

$$\dot{\lambda}_2 = \lambda_4 r y^2 \sin 2\theta + \lambda_5 y^2 \cos 2\theta$$
$$- \lambda_6 \left[2xy \sec^2\theta + c\beta \cos\psi \sin\nu \tan\theta \sec\theta/(mr)\right] \tag{19}$$

$$\dot{\lambda}_3 = 0 \tag{20}$$

$$\dot{\lambda}_4 = -\lambda_1 + 2 (\lambda_5 x + \lambda_6 y)/r \tag{21}$$

$$\dot{\lambda}_5 = - \lambda_2 - 2 \lambda_4 rx + 2 \lambda_5 \rho/r - 2 \lambda_6 y \tan\theta \tag{22}$$

$$\dot{\lambda}_6 = - \lambda_3 - 2 \lambda_4 \, ry \cos^2 \theta + \lambda_5 \, y \sin 2\theta$$
$$- 2 \lambda_6 (x \tan \theta - \rho/r) \tag{23}$$

$$\dot{\lambda}_7 = \frac{c\beta (\lambda_4 \, r \cos \psi \cos \nu + \lambda_5 \sin \psi + \lambda_6 \cos \psi \sin \nu \sec \theta)}{(m^2 r)} \tag{24}$$

The differential equation for λ_7 may also be written in the following useful form:

$$\dot{\lambda}_7 = \frac{\beta}{m}\left[k + \lambda_7\right] \tag{25}$$

The "switching function" (k), which appears in Equation 25, governs thrust on-off control and is defined as follows:

$$k = - \lambda_7 + c (\lambda_4 \, r \cos \psi \cos \nu + \lambda_5 \sin \psi + \lambda_6 \cos \psi \sin \nu \sec \theta)/(mr) \tag{26}$$

where

$$\text{for} \qquad k > 0 \qquad \beta = \beta_{max} \tag{27}$$

$$\text{and for} \qquad k < 0 \qquad \beta = 0 \tag{28}$$

Equations for the steering angles (ν and ψ) are as follows:

$$\cos \psi \cos \nu = r \, \lambda_4 \cos \theta / D \tag{29}$$

$$\cos \psi \cos \nu = \lambda_6 / D \tag{30}$$

$$\sin \psi = \lambda_5 \cos \theta / D \tag{31}$$

where

$$D = + \left(r^2 \lambda_4^2 \cos^2 \theta + \lambda_5^2 \cos^2 \theta + \lambda_6^2 \right)^{1/2} \tag{32}$$

Even though the steering angles have no physical significance when the vehicle is on a coasting arc they will be computed over the entire trajectory. Further, we note that Equation 26 becomes

$$k = cD \sec \theta/(mr) - \lambda_7 \tag{33}$$

when Eq. 29 to 31 are substituted.

Since the differential equations do not involve time explicitly, one obtains a first integral (Hamiltonian):

$$\lambda_1 \dot{r} + \lambda_2 \dot{\theta} + \lambda_3 \dot{\phi} + \lambda_4 \dot{\rho} + \lambda_5 \dot{x} + \lambda_6 \dot{y} + \lambda_7 \dot{m} = A = \text{constant} \tag{34}$$

The Hamiltonian could be used to replace one of the Euler-Lagrange differential equations and thereby reduce the order of the system by one. But this is not done. Rather, the behavior of A is used as an indication of the accuracy of the integration process.

BOUNDARY AND CORNER CONDITIONS

Note that the system is described by 14 equations for the variables:

$$r, \theta, \phi, \rho, x, y, m, \lambda_1, \lambda_2, \lambda_3, \lambda_4, \lambda_5, \lambda_6, \lambda_7$$

This system thus requires 14 boundary conditions. Eleven are specified by the physics of the problem; that is, by the required orbital elements and the initial mass. The three remaining boundary conditions are derived from the transversality condition

$$\left[s dG + \lambda_1 \, dr + \lambda_2 \, d\theta + \lambda_3 \, d\phi + \lambda_4 \, d\rho + \lambda_5 \, dx + \lambda_6 \, dy + \lambda_7 \, dm - A dt \right]_0^{t_F} = 0 \tag{35}$$

where s is an arbitrary positive constant (we are minimizing sG instead of G). We consider first the "dt" term separately equal to zero, and since the initial time is fixed ($dt_o = 0$) there remains $A \, dt_F = 0$. Thus, if the time open problem is considered ($dt_F \neq 0$) we get $A = 0$. Or if the final time is fixed ($dt_F = 0$) and A is not known.

We chose the time open situation even though in the program the final time is fixed for any given iteration. Thus we use

$$A = 0 \tag{36}$$

Next we consider the dG and dm terms together and we find as the first of the three boundary conditions that

$$- sc \, dm_{t_F}/m + \lambda_7 \, dm_{t_F} = 0 \tag{37}$$

or

$$\lambda_7 = \frac{sc}{m} \qquad \text{at } t = t_F \tag{38}$$

where s is an arbitrary positive factor (i. e., we minimize sG instead of G). Now in view of Equation 33, the λ_7 boundary conditions is equivalent to

$$k = c(D \sec \theta/r - s)/m \qquad \text{at } t = t_F \tag{39}$$

By requiring that $k < 0$ at the end point, the λ_7 condition (Equation 38) which requires only that λ_7 be positive is clearly satisfied. (Note that s, c, D, m, r, and sec θ are always positive.)

The remaining parts of the transversality condition are now obviously satisfied by the condition $A = 0$ when the position and velocities used correspond to the orbits given.

From the point of view of the program there are 12 boundary conditions taken from the position and velocities at the end points, one condition taken as the value of the initial mass, and one is the requirement that $k < 0$ at the end. The 12 position and velocity conditions are equivalent to the 10 from the orbital elements plus the second and third from the transversality condition.

A somewhat different form of Eq. 35 may be derived for the case when $A \neq 0$:

$$\dot{r}\lambda_1 + \dot{\theta}\lambda_2 + \dot{\phi}\lambda_3 + \dot{\rho}\lambda_4 + \dot{x}\lambda_5 + \dot{y}\lambda_6 \Big]_0^F = \beta k \Big]_0^F \tag{40}$$

From this form of the equation, it is clear that the Hamiltonian (A in Eq. 25) must be equal to βk at the end points. Further, (if $A \neq 0$) Equation 40 implies that $k(0) = k(t_F)$.

We have obtained a system of 14 first-order differential equations which yield as a solution the required optimal trajectory. Seven of the boundary conditions (initial position, velocity, and mass) are known at the initial time and seven (final position, velocity, and λ_7 or k) at the final time.

In the computer solution it was found to be advantageous to employ k rather than λ_7. Thus Eq. 24 is replaced with

$$\dot{k} = cD \sec \theta \, (\dot{D}/D - \rho/r + x \tan \theta)/(mr) \qquad (41)$$

This substitution is significant because it replaces λ_7 with a more easily visualized function and one which is very well approximated from the impulsive case.

For this particular problem, the corner conditions are such that the multipliers associated with each of the state variables and A, the first integral, must have the same value immediately preceding and following a corner.

$$(\lambda_i)_- = (\lambda_i)_+ \qquad (42)$$

$$A_- = A_+ \qquad (43)$$

Because this formulation is in spherical coordinates two of the multipliers, λ_1 and λ_2, behave as do the velocity components. (See Equations 18 and 19.) Thus there is a discontinuity in the rates when the engine is turned off or on which is equivalent to a finite discontinuity in the impulsive approximation.

MULTIPLIERS ALONG A KEPLERIAN ARC

As it was shown by Lawden[14] it is possible to integrate in closed form the Euler-Lagrange equations for the first six multipliers along a coast or Keplerian arc if they are expressed in a system of coordinates in which the orbit lies in the reference plane. The results given agree with those of Lawden but there is a different nomenclature and in addition here the coordinate system is spherical where Lawden uses a locally level cartesian system. Thus, we reduce Equations 18 to 23 by setting $\beta = 0$ and $\theta = 0$, and we find:

$$\dot{\lambda}_1^{\,o} = - \lambda_4^{\,o} \left(y_o^{\,2} + 2 \mu / r^3 \right) - 2 \lambda_6^{\,o} \rho \, y_o \tag{44}$$

$$\dot{\lambda}_2^{\,o} = \lambda_5^{\,o} \, y_o^{\,2} \tag{45}$$

$$\dot{\lambda}_3^{\,o} = 0 \tag{46}$$

$$\dot{\lambda}_4^{\,o} = - \lambda_1^{\,o} + 2 \lambda_6^{\,o} \, y_o / r \tag{47}$$

$$\dot{\lambda}_5^{\,o} = - \lambda_2^{\,o} + 2 \lambda_5 \, \rho / r \tag{48}$$

$$\dot{\lambda}_6^{\,o} = - \lambda_3^{\,o} - 2 \lambda_4^{\,o} r y_o + 2 \lambda_6^{\,o} \rho/r \tag{49}$$

The superscript "o" and subscript "o" are used to indicate that the orbit lies in the reference plane. To begin the solution we change the independent variable to true anomaly, f, by using

$$y_o = df/dt = h/r^2 \tag{50}$$

The equations become (where the "′" indicates d/df and when the superscript "o" and subscript "o" have been temporarily omitted):

$$y \lambda_1' = - \lambda_4 (y^2 + 2\mu/r^3) - 2 \rho y \lambda_6 \tag{51}$$

$$\lambda_2' = y \lambda_5 \tag{52}$$

$$\lambda_3' = 0 \tag{53}$$

$$y \lambda_4' = - \lambda_1 + 2 \lambda_6 y/r \tag{54}$$

$$y \lambda_5' = - \lambda_2 + 2 \lambda_5 \rho/r \tag{55}$$

$$y \lambda_6' = - \lambda_3 - 2 ry \lambda_4 + 2 \rho \lambda_6/r \tag{56}$$

We note first that λ_3 is a constant along the orbit. We choose to write:

$$\lambda_3 = e h C_1/p \tag{57}$$

and C_1 becomes the first of the six constants of integration.

We note next that the out-of-plane multipliers λ_2 and λ_5 are separated from the others and that they can be obtained by writing:

$$q = \lambda_5 \, p^2/r^2 = \lambda_5 \, (1 + e \cos f)^2 \tag{58}$$

The two equations become;

$$\lambda_2' = h \, q/p^2 \tag{59}$$

$$\lambda_5' = p^2 \, \lambda_2/h \tag{60}$$

This pair of equations is easily solved by assuming:

$$\lambda_2 = A_2 \sin f + B_2 \cos f \tag{61}$$

where A_2 and B_2 are constants of integration. Then we find:

$$q = p^2 \, (A_2 \cos f - B_2 \sin f)/h \tag{62}$$

and

$$\lambda_5 = r^2 \, (A_2 \cos f - B_2 \sin f)/h \tag{63}$$

We know that one integral of the set of equations is the Hamiltonian, A, which is

$$A = \lambda_1 \dot{r} + \lambda_3 \dot{\phi} + \lambda_4 \dot{\rho} + \lambda_6 \dot{y} \tag{64}$$

Using this and the well known orbit equations we can eliminate λ_1 from Equation 54 and in doing so the coefficient of the λ_6 happens to vanish so that the result is

$$\lambda_4' - \lambda_4 \cot f = \lambda_3 p/(eh \sin f) - Ap^3/\left[eh^2 \sin f\,(1 + e \cos f)^2\right] \quad (65)$$

We note that this equation has singularities at θ and π and is not meaningful for circular orbits. For the moment we ignore these difficulties hoping they will go away (they do go away as a matter of fact!). The homogeneous portion is seen to have the solution

$$\lambda_4 = K_1 \sin f \quad (66)$$

We use this substitution in the variation of constants method for integrating Equation 65 and we find that the differential equations for K_1 is:

$$K_1' = C_1/\sin^2 f - C_2/\left[\sin^2 f\,(1 + e \cos f)^2\right] \quad (67)$$

where C_1 and C_2 are constants of integration. C_2 is defined by

$$C_2 = A p^3/(eh^2) \quad (68)$$

Note that in this analysis C_2 has replaced A as one of the six constants of integration.

Thus we find

$$K_1 = A_4 - C_1 \cot f - C_2 I \quad (69)$$

where A_4 is another (the fifth so far) constant of integration and where

$$I = \int \frac{df}{\sin^2 f\,(1 + e \cos f)^2} \quad (70)$$

The evaluation of I is easily managed by changing to eccentric anomaly.

The result is:

$$I = \frac{1}{(1 - e^2)^{5/2}} \int \frac{(1 - e \cos E)^3 \, dE}{\sin^2 E} \tag{71}$$

$$= \frac{1}{(1 - e^2)^{5/2}} \frac{1}{\sin E} \left[- \cos^2 E + 3e - 3e^2 (\cos E + E \sin E) \right.$$

$$\left. + e^3 (1 + \sin^2 E) \right] \tag{72}$$

Upon substitution of the expression for K_1 in Equation 66 we obtain the solution for λ_4:

$$\lambda_4 = A_4 \sin f - C_1 \cos f + C_2 \sin f \cdot I \tag{73}$$

and we note that there are no singularities remaining. The fact that Equation 65 has singularities is thus unimportant and the given expression for λ_4 which has finite values everywhere is the solution for λ_4 everywhere.

The equation for λ_6 is next investigated and when the coefficients are expressed in terms of true anomoly there results:

$$\lambda_6' - \frac{2e\lambda_6 \sin f}{1 + e \cos f} = \frac{-\lambda_3 p^2}{h(1 + e \cos f)^2} - \frac{2p\lambda_4}{(1 + e \cos f)} \tag{74}$$

The integral of the homogeneous part is

$$\lambda_6 = K_2 (1 + e \cos f)^{-2}$$

and again we utilize the variation of parameters method. We find that the differential equation for K_2 is

$$K_2' = -peC_1 - 2p\lambda_4 (1 + e \cos f) \tag{75}$$

The integral for K_2 is then found to be:

$$K_2 = A_6 + 2p (1 + e \cos f) (A_4 \cos f + C_1 \sin f) + 2p C_2 I_2 \tag{76}$$

where A_6 is the final constant of integration and where

$$I_2 = \int \sin f (1 + e \cos f) I \, df \tag{77}$$

$$= - \frac{1}{2e} \left[I (1 + e \cos f)^2 + \cot f \right] \tag{78}$$

Again, singularities seem to appear but in the final result for λ_6 there are none and so we will be able to use the results for any position in the orbit. We find:

$$\lambda_6 = \frac{A_6}{(1 + e \cos f)^2} + \frac{p (2 + e \cos f)}{(1 + e \cos f)^2} (A_4 \cos f + C_1 \sin f)$$

$$+ \frac{pC_2}{(1 - e^2)^2} \left[\frac{3eE}{\sqrt{1 - e^2}} - \frac{\sin f}{1 + e \cos f} \left(1 + 2e^2 + \frac{1 - e^2}{1 + e \cos f} \right) \right] \tag{79}$$

Finally, the expression for the Hamiltonian is solved for λ_1. The intermediate results again show possible zero divisors for some orbit positions but they do not exist in the final result for λ_1. We obtain first:

$$\lambda_1 = pA/(eh \sin f) - h/(r^2 \sin f) \left[h\lambda_3/(e\mu) + \lambda_4 \cos f - 2\lambda_6 \sin f/r \right] \tag{80}$$

The result of substituting the expressions for λ_3, λ_4, and λ_6 into Equation 80 is:

$$\lambda_1^o = 2A_6 h/(rp^2) + (1 + 2r/p)(h/r^2)(A_4 \cos f + C_1 \sin f) + AL_1 \quad (81)$$

where

$$L_1 = \frac{3\,p^3\,E\,(2 + e\cos f)}{hr^2\,(1 - e^2)^{5/2}} - \frac{p \sin f}{(1 - e^2)^2\,eh}\left[3 + 4e\cos f + e^2(3 + \cos^2 f) + 4e^3 \cos f - e^4(1 - 2\cos^2 f)\right] \quad (82)$$

Except for λ_1 which is directly above we now collect the results (reverting to A, the Hamiltonian, instead of C_2, restoring the superscript "o" and utilizing $p = r(1 + e\cos f)$ frequently).

$$\lambda_2^o = A_2 \sin f + B_2 \cos f \quad (83)$$

$$\lambda_3^o = eh\,C_1/p \quad (84)$$

$$\lambda_4^o = A_4 \sin f - C_1 \cos f + AL_4 \quad (85)$$

where

$$L_4 = +\frac{3\,e\,E\,p^3 \sin f}{(1 - e^2)^{5/2}} - \frac{p^3}{eh^2\,(1 - e^2)^2}\left[2e - \cos f\,(1 + e^2) + \frac{e^3 \sin^2 f}{1 + e\cos f}\right] \quad (86)$$

$$\lambda_5^o = r^2\,(A_2 \cos f - B_2 \sin f)/h \quad (87)$$

$$\lambda_6^{\circ} = A_6 r^2/p^2 + r^2 (1 + p/r)/p (A_4 \cos f + C_1 \sin f) + AL_6 \qquad (88)$$

where

$$L_6 = \frac{p^4}{eh^2 (1 - e^2)^2} \left[\frac{3eE}{\sqrt{1 - e^2}} - r \sin f/p \left[1 + 2e^2 + r(1 - e^2)/p \right] \right] \qquad (89)$$

We note that the singularities at f = 0 and π which appeared along the way have disappeared in the final expressions for all multipliers and thus it is legitimate to have ignored them. For circular orbits, however, there are terms in L_1, L_4, and L_6 which are unbounded for e = 0. We note that $\lambda_3 = eh\, C_1/p$. We assume that C_1 will remain finite as we consider a series of orbits of smaller and smaller eccentricity. Thus we are led to take

$$\lambda_3 = 0 \text{ for circular orbits.} \qquad (90)$$

It follows that A = 0 for circular orbits and hence that the terms in L_1, L_4 and L_6 do not exist for circular orbits. In the general case as well L_1, L_4 and L_6 are not needed since we have chosen A = 0. Thus we find that for circular orbits the equations used are entirely satisfactory and one needs only to add Equation 90 for this case.

III. QUASILINEARIZATION

GENERAL DESCRIPTION

Having obtained a nonlinear two-point boundary value problem, the powerful method of quasilinearization (2, 3, 4, 10, 12) may be used to generate the required numerical solutions. The previously derived differential equations may be written as a set of 14 first-order equations, each of which may be considered to be one component of the vector equation:

$$\dot{\underset{\sim}{X}} = g(\underset{\sim}{X}) \tag{91}$$

where

$$\underset{\sim}{X}^T = (r,\ \phi,\ \rho,\ y,\ m,\ \lambda_1,\ k,\ \lambda_4,\ \lambda_6,\ \theta,\ x,\ \lambda_2,\ \lambda_5,\ \lambda_3)$$

The differential equations are Equations 8, 10, 11, 13, 14, 18, 41, 21, 23, 9, 12, 19, 22, 20, respectively.

The sequence and numbering of variables used here and in the computer program is governed by two considerations. λ_3 is placed last since it is constant and does not require integration. The full set of equations for a two-dimensional problem is placed first so that the program could easily be programmed for either two- or three-dimensions. In two dimensions λ_3, the last variable, becomes variable 10 (instead of θ).

The quasilinearization method may be regarded as an extension of the Newton-Raphson method for algebraic equations to ordinary differential equations. Suppose that the "n'th" approximation to the time history of the solution vector, $\underset{\sim}{X}^{(n)}(t)$, is known. A Taylor series expansion about this approximation, truncated with linear terms, may then be made to obtain the derivative of the (n + 1)'st approximation, $\underset{\sim}{X}^{(n+1)}$:

$$\dot{x}_i^{(n+1)} = g_i(\underset{\sim}{X}^{(n)}) + \sum_{j=1}^{N} \left[x_j^{(n+1)} - x_j^{(n)} \right] \frac{\partial g_i(\underset{\sim}{X}^{(n)})}{\partial x_j} \qquad (92)$$

$$i = 1, 2 \ldots N$$

where N is the number of differential equations. This is the fundamental equation of quasilinearization[2, 3].

The Jacobian matrix of partial derivatives, $\partial g_i / \partial x_j$, is given below in Equation 118 for this problem. It does not contain the terms depending on the switching function which are required at switch points. These are handled in a special procedure as indicated below in Equations 96 through 114.

Because Equation 92 is linear in $x_i^{(n+1)}$, the solutions may be added to satisfy all the boundary conditions. At each stage of the computations the particular solution, $P_i(t)$, is required to satisfy the known initial boundary conditions and the "best" initial values of the remaining initial conditions while the homogeneous solutions, H_{ij}, utilize a non-zero initial value of only one of the non-specified initial conditions. Thus it is possible to generate a particular solution plus as many homogeneous solutions as

there are unknown initial boundary conditions and to form the (n+1)st approximation, $x_i^{(n+1)}(t)$, by summing the stored values of $P_i(t)$ and $H_{ij}(t)$:

$$x_i^{(n+1)}(t) = P_i(t) + \sum_{j=1}^{6} C_j H_{ij}(t) ; \quad i = 1, 2 \ldots N \tag{93}$$

The coefficients C_j are first obtained by requiring that the final boundary conditions (B_j) are satisfied; that is by

$$B_j(t_F) - P_j(t_F) = \sum_{k=1}^{6} C_k H_{jk}(t_F) , \quad j = 1, \ldots, 6 \tag{94}$$

If the process converges it will do so quadratically[2, 3]. However, in order for the process to converge, one must have a sufficiently accurate first approximation, $\underset{\sim}{X}^{(0)}(t)$, to the time history of the solution vector. The technique used to generate this first iterate is described in the following section. Convergence of the process may be examined by evaluating the following relationships at the end of each iteration:

$$\xi_i(\underset{\sim}{X}^{(n)}, \underset{\sim}{X}^{(n+1)}) = \underset{t}{MAX} \left| x_i^{(n)}(t) - x_i^{(n+1)}(t) \right| \tag{95}$$

$$i = 1, 2 \ldots N$$

The abrupt changes in mass flow rate require special consideration. Whenever the switching function exhibits a zero ("switch point"), the mass flow rate ($\dot{m} = -\beta$) must undergo a discontinuous change. This property can be expressed by employing a unit step function in the definition of β.

$$\beta = \beta_0 u\left[k^{(n)}(t)\right] \tag{96}$$

where

$$u\left[k^{(n)}(t)\right] = \int_{k-}^{k+} \delta\left[k^{(n)}(t)\right] dk \tag{97}$$

At a switch point ($k^{(n)} = 0$, and $t = t_s$) Eq. 92 must be integrated as follows:

$$\int_{t_s-}^{t_s+} \dot{x}_i^{(n+1)}\, dt = \int_{t_s-}^{t_s+} \left\{ g_i(\underset{\sim}{X}^{(n)}) + \sum_{j=1}^{N} \left[x_j^{(n+1)} - x_j^{(n)} \right] \frac{\partial g_i(\underset{\sim}{X}^{(n)})}{\partial x_j} \right\} dt \tag{98}$$

$$i = 1, 2 \ldots N$$

The $g_i(\underset{\sim}{X}^{(n)})$ term offers no net contribution and may therefore be excluded from further consideration. Next, observe that at a switch point the Jacobian matrix will have some non-zero terms appearing in the column containing partial derivatives with respect to the switching function (k). Note that this is only true at the switch points and that Eq. 118 is valid elsewhere. Thus, Eq. 98 becomes:

$$\Delta x_i^{(n+1)} = \int_{t_s-}^{t_s+} \left[k^{(n+1)} - k^{(n)} \right] \frac{\partial g_i(\underset{\sim}{X}^{(n)})}{\partial k}\, dt \tag{99}$$

Since the switching time is determined by the n'th iteration wherein $k^{(n)} = 0$ one obtains:

$$\Delta x_i^{(n+1)} = \int_{t_s-}^{t_s+} k^{(n+1)} \frac{\partial g_i(\underset{\sim}{X}^{(n)})}{\partial k}\, dt, \quad i = 1, 2 \ldots N \tag{100}$$

At this point it should be noted that only those terms in Equation 100 which contain mass flow rate will contribute to the solution at a discontinuity. Substitution of Equations 96, and 97 into Eq. 100 yields expressions of the following form:

$$\Delta x_i^{(n+1)} = S_i \beta_0 k^{(n+1)} \int_{t_{s-}}^{t_{s+}} \frac{\partial}{\partial k} \int_{k-}^{k+} \delta\left[k^{(n)}(t)\right] dt\, dt \qquad (101)$$

or

$$\Delta x_i^{(n+1)} = S_i \beta_0 k^{(n+1)} \int_{t_{s-}}^{t_{s+}} \delta\left[k^{(n)}(t)\right] dt, \quad i = 1, 2 \ldots N \qquad (102)$$

where $S_i\beta_0$ represents the rate of change of the variable due to thrust.

In order to integrate with respect to the argument of the delta function one may adopt the following definitions:

$$k = F(t), \quad t = F^{-1}(k) \qquad (103)$$

$$dt = \left[\frac{d}{dk} F^{-1}(k)\right] dk \qquad (104)$$

Equation 101 may now be rewritten and integrated with respect to k:

$$\Delta x_i^{(n+1)} = S_i \beta_0 k^{(n+1)} \int_{k-}^{k+} \delta\left[k^{(n)}(t)\right] \frac{d}{dk} F^{-1}(k)\, dk \qquad (105)$$

$$\Delta x_i^{(n+1)} = S_i \beta_0 k^{(n+1)} \frac{dt}{dk} \int_{k-}^{k+} \delta\left[k^{(n)}(t)\right] dk \qquad (106)$$

$$\Delta x_i^{(n+1)} = S_i \beta_0 k^{(n+1)} / \dot{k}, \quad i = 1, 2 \ldots N \qquad (107)$$

Equation 107 now may be applied to derive the following final expressions for the contributions to the solution at a corner point:

$$\Delta\rho = \Delta t\ c\beta\ \cos\nu\ \cos\psi/m \tag{108}$$

$$\Delta x = \Delta t\ c\beta\ \sin\psi/(mr) \tag{109}$$

$$\Delta y = \Delta t\ c\beta\ \sin\nu\ \cos\psi\ \sec\theta/(mr) \tag{110}$$

$$\Delta m = -\beta\Delta t \tag{111}$$

$$\Delta\lambda_1 = \Delta y\ \lambda_6/r \tag{112}$$

$$\Delta\lambda_2 = -\Delta y\ \lambda_6\ \tan\theta \tag{113}$$

where

$$\Delta t = k^{(n+1)}/|\dot{k}| \tag{114}$$

During the integration of the particular and homogeneous solutions required for Equation 94, $k^{(n+1)}$ will, in general, be non-zero at the switch points determined from the n'th iteration. The quantity, Δt, defined in Equation 114, may be regarded as an incremental change in the length of a burn period called for since $k^{(n+1)}$ is now non-zero at the switch points. The incremental changes defined by Eqs. 108 through 113 compensate for the changes which will result from this small changes in burning time. The need for the absolute value sign can be established by considering the physical implications of a non-zero value of $k^{(n+1)}$ at points where $k^{(n)} = 0$.

THE JACOBIAN

The matrix of partial derivatives of the differential equations is the "brains" of the quasilinearization program. For this problem we have a 14×14 matrix and it is illustrated in Equation 118. Only the zeros, ones

and very short terms are given in Equation 118. The 55 other longer terms are indicated and are given in order as Equations 119 to 173. For economies in writing and deriving these equations we have utilized the direction cosines of the thrust or primer vector, namely:

$$\ell_1 = \cos\psi \cos\nu = r\lambda_4 \cos\theta/D \tag{115}$$

$$\ell_2 = \cos\psi \sin\nu = \lambda_6/D \tag{116}$$

$$\ell_3 = \sin\psi = \lambda_5 \cos\theta/D \tag{117}$$

(118)

$$\frac{\partial g_i}{\partial x_j} =$$

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial \dot{\rho}}{\partial r} & \frac{\partial \dot{\rho}}{\partial \theta} & 2r\dot{x} & 0 & 0 & \frac{\partial \dot{\rho}}{\partial \lambda_5} & \frac{\partial \dot{k}}{\partial \lambda_3} & \frac{\partial \dot{\rho}}{\partial \lambda_4} \\
\frac{\partial \dot{y}}{\partial r} & \frac{\partial \dot{y}}{\partial \theta} & \frac{\partial \dot{y}}{\partial x} & 0 & 0 & \frac{\partial \dot{y}}{\partial \lambda_5} & \frac{\partial \dot{y}}{\partial m} & \frac{\partial \dot{y}}{\partial \lambda_4} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial \dot{\lambda}_1}{\partial r} & \frac{\partial \dot{\lambda}_1}{\partial \theta} & \frac{\partial \dot{\lambda}_1}{\partial x} & 0 & 0 & \frac{\partial \dot{\lambda}_1}{\partial \lambda_5} & \frac{\partial \dot{\lambda}_1}{\partial m} & \frac{\partial \dot{\lambda}_1}{\partial \lambda_4} \\
\frac{\partial \dot{k}}{\partial r} & \frac{\partial \dot{k}}{\partial \lambda_6} & \frac{\partial \dot{k}}{\partial x} & \frac{-c\ell_3}{mr} & \frac{\partial \dot{k}}{\partial \lambda_5} & \frac{\partial \dot{k}}{\partial \lambda_3} & \frac{-\dot{k}}{m} & \frac{\partial \dot{k}}{\partial \lambda_4} \\
\frac{\partial \dot{\lambda}_4}{\partial r} & 0 & \frac{2\lambda_5}{r} & 0 & \frac{2x}{r} & 0 & \frac{-c\ell_1}{m} & 0 \\
\frac{\partial \dot{\lambda}_6}{\partial r} & \frac{\partial \dot{\lambda}_6}{\partial \theta} & \frac{\partial \dot{\lambda}_6}{\partial x} & 0 & \frac{\partial \dot{\lambda}_6}{\partial \lambda_5} & -1 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
\frac{\partial \dot{x}}{\partial r} & \frac{\partial \dot{x}}{\partial \theta} & \frac{-2\rho}{r} & 0 & \frac{\partial \dot{x}}{\partial \lambda_5} & 0 & \frac{-c\beta\ell_3}{m^2 r} & \frac{\partial \dot{x}}{\partial \lambda_4} \\
\frac{\partial \dot{\lambda}_2}{\partial r} & \frac{\partial \dot{\lambda}_2}{\partial \theta} & \frac{\partial \dot{\lambda}_2}{\partial x} & 0 & \frac{\partial \dot{\lambda}_2}{\partial \lambda_5} & 0 & \frac{\partial \dot{\lambda}_2}{\partial m} & \frac{\partial \dot{\lambda}_2}{\partial \lambda_4} \\
\frac{\partial \dot{\lambda}_5}{\partial r} & \frac{\partial \dot{\lambda}_5}{\partial \theta} & -2r\lambda_4 & -1 & \frac{2\rho}{r} & 0 & 0 & -2r\dot{x} \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}$$

$$\frac{\partial \dot{\rho}}{\partial r} = x^2 + y^2 \cos^2 \theta - 2\mu/r^2 + c\beta \ell_1 (1 - \ell_1^2)/(mr) \tag{119}$$

$$\frac{\partial \dot{\rho}}{\partial y} = 2ry \cos^2 \theta \tag{120}$$

$$\frac{\partial \dot{\rho}}{\partial \lambda_4} = c\beta \ell_1 (1 - \ell_1^2)/(m \lambda_4) \tag{121}$$

$$\frac{\partial \dot{\rho}}{\partial \lambda_6} = -c\beta \ell_1 \ell_2/(mD) \tag{122}$$

$$\frac{\partial \dot{\rho}}{\partial \theta} = -ry^2 \sin 2\theta - c\beta \lambda_4 \ell_2^2 \, r \sin \theta/(mD) \tag{123}$$

$$\frac{\partial \dot{\rho}}{\partial \lambda_5} = -c\beta \ell_1 \ell_3^2/(m\lambda_5) \tag{124}$$

$$\frac{\partial \dot{y}}{\partial r} = 2\rho y/r^2 - c\beta \ell_2 (1 + \ell_1^2) \sec \theta/(mr^2) \tag{125}$$

$$\frac{\partial \dot{y}}{\partial y} = 2x \tan \theta - 2\rho/r \tag{126}$$

$$\frac{\partial \dot{y}}{\partial m} = c\beta \ell_2 \sec \theta/(m^2 r) \tag{127}$$

$$\frac{\partial \dot{y}}{\partial \lambda_4} = -c\beta \ell_1 \ell_2/(mD) \tag{128}$$

$$\frac{\partial \dot{y}}{\partial \lambda_6} = c\beta (1 - \ell_2^2) \sec \theta/(mDr) \tag{129}$$

$$\frac{\partial \dot{y}}{\partial \theta} = \left[2xy + c\beta \ell_2 \sin\theta \,(1 + \ell_1^2 + \ell_3^2)/(mr)\right] \sec^2\theta \tag{130}$$

$$\frac{\partial \dot{y}}{\partial \lambda_5} = -c\beta \ell_2 \ell_3/(mDr) \tag{131}$$

$$\frac{\partial \dot{\lambda}_1}{\partial r} = 6\mu\lambda_4/r^4 + 4\rho\,(\lambda_5 x + \lambda_6 y)/r^3 - c\beta D\,(2 + \ell_1^2)\,(1 - \ell_1^2)\sec\theta/(mr^3) \tag{132}$$

$$\frac{\partial \dot{\lambda}_1}{\partial \rho} = -2\,(\lambda_5 x + \lambda_6 y)/r^2 \tag{133}$$

$$\frac{\partial \dot{\lambda}_1}{\partial y} = -2\,(\lambda_4 y \cos\theta + \lambda_6 \rho/r^2) \tag{134}$$

$$\frac{\partial \dot{\lambda}_1}{\partial m} = -c\beta D\,(\ell_2^2 + \ell_3^2)\sec\theta/(mr)^2 \tag{135}$$

$$\frac{\partial \dot{\lambda}_1}{\partial \lambda_4} = -\left[x^2 + y^2\cos^2\theta + \frac{2\mu}{r^3}\right] - c\beta\,\ell_1\,(\ell_2^2 + \ell_3^2)/(mr) \tag{136}$$

$$\frac{\partial \dot{\lambda}_1}{\partial \lambda_6} = -2\rho y/r^2 + c\beta\,\ell_2\,(1 + \ell_1^2)\sec\theta/(mr^2) \tag{137}$$

$$\frac{\partial \dot{\lambda}_1}{\partial \theta} = \lambda_4 y^2 \sin 2\theta + c\beta\, D\,\ell_2^2\,(1 + \ell_1^2)\sin\theta\sec^2\theta/(mr^2) \tag{138}$$

$$\frac{\partial \dot{\lambda}_1}{\partial x} = -2\,(\lambda_4 x + \rho\lambda_5/r^2) \tag{139}$$

$$\frac{\partial \dot{\lambda}_1}{\partial \lambda_5} = -2\rho x/r^2 + c\beta \ell_3 (1 + \ell_1^2)/(mr^2) \tag{140}$$

$$\frac{\partial \dot{k}}{\partial r} = c\left[-r\ell_1\lambda_1 (1 - \ell_1^2) + \lambda_2\ell_3 (1 + \ell_1^2) + \lambda_3\ell_2 (1 + \ell_1^2) \sec\theta\right]/(mr^2)$$

$$- c\rho D (1 - \ell_1^2)(2 + \ell_1^2) \sec\theta/(mr^3) \tag{141}$$

$$+ cx D\ell_2^2 (1 + \ell_1^2) \tan\theta \sec\theta/(mr^2)$$

$$\frac{\partial \dot{k}}{\partial \rho} = cD (1 - \ell_1^2) \sec\theta/(mr^2) \tag{142}$$

$$\frac{\partial \dot{k}}{\partial \lambda_4} = c\left[-\lambda_1 (1 - \ell_1^2) r \cos\theta + \lambda_2\ell_1\ell_3 \cos\theta + \lambda_3\ell_1\ell_2\right]/(mD) \tag{143}$$

$$+ cx\ell_1\ell_2^2 \tan\theta/m - c\rho\ell_1 (1 - \ell_1^2/(mr)$$

$$\frac{\partial \dot{k}}{\partial \lambda_6} = c\left[r\lambda_1\ell_1\ell_2 + \lambda_2\ell_2\ell_3 - \lambda_3 (1 - \ell_2^2) \sec\theta\right]/(mDr)$$

$$- cx\ell_2 (2 - \ell_2^2) \tan\theta \sec\theta/(mr) \tag{144}$$

$$+ c\rho\ell_2 (1 + \ell_1^2) \sec\theta/(mr^2)$$

$$\frac{\partial \dot{k}}{\partial x} = cD\ell_2^2 \tan\theta \sec\theta/(mr) \tag{145}$$

$$\frac{\partial \dot{k}}{\partial \lambda_5} = c\left[r\lambda_1\ell_1\ell_3 \cos\theta - \lambda_2 (1 - \ell_3^2) \cos\theta + \lambda_3\ell_2\ell_3\right]/(mDr) \tag{146}$$

$$+ cx\ell_2^2 \ell_3 \tan\theta/(mr) + c\rho\ell_3 (1 + \ell_1^2)/(mr^2)$$

$$\frac{\partial \dot{k}}{\partial \lambda_3} = - c \ell_2 \sec \theta/(mr) \tag{147}$$

$$\frac{\partial \dot{\lambda}_4}{\partial r} = -2 (\lambda_5 x + \lambda_6 y)/r^2 \tag{148}$$

$$\frac{\partial \dot{\lambda}_6}{\partial r} = -2(\lambda_4 y \cos^2 \theta + \lambda_6/r^2) \tag{149}$$

$$\frac{\partial \dot{\lambda}_6}{\partial y} = -2 \cos \theta (r\lambda_4 \cos \theta - \lambda_5 \sin \theta) \tag{150}$$

$$\frac{\partial \dot{\lambda}_6}{\partial \lambda_4} = -2ry \cos^2 \theta \tag{151}$$

$$\frac{\partial \dot{\lambda}_6}{\partial \lambda_4} = -2 (x \tan \theta - \rho/r) \tag{152}$$

$$\frac{\partial \dot{\lambda}_6}{\lambda \theta} = 2ry\lambda_4 \sin 2\theta + 2\lambda_5 y \cos 2\theta - 2x\lambda_6 \sec^2 \theta \tag{153}$$

$$\frac{\partial \dot{\lambda}_6}{\partial x_5} = -2\lambda_6 \tan \theta \tag{154}$$

$$\frac{\partial \dot{\lambda}_6}{\partial x} = y \sin 2\theta \tag{155}$$

$$\frac{\partial \dot{x}}{\partial r} = 2\rho x/r^2 - c\beta \ell_3 (1+\ell_1^2)/(mr^2) \tag{156}$$

$$\frac{\partial \dot{x}}{\partial y} = -y \sin 2\theta \tag{157}$$

$$\frac{\partial \dot{x}}{\partial \lambda_4} = -c\beta\ell_1 (1-\ell_1^2)/(m\lambda_4) \tag{158}$$

$$\frac{\partial \dot{x}}{\partial \lambda_6} = -c\beta\ell_1\ell_2/(mD) \tag{159}$$

$$\frac{\partial \dot{x}}{\partial \theta} = -y^2 \cos 2\theta - c\beta\ell_2^2 \lambda_5 \sin \theta/(mDr) \tag{160}$$

$$\frac{\partial \dot{x}}{\partial \lambda_5} = -c\beta\ell_1\ell_3^2/(m\lambda_5) \tag{161}$$

$$\frac{\partial \dot{\lambda}_2}{\partial r} = \lambda_4 y^2 \sin 2\theta + c\beta D\ell_2^2 (1+\ell_1^2) \sin \theta \sec^2 \theta/(mr^2) \tag{162}$$

$$\frac{\partial \dot{\lambda}_2}{\partial y} = 2\lambda_4 ry \sin 2\theta + 2\lambda_5 y \cos 2\theta - 2x\lambda_6 \sec^2 \theta \tag{163}$$

$$\frac{\partial \dot{\lambda}_2}{\partial m} = -c\beta D\ell_2^2 \sin \theta \sec^2 \theta/(m^2 r) \tag{164}$$

$$\frac{\partial \dot{\lambda}_2}{\partial \lambda_4} = ry^2 \sin 2\theta + c\beta\ell_1\ell_2^2 \tan \theta/m \tag{165}$$

$$\frac{\partial \dot{\lambda}_2}{\partial \lambda_6} = -2xy \sec^2 \theta - c\beta\ell_2^2 (2-\ell_2^2) \tan \theta \sec \theta/(mr) \tag{166}$$

$$\frac{\partial \dot{\lambda}_2}{\partial \theta} = 2ry^2\lambda_4 \cos 2\theta - 2y^2\lambda_5 \sin 2\theta - 4xy\lambda_6 \tan\theta \sec^2\theta$$

$$-c\beta D\ell_2{}^2 \left[1 + \sin^2\theta\,(2-\ell_2{}^2)\right] \sec^3\theta/(mr) \qquad (167)$$

$$\frac{\partial \dot{\lambda}_2}{\partial x} = -2y\,\lambda_6 \sec^2\theta \qquad (168)$$

$$\frac{\partial \dot{\lambda}_2}{\partial \lambda_5} = y^2 \cos 2\theta + c\beta\ell_2{}^2\ell_3 \tan\theta/(mr) \qquad (169)$$

$$\frac{\partial \lambda_5}{\partial r} = -2\,(x\lambda_4 + \lambda_5\,\rho/r^2) \qquad (170)$$

$$\frac{\partial \dot{\lambda}_5}{\partial y} = -2\lambda_6 \tan\theta \qquad (171)$$

$$\frac{\partial \dot{\lambda}_5}{\partial \lambda_6} = -2y \tan\theta \qquad (172)$$

$$\frac{\partial \dot{\lambda}_5}{\partial \theta} = -2y\lambda_6 \sec^2\theta \qquad (173)$$

COMPUTATION OF THE INITIAL ITERATE

The time scale and the initial time histories of all variables are computed from the numerical results for an optimum two-impulse transfer (see Reference 11). We describe first the procedures for constructing the time scale and the mass history, then those for computing the state variables, and finally those for computing the Lagrange multipliers and the switching function.

Time Scale and Mass History

The time scale is based on the impulses (I_1, I_2) and the transfer time (t_t) from the two-impulse computation and on the vehicle and engine characteristics. It is illustrated symbolically in Figure 2. There is an initial coasting arc on the first orbit of some given duration before the engine is first ignited; e.g., t_A = three percent of a revolution). The mass at the end of the first burn (m_1) and the duration of the first burn (t_1) are determined from

$$I_1 = g_o \cdot I_{sp} \ln (m_o/m_1) \tag{174}$$

and

$$t_1 = (m_o - m_1)/\beta \tag{175}$$

where m_o, is the initial mass, I_{sp} is the specific impulse of the fuel, β is the mass flow rate, and g_o is the acceleration of gravity at the Central Body's surface. (The value of g_o is included in the program for either British or metric units as desired, but if some other system is called for, g_o is set to 1.0 and I_{sp} must then be the exhaust velocity of the fuel in the desired units.)

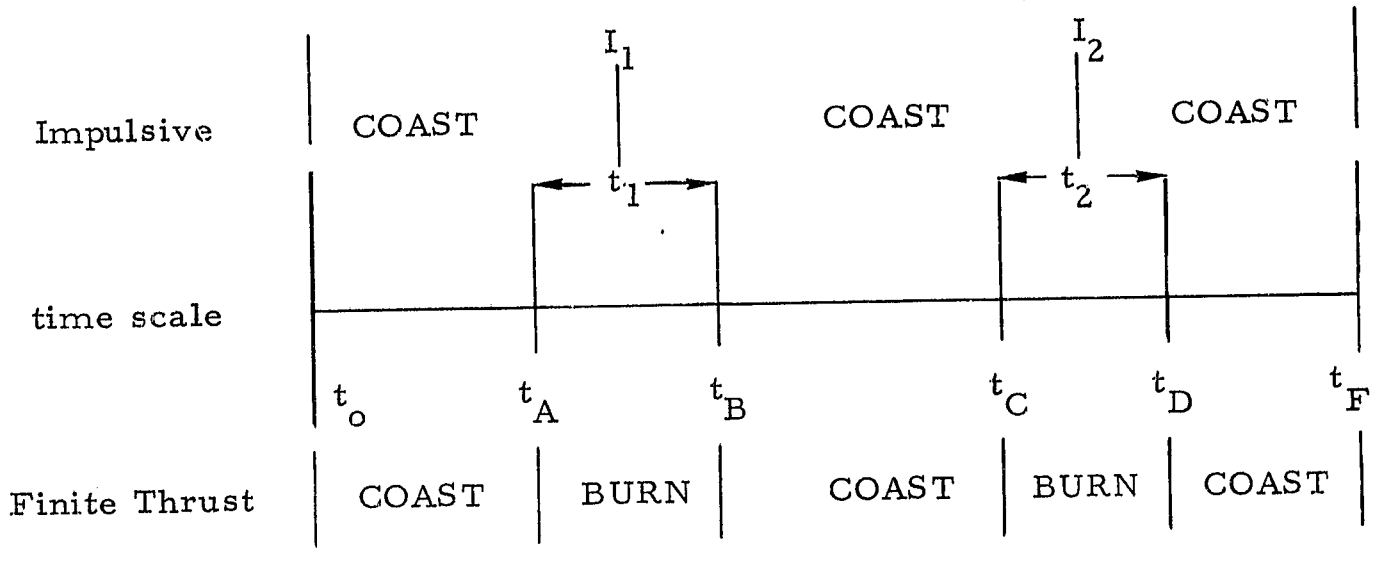


Figure 2. Symbolic Presentation of the Time Scale

The final mass (m_2) and the second burn time (t_2) are similarly determined.

$$I_2 = g_o I_{sp} \ln (m_1/m_2) \tag{176}$$

$$t_2 = (m_1 - m_2)/\beta \tag{177}$$

The two burn intervals (t_1 and t_2) are centered at the impulse times as called for by the impulsive trajectory. We require that t_B does not exceed t_C; that is, a coasting time on the transfer orbit should occur. Finally, there is a period of coasting along the final orbit ($t_F - t_D$) until the final point is reached.

The program makes use of 100 values of the time. The 99 steps are divided among the five intervals (coast, burn, coast, burn, coast) by requiring a minimum number in any interval and by specifying the desired

step size. Any excess available is automatically assigned to the coast arc along the transfer orbit. If too many steps are called for the numbers of steps assigned are multiplied by the fraction needed to make the total 99.

The Initial Iterate of the State Variables

Since orbital elements and times are now available along the three coast arcs, it is easy to compute the position and velocity of the vehicle for these three intervals; that is from 0 to t_A, from t_B to t_C, and from t_D to t_F.

For the two burn intervals (t_A to t_B and t_C to t_D) the curves for each variable are bridged by means of a straight line or by means of a cubic fitting function such that the rates at the two ends and the curves themselves match.

The Initial Iterate of the Lagrange Multipliers and the Switching Function

In this section the computational procedure to produce initial time histories of the six multipliers for the position and velocity equations is given. These are equivalent to primer vector and its derivative as suggested by Lawden[14]. The equations used are given without derivation, although the procedure to obtain them is indicated so that an interested reader can verify the results. A block diagram indicating the computational steps required is given in Figure 3.

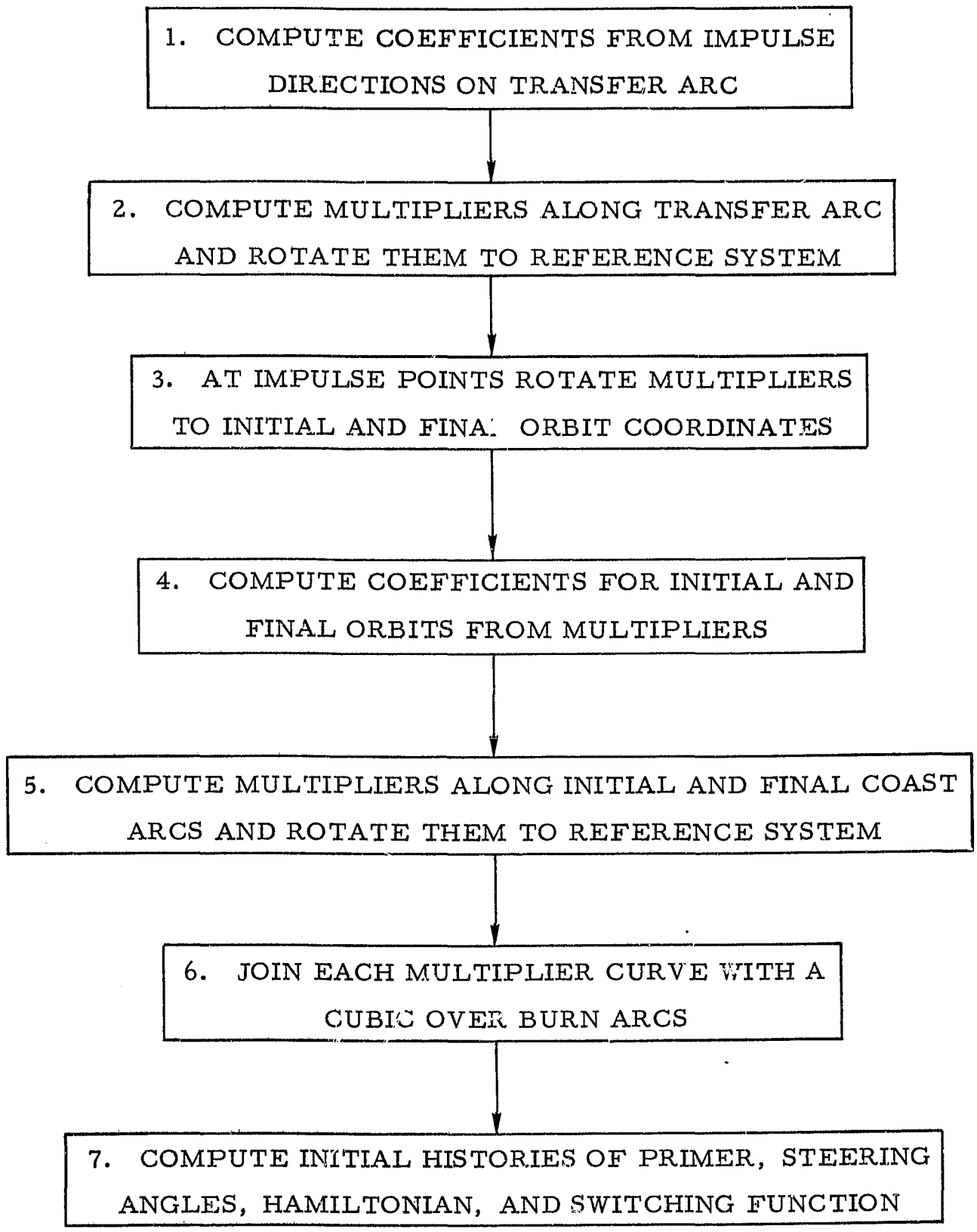


Figure 3. Procedure for Computation of Initial Time Histories of Multipliers Primer, Steering Angles, Hamiltonian, and Switching Function

It is possible to compute the values of the Lagrange multipliers along coast arcs because it is possible to integrate the differential equations for them in a closed form as shown above on pages 12 through 19. The integration is done in a system of coordinates in which the orbit lies in the reference plane. The mutlipliers or the components of the primer vector and of its derivative are then computed in the desired frame of reference by the proper coordinate rotation at each point of computation along the coast trajectories. The variations of the multipliers over the burn arcs are then handled like the state variables; that is, by a linear or cubic bridge.

Before proceeding with computational matters, we summarize the equations for the multipliers and the primer vector. As has been stated we have assumed that the Hamiltonian, A, is zero which implies the assumption that the impulsive result is close to the time-open optimal solution. Thus we begin with the set of Equations 81, 83, 84, 85, 87, 88 with A = 0.

These are:

$$\lambda_1^o = 2A_6 h/(rp^2) + (1 + 2r/p)\, h/r^2\, (A_4 \cos f + C_1 \sin f_1) \qquad (178)$$

$$\lambda_2^o = A_2 \sin f + B_2 \cos f \qquad (179)$$

$$\lambda_3^o = eh\, C_1/p \qquad (180)$$

$$\lambda_4^o = A_4 \sin f - C_1 \cos f \qquad (181)$$

$$\lambda_5^o = r^2/h\, (A_2 \cos f - B_2 \sin f) \qquad (182)$$

$$\lambda_6^o = A_6 r^2/p^2 + r/p\,(1 + r/p)\,(A_4 \cos f + C_1 \sin f) \qquad (183)$$

where e, h, p are eccentricity, angular momentum/mass, and semi-latus rectum of the orbit and where f is the true anomaly along the orbit. The direction of the thrust, when it is called for, is given by the primer vector, $\underline{Q}$, which is:

$$\underline{Q} = \lambda_4^o \underline{U}_r + (\lambda_5^o \underline{U}_h + \lambda_6^o \underline{U}_p)/r \qquad (184)$$

where $\underline{U}_r$, $\underline{U}_h$, $\underline{U}_p$ are radial, binormal, circumferential unit vectors, respectively. When the primer is expressed in the more general coordinate system (that of the overall problem) it is given by:

$$\underline{Q} = \lambda_4 \underline{U}_r + \lambda_5/r \underline{U}_\psi + \lambda_6/(r \cos\theta) \underline{U}_\theta \qquad (185)$$

where $\underline{U}_r$, $\underline{U}_\phi$, $\underline{U}_\theta$ are radial, east, north unit vectors, respectively. It is the vector property of $\underline{Q}$ and its derivative $\dot{\underline{Q}}$ which are used to determine the scheme for changing coordinate systems. Thus we find:

$$\lambda_4 = \lambda_4^o \qquad (186)$$

$$\lambda_6/\cos\theta = \lambda_6^o \cos\alpha - \lambda_5^o \sin\alpha \qquad (187)$$

$$\lambda_5 = \lambda_6^o \sin\alpha + \lambda_5^o \cos\alpha \qquad (188)$$

$$\lambda_1 = \lambda_1^o \qquad (189)$$

$$\lambda_3/\cos\theta = \lambda_3^o \cos\alpha - \lambda_2^o \sin\alpha + \lambda_5^o y_o \tan\theta \qquad (190)$$

$$\lambda_2 = \lambda_3^o \sin\alpha + \lambda_2^o \cos\alpha - y \sin\theta\,(\lambda_6^o \cos\alpha - \lambda_5^o \sin\alpha) \qquad (191)$$

where α = heading of the subsatellite point north of east, that is,

$$\tan\alpha = x/(y\cos\theta) \tag{192}$$

and where y_o is the total angular velocity in the orbit plane x/(y as θ), that is,

$$y_o = \sqrt{x^2 + y^2 \cos^2\theta} \tag{193}$$

The inverse to the transformation given by Equations 186 through 191 is easily found to be:

$$\lambda_4^o = \lambda_4 \tag{194}$$

$$\lambda_5^o = \lambda_5 \cos\alpha - \frac{\lambda_6 \sin\alpha}{\cos\theta} \tag{195}$$

$$\lambda_6^o = \lambda_5 \sin\alpha + \frac{\lambda_6 \cos\alpha}{\cos\theta} \tag{196}$$

$$\lambda_1^o = \lambda_1 \tag{197}$$

$$\lambda_2^o = \lambda_2 \cos\alpha - \frac{\lambda_3 \sin\alpha}{\cos\theta} + y\lambda_6 \tan\theta \cos\alpha + y_o\lambda_5^o \tan\theta \sin\alpha \tag{198}$$

$$\lambda_3^o = \lambda_2 \sin\alpha + \frac{\lambda_3 \cos\alpha}{\cos\theta} + y\lambda_6 \tan\theta \sin\alpha - y_o\lambda_5^o \tan\theta \cos\alpha \tag{199}$$

The fact that λ_3 is a constant is checked by noting that equation 190 reduces to:

$$\lambda_3 = \lambda_3^o \cos i + A_2 \sin i \sin\omega - B_2 \sin i \cos\omega \tag{200}$$

when it is expressed in terms of the orbital elements.

Another manipulation that is needed is to obtain the constants A_2, B_2, A_4, A_6, and C_1 from the multipliers, $\lambda_1^o, \ldots, \lambda_6^o$ at a point. This is an over-determined situation, for it has been assumed that A = 0 for the problem. The equations can be written in the following form:

$$A_2 = \lambda_2^o \sin f + \lambda_5^o h/r^2 \cos f \tag{201}$$

$$B_2 = \lambda_2^o \cos f - \frac{\lambda_5^o h}{r^2} \sin f \tag{202}$$

$$A_4 = \frac{2\lambda_6^o}{r} \cos f - \frac{\lambda_1^o r^2}{h} \cos f + \lambda_4^o \sin f \tag{203}$$

$$A_6 = \frac{\lambda_1^o r p^2}{h}\left(1 + \frac{r}{p}\right) - \lambda_6^o \frac{p^2}{r^2}\left(1 + \frac{2r}{p}\right) \tag{204}$$

$$C_1 = \frac{\lambda_3^o p}{he} \tag{205}$$

$$C_1 = \frac{2\lambda_6^o \sin f}{r} - \frac{\lambda_1^o r^2}{h} \sin f - \lambda_4^o \cos f \tag{206}$$

The last two equations may be considered as yielding a computational check on the validity of the assumption that A = 0.

Except for the first step itself the mathematical details required for the computations indicated in Figure 3 have now been presented. Thus let us return to the outline of Figure 3 and consider the computational procedure.

Consider the transfer orbit arc between the two impulses. Since A = 0 and since only the ratios of the multipliers are utilized in obtaining the steering angles, there remain only four conditions needed to determine the five coefficients A_2, B_2, A_4, A_6, and C_1. Except for the special case of at least one circular orbit in the coplanar case we choose an initial arbitrary value of λ^{o}_{41}.* The sign must be that of the radial thrust at the first impulse in order for the steering angle to lie in the proper direction (outward is positive). Next, λ^{o}_{61} and λ^{o}_{51} are obtained by making the primer lie in the thrust direction. That is:

$$\lambda^{o}_{61} = \lambda^{o}_{41}\, r_1\, (F_1)_C/(F_1)_R \tag{207}$$

$$\lambda^{o}_{51} = \lambda^{o}_{41}\, r_1\, (F_1)_B/(F_1)_R \tag{208}$$

For the second impulse we use:

$$R = \lambda^{o}_{52}/\lambda^{o}_{42} = r_2\,(F_2)_B/(F_2)_R \tag{209}$$

$$S = \lambda^{o}_{62}/\lambda^{o}_{42} = r_2(F_2)_C/(F_2)_R \tag{210}$$

Thus to determine the coefficients the following set of equations has to be solved:

$$A_4 \sin f_1 - C_1 \cos t_1 = \lambda^{o}_{41} \tag{211}$$

$$r_1 \left(1 + \frac{r_1}{p}\right)(A_4 \cos f + C_1 \sin t) + A_6 \frac{r_1^2}{p^2} = \lambda^{o}_{61} \tag{212}$$

---

*The additional subscript indicates which of the two impulses is being considered.

$$A_2 \cos f_1 - B_2 \sin f_1 = \lambda^o_{51} \, h/r_1^2 \tag{213}$$

$$\frac{-r_2^2}{h} A_2 \cos f_2 + \frac{r_2^2}{h} B_2 \sin f_2 + R A_4 \sin f_2 - R C_1 \cos f_2 = 0 \tag{214}$$

$$\left[r_2 \left(1 + \frac{r^2}{p} \right) \cos f_2 - S \sin f_2 \right] A_4 + A_6 \frac{r_2^2}{p^2} + \left[r_2 \left(1 + \frac{r^2}{p} \right) \sin f_2 - S \cos f_2 \right] C_1 = 0 \tag{215}$$

The results are:

$$A_2 = \left[\lambda^o_{41} \sin f_1 \cdot D_3 + \lambda^o_{61} \frac{r_2^2 R}{p^2} \sin f_1 \sin \Delta 0 + \lambda^o_{51} \frac{r_2^2}{r_1^2} \sin f_2 \, D_4 \right] / D_o \tag{216}$$

$$B_2 = \left[\lambda^o_{41} \cos f_1 \cdot D_3 + \lambda^o_{61} \frac{r_2^2 R}{p^2} \cos f_1 \sin \Delta 0 + \lambda^o_{51} \frac{r_2^2}{r_1^2} \cos f_2 \, D_4 \right] / D_o \tag{217}$$

$$A_4 = \lambda^o_{41} \left\{ \frac{r_1^2}{p^2} \left[S \cos f_2 + r^2 \left(1 + \frac{r^2}{p} \right) \sin f_2 \right] - \frac{r_2^2 r_1}{p^2} \left(1 + \frac{r_1}{p} \right) \sin f_2 \right\} / D_4 - \lambda^o_{61} \frac{r_2^2}{p^2} \cos f_1 / D_4 \tag{218}$$

$$A_6 = -\lambda^o_{41} r_1 \left(1 + \frac{r_1}{p} \right) \left[S \cos \Delta 0 + r_2 \left(1 + \frac{r_2}{p} \right) \sin \Delta\theta \right] / D_4 + \lambda^o_{61} \left[-S \sin \Delta 0 + r_2 \left(1 + \frac{r_2}{p} \right) \cos \Delta\theta \right] / D_4 \tag{219}$$

$$C_1 = \lambda_{41}^{o} \left\{ \frac{r_1^2 S}{p^2} \sin f_2 + \frac{r_1 r_2}{p^2} \left[-r_1 \left(1 + \frac{r_1}{p} \right) \cos f_2 \right. \right. \tag{220}$$

$$\left. \left. + r_2 \left(1 + \frac{r_2}{p} \right) \cos f_1 \right] \right\} / D_4 - \lambda_{61}^{o} \frac{r_2^2}{p^2} \sin f_1 / D_4$$

where the determinant of the coefficients is D and it is expressed as

$$D_o = D_2 + D_4, \text{ (determinant of the coefficients)} \tag{221}$$

with

$$D_2 = \frac{r_2^2}{h} \sin \Delta \theta \tag{222}$$

$$D_3 = \frac{R\, r_1\, r_2}{p^2} \left[r_2 \left(1 + \frac{r_1}{p} \right) \cos \Delta \theta - r_1 \left(1 + \frac{r_2}{p} \right) \right] \tag{223}$$

$$D_4 = \frac{r_1^2}{p^2} \left[-S \sin \Delta \theta + r_2 \left(1 + \frac{r_2}{p} \right) \cos \Delta \theta \right] - \frac{r_2^2}{p^2} r_1 \left(1 + \frac{r_1}{p} \right) \tag{224}$$

and

$$\Delta \theta = f_2 - f_1 \tag{225}$$

Zero divisors will occur if the radial thrust happens to equal zero at either impulse. Note that if either or both orbits are circular for the coplanar case a different technique is used (see the next paragraph). If the radial thrust is less than an assumed small value it is automatically set to 10^{-4} times the circumferential thrust. This will cause only a very small

error in the steering angles and is not expected to affect the convergence materially or to affect the solution at all.

If the initial and final orbits are coplanar no rotation of the multipliers is required, i.e., $\lambda_i = \lambda_i^o$ at all points and $A_2 = B_2 = 0$ for all orbits. If, in addition, one of the orbits is circular, we find that along it $\lambda_3 = 0$ (provided C_1 is not infinite). Consequently, $\lambda_3 = 0$ for all orbits and we may assume that $C_1 = 0$ for the transfer orbit. There remains only one undetermined value. We assume that the given constant is A_6 (instead of λ_{41}^o) and then we determine A_4 from the initial thrust direction. The final thrust direction is not used since the set of values of the coefficients is already complete.

In Steps 2 and 3 of Figure 3 the multiplier values are computed along the coasting part of the transfer trajectory (from t_B to t_C) and at the two impulse application points. The former are rotated to the coordinate system of the problem and the latter to the coordinate systems of the initial and final orbits by use of Equations 186 through 191 and Equations 194 through 199.

As has been mentioned, with the spherical coordinate system the multipliers λ_1 and λ_2 behave as do the velocity components (Equations 18 and 19). Thus it is necessary to introduce steps in λ_1 and λ_2 at the impulse points when these are to be referred to the initial, final orbits. From Equations 18 and 19 these steps are easily seen to be given by the following equations (note that each $(\lambda_i)_1$ corresponds to a step backward in time).

$$(\Delta \lambda_1)_1 = -\frac{I_1}{r_1^3 Q} (\lambda_5^2 + \lambda_6^2 \sec^2 \theta) \qquad (226)$$

$$(\Delta \lambda_2)_1 = + \frac{I_1}{r_1^2 Q} \lambda_6^2 \sec^2 \theta \tan \theta \qquad (227)$$

$$(\Delta \lambda_1)_2 = + \frac{I_2}{r_2^3 Q} (\lambda_5^2 + \lambda_6^2 \sec^2 \theta) \qquad (228)$$

$$(\Delta \lambda_2)_2 = - \frac{I_2}{r_2^2 Q} \lambda_6^2 \sec^2 \theta \tan \theta \qquad (229)$$

The inclusion of this feature as the program was being developed produced a remarkable improvement in the program behavior inasmuch as it yielded very good primer and switching function time history estimations.

In step 4 (Figure 3) the coefficients are computed for the initial and final orbits from Equations 164 through 169. Results from both equations for C_1 are printed. Approximately equality of the two results indicates a valid set of initial conditions for time open problems (A = 0). The failure of these two numbers to agree, as sometimes occurs, is due to a non-zero value of the Hamiltonian being required for consistency with the impulsive data. This implies that the impulsive data is not really optimal. The final time to reach the specified point is an important sensitive element for this comparison as is illustrated particularly by cases where one impulse is very much larger than the other. The finite thrust trajectory tends to "cut the corner" compared to the impulsive trajectory and thus requires a slightly smaller total time.

In step 5 (Figure 3) the multipliers are computed along the coasting portions of the initial, final orbits, and are rotated as needed into the coordinate system for the problem. The burn arc portions of the variations of the multipliers are determined in step 6 (Figure 3) by fitting a cubic (in time) over the interval in such a way as to provide continuity in both value and rate for each multiplier. With a complete set of the six multipliers over the entire trajectory the steering angles are directly computed using Equations 148, 115, 116, 117. Finally, the initial history of the primer length, P, and the switching function, h, are computed by using

$$Q = D \sec \theta / r = \sqrt{\lambda_4^2 + \left(\lambda_5^2 + \lambda_6^2 \cos^2 \theta\right)/r^2} \qquad (230)$$

and

$$k = \frac{c}{m} (Q - Q_o) \qquad (231)$$

where Q_o is the value of the length of the primer at the impulses (as computed along the coasting trajectory). The length of the primer generally has two equal maxima at these points. If the two are not equal, however, the value of Q_o in Equation 231 is linearly varied with time along the coast arc from one value to the other so as to produce a proper shape for the switching function.

The behavior of the primer vector length or the switching function gives a significant indication as to whether an optimum finite-thrust transfer can

be found. As is well known for the impulsive case, the thrust is to be used where the primer length is a maximum, and in fact, if any thrust is suggested away from such a maximum it will not represent an optimal situation(15). The fact is that in the great majority of cases the program yields a curve having two equal maxima for both the primer and the switching function. The program then succeeds in finding locally optimal finite-thrust trajectories.

Note that in computing the multipliers the condition that the primer length possess a maximum was not utilized. It would be extremely difficult to do this because of the quadratic nature of the expressions required. Rather, we made use only of the direction of the thrust as determined from an optimal impulse computation. Much of the success of the finite-thrust program is thus due to the fact that the impulsive computation does yield a locally optimal transfer as indicated by the results of the primer length computation described above.

Where the behavior of the primer length of the initial iterate is different from that described, then no optimal finite-thrust transfer near the situation being attempted exists or, if one does, special techniques will be required. This is because it is indicating that to be optimal the maneuver requires one or three burns, but not the two burns the program is attempting to find.

IV. NUMERICAL SOLUTION BY QUASILINEARIZATION

COMPUTATION TECHNIQUES

Equations 1 through 230 were programmed in FORTRAN H for solution by an IBM System 360 digital computer. A double-precision integration program (Ref. 12) employing Runge-Kutta starting procedures and a variable step-size difference integration scheme was utilized. The current approximation to the solution, $\underset{\sim}{X}^{(n)}$, as well as $\underset{\sim}{P}$ and the $\underset{\sim}{H}_j$, were generated by integration and stored at fixed tabular intervals. During the integration it was necessary to determine values of the variables between the tabular points. These intermediate values were obtained by Sterling interpolation truncated with second differences. The 98 equations for $\underset{\sim}{P}$ and the $\underset{\sim}{H}_j$ were integrated simultaneously, requiring only one evaluation of g_i and the Jacobian matrix at each value of time (t). After the integration was terminated at the final time (t_F) the combination coefficients, C_j, were evaluated from Equation (93), and the stored values substituted into Equation (94) to obtain the new approximation: $\underset{\sim}{X}^{(n+1)}$.

Computation time was saved by relaxing truncation error requirements where possible. At the beginning of each iteration the best available initial conditions were used for $\underset{\sim}{P}$. The contributions of the $\underset{\sim}{H}_j$ to $\underset{\sim}{X}^{(n+1)}$ therefore diminished as the process converged. Thus, the accuracies of the $\underset{\sim}{H}_j$ were not as important as that of $\underset{\sim}{P}$.

Convergence of the quasilinearization process was tested by comparing the values of $x_i^{(n)}$ and $x_i^{(n+1)}$ at the storage interval according to Equation 95. The entire process described above was performed by a general purpose quasilinearization subroutine which utilized double-precision FORTRAN H[12].

INITIAL CONDITIONS

The extreme sensitivity of the orbital transfer maneuvers considered here was promptly discovered in the early stages of development of this program when solution was attempted without the initial iterate scheme described. Partly because the multipliers were so poorly approximated at that time, small changes in the durations of the thrust periods were found to produce large variations in the trajectory and control variables. Such small variations would often cause the process to diverge. It was therefore necessary to obtain a realistic approximation to the optimal trajectory including the multipliers prior to the initiation of the quasilinearization computations.

It was found that the optimum two-impulse orbital transfer yielded an excellent initial approximation to its finite thrust counterpart. Accordingly, an initial conditions subroutine which utilized a steep descent numerical optimization program (Reference 11) was employed to generate an initial approximation to the trajectory: $\underset{\sim}{X}^{(0)}$. This procedure yielded excellent time histories for the state variables r, θ, ϕ, ρ, x, y, and m. Utilizing the results of pages 34 through 38 and noting the impulses (velocity change) given by the two-impulse program, it was possible to accurately predict the duration of each "burn period." (Note that one first obtains the change

in mass associated with the burn period, and that one must specify a set of rocket parameters; e.g., specific impulse, exhaust velocity, mass flow rate, etc.). In order to generate the initial conditions it was assumed that the impulsive take-off and arrival points occurred at the center of each burning period (see Figure 4).

To avoid computational problems which arise when a thrust initiation or termination point occurs at $t = 0$ or $t = t_F$, a coasting arc of several percent of t_F was assumed to occur at the beginning and end of the final optimal trajectory. Under this assumption it was a straightforward matter to compute exact initial and final values of the first six state variables. An initial approximation to the total time (t_F) required for the maneuver was obtained by summing the transfer times corresponding to the three impulsive coasting arcs. (See Figure 4.)

Although the initial shape of the switching function (k) was unknown the impulsive solution gave an excellent approximation to the "switch points" (i.e., $k = 0$). In prior arguments it was established that only these critical values of the switching function are required for the generation of $\underset{\sim}{X}^{(k+1)}$. One may verify this by referring to Equations 107 to 113 and by noting that the Jacobian matrix contains no partial derivatives with respect to k except at a switching point. Note, however, that Equation 107 also requires an initial guess as to the value of $\dot{k}$ at the switching points. These values of $\dot{k}$ are obtained by differencing the initial iterate for k.

Having determined the time histories of the state variables, the switching times, and the Lagrange multipliers, it would seem that we are ready to turn the computations over to the quasilinearization procedure. As it turns out this is almost true. However, it was found that efficiency and accuracy demanded the development of several numerical "tricks." These refinements, which are necessary to the successful solution of this problem, are presented in the next few sections.

VARIABLE LENGTH STORAGE TABLE

Although straightforward application of quasilinearization will result in a solution of the problems considered here, it was found necessary to employ a number of refinements to assure accuracy and proper convergence. For instance, when storing the tabular values of the solution $\underset{\sim}{X}^{(n)}$ for use in producing $\underset{\sim}{X}^{(n+1)}$ it was necessary to provide a method of maintaining integration accuracy over all portions of the trajectory. That is, it was found necessary to increase the data point storage density during the burning periods. For this reason, the quasilinearization process was constructed about a storage table having variable storage intervals. (See Reference 12.) Figure 4a illustrates this concept. The basic quasilinearization subroutine described in Reference 12 was programmed to integrate over each segment of the table and to stop at the boundaries. At this point a new storage interval would be introduced and the process continued.

SHIFTING STORAGE TABLE

Because the "bang-bang" control problem is inherently discontinuous at the corner points significant numerical problems are encountered. At such points one enters a new flight regime wherein $\dot{\underset{\sim}{X}}^{(n)}$ has entirely different properties. For this reason, the performance of valid numerical interpolation across a switching point is not normally possible.

As the quasilinearization process converges the switching times indicated by the n'th iteration will not coincide with those given by the n+1'st iteration. If the switching point occurs between two table entries it is necessary to perform forward interpolation to arrive at appropriate stopping conditions and backward interpolation to arrive at appropriate numerical values to restart the integration. Since two different flight regimes are involved one finds that the values interpolated from different directions at the switching point, in general, do not agree. In many instances the above problem results in serious numerical errors or convergence failure.

One may employ another "trick" to insure numerical integration accuracy at the corner points. After each iteration of quasilinearization new switching times are determined which, in general, will not occur at the previous tabular values. A new table then may be constructed wherein the new switching times are used to define the boundaries of each variable length storage array. This makes it possible to perform a new iteration of quasilinearization with the assurance that each stopping point coincides

with a table entry. This new table and its relationship with that used in the prior iteration is illustrated in Figure 4b. As the quasilinearization process converges the changes in the switching times (Δt_s) approach zero and the table becomes nearly constant. Note that this method eliminates the necessity of interpolating to obtain the values at the stopping points. This numerical continuity across the corner points was found to be essential for the accurate convergence of the orbital transfers considered here.

SWITCH POINT ANALYSIS

As was pointed out earlier, the "bang-bang" control process produces trajectories which are very sensitive to the initiation, termination and duration of thrusting periods. It was therefore necessary to employ a rather sophisticated process for determining and controlling the switching times to be utilized in the determination of $\underset{\sim}{X}^{(n+1)}$.

A numerical procedure for determining the zeroes of the switching function was employed at the end of each quasilinearization iteration. If the new switching times showed large deviations from the previously used values (those illustrated in Figure 4a) the new times would not be adopted. Instead, the program would shift the thrust initiation and termination times by a small portion of the indicated change.

Newly determined switching times would be fully utilized only when close agreement with the previous iteration had been achieved. In general, such close agreement could only be expected to occur after the quasilinearization process had proceeded through several iterations However,

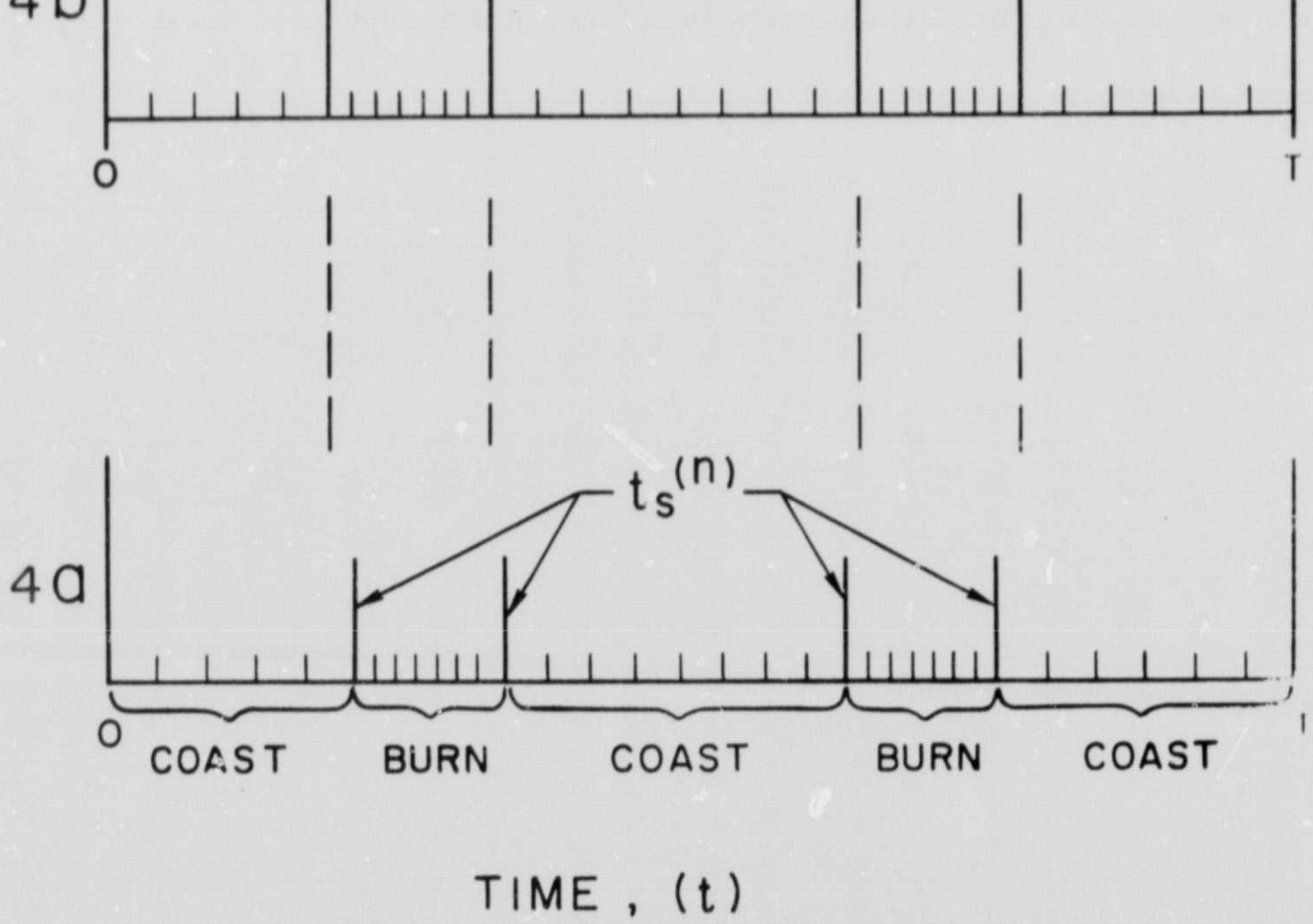


Figure 4. Variable Length Storage Table (4a) and Table With Switching Points, t_s, Shifted for Next Iteration (4b)

once this requirement was met, the program was completely free to use the switching times indicated by the n'th iteration during the computation of the n+1'st. The above programmed constraints forced the solution to conform to impulsive initial conditions until the process had achieved sufficient convergence to adequately control itself. Without this constraint and without the judicious use of the impulsive initial conditions it was usually impossible to obtain convergence.

V. NUMERICAL STUDIES

INTRODUCTION

This section is included to give the reader an impression of the kinds of results one might typically expect to generate using this program. It has been found that the optimum finite-thrust maneuver closely parallels its impulsive counterpart. This fact has been used to considerable advantage in the generation of initial iterates. Figures 5 through 21 aptly demonstrate this last point. Figure 5 is a diagram showing a typical finite-thrust maneuver between two inclined, elliptical, non-coapsidal orbits. The orbit and vehicle parameters are as follows: $p_1 = 5000$ mi, $p_2 = 6000$ mi, $e_1 = e_2 = 0.2$, $\omega_1 = -90°$, $\omega_2 = +30°$, $\beta = 0.001\ m_o$/sec, and initial $F/W = 0.04$. In this diagram, wherein the initial and transfer orbits are projected upon the final orbit's plane, the finite-thrust transfer coincides with its impulsive counterpart when plotted to the scale shown. The small arcs over which the engine is burning are also noted. In this case, they represent a small portion of the total transfer arc.

The benefits of the effort expended upon the previously discussed methods for accurately determining the initial iterate are concisely illustrated in Figures 6 through 21. In these figures, state variables, Lagrange multipliers, or control variables for the converged solution are represented by a solid line. The starting values corresponding to each variable are

	*** FINAL TRANSFER CONDITIONS ***		CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC	
IMPULSE	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
FIRST VECTOR	-0.268284212940 04	0.951074569290 03	0.353476692410 03	0.288829749150 04
SECOND VECTOR	-0.953404750060 03	0.704835692500 03	0.165312465860 04	0.203435373240 04
TOTAL				0.480265122390 04
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT	
P	0.500000000000 04	0.600000000000 04	0.664484936610 04	
E	0.200600000000 00	0.200000000000 00	0.104330532960 00	
I (DEG)	0.500000000000 01	0.0	0.524524587510 01	
R A NODE (DEG)	0.0	0.0	0.755679123660 01	
ARG PER (DEG)	-0.900000000000 02	0.300000000000 02	0.212396622870 02	
A	0.520833333330 04	0.625000000000 04	0.671797356860 04	
PERIOD	0.763696370720 04	0.100390495040 05	0.111874216420 05	
TRAN PT RAD	0.618868173590 04	0.736062363290 04		
PHI-1, PHI-2, P	0.738151617510 02	0.187556791240 03	0.664484936610 04	

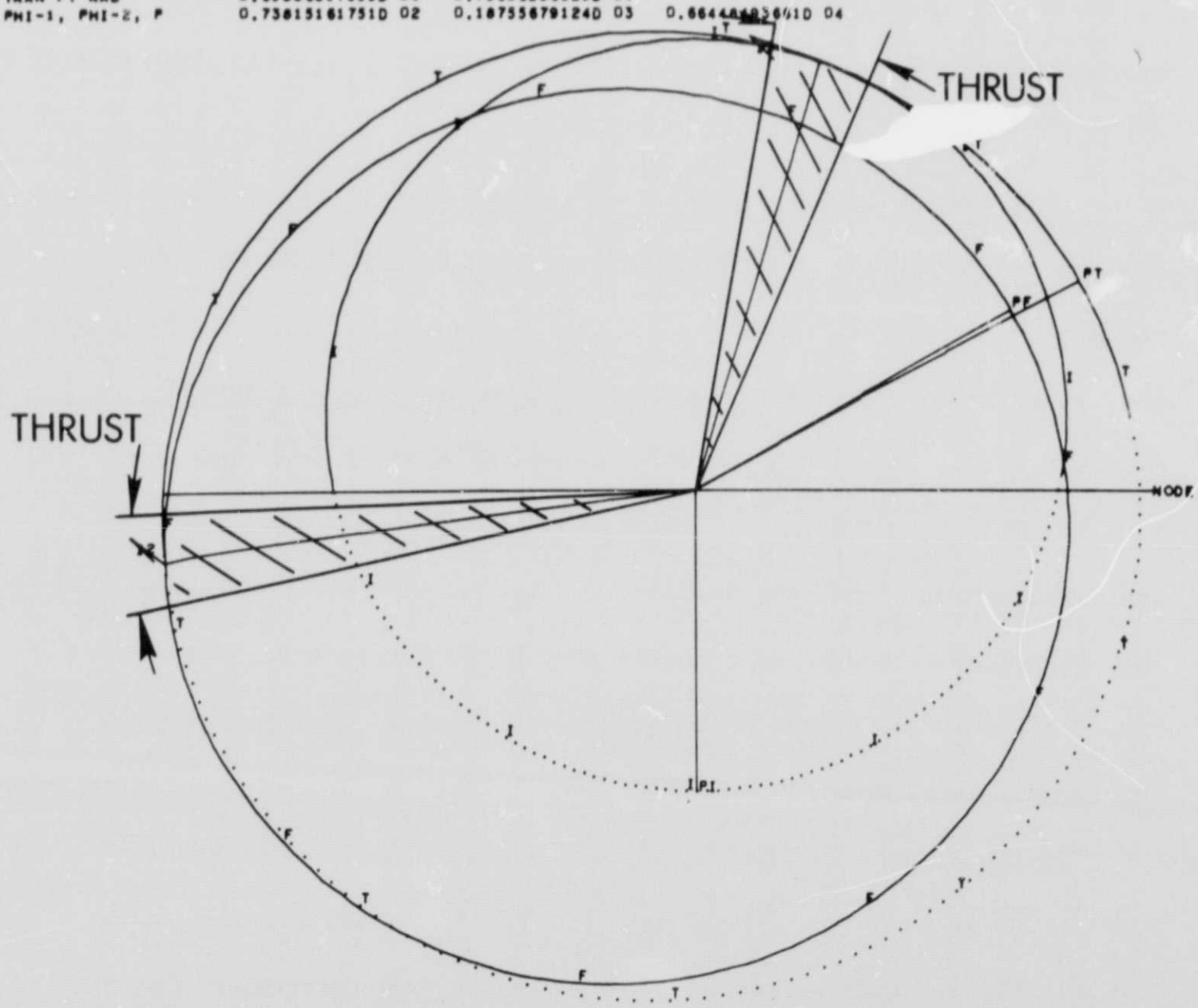


Figure 5. The Orbits (Initial (I) and transfer(T) orbits are projected upon the plane of the final orbit. Dotted portions are below the plane of the final orbit. The thrust arcs showing the finite-thrust maneuver are indicated.)

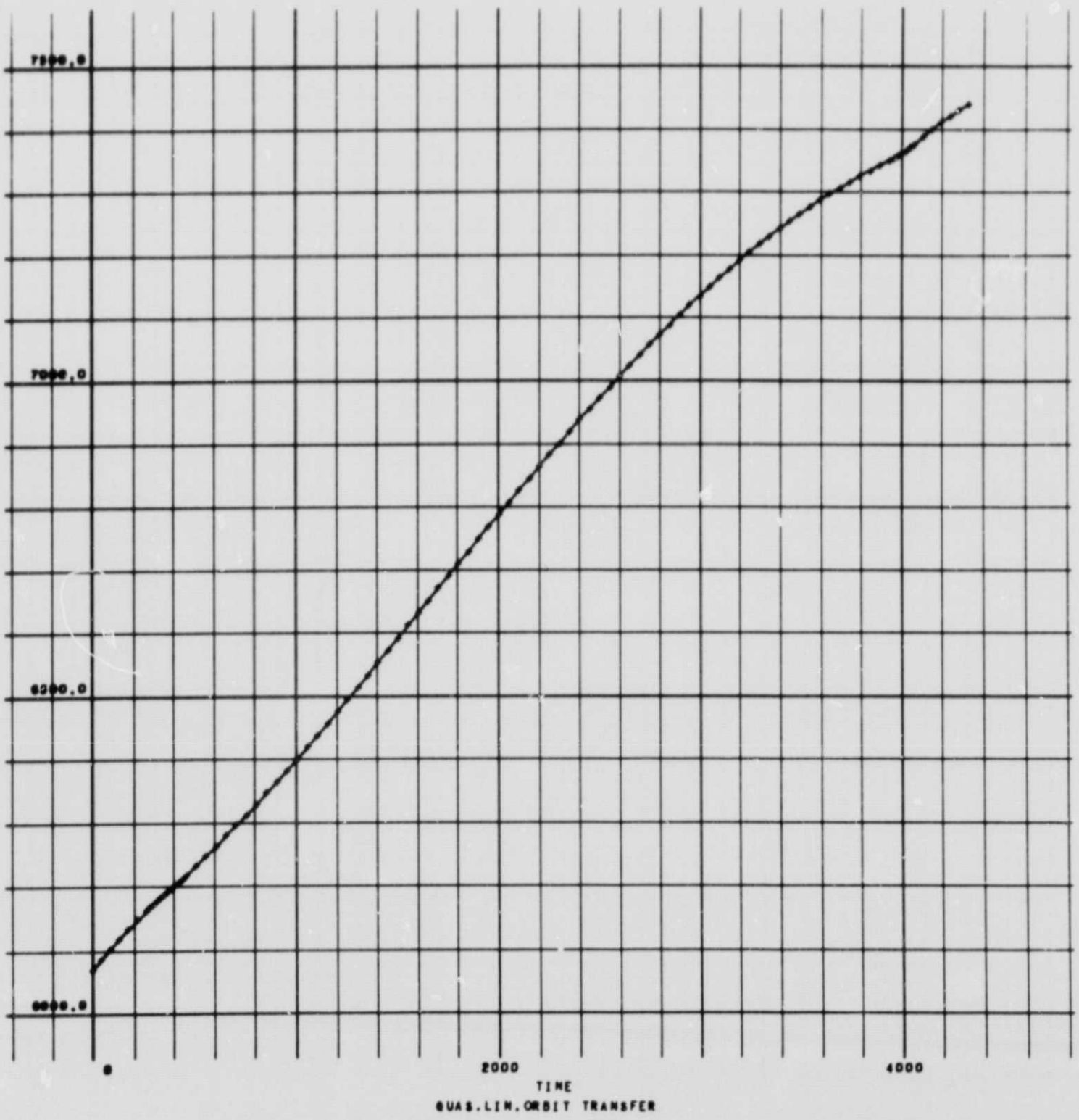


Figure 6. Radius - Converged Time History (Solid Line) Compared to Initial Iterate (+)

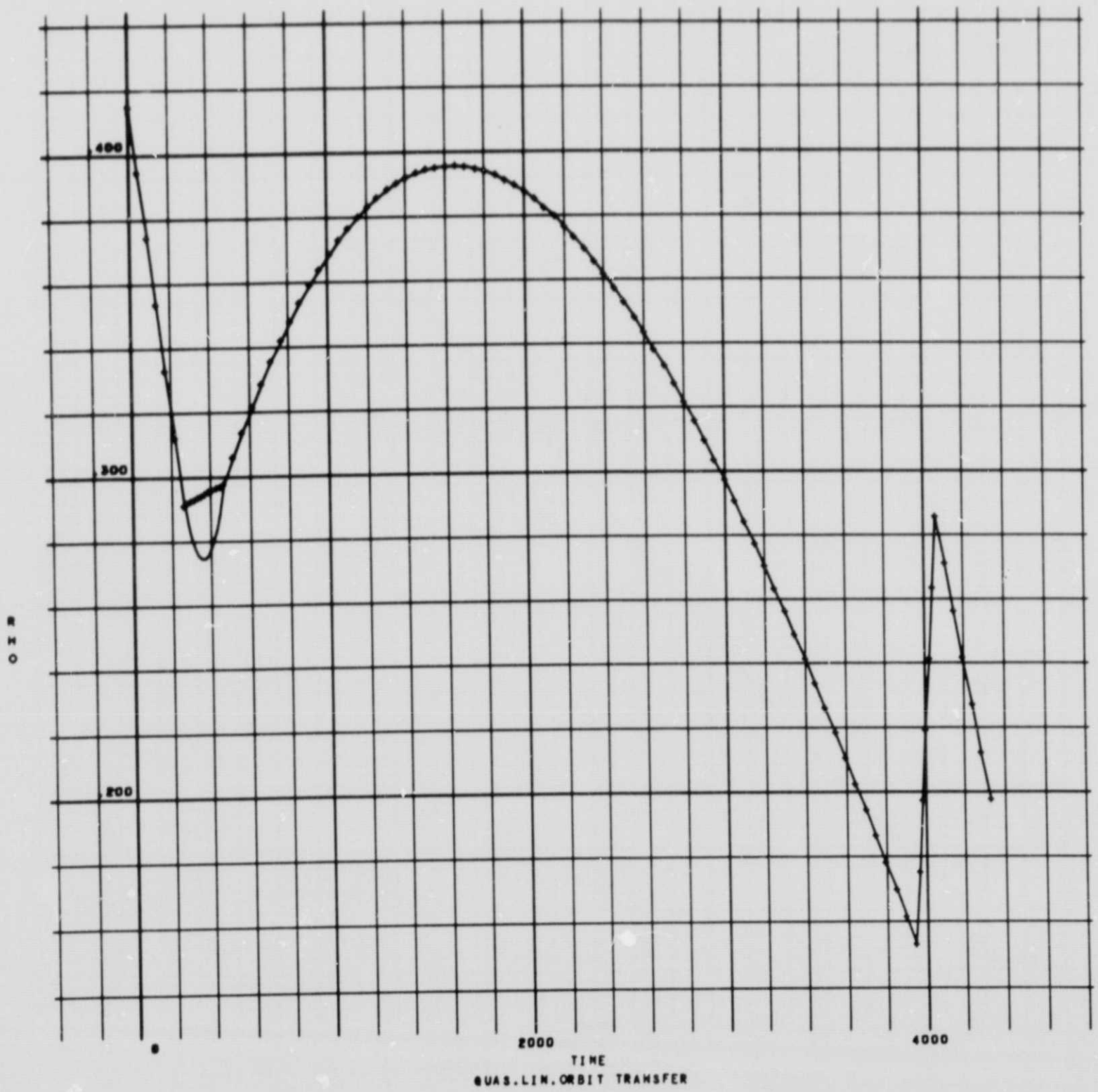


Figure 7. Derivative of Radius - Converged Time History (Solid Line) Compared to Initial Iterate (+)

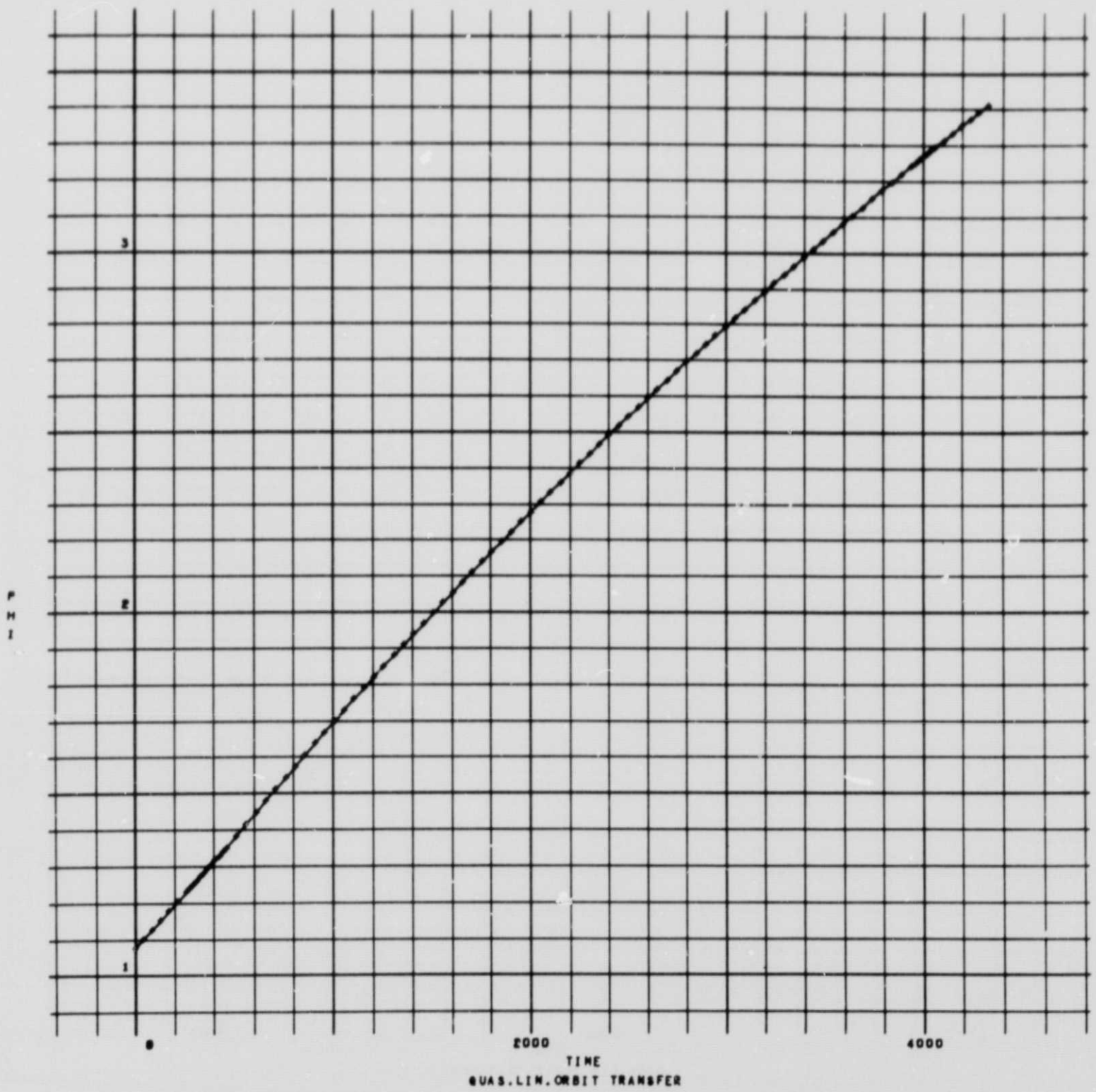


Figure 8. Longitude - Converged Time History (Solid Line) Compared to Initial Iterate (+)

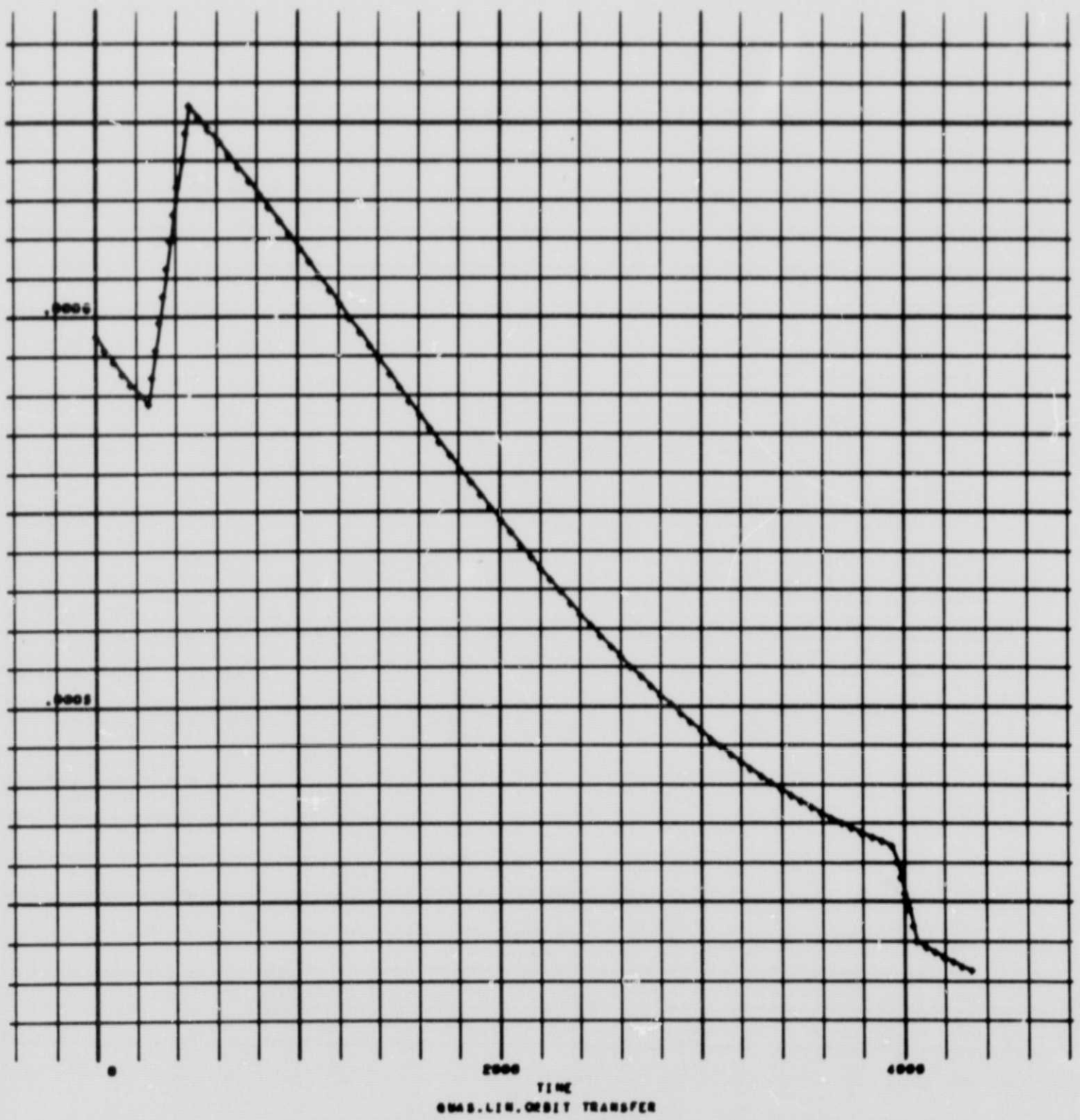


Figure 9. Derivative of Longitude - Converged Time History (Solid Line) Compared to Initial Iterate (+)

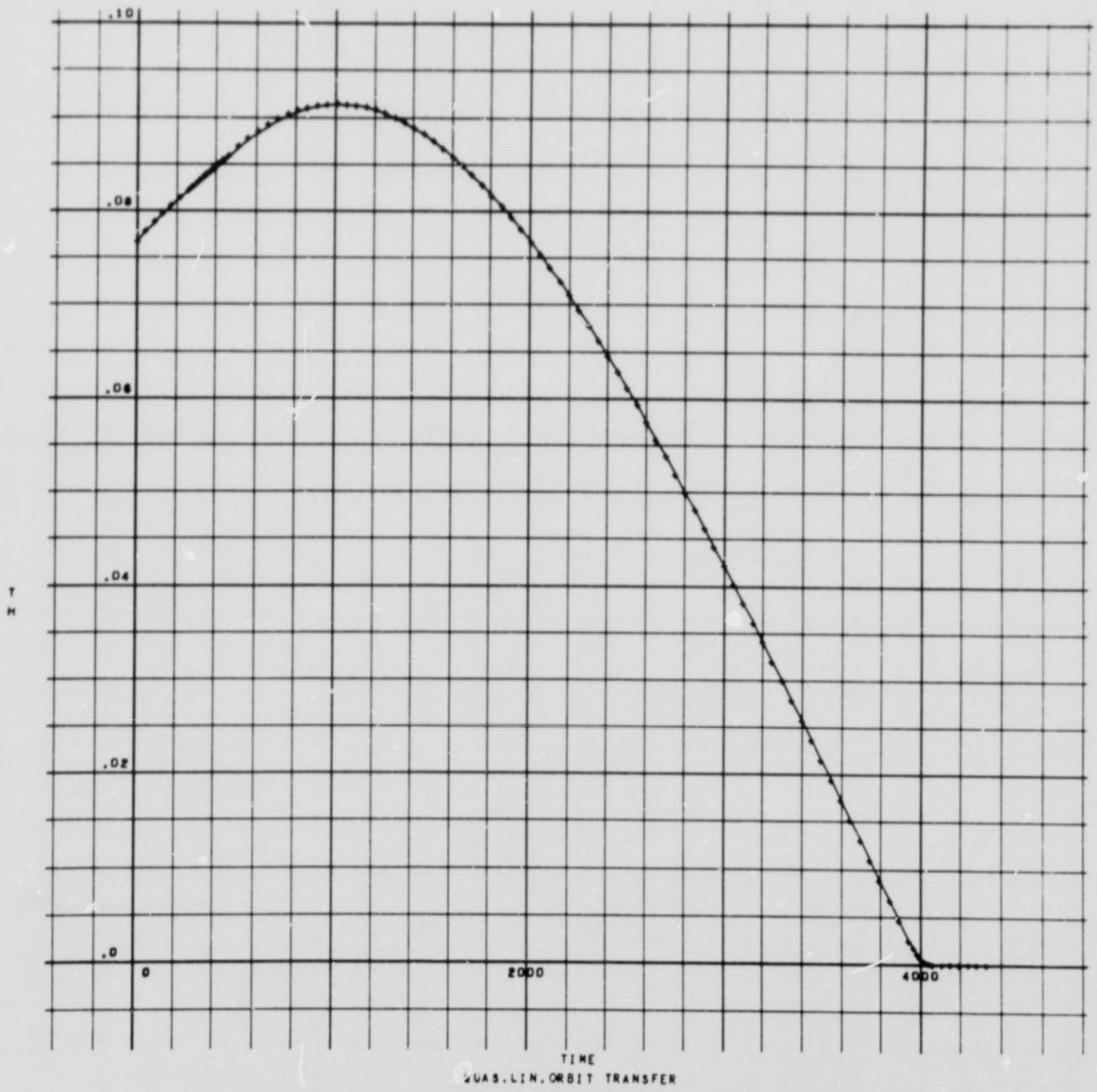


Figure 10. Latitude Angle - Converged Time History (Solid Line) Compared to Initial Iterate (+)

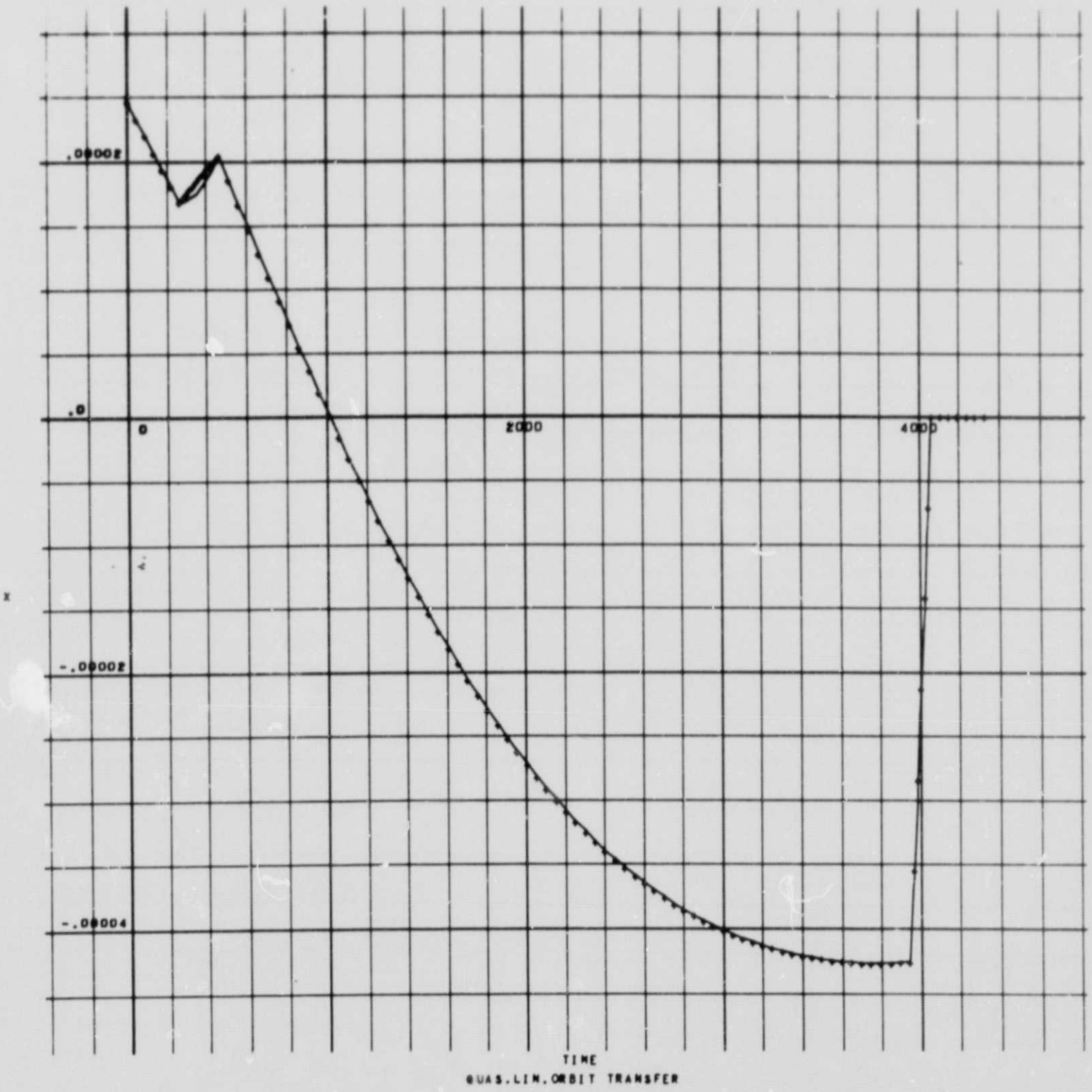


Figure 11. Derivative of Latitude - Converged Time History (Solid Line) Compared to Initial Iterate (+)

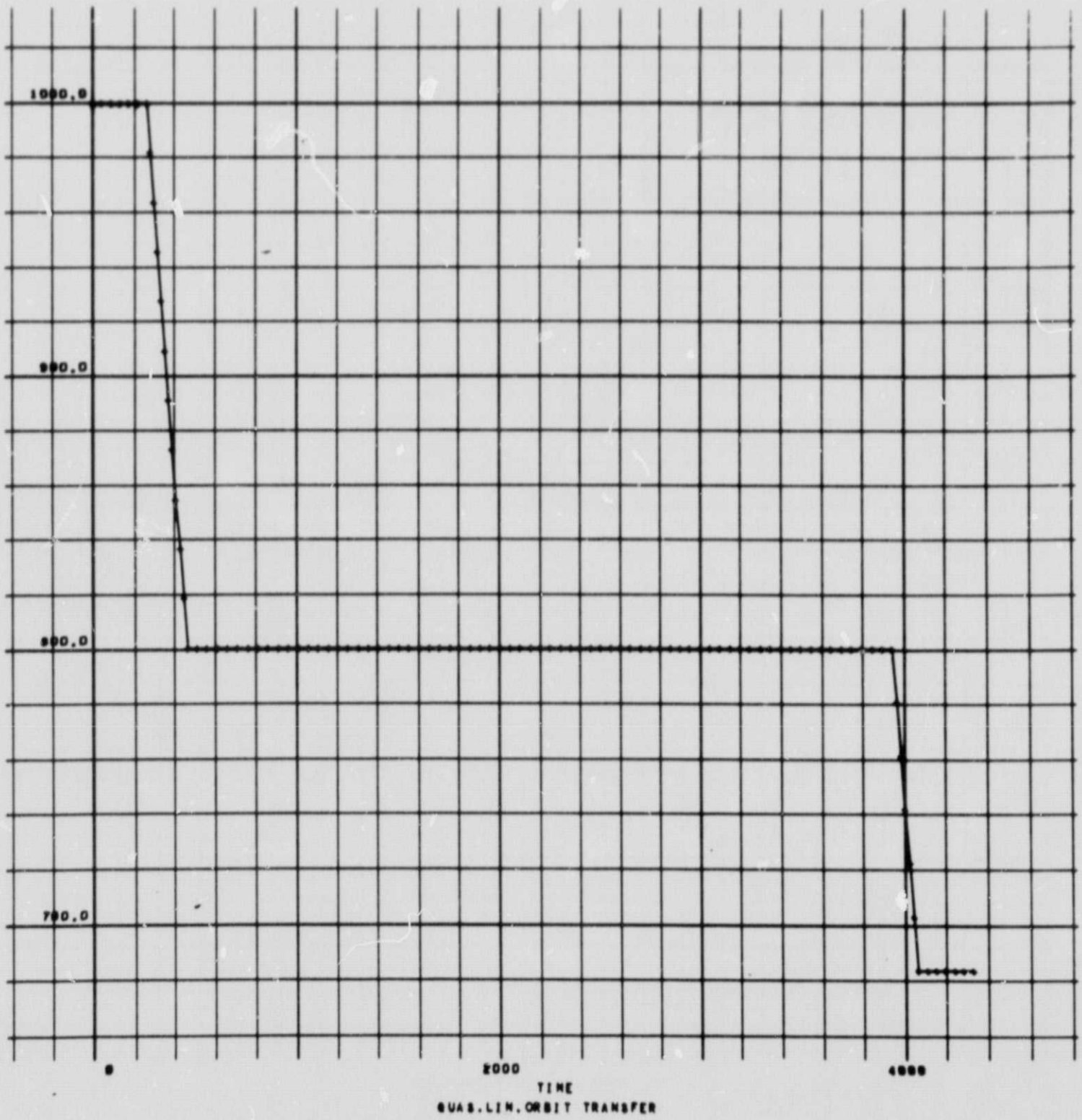


Figure 12. Mass - Converged Time History (Solid Line) Compared to Initial Iterate (+)

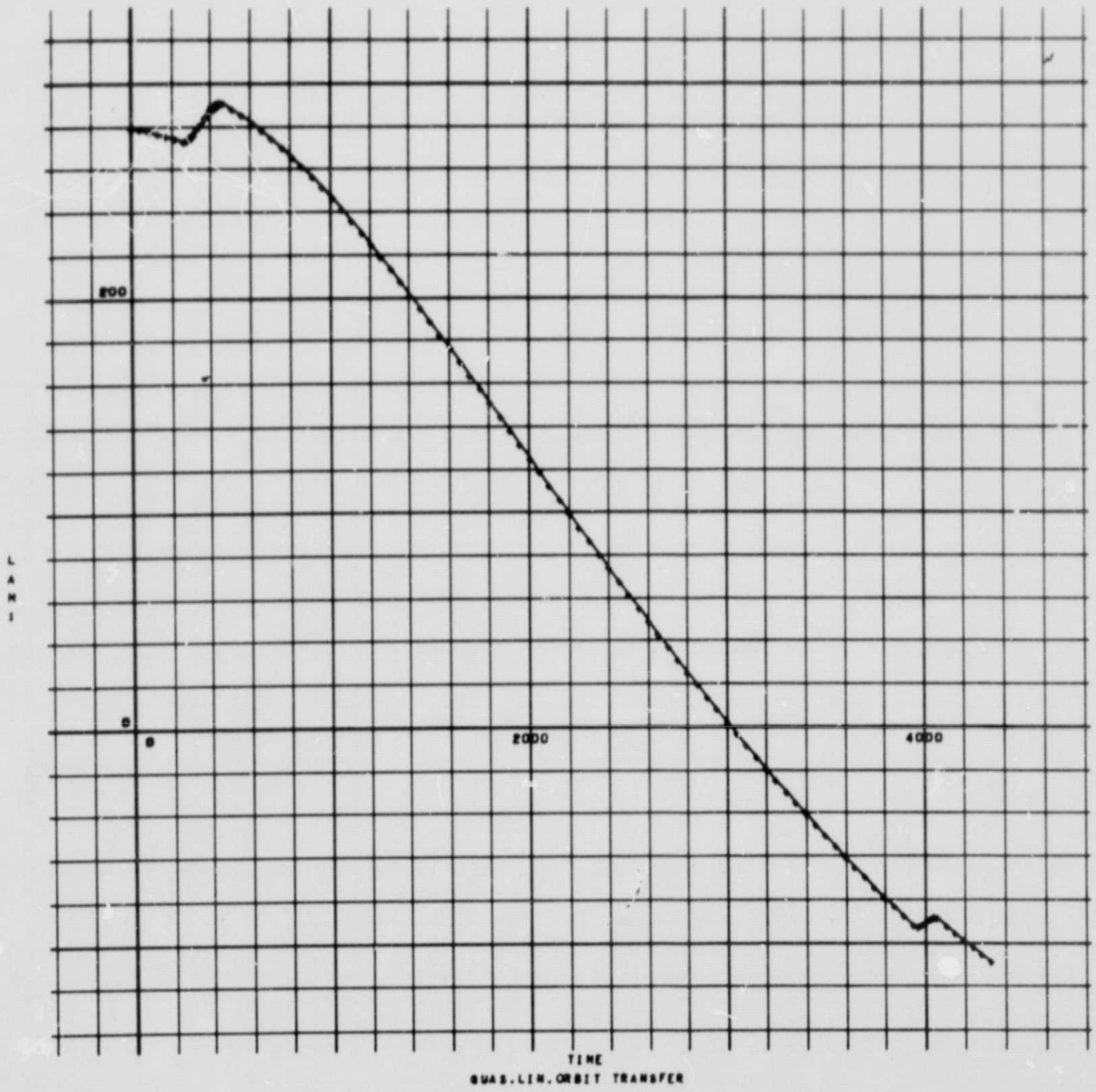


Figure 13. Lagrange Multiplier λ_1 - Converged Time History (Solid Line) Compared to Initial Iterate (+)

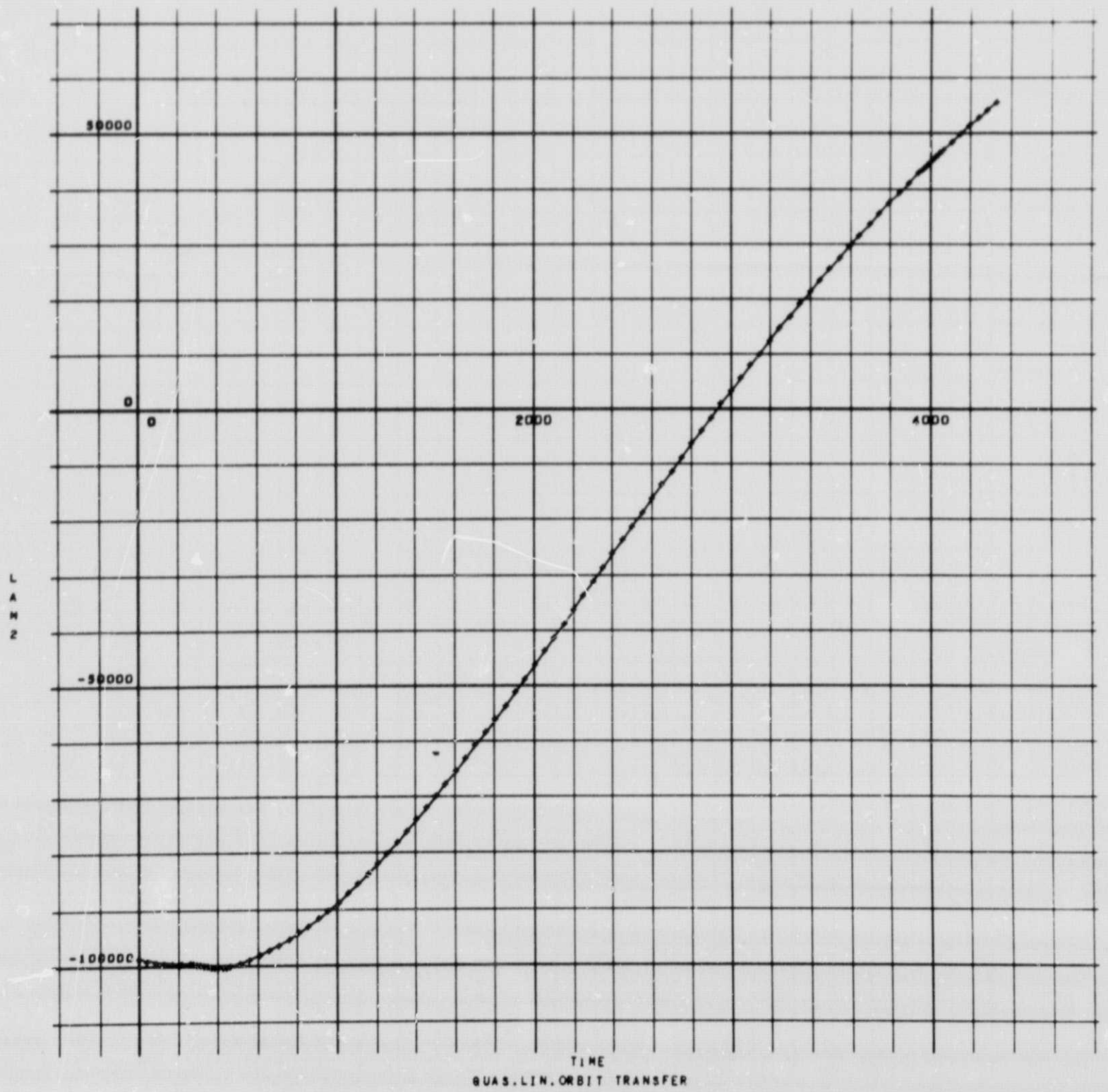


Figure 14. Lagrange Multiplier λ_2 - Converged Time History (Solid Line) Compared to Initial Iterate (+)

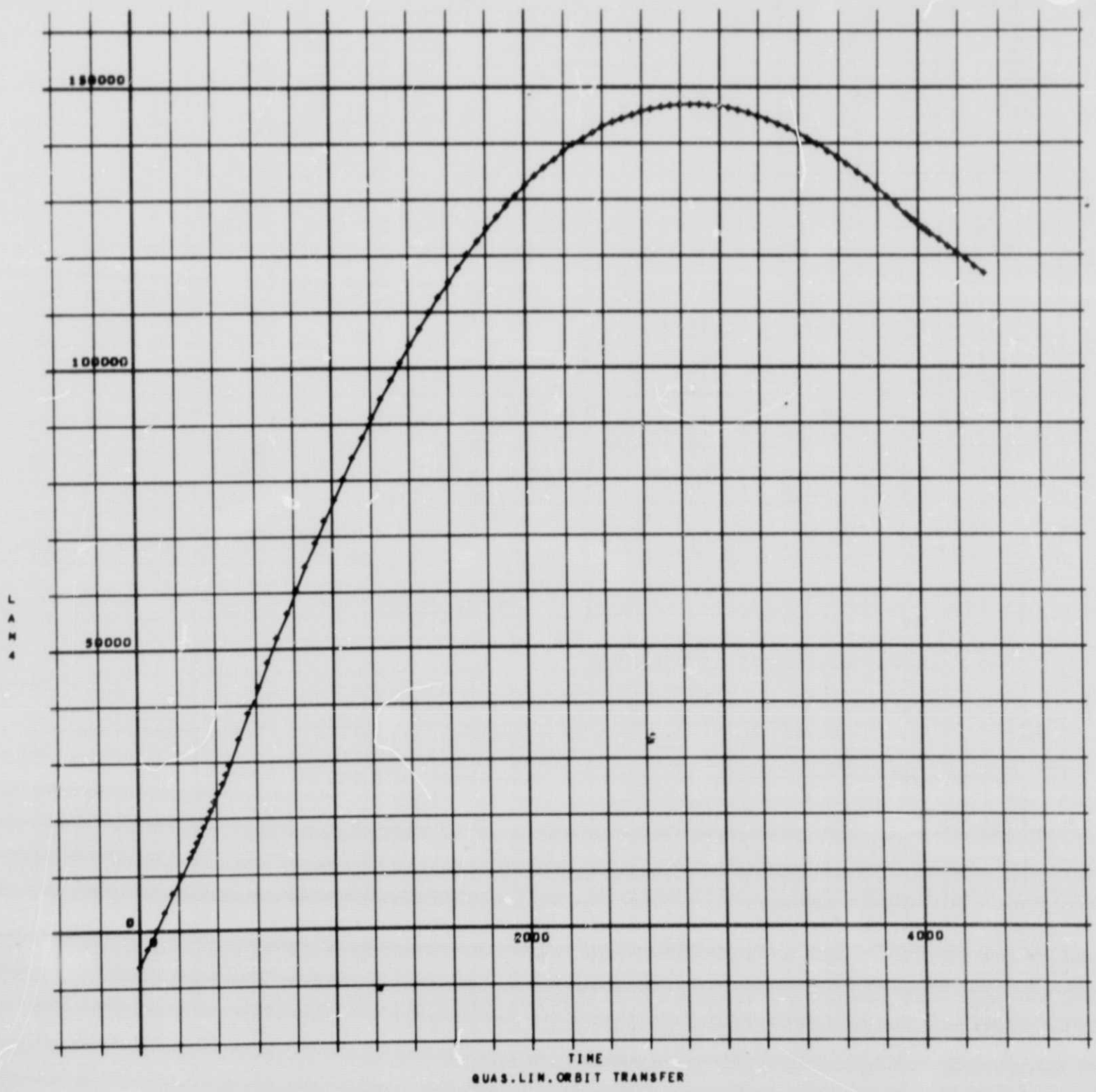


Figure 15. Lagrange Multiplier λ_4 - Converged Time History (Solid Line) Compared to Initial Iterate (+)

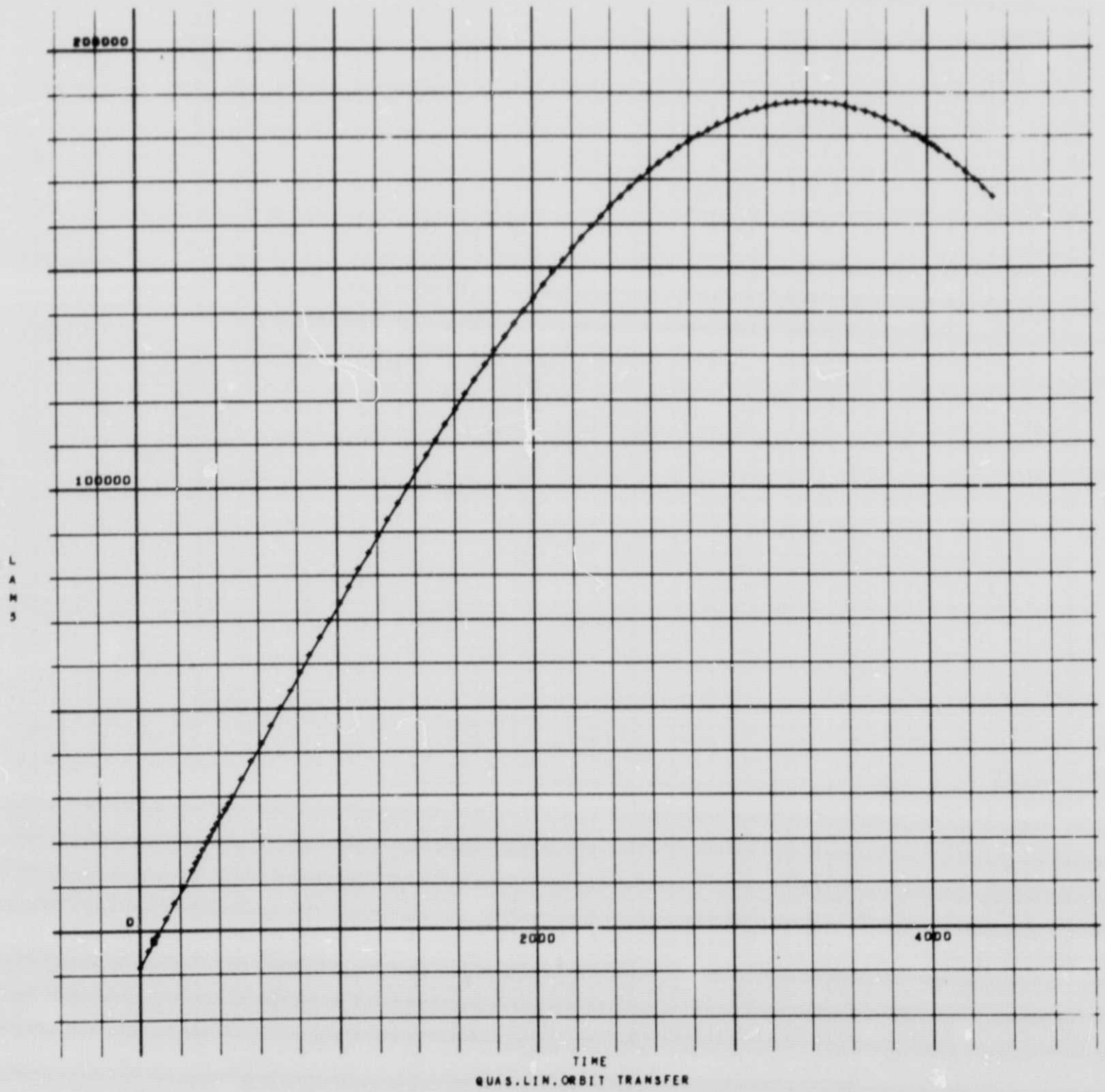


Figure 16. Lagrange Multiplier λ_5 - Converged Time History (Solid Line) Compared to Initial Iterate (+)

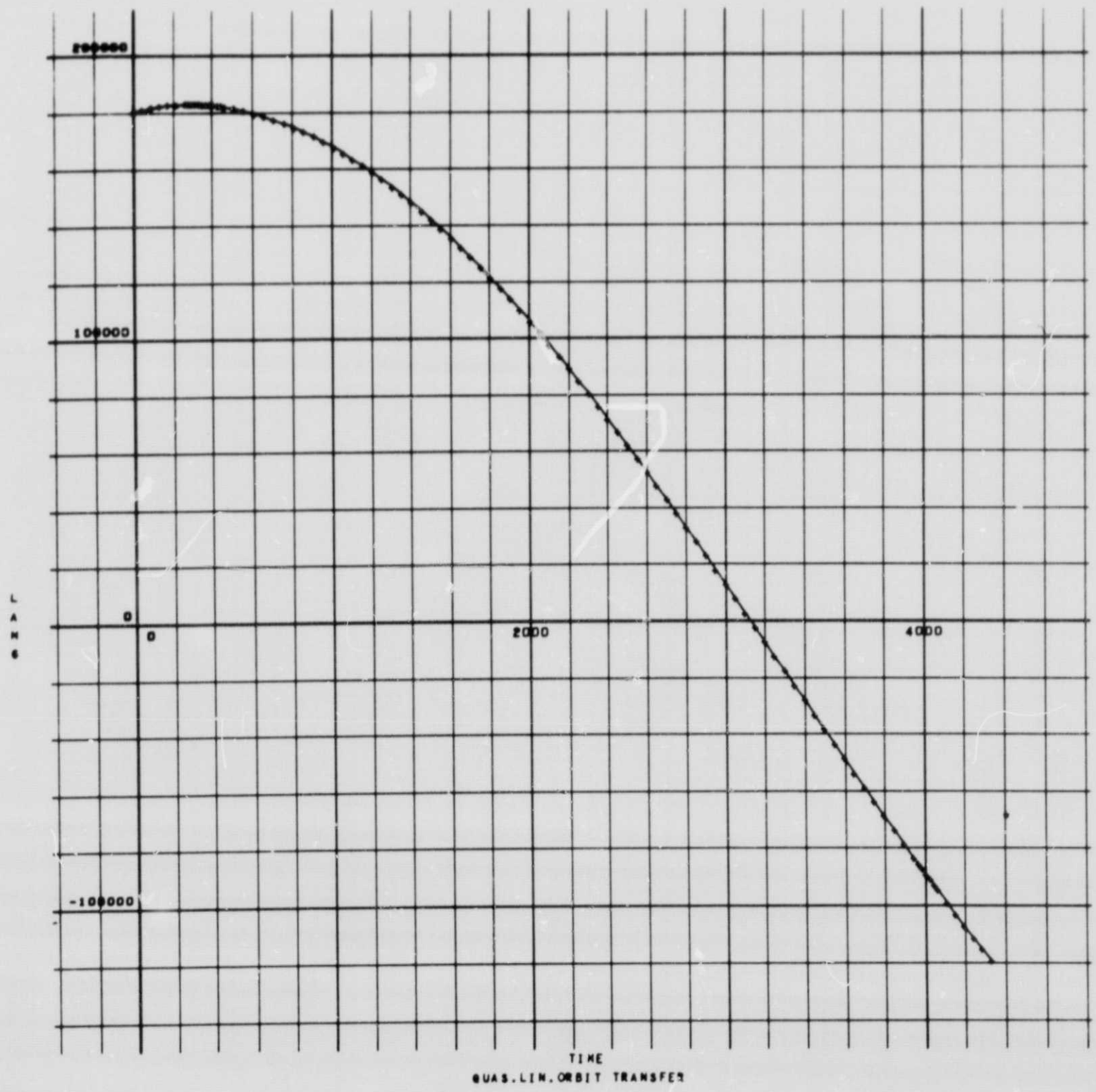


Figure 17. Lagrange Multiplier λ_6 - Converged Time History (Solid Line) Compared to Initial Iterate (+)

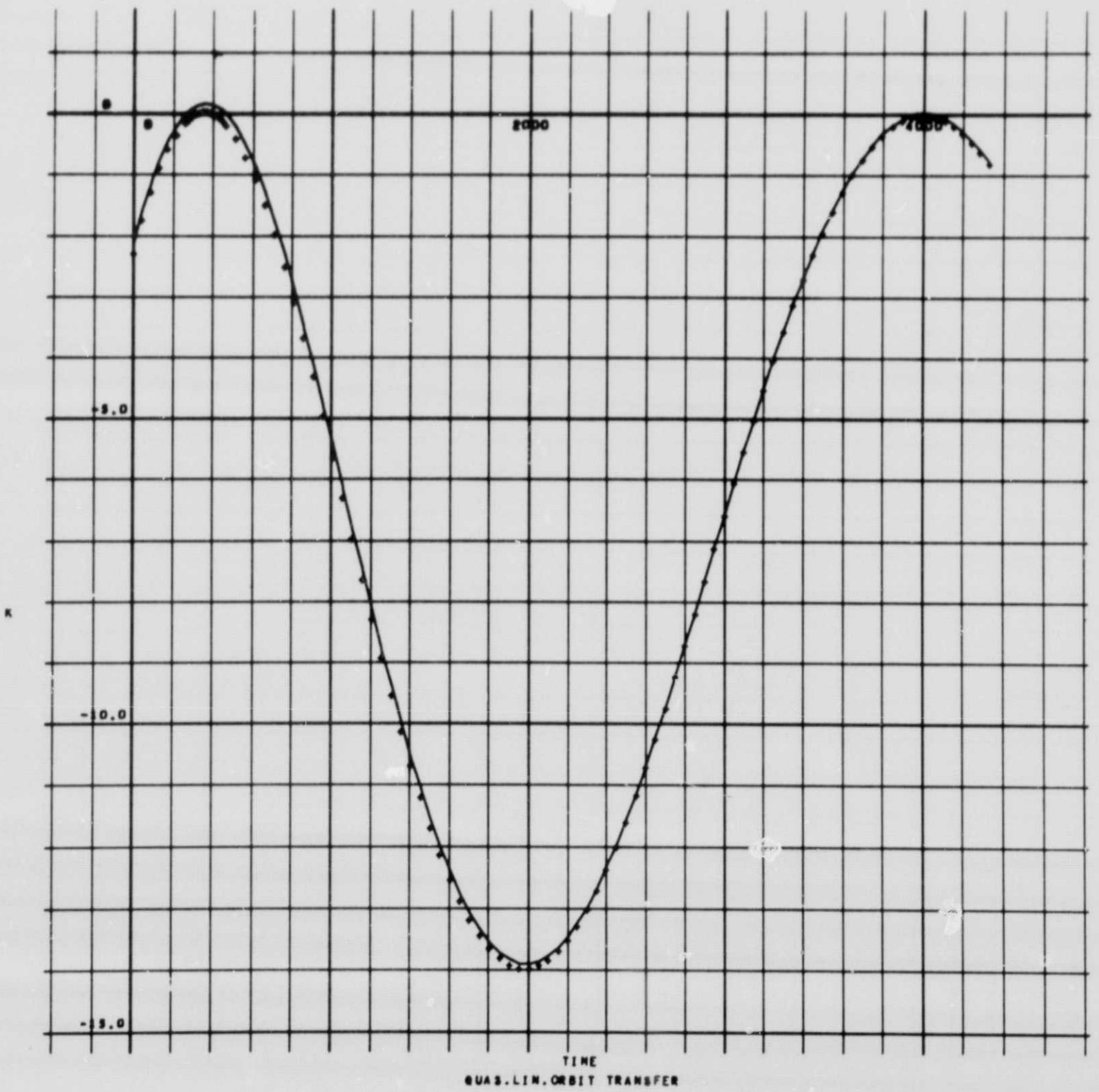


Figure 18. Switching Function - Converged Time History (Solid Line) Compared to Initial Iterate (+)

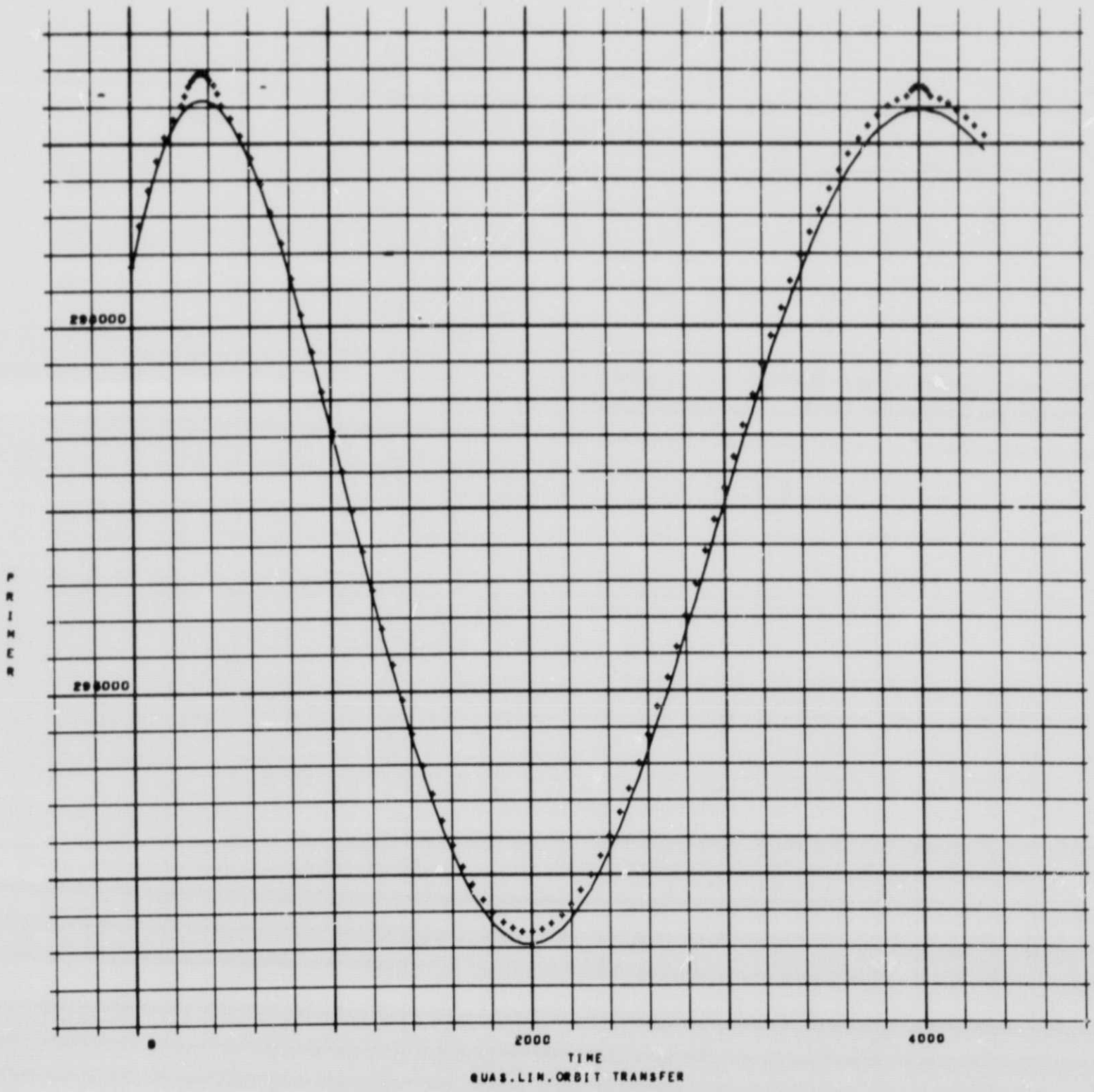


Figure 19. Primer Vector Magnitude (Prime Components are λ_4, λ_5/r, $\lambda_6/r\cos\theta$) - Converged Time History (Solid Line) Compared to Initial Iterate (+)

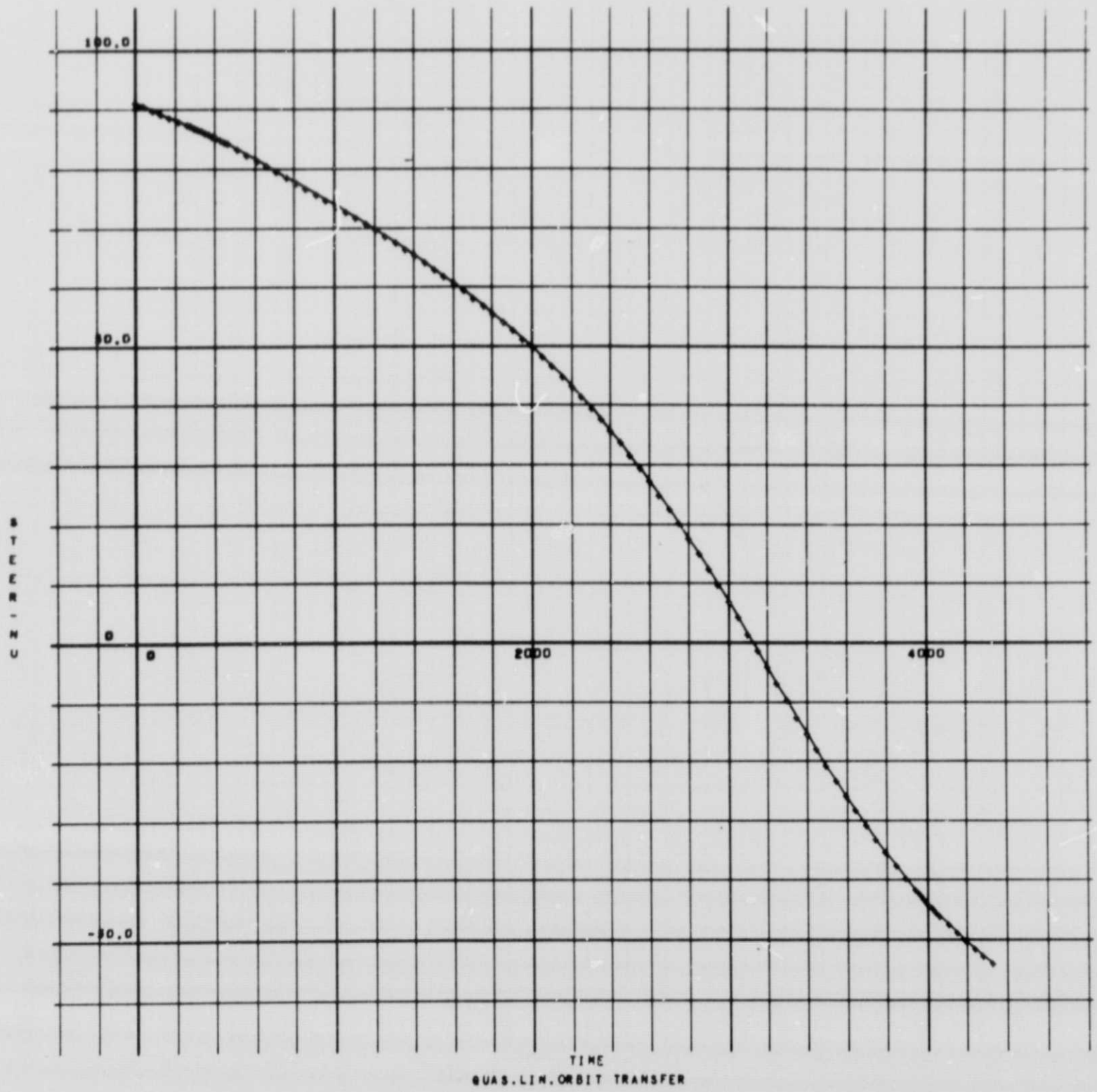


Figure 20. Steering Angle ν - Converged Time History (Solid Line) Compared to Initial Iterate (+)

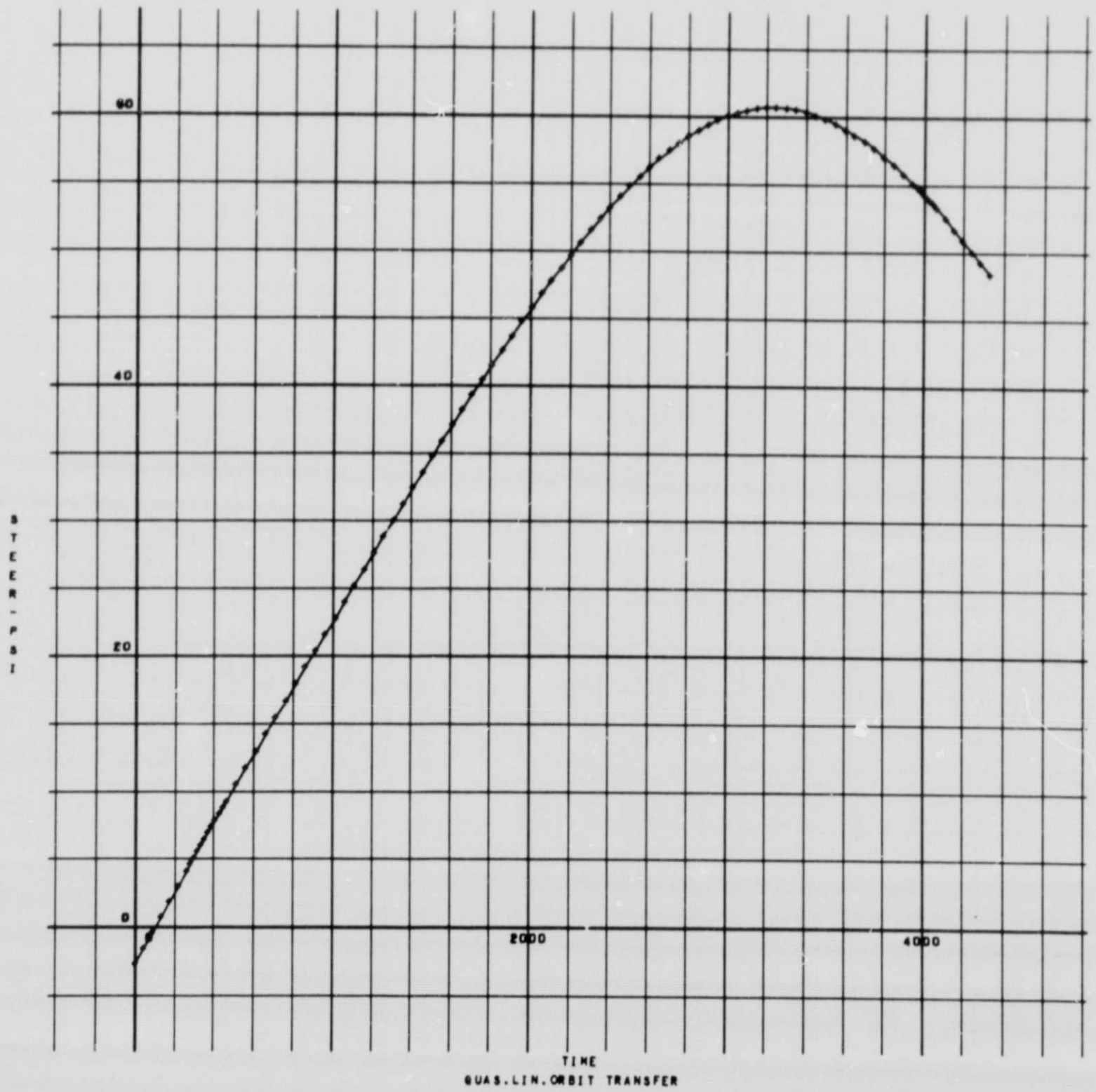


Figure 21. Steering Angle ψ - Converged Time History (Solid Line) Compared to Initial Iterate (+)

represented by a series of "+" signs. The close agreement of the initial iterate and the converged solution is apparent. Note especially the excellent initial iterates obtained for the Lagrange multipliers' time histories. This information is obtained completely from the impulsive solution and in many cases its accuracy means the difference between success and failure. Even the primer vector magnitude, Figure 19, which is particularly sensitive to small changes in its components, shows extremely good agreement with the converged solution. The above results may seem too perfect, but they are typical of what one obtains when running normal cases with moderate thrust levels.

The switching function for this maneuver appears as Figure 18. It is clear that the engine is burning ($k > 0$) for a small portion of the total time required for the orbital transfer. For this intermediate thrust case, the thrust initiation and termination times derived from the impulsive solution differed by less than a few seconds from those indicated by the quasilinearization solution and the total ΔV was only about one percent greater. In order to achieve adequate convergence of the quasilinearization process the switching times must be determined to approximately 0.001 seconds. Referring to Figure 18 and noting that the maneuver may involve several thousand seconds, one obtains an appreciation for the accuracy which must be maintained. Furthermore, when one considers that this accuracy must be maintained during the computations inherent in Equation 41, it is clear that double precision arithmetic is a necessity.

CONVERGENCE AND VALIDITY TESTS

Several tests of a solution's convergence and validity are available. First, because the best approximation to $\underset{\sim}{P}(0)$ is utilized for each iteration, the C_j appearing in Equation 93 should approach zero as the process converges. This provides the first test of convergence. Another test of convergence may be performed by noting successive values of the ξ_i given by Equation 95. During successive iterations the ξ_i should also approach zero. One should also observe the switching times approaching appropriate constant values during successive iterations. Also, since k must be zero at each switch point one should observe appropriate tabular values of k becoming successively smaller with each iteration. All of the above convergence criteria were achieved during the computation of the solutions presented here.

One may compute certain constants of motion as a test of the solution's validity, accuracy, etc. For instance, the Hamiltonian which is given by Equation 34 must remain constant. Of course, it is also possible to compute energy and angular momentum along the coasting arcs and note if these constants are really constant. For all of the results presented here, solutions wherein energy and angular momentum remained constant to more than eight significant figures were achieved. Similar accuracy was achieved for the Hamiltonian. Although this extreme accuracy would not ordinarily be necessary for engineering studies, it was required for the accurate comparison of finite thrust maneuvers and their impulsive counterparts.

As a final test, the converged solution $\underset{\sim}{X}^{(\infty)}(0)$, was utilized as initial conditions for a straightforward integration of the Euler Lagrange differential equations. The solution produced by this method was then compared with that generated by quasilinearization. In all cases this test confirmed the validity of the quasilinearization solution.

MINIMUM FUEL WITH FINAL TIME OPEN

Because the transfer time (t_F) derived from the impulsive solution is slightly nonoptimal additional computations must be performed to determine that trajectory which is time-optimal as well as fuel-optimal. Since the Hamiltonian may be thought of as the partial derivative of final mass with respect to t_F it is necessary to adjust t_F until the Hamiltonian approaches zero. This was accomplished by perturbing t_F and computing an additional fuel-optimal trajectory. This required several additional quasilinearization iterations. The Hamiltonian corresponding to each of the fuel-optimal trajectories was examined and simple linear extrapolation was used to predict a new value of t_F corresponding to A = 0. Thus, it was possible to compute a trajectory for which t_F was "locally" optimal.

ΔV REQUIREMENTS (IMPULSIVE THRUST VS FINITE THRUST)

Numerous optimal trajectories were computed in order to produce a comparison of optimal impulsive transfers and corresponding finite-thrust maneuvers. The results of some prior two-dimensional studies are presented here for illustration. Figure 22 compares the velocity change (ΔV) required for finite-thrust and two-impulse maneuvers over a wide range of

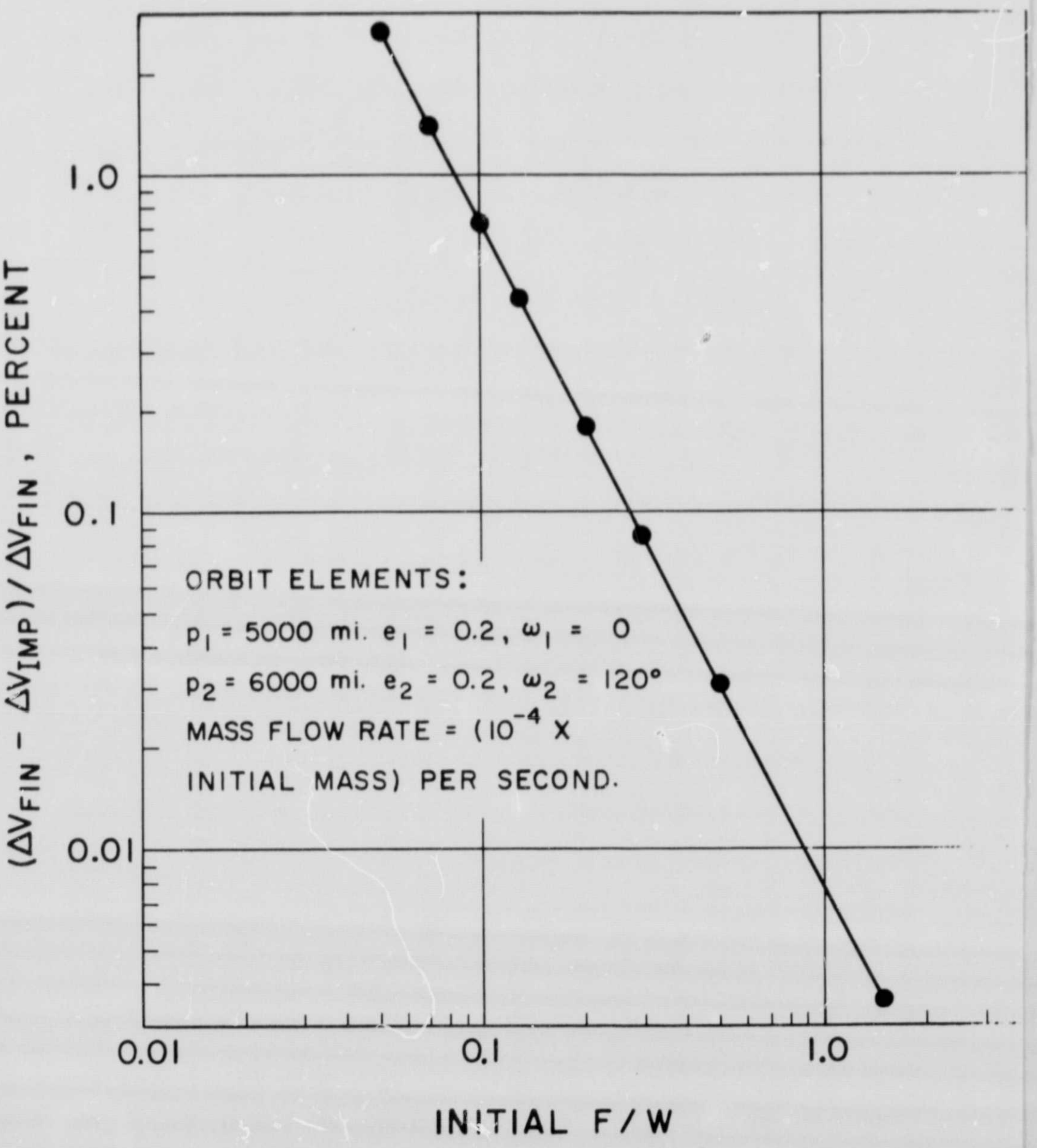


Figure 22. Difference in ΔV Required for Finite - Thrust and Impulsive Transfers Versus Initial Thrust-to-Weight Ratio

initial thrust-to-weight ratios (F/W). It was produced by beginning with the transfer maneuver depicted in Figure 5 and parametrically varying the specific impulse (note that β was held constant). The original log-log plot of this data showed no deviation from a straight line (parabola) over the range shown. The fact that the impulsive orbital transfer is a very close approximation to the finite thrust maneuver is verified by the small percentage differences in Figure 22.

Another interesting comparison was produced by varying the relative perigee angle ($\Delta\omega$) of the two coplanar elliptical orbits of Figure 5. Figure 23 demonstrates that the difference in velocity change required for impulsive and the finite thrust maneuvers exhibits a strong dependence upon $\Delta\omega$. The curve is divided into two regimes corresponding to intersecting and non-intersecting orbits. Near the value of $\Delta\omega$ which corresponds to tangency ($\Delta\omega = 53°.1301$) the curve abruptly, but continuously, changes.

This particular phenomenon is best explained by reference to Figure 24 which contains curves for the two-impulse and finite thrust maneuvers as separate plots. Figure 24 concerns a small range of $\Delta\omega$ over which the orbits are "almost tangent." Since it was known that the class of "almost tangent" orbits produced a number of interesting results (References 7 and 8) considerable effort was devoted to accuracy in examining these critical orientations. Note that the two curves are nearly coincident as long as the orbits do not intersect and that the curves become separate as intersection deepens.

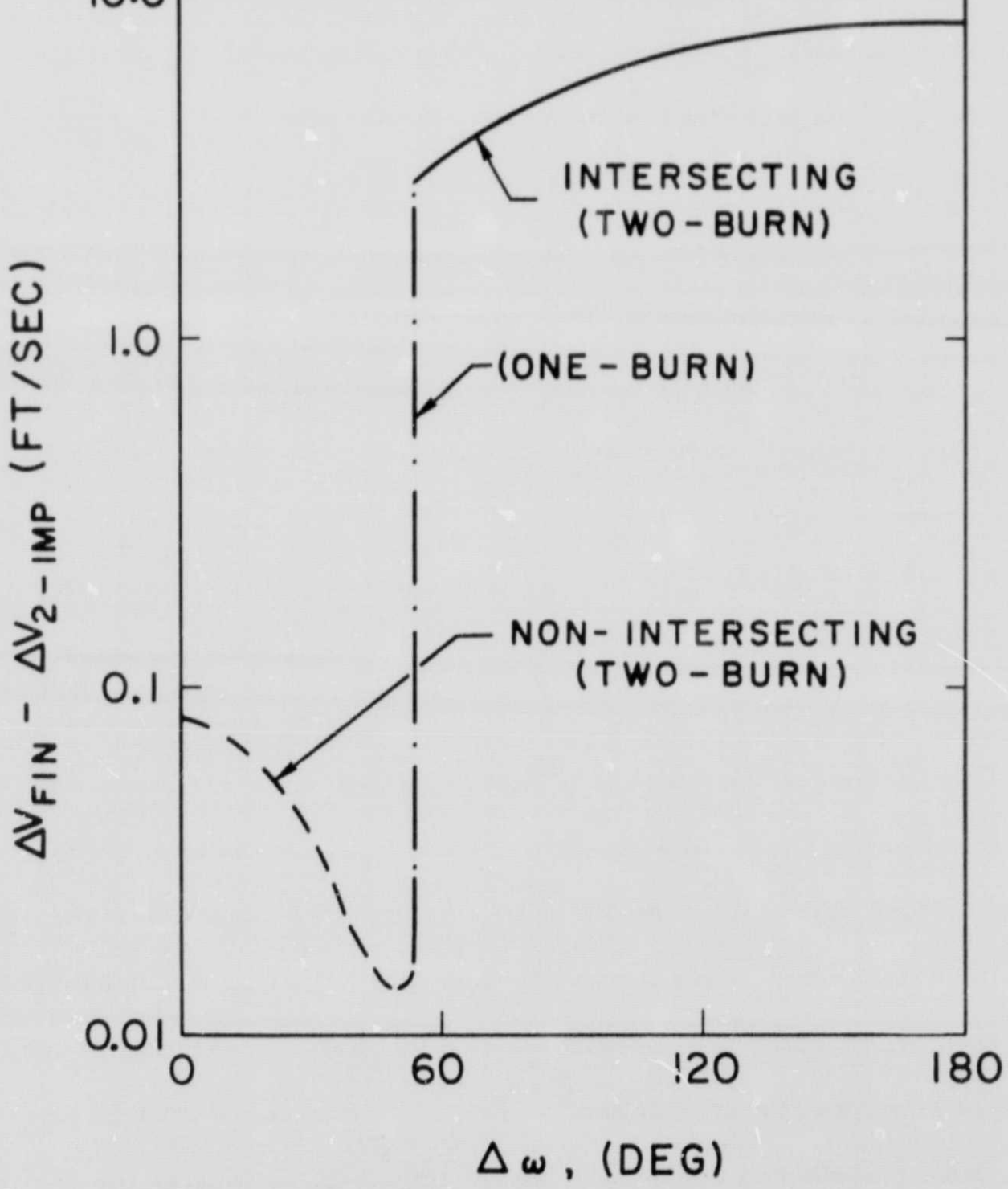


Figure 23. ΔV Difference as a Function of the Relative Perigee Orientation of Two Elliptical Orbits

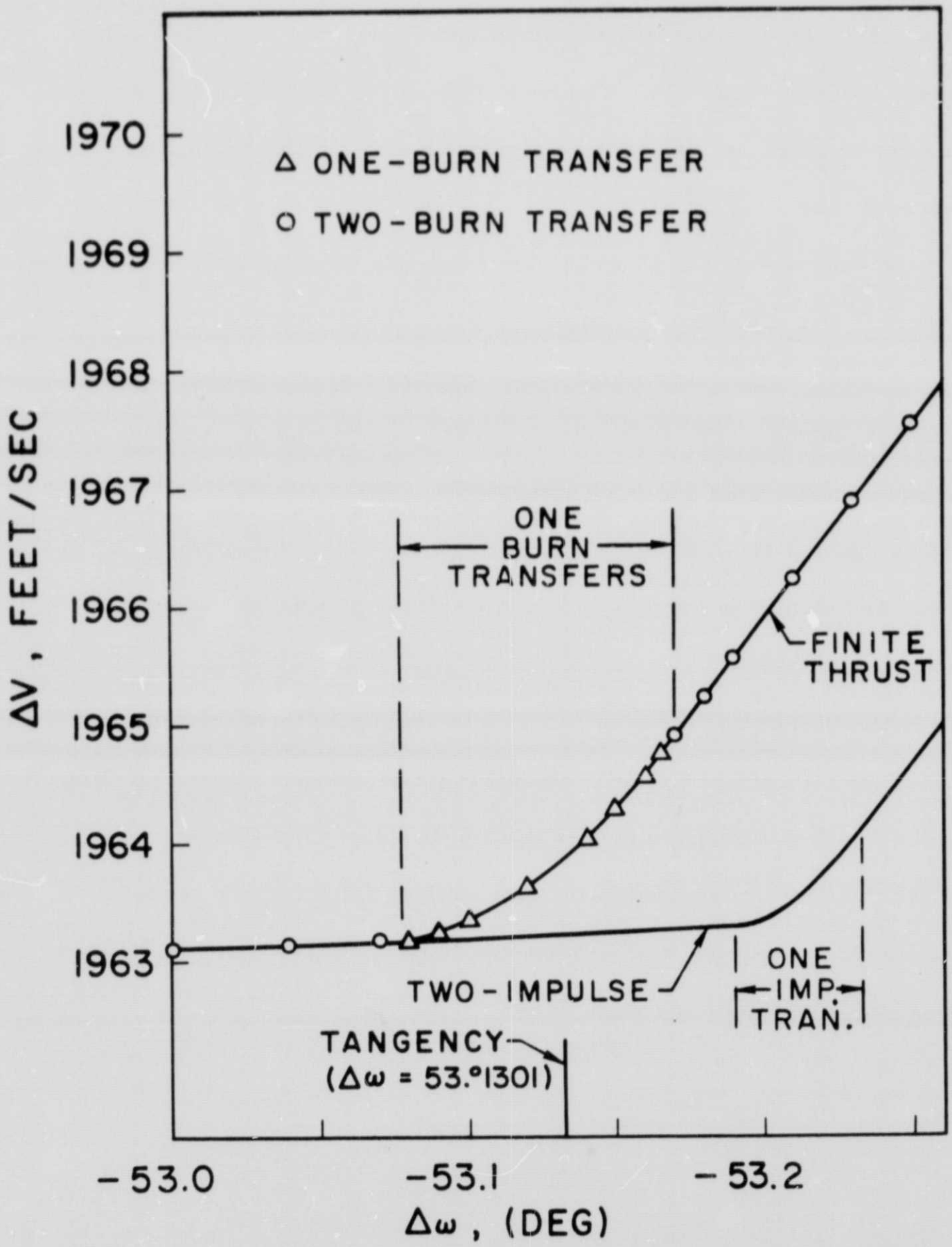


Figure 24. Velocity Change Required for Finite - Thrust and Impulsive Transfers Between "Almost Tangent" Orbits

This sudden diverging of the two curves in Figure 24 explains the abrupt change noted in Figure 23. The magnitude of this difference depends upon the particular rocket parameters employed (e.g., specific impulses, mass flow rate, etc.).

In References 14, 16 and 17 the existence of an optimal one-impulse maneuver was discussed. The two-impulse curve shown in Figure 24 contains a small region over which a one-impulse transfer between the two orbits is optimal. The finite thrust curve is composed of a number of points which are designated one-burn maneuvers or two-burn maneuvers. Note that an optimal one-burn maneuver also exists over a small range of $\Delta\omega$. Thus, one obtains conditions which are again analogous to those found for impulsive transfer.

An attempt was made to extend this analogy to three-dimensions. The procedure was to find an optimum one-impulse transfer between inclined orbits and then attempt to converge upon its finite-thrust counterpart. Unlike the two-dimensional case, this attempt did not prove successful. The question of the existence of optimal one-burn transfers between inclined orbits therefore remains open.

VI. CONCLUSION

This document contains a description of an extremely valuable computer program for performing comprehensive studies of orbital transfer maneuvers. It is now possible, in a single computer run, to obtain complete two-impulse transfer optimization results, the absolutely optimum two-impulse maneuver and a complete finite thrust solution for this absolute optimum.

The numerical results obtained demonstrate that the quasilinearization technique can be a powerful tool for the optimization of "bang-bang" control problems. However, a good first approximation to the solution definitely is required to insure convergence. In general, it appears that impulsive orbital transfer maneuvers provide an excellent approximation to their finite thrust counterparts. For this reason most preliminary engineering design studies could be performed with simple economical impulsive transfer optimization procedures, thus avoiding a more time consuming and difficult finite-thrust optimization.

VII. REFERENCES

1. Jurovics, Stephen A., Orbital Transfer by Optimum Thrust Direction and Duration, North American Aviation, Inc., SID 64-29 (12 February 1964).

2. Kalaba, R., Some Aspects of Quasilinearization, (Nonlinear Differential Equations and Nonlinear Mechanics) New York; Academic Press, Inc., pp. 135-146 (1963).

3. Bellman, R., Kagiwada, H., and Kalaba, R., Quasilinearization, System Identification and Prediction, RAND CORP., RM - 3812 PR (August 1963).

4. McGill, R., and Kenneth, P., "Solution of Variational Problems by Means of a Generalized Newton-Raphson Operator," AIAA J., 2, pp. 1761-1766 (October 1964).

5. McCue, G.A., "Optimum Two-Impulse Orbital Transfer and Rendezvous Between Inclined Elliptical Orbits," AIAA J., 1, pp. 1865-1872 (1963).

6. McCue, G.A., "Optimization and Visualization of Functions," AIAA J., 2, pp. 99-100 (January 1964).

7. McCue, G.A., and Bender, D.F., Optimum Transfers Between Nearly Tangent Orbits, North American Aviation, Inc., SID 64-1097 (May 1, 1965).

8. McCue, G. A., and Bender, D. F., "Numerical Investigation of Minimum Impulse Orbital Transfer," AIAA J., 3, pp. 2328-2334 (December 1965).

9. McCue, G. A., "Quasilinearization Determination of Optimum Finite-Thrust Orbital Transfer," AIAA J., 5, pp 755-763 (1967).

10. Radbill, J. R. and McCue, G. A., Quasilinearization and Nonlinear Problems in Fluid and Orbital Mechanics, New York, American Elsevier Publishing Company, Inc. (1969).

11. McCue, G. A., Bender, D. F., and Morford, J. M., Program for Optimization of Two-Impulse Transfers by Contouring and Steep Descent, North American Rockwell Corporation, SD 69-3 (3 January 1969).

12. Radbill, John R. and McCue, Gary A., Generalized Quasilinearization Routines for Solving Nonlinear Boundary-Value Problems, North American Rockwell Corporation, SD 68-1055 (15 December 1968).

13. Leitman, G., "Variational Problems With Bounded Control Variables," Optimization Techniques, edited by G. Leitman (Academic Press, New York) pp. 171-204 (1962).

14. Lawden, D. F., Optimal Trajectories for Space Navigation, Butterworths Scientific Publications, Ltd., (London, 1963).

15. Lion, P. M., "A Primer on the Primer," Second Compilation of Papers on Trajectory Analysis and Guidance Theory, PM-67-21, Prepared by Contractors for the Guidance Laboratory, NASA Electronics Research Center, Cambridge, Massachusetts, pp. 87-103 (January 1968).

16. Contensou, P., "Etude theorique des trajectories optimales dans un champ de gravitation. Application au cas d'un center d'attraction unique," Astronaut. Acta 8, 134-150 (1963); also Grumman Research Dept. Translation TR-22 translated by P. Kenneth (August 1962).

17. Breakwell, J. V., "Minimum Impulse Transfer," AIAA Preprint 63-416 (1963).

APPENDIX I

COMPUTER PROGRAM DESCRIPTION

APPENDIX I

COMPUTER PROGRAM DESCRIPTION

This appendix presents a description of the various subroutines that comprise the finite-thrust orbital transfer program. Throughout the development of this program the authors drew from their experience with impulsive transfer and with the application of quasilinearization to various engineering problems. For this reason major portions of the complete program were adapted from previous and concurrent projects.

The program can be logically segmented into three distinct parts: (1) the routines that perform contouring and steep descent and which yield an optimum two-impulse transfer, (2) the routines which perform the quasilinearization operations, and (3) the numerous routines that are required for solving this particular finite-thrust problem. Because of this logical segmentation and because of the extreme length of the documentation, it was deemed necessary and desirable to divide the documentation into three separate volumes:

1. "Program for Optimization of Two-Impulse Transfers by Contouring and Steep Descent," SD 69-3.
2. "Generalized Quasilinearization Routines for Solving Nonlinear Boundary Value Problems," SD 68-1055.

3. The present volume which is entitled "Quasilinearization Program for Determining Optimum Finite-Thrust Transfers Between Inclined Orbits," SD 69-4.

Ordinarily the documentation contained in the present volume will suffice for utilizing this program to solve finite-thrust orbital transfer problems. Users wishing to perform studies which require modifications will find it necessary to refer to the first and second reports noted above.

Users will immediately note that the finite-thrust program is really a very comprehensive orbital transfer analysis program which incorporates numerous useful features. It has been written to allow a relatively untrained analyst to input a minimum amount of data concerning his problem and automatically receive a maximum amount of useful information. In the limit, if he wishes to specify only the central body name, the unit system, the vehicle's parameters, and the elements of the two orbits, he may do so. In this case, the program will supply all necessary remaining parameters to perform the optimization. It is also designed so that the output will provide considerable information for the running of additional cases. This is necessary because the results of one case usually suggest a number of additional problems which should be investigated in more detail.

PROGRAM STRUCTURE

The program has been written in the FORTRAN H programming language for the IBM 360 Model 65 digital computer. Of course, many of

the subroutines were originally written in FORTRAN II and FORTRAN IV and therefore reflect some of the restrictions inherent in those earlier languages.

The above mentioned division of the program into three distinct parts is perhaps best illustrated by the diagram presented in Figure 25. This particular figure illustrates the modular tree structure of the main program and its 45 subroutines. Note that those subroutines which are subordinate to subroutine STPCØN are documented in report number SD 69-3 (Reference 11). Also note that the subroutines subordinate to subroutine QASLN are documented in report number SD 68-1055 (Reference 12). The remaining major function is the computation of initial conditions for the complete time histories of the state variables and Lagrange multipliers. All of these computations are supervised by subroutine XFIRST. Note next that the main program also calls subroutines JACØB, ØUTPUT, KHIST, and QPLØT. The first three routines are called primarily for initialization while QPLØT is called in order to provide a graphical summary of the initial iterate and converged solution.

Figure 25 should suffice to illustrate the relationships and lines of communication which tie the complete program together. Tables 1, 2, and 3 provide a brief description of the duties of each individual subroutine module. Table 1 lists the subroutines which are of particular importance for the finite-thrust program, while Tables 2 and 3 list the subroutines subordinate to the two-impulse optimization and quasilinearization packages.

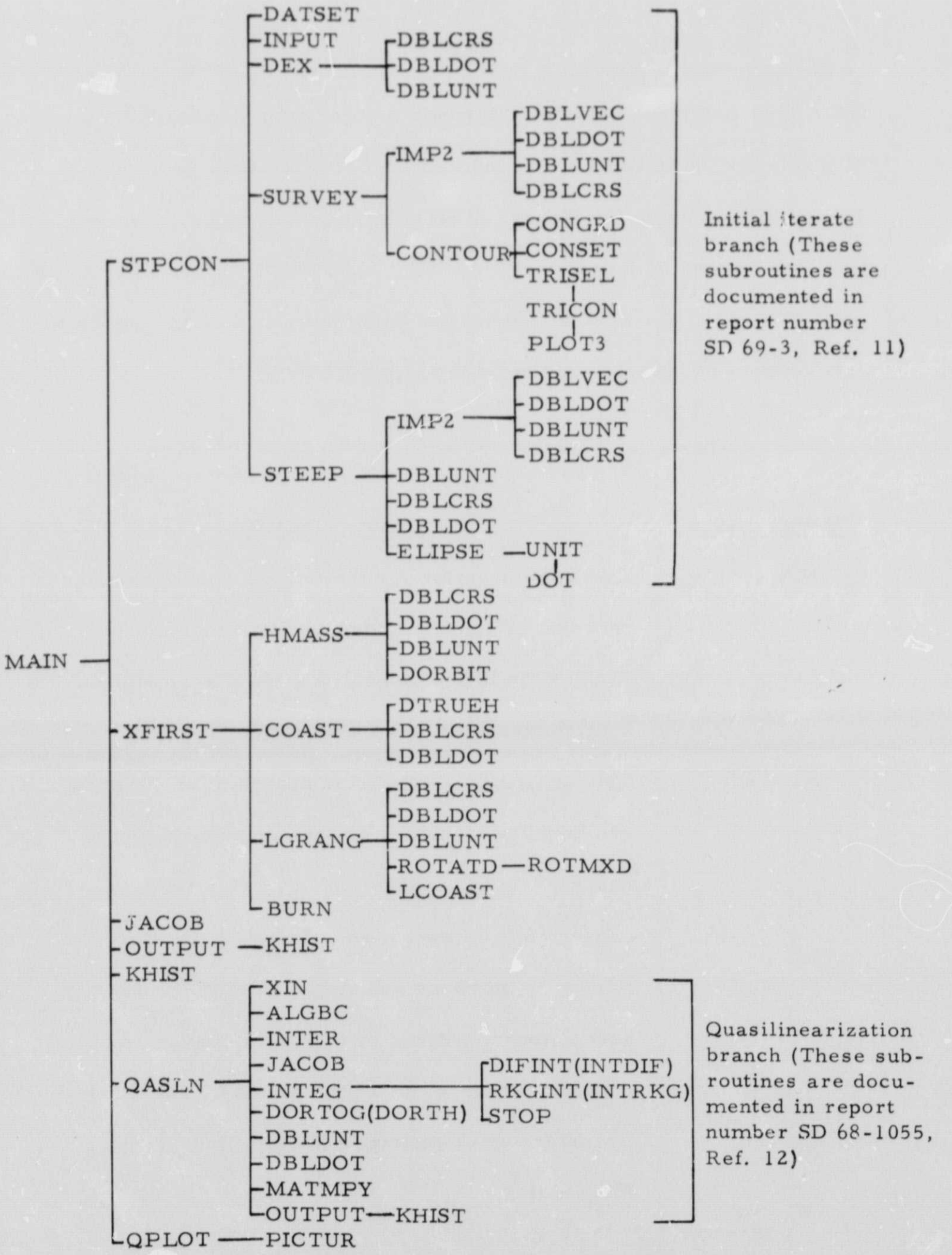


Figure 25. Modular Tree Structure of Two-Impulse Orbital Transfer Optimization Program

If the user requires additional information concerning their functions he may refer to the listings in Appendix III and/or to References 11 and 12. Appendix III contains the listings of all subroutines in Table 1. It also provides listings of subroutines DATSET, JACØB, ØUTPUT, XIN, ALGBC, and STØP which are also documented in References 11 and 12, but which have been modified to solve the finite-thrust problem.

Table 1. Functions of Principal Program Modules

Module Name	Module Function
MAIN	Executive program for the finite-thrust optimizing package. Performs common assignments and handles the logical operations associated with solving a particular optimization problem.
STPCON	This subroutine is the main program for analyzing two-impulse transfers by contouring and steep descent. In the present application it has been converted to an executive subroutine. Documentation of its functions and of the subroutines directly associated with it can be found in Reference 11 entitled "Program for Optimization of Two-Impulse Transfers by Contouring and Steep Descent" (SD 69-3).

Table 1. Functions of Principal Program Modules (Cont)

Module Name	Module Function
XFIRST	This is an executive subroutine for handling the computation of the initial iterate ($\underset{\sim}{X}(t)$).
HMASS	This subroutine computes the burning time histories from the impulsive approximation and determines the sizes and organization of the storage table for the initial iterate. It also determines the perigee passage times for all three orbits.
COAST	This module computes the state variables on the three coasting arcs, in spherical coordinates, from orbital elements and time.
LGRANG	This subroutine performs the computations for estimating the values of the Lagrange multipliers prior to initiating quasilinearization. It also computes the primer vector which is used to compute the initial time histories of the switching function.
LCOAST	This subroutine computes the Lagrange multipliers over the coast arcs from information supplied by subroutine LGRANG.
BURN	This subroutine constructs the initial time histories of all variables over the short burn arcs. It does so by fitting a cubic or linear curve between end points at the beginning and end of each burn.

Table 1. Functions of Principal Program Modules (Cont)

Module Name	Module Function
QASLN	QASLN performs logical operations and computations to implement the quasilinearization algorithm.
ØUTPUT	This subroutine provides standard output at various locations in the computations. It is usually called by the QASLN subroutine.
KHIST	At the end of each quasilinearization iteration it is necessary to examine the switching function and make appropriate adjustments to the thrust initiation and termination times. This subroutine performs these adjustments through the judicious use of control parameters supplied as input data.
QPLØT	This subroutine allows the programmer to select plotted output of the initial iterate and converged solution for each variable. To plot the actual graphs it calls subroutine PICTUR.
PICTUR	This is a standard subroutine for plotting several kinds of data using the SC-4020 CRT output device.
DTRUEH	This double precision subroutine solves Kepler's equation for all types of orbits—given mean anomaly and eccentricity. If E is close to one (ABS(E-1) less than 10^{-10}) a parabolic orbit is assumed. There is

Table 1. Functions of Principal Program Modules (Cont)

Module Name	Module Function
DTRUEH (Cont)	excellent convergence for all elliptical orbits. However, for hyperbolic orbits the program fails if the mean anomaly is too large. The argument list returns true anomaly, eccentricity (or hyperbolic) anomaly, and the sine and cosine of the true anomaly.
RØTATD	Subroutine RØTATD takes a given vector in Cartesian coordinates and rotates it in a coordinate system or rotates the coordinate system. As many as ten rotations are allowed and each one calls RØTMXD (angles in degrees).
RØTMXD	This subroutine provides the matrix for a single axis rotation.
DØRBIT	This subroutine determines mean anomaly in revolutions from true anomaly (in radians) and eccentricity for elliptic orbits.

Table 2. Functions of Important Program Modules That Are Subordinate to Subroutine STPCØN. (Their functions are discussed in detail in report number SD 69-3.)

Module Name	Module Function
INPUT	This subroutine reads and prints initial orbit geometry and other data parameters to be used in impulse computations. It reads and changes only those parameters that are specified on data cards.
DEX	DEX computes orbit geometry information to be used in impulse computations, and performs coordinate transformations.
SURVEY	This module generates an array of the impulse and p_t values which are to be contoured, calls the contouring package, and generates a p-optimized map of the impulse function. It then scans the array of impulse points and identifies possible local minima for further investigation.
STEEP	This is an executive subroutine for conducting an adaptive steep-descent search for a local optimum.
ELIPSE	ELIPSE projects initial and transfer orbits on final orbit plane, plots all three orbits, node line, impulse vectors, perigee lines, radius vectors to the transfer points, and lables the resulting figure.

Table 2. Functions of Important Program Modules That Are Subordinate to Subroutine STPCØN (Cont)

Module Name	Module Function
IMP2	IMP2 performs optional p-optimization and computes impulse and its gradient as a function of $\emptyset_1$, $\emptyset_2$, p_t.
CØNTUR	This is an executive subroutine for the contouring package (see Ref. 11 for further details).
CØNGRD	This module generates grid (if desired) and scaling information for subsequent plotting of contour map. It then plots a contour identification table, a parameter table, and titles.
CØNSET	Subroutine CØNSET computes and/or sets appropriate contour intervals.
TRISEL	This subroutine selects the triangle to be contoured.
TRICØN	This is the routine which performs the actual contouring operation including interpolation to compute the end points of the line segments which are then plotted by PLØT3.
PLØT3	PLØT3 is of routine that plots individual line segments which form contour lines and labels contour lines.
DBLDØT	DBLDØT is a double precision function that computes the dot-product of two vectors.

Table 2. Functions of Important Program Modules That Are Subordinate to Subroutine STPCØN (Cont)

Module Name	Module Function
DBLCRS	This double-precision subroutine takes the cross-product of two vectors.
DBLUNT	DBLUNT is a double-precision function that unitizes a vector.
DBLVEC	This double-precision subroutine sets up a vector by stipulation of its components.
DØT	This single-precision function computes the dot-product of two vectors.
UNIT	UNIT is a single-precision function that unitizes a vector.

Table 3. Functions of Important Program Modules That Are Subordinate to Subroutine QASLN. (Their functions are discussed in report number SD 68-1055)

Module Name	Module Function
XIN	Subroutine XIN reads data and constructs polygonal initial approximation to solution. (This is a dummy subroutine in this program.)
ALGBC	This subroutine computes initial conditions on variables which are dependent on other variables or

Table 3. Functions of Important Program Modules That Are Subordinate to Subroutine QASLN (Cont)

Module Name	Module Function
ALGBC (cont)	parameters and the Jacobian matrix of these conditions. Specifies some initial conditions. (This is mainly a dummy subroutine in this program.)
INTER	INTER interpolates for state vector in table of last solution.
JACØB	This subroutine contains all of the differential equations required by this particular problem. When it is called, it returns the current estimate of the derivative of each variable and evaluates the Jacobian matrix as well. (Described in report number SD 68-1055)
INTEG	Subroutine INTEG controls integration: starting of difference integration with Runge-Kutta integration, stopping at tabular points and special stopping procedure at prescribed points.
INTRKG	Runge-Kutta-Gill Integration: Starts integration for INTDIF.
INTDIF	Difference integration: Builds table integrates with Adams-Moulton formula.

Table 3. Functions of Important Program Modules That Are Subordinate to Subroutine QASLN (Cont)

Module Name	Module Function
STØP	Subroutine STØP carries out special procedure at stopping points. Each time the thrust is initiated or terminated the QASLN program calls subroutine STØP, which provides the small corrections noted in Equations 108 to 113. It also stores the derivatives for later use by subroutine KHIST.
IDETRM	This subroutine evaluates determinant (used in calculating "condition" of homogeneous solution matrix).
DØRTH	This subroutine orthogonalizes solution vectors in homogeneous solution matrix.
MATMPY	MATMPY is a subroutine which performs matrix multiplication
ISIMDD	ISIMDD is a FØRTRAN H library subroutine which solves a set of linear equations

DESCRIPTION OF PROGRAM FUNCTIONS

In a typical application the main program first transfers control to subroutine STPCØN. STPCØN is the subroutine equivalent of the main program described in Ref. 11. It calls five higher level subroutines directly; i.e., DATSET, INPUT, DEX, SURVEY, and STEEP. The DATSET subroutine simply provides reasonable values for all input data parameters which would otherwise be read through the use of subroutine INPUT. The INPUT subroutine functions in such a manner that it is necessary to enter only those data parameters which are being modified for a particular run. That is, a data field may be left blank and the program will not change the previously existing value of the parameter which is in storage. The data cards are each numbered in columns 79 and 80. The number appearing in these columns identifies the data card and thereby tells the program what it should expect to be on that particular card. For instance, card 01 contains the elements of the initial orbit. Because of this procedure, the data cards may be read into the program in any order desired. The program will then read and translate each card individually. A non-zero entry in column 78 instructs the program to terminate the card-reading operation and return to the calling program. The above procedures allow one to process an additional case by changing only one data parameter on one card and by simply adding that data card at the end of the data deck. This process is illustrated further in the sample data sheets presented

later. Once the data are entered, the DEX subroutine performs transformations which return the orbital elements to a standard form. That is, the fundamental plane is transformed to the plane of the final orbit and the reference direction is the line of intersection of the initial and final orbit planes.

If the program has been instructed to produce a contour map of the p-optimized impulse function for the entire function space, the main program will transfer control to subroutine SURVEY. Within SURVEY numerous two-impulse transfers are computed so as to produce a grid of data points for subsequent contouring. Each point is produced by entering the IMP2 subroutine and finding a two-impulse transfer for the given take-off and arrival points such that impulse is minimized with respect to p_t. Once the grid of points is available the contouring package (Reference 11) is instructed to produce a contour map similar to that of Figure 30. Since the entire space is usually searched during this initial optimization, and since one usually desires to know if local minima exist within the survey space; the program next examines each point to determine if it exhibits less impulse than its immediate neighbors. Since such points usually correspond to local minima they are tabulated and ranked for further computations.

At this point in the computations the user has the option of instructing the program to perform a detailed examination of each suspected minima. If desired the "n" suspected minima with least impulse will be examined by first producing a small scale contour map of the region immediately

surrounding the suspected minimum and then performing a steep descent optimization using the suspected minimum as an initial iterate. If no contour maps are desired the user still has the option of conducting a steep descent search beginning at each suspected minimum. The program also draws a picture of the initial, final, and transfer orbits associated with each minimum found by steep descent. Parameter tables giving the user many necessary input and optimized results are included in the final output summary for each optimization.

From the above description it should be clear that the program is extremely versatile. The above quick description will be supplemented in Appendix II by examining the results of running a typical sample problem. The reader should note that considerable attention has been given to providing output which is adequate and meaningful but not superfluous. At this point in the computations the program either picks the steep descent result having the least impulse or performs a steep descent using the search initiation point which is supplied as input. The results of this final steep descent are used to manufacture the initial iterate for the quasi-linearization procedure.

The computations are then transferred to subroutine XFIRST. XFIRST draws upon the computations of subroutines HMASS, CØAST, LGRANG, LCØAST, and BURN in order to piece together the initial iterate. The principal function of HMASS is to determine the time history of the vehicle's mass from the two-impulse transfer and to construct the tabular values of

the independent variable at which the value of the solution will be stored. The reader is referred to Figure 4 which indicates the structure of the storage table. Recall that the program is constructed to include a coasting arc prior to the first burn and at the end of the second burn.

The initial iterate for the Lagrange multipliers is then computed along the coast arcs within the subroutines LGRANG and LCOAST which make use of Equations 178-229. This procedure is outlined in Figure 3. For each state variable and multiplier the values along the burn arcs are computed by the subroutine BURN. Finally the values of the steering angles, the switching function, and the primer vector length are computed along the complete trajectory.

At this point the main program performs a number of computations which are required to initialize and to set up the quasilinearization procedure. Note that subroutine QASLN contains variable dimensions and that it is necessary to provide a number of storage tables and parameters which it will utilize. Depending upon whether a two-dimensional or three-dimensional case is to be solved the main program takes one of two major branches and transposes the data provided by XFIRST.

At this point subroutine QASLN is called upon to solve the complete set of differental equations. The important subroutines of the QASLN package are thoroughly described in Reference 12 along with their functions and interrelationships. Because some minor modifications were performed upon QASLN to adapt it to this particular problem a current listing is

included in Appendix III. Note that these changes were mainly in procedure and in formatting, and that they do not affect the overall functioning of the subroutine which is documented in Reference 12.

As noted above, subroutine QASLN is called upon to determine the conditions for producing an optimum transfer between the two given orbits in a fixed total time interval (t_F). Therefore when it is called it will perform a number of iterations in order to converge upon a fixed time solution—three to six iterations are usually sufficient. When it exceeds the given number of iterations or if it converges upon a solution, QASLN will transfer control back to the main program. If it is desired to minimize the fuel required with respect to the total time for the maneuver the main program will add a time increment to the total time and transfer control back to QASLN which must then perform another set of iterations. Upon completing this second set of iterations, the main program has information for numerically estimating the partial derivative of ΔV with respect to total time. It makes use of the fact that the Hamiltonian, which is the partial derivative of the payoff function with respect to total time, must be zero if the maneuver is to be time optimal as well as fuel optimal. The two sets of iterations now provide information for extrapolating the total time so as to attempt to reduce the Hamiltonian to zero. The program therefore estimates a new total time and transfers control back to QASLN to perform another set of iterations. This cycling procedure is repeated as many

times as is specified by input data parameter IQCYCL. Four of these cycles are usually sufficient to reduce the Hamiltonian to essentially zero, thereby minimizing the fuel with respect to total time.

During the last call to the quasilinearization procedure the user has the option of requesting that QASLN compute the complete trajectory using the initial conditions which it has determined and bypassing the quasilinearization equations. That is, QASLN will do a straightforward integration of the differential equations similar to that performed by standard numerical integration programs. QASLN will then compare this integrated solution with the solution obtained through the use of the quasilinearization algorithm. That is, it will evaluate an equation similar to Equation 95 and write out the maximum difference between the two solutions for each variable. This procedure provides an independent check of the accuracy and convergence of the quasilinearization solution.

As a final step in the computations, the program can be instructed to call subroutine QPLØT which will provide CRT output of the initial iterate and the converged solution on the same graph. The graphs are similar to Figures 6 through 21. The program then returns control to subroutine STPCØN which will attempt to read and process a new case.

DATA STRUCTURE

Because of the finite-thrust program's extreme complexity and versatility it requires a number of user supplied control parameters. The data are perhaps best explained by referring to Figure 26 which contains a

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 1 of 8 JOB NO. _____

	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1	TRANSFER		
13	N EARTH MOON		
25	T		TITLE: Case title information
37			
49		73 80	
61		TITLE	
1	EARTH		BODY: Name of body (e.g., EARTH, MOON, SUN, ...) (left justified)
13	B O I T I C N		UNITS: Unit system ("ABSOLUTE", "METERS", "KILOMETERS", or "BODY") (left justified)
25			GMU: Gravitation constant used with "BODY" specification
37			
49		73 80	
61		BODY UNITS	
1	5 0 0 0 . 0		a
13	0 . 2		e
25	0 . 5		i Elements of initial orbit
37	0 . 0		Ω
49	- 0 0 . 0	73 80	ω
61		0 1	XOPB1(6): Not currently used
1	6 0 0 0 . 0		a
13	0 . 2		e
25	0 . 0		i Elements of final orbit
37	0 . 0		Ω
49	3 0 . 0	73 80	ω
61		0 2	XOPB2(6): Not currently used

FORM 114-C-17 REV. 7-58 - VELLUM

Figure 26. Data Input Format and Data Description

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 2 of 8 JOB NO. _____

NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1 7 0 . 0		α_1
13 1 8 0 . 0		α_2 } Search initiation point for steep descent
25 5 0 0 0 . 0		P_i
37		AFST(4).
49	73 80	AFST(5). } Not currently used
61	0 3	AFST(6)
1 2 .		AMPINC: Reward factor to multiply (increase) step size α
13 . 5		AMPDEC: Penalty factor to multiply (decrease) step size α
25 1 . - 0 8		EPSEPS: Step size amplitude tolerance, ε_3, for iteration step
37 2		XAMP: Initial value of α
49 3 .	73 80	SAVE: Maximum factor by which any component of the steering vector may increase from one iteration to the next
61 1 .	0 4	DIAG: If nonzero ellipsoidal detour on CRT
1		SCLINC: Not currently used
13 . 8 5		SCLDEC: Factor to decrease scale s_i; changing step direction
25 1		ISCL: n_i, determines to which s_i the scaling factors are normalized
37 5		IAMP: Number of successful iterations before α is rewarded
49	73 80	PLSS } Not currently used
61	0 5	BIAS
1 1 .		s_1
13 1 .		s_2 } Scale factors for movement in parameter space (initial values) (if $s_2 = 0$, program will not maximize)
25 1 0 0 .		s_3
37 1 . - 0 5		ε_1 Tolerance for partial derivative convergence
49 2 0 0	73 80	MAXIT: Maximum number of iterations allowed
61 5	0 6	IPRIN: Controls printout of iterations: e.g., every 5th iteration

FORM 114-C-17 REV. 7-58-VELLUM

Figure 26. Data Input Format and Data Description (Continued)

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 3 of 8 JOB NO. _____

	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1			XIN(1)
13			XIN(2)
25			XIN(3) Not currently used
37			XIN(4)
49		73 80	XIN(5)
61		0 7	XIN(6)
1	1 0 0 0.		XIN(7): Initial vehicle mass
13			XIN(8)
25			XIN(9) Not currently used
37			XIN(10)
49	2 0 0 0 0.	73 80	XIN(11): Estimate of λ_4 at first burn
61		0 8	XIN(12): Not currently used
1			XIN(13)
13			XIN(14)
25			XFN(1) Not currently used
37			XFN(2)
49		73 80	XFN(3)
61		0 9	XFN(4)
1			XFN(5)
13			XFN(6)
25			XFN(7) Not currently used
37			XFN(8)
49		73 80	XFN(9)
61		1 0	XFN(10)

FORM 114-C-17 REV. 7-58 - VELLUM

Figure 26. Data Input Format and Data Description (Continued)

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 4 of 8 JOB NO. _____

	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH	
1			XFN(11)	
13			XFN(12)	
25			XFN(13)	Not currently used
37			XFN(14)	
49		73 80	RATES	
61	3	11	IDIMEN:	Specifies two or three-dimensional space (e.g., 2 = coplanar orbits) (not required)
1	0.1		BETASV:	Mass flow rate per second
13	4000.0		XISP:	Specific Impulse
25	0.1		TFPERT:	Time increment added to impulsive approximation (percentage of total time)
37	0.1		PERTX:	Time perturbation at the end of the first iteration se (percentage of total time)
49	3.0	73 80	CØASTI:	Duration of initial and final coast arcs (percentage of total time)
61		12	BL126:	Not currently used
1	0.0		BRNINC:	A priori percentage change in burn arc durations
13	1.0		BMAX1:	First burn thrust initiation and termination point changes allowable (maximum percentage of burn arc)
25	1.0		BMAX2:	Second burn thrust initiation and termination point changes allowable (maximum percentage of burn arc)
37	1.0		BSH1:	First burn maximum allowable change in total burn arc (percentage of total time)
49	1.0	73 80	BSH2:	Second burn maximum allowable change in total burn arc (percentage of total time)
61	0.0	13	TWØBRN:	One-burn transfer control (0 = two burns, 1 = first burn nulled, 2 = second burn nulled)
1			RESET	
13			KMULT	
25			NØPRT	
37			PRNJ	Not currently used
49		73 80	BL145	
61		14	BL146	

FORM 114-C-17 REV. 7-58 - VELLUM

Figure 26. Data Input Format and Data Description (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

FORTRAN FIXED 10 DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 5 of 8 JOB NO. _____

NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH	
1		INSTP:	Number of integration steps per table entry
T		REINT:	If true, solution will be reintegrated and compared to quasilinearization solution (logical variable)
T		CRTØUT:	Logical variable set to true if CRT desired
4		ITSQL:	Number of quasilinearization iterations per cycle
1	73 80	IOPRIN:	Print control (0, 1, 2, 3: print increases in detail as IOPRIN increases)
3	15	IQCYCL:	Number of cycles of quasilinearization for final time iteration
		HBARTL:	Average Hamiltonian tolerance if tolerance exceeded quasilinearization is terminated (not currently used)
.001		RHØSI:	Convergence tolerance on radius for quasilinearization iteration termination (units = units of radius)
		BL163	Not currently used
		BL164	
	73 80	BL165	
	16	BL166	
6.		XNB:	Minimum number of table entries during burn arcs
6.		XNC:	Minimum number of table entries during coast arcs
0.2		DTB:	Maximum time increment during burn arcs (percentage of average orbital period)
2.0		DTC:	Maximum time increment during coast arcs (percentage of average orbital period)
	73 80	JMAX:	Storage table length (not required)
	17	IBL176:	Not currently used
			Not currently used
	73 80		
	18		

FORM 114-C-17 REV. 7-68 - VELLUM

Figure 26. Data Input Format and Data Description (Continued)

FORTRAN FIXED 10 DIGIT DECIMAL DATA

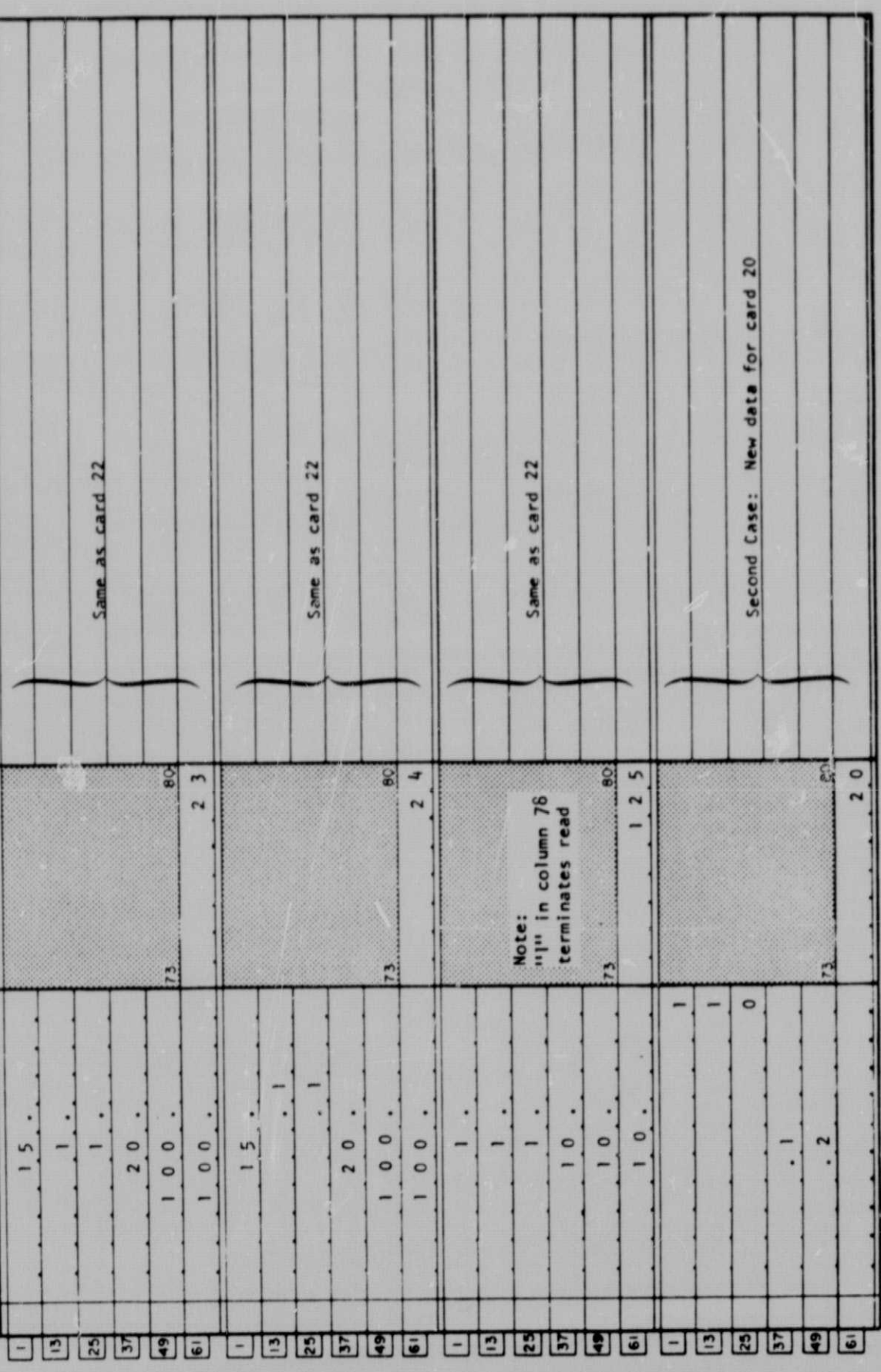
DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 6 of 8 JOB NO. _____

NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
		Not currently used
	73 80	
	1 9	
4		IMAPS: Maximum number of best rank suspected minima to be contoured in detail
5		NSTEEP: Maximum number of steep descents into the best ranking suspected minima
0		IUSE: Perform a final steep descent using the search initiation point of data card 03 if IUSE = 0
.2		SCXY: $\emptyset_1$ and $\emptyset_2$ scale factor for local area maps
.1		SCIMP: Impulse scale factor for local area maps
	73 80	
	2 0	Not currently used
20.		SPAR(1,1): Number of points in contour grid (e.g., 20 x 20)
18.		SPAR(2,1): $\emptyset_1$ interval for contour grid
18.		SPAR(3,1): $\emptyset_2$ interval for contour grid
180.5		SPAR(4,1): $\emptyset_1$ of center of contour map
181.5		SPAR(5,1): $\emptyset_2$ of center of contour map
500.		SPAR(6,1): Impulse contour interval
	73 80	
	2 1	
12.		Same as card 21 -- this card specifies an additional contour map if desired, but there is no ranking of minima and generation of small scale maps.
15.		
15.		
70.		
181.		
	73 80	
150.		
	2 2	

FORM 114-C-17 REV. 7-58 - VELLUM

Figure 26. Data Input Format and Data Description (Continued)

SPACE DIVISION · NORTH AMERICAN ROCKWELL CORPORATION



FORTRAN FIXED IO DIGIT DECIMAL DATA

DECK NO. _____ PROGRAMMER _____ DATE _____ PAGE 7 of 8 JOB NO. _____

	NUMBER	IDENTIFICATION	DESCRIPTION DO NOT KEY PUNCH
1	15.		
13	1.		
25	1.		Same as card 22
37	20.		
49	100.	73 80	
61	100.	23	
1	15.		
13	.1		
25	.1		Same as card 22
37	20.		
49	100.	73 80	
61	100.	24	
1	1.		
13	1.		
25	1.	Note: "1" in column 76 terminates read	Same as card 22
37	10.		
49	10.	73 80	
61	10.	125	
1	1		
13	1		
25	0		Second Case: New data for card 20
37	.1		
49	.2	73 80	
61		20	

FORM 114-C-17 REV. 7-68 - VELLUM

Figure 26. Data Input Format and Data Description (Continued)

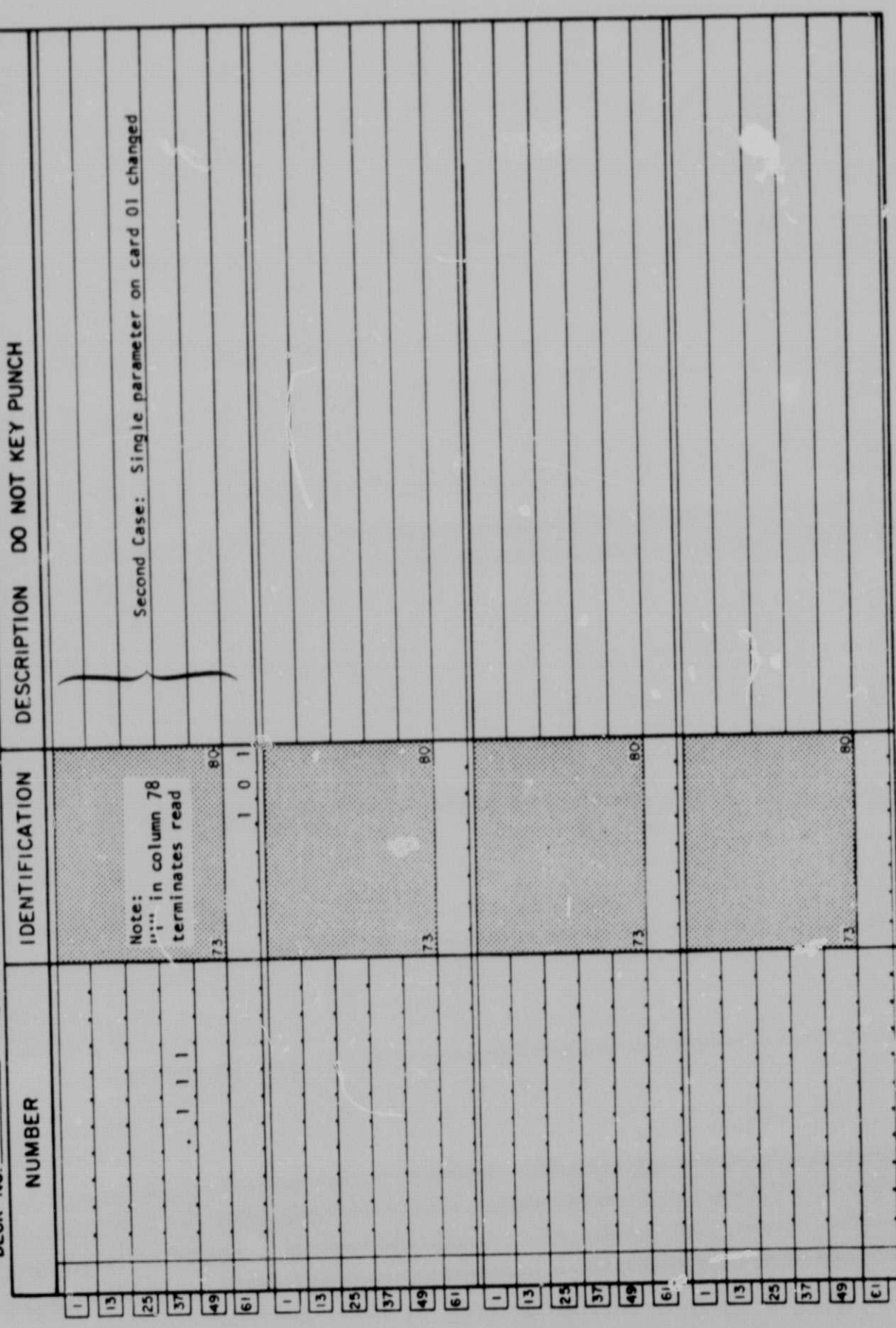


Figure 26. Data Input Format and Data

description of the input formats. The description of Figure 26 is further amplified by Table 4 which lists the FORTRAN equivalent of each of the formats required for the complete set of 27 possible input cards. Note that the data cards are classified according to the number appearing in columns 79 and 80 except for the first two cards which are read independently and only once at the beginning of each computer run. The first card is merely for case description and allows the programmer to insert his own alpha-numeric identification. The second card is interesting in that it allows the user to specify the central body and the unit system he wishes. For instance, he may write EARTH, MØØN, SUN, VENUS, MARS, or JUPITER in the first field and the program will utilize the appropriate gravitational constant for those bodies. Similarly, BRITISH, METERS, KILØMETERS, or BØDY may be written in the second field and the program will use the corresponding unit system. In the case of the BØDY specification, the program will expect the user to supply an appropriate constant for g-times-mass of the attracting center. These first two cards are unique in that they are read only once at the beginning of the computer run and may not be reread. All other cards may be repeated as often as desired to produce additional cases.

At this point in the computations the program enters subroutine INPUT and reads the orbital elements, constants and control parameters from the remaining cards. As noted earlier, each card and its format are identified by the number appearing in columns 79 and 80. The program continues

Table 4. FORTRAN Data Input Formats

Card Identification	Format
TITLE	(18A4)
BØDYUNIT	(6A4, D12.0)
01	(6D12.0)
02	(6D12.0)
03	(6D12.0)
04	(6D12.0)
05	(2D12.0, 2I12, 2D12.0)
06	(4D12.0, 2I12)
07	(6D12.0)
08	(6D12.0)
09	(6D12.0)
10	(6D12.0)
11	(4D12.0, L12, I12)
12	(6D12.0)
13	(5D12.0, E12.0)
14	(E12.0, 2I12, 3D12.0)
15	(I12, 2I12, 3I12)
16	(6D12.0)
17	(4D12.0, 2I12)

Table 4. FORTRAN Data Input Formats (Cont)

Card Identification	Format
18	(6E12.0)
19	(6E12.0)
20	(3I12, 2E12.0, I1[illegible],
21	(6E12.0)
22	(6E12.0)
23	(6E12.0)
24	(6E12.0)
25	(6E12.0)

reading until it encounters a non-zero entry in column 78. Figure 26 also includes a description of each data parameter beside the field where it appears on the data card. Of course, one who requires complete knowledge of the functions of the input parameters may resort to examining the program's listings in Appendix III and in Refs. 11 and 12. The descriptions of Fig. 26 should suffice, but a few additional words of clarification are listed below:

cards 01 and 02 — The first two cards allow the programmer to specify the elements of the initial and final orbits. The program is expected to find the minimum fuel transfer between these two orbits. At the present time the orbits are limited to circular or elliptical.

card 03

This card contains the initial iterate for a steep descent optimization. That is, ϕ_1 is the estimated take-off angle on the initial orbit, ϕ_2 the estimated arrival point on the final orbit and p_t is the semi-latus rectum of the transfer orbit. These parameters are used to initiate the two-impulse search which generates an optimum two-impulse transfer. The two-impulse transfer circumstances then are used to compute the initial iterate for finite-thrust optimization.

cards 04, 05, 06

These three cards contain parameters for controlling the steep descent optimization process. A more complete description of the steep descent process will be found in Ref. 11.

cards 07, 08, 09 10, 11

These five cards provide a method of inputting an estimate of the initial (XIN) and final (XFN) values of 14 state variables and Lagrange multipliers. Fortunately, the excellent performance of the initial conditions determination subroutines has eliminated the necessity for inputting most of these boundary conditions. The exceptions appear on card 08 and are a specification of the vehicle's initial mass and an estimate of the value of λ_4 at the first burn which is used to scale the Lagrange multipliers. Since the

other boundary conditions are not required, the provision for inputting them could be removed. However, these storage locations have been left in the present version of the program. Future modifications to the program, in order to solve more difficult problems such as rendezvous, no doubt, will require provisions for additional input. These locations can easily be reallocated at that time. In the meantime, users wishing to make minor modifications to the program which require additional data input may find that it is convenient to read the data using these available locations.

cards 12 and 13

Most of the parameters appearing on card 12 have obvious meanings. An exception is TFPERT which allows the user to estimate how the total time for the impulsive maneuver should be slightly modified to more closely approximate the optimum time for the finite-thrust maneuver. At the end of the first QASLN iteration set it is necessary to perturb the total time by a small amount in order to generate an approximation to the partial derivative of ΔV with respect to total time. PERTX is this perturbation, and it is applied prior to the

computation of the next iteration set. BRINC on card 13 is similar to TFPERT but it allows the user to specify that the burn arc durations are slightly different than those specified by the impulsive approximation. The next four parameters, BMAX1, BMAX2, BSH1, BSH2, are tolerances used by subroutine KHIST when it decides how to adjust the thrust initiation and termination points between quasilinearization iterations. They should generally be smaller than a few percent of the burn arc duration.

The parameter TWØBRN allows the user to specify that a one-burn transfer is adequate. Users should be careful with this control, since if it is set to call for a one-burn transfer and the maneuver is really a two-burn transfer, the quasilinearization procedure will usually diverge.

card 14 — No currently used input parameters appear on card 14. These input fields are therefore available for the user to modify the program and to input his additional data.

cards 15 and 16 — These cards contain several input and control parameters which are adequately described in Figure 26.

card 17 — The controls of card 17 are used to specify the nominal structure of the solution storage table (See Figure 4). They are provided so that the user may assure that some minimum number of storage points occur during a burn arc or a small coast arc. In order to assure accurate interpolation XNB and XNC should never be smaller than 4. DTB and DTC are included in order to provide some sort of limit on the size of allowable integration steps. That is, if an extremely-high-thrust vehicle is being simulated the user may want integration step sizes no larger than five percent of the burn time.

cards 18, 19 — These cards have no current use in the program and are available to the user.

cards 20, 21, 22, 23, 24, 25 — These cards control the generation of contour maps which is part of the initial optimization procedure. The descriptive definition in Figure 26 should suffice. All necessary additional details are provided in Reference 11.

As stated above only those parameters necessary for a particular problem need be included on the data cards. In the event that parameters are deleted or that entire cards are deleted, subroutine DATSET will supply

nominal and normally sufficient data and parameters for proper processing. It is therefore possible to run a case by supplying only the first five cards listed in Figure 26, plus the parameters which describe the vehicle's physical characteristics. This means that a relatively untrained analyst may be able to obtain reasonable answers to his problem with one computer run.

APPENDIX II

EXAMPLE OF THE USE OF THE FINITE-THRUST ORBITAL TRANSFER OPTIMIZATION PROGRAM

APPENDIX II

SAMPLE OPTIMIZATION PROBLEM

Much of the foregoing discussion is difficult to understand without a realistic example. For this reason this appendix presents a comprehensive sample problem. Figure 27 contains the images of the data cards which were used to produce the computer output appearing as Figures 27 through 55. The orbits utilized for this example are inclined, elliptical, and non-coapsidal. Since the British unit system was used the output parameters will appear in miles and ft/sec. Figure 27 represents the first and second pages of printed output and it contains the image of the data cards in the order in which they were used. Note that not all of the data cards defined in Appendix I Figure 26, are included and that this means that many of the deleted input data parameters were supplied by subroutine DATSET. Figure 28 contains a printout of the data array common block showing the status of each data parameter just prior to execution of the case. As noted earlier subroutine DEX transforms the elements to a standard orientation. These transformed elements appear next in the printed output (Figure 29).

The program then generates the contour map of the entire function space which appears as Figure 30. In this case, four possible minima were identified and noted by printing the numbers 1, 2, 3, and 4 (in groups of five) on the contour map and by the printed output appearing in Figure 29.

```
 DATE 02/04/69,TIME =  96.553


CASE IDENTIFICATION
  TRANSFER IN EARTH ORBIT

BODY IS EARTH
UNIT SYSTEM IS BRITISH
UXM = 0.956345E 05
```

```
 DATE 02/04/69,TIME =  96.553

 DATA CARD IMAGES
5000.          .2              5.              0.              -90.                            0 0 1
6000.          .2              0.                0.            30.                             0 0 2
70.            180.            5000.                         20000.                          0 0 3
1000.                                                        3.                              0 0 8
1.0            400.            0.1             0.1           1.              0.0             0 0 12
0.             1.              1.              1.                                            0 0 13
        1               T               T               6               1               3     0 0 15
                .001                                                    108                   0 0 16
  6.                    6.              .2              2.                                    0 0 17
          4                     5               0               .2              .1            0 0 20
20.            18.             18.             180.5           181.1           90C.          1 1 21
```

Figure 27.

```
  DATA ARRAY IMAGE
CARD NO.=  1    5.000000CCCD 03   2.000000C0D-01   5.000000G0D 00   0.0              -9.00C000CCD 01   0.0
CARD NO.=  2    6.000000C0D 03    2.0CCC00C0D-01   0.0              0.0               3.C0C000CCD 01   0.0
CARD NO.=  3    7.000000CCD 01    1.8CC000000D 02  5.000000000D 03  0.0               0.0              0.0
CARD NO.=  4    2.C0C000CCD 00    5.0CCC00000D-01  1.000000000D-05  2.000000000D 00   2.C0C000CCD 00   1.0000C0000 00
CARD NO.=  5    0.0               8.5CC000000D-01                5                 0.0              0.0
CARD NO.=  6    1.0000000CD 00    1.0CCC00000D 00  1.000000000D 02  1.000000000D-05                200              5
CARD NO.=  7    0.0               0.0              0.0              0.0               0.0              0.0
CARD NO.=  8    1.000000000D 03   0.0              0.0              0.0               2.C0C000CCD 04   0.0
CARD NO.=  9    0.0               0.0              0.0              0.0               0.0              0.0
CARD NO.= 10    0.0               0.0              0.0              0.0               0.0              0.0
CARD NO.= 11    0.0               0.0              0.0              0.0                              T                3
CARD NO.= 12    1.000000CCD 00    4.000000000D 02  1.000000000D-01  1.000000000D-01   3.00C000CCD 00   0.0
CARD NO.= 13    0.0               1.000C00000D 00  1.000000000D 00  1.000000000D 00   1.00C000CCD 00     0.0
CARD NO.= 14      0.0                              0                0.0               0.0              0.0
CARD NO.= 15                   1                  T                                  1                3
CARD NO.= 16    0.0               1.000000000D-03  0.0              0.0               0.0              0.0
CARD NO.= 17    6.000000000D 00   6.0CC000000D 00    0.0            2.000000000D-01   2.000000000D 00                100      0.0
CARD NO.= 18      0.0               0.0              0.0              0.0               0.0              0.0
CARD NO.= 19      0.0               0.0              0.0              2.000000E-01      9.599996E-02     0.0
CARD NO.= 20                   4                  5                0                 1.805000E 02     1.811CC0E 02     5.00C000E 02
CARD NO.= 21    2.000000CE 01     1.8C0000E 01     1.800000E 01     0.0               0.C              0.0
CARD NO.= 22    0.0               0.0              0.0              0.0               0.0              0.0
CARD NO.= 23    0.0               0.0              0.0              0.0               0.0              0.0
CARD NO.= 24    0.0               0.0              0.0              0.0               0.0              0.0
CARD NO.= 25    0.0               0.0              0.0              0.0               0.0              0.0
```

Figure 28.

```
DATE 02/04/69,TIME =  96.553


TRANSFORMED ELEMENTS
          P1                    E1                    W1                  INCL
 0.5000CCC0000D C4  0.2CCC00C0000D 00 -0.900000000000D 02  0.5000000000000 01
          P2                    E2                    W2
 0.6000CCC000CD C4  0.2CCC00000000D CC  0.30CC00000C00D 02
```

```
GENERATE CONTOUR MAP
   0.200000E 02    0.180000E 02    0.180000E 02    0.180500E 03    0.181100E 03    0.500000E C3

LOCAL MINIMA AT

      PHI-1           PHI-2             P             IMPULSE        RANK
 0.3839470 02    0.3042570 03    0.6618960 04    0.5371050 04        2
 0.7628540 02    0.1905740 03    0.6661790 04    0.4915250 04        1
 0.1710260 03    0.7688540 02    0.4595740 04    0.5439350 04        3
 0.3036570 03    0.3899470 02    0.4597940 04    0.5667560 04        4
```

Parameters on cards 21-25 are printed for each basic map generated.

Figure 29.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

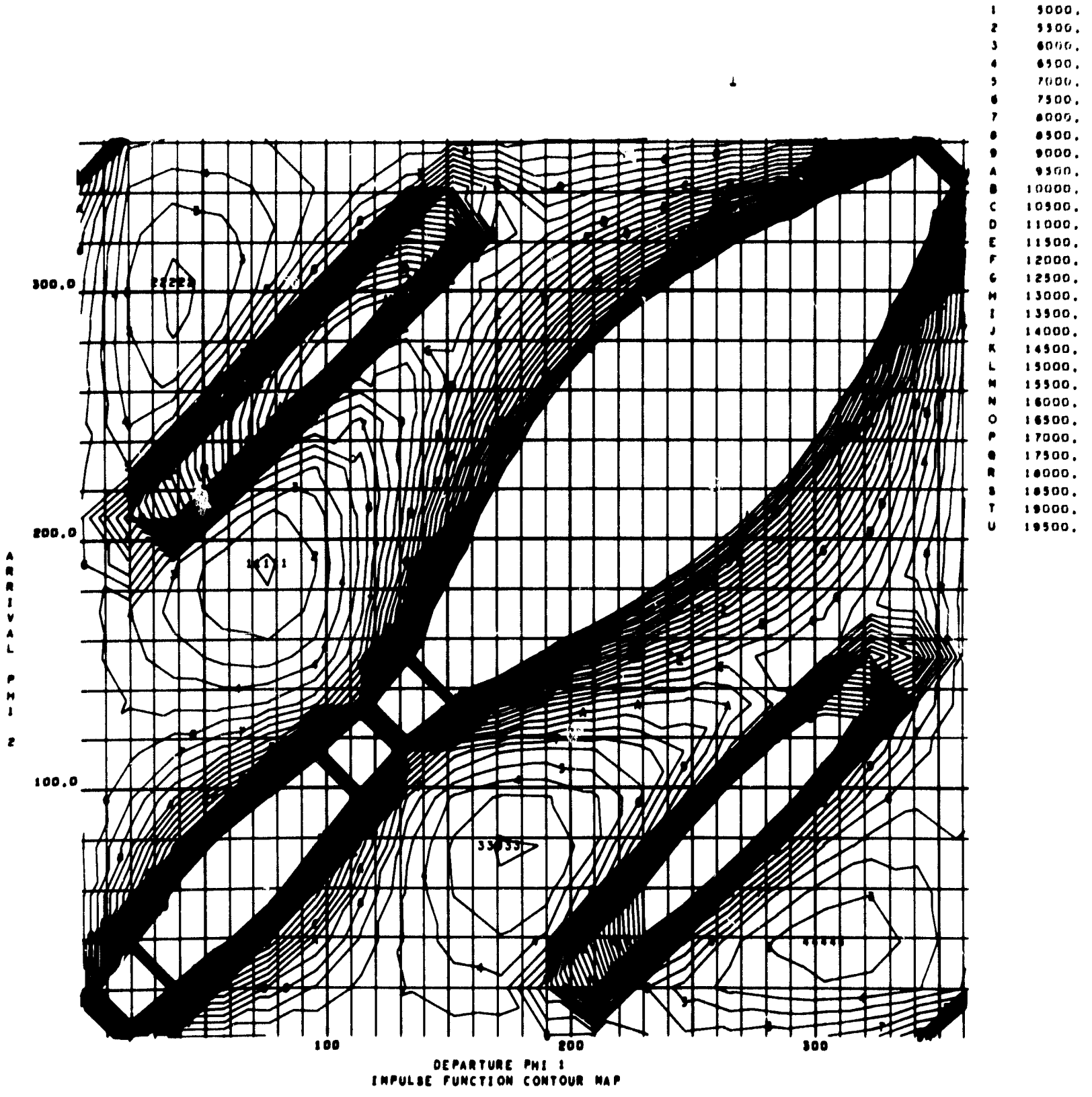

1 9000.
2 9500.
3 6000.
4 6500.
5 7000.
6 7500.
7 8000.
8 8500.
9 9000.
A 9500.
B 10000.
C 10500.
D 11000.
E 11500.
F 12000.
G 12500.
H 13000.
I 13500.
J 14000.
K 14500.
L 15000.
M 15500.
N 16000.
O 16500.
P 17000.
Q 17500.
R 18000.
S 18500.
T 19000.
U 19500.
300.0
200.0
100.0
ARRIVAL PHI 2
100
200
300
DEPARTURE PHI 1
IMPULSE FUNCTION CONTOUR MAP

Figure 30.

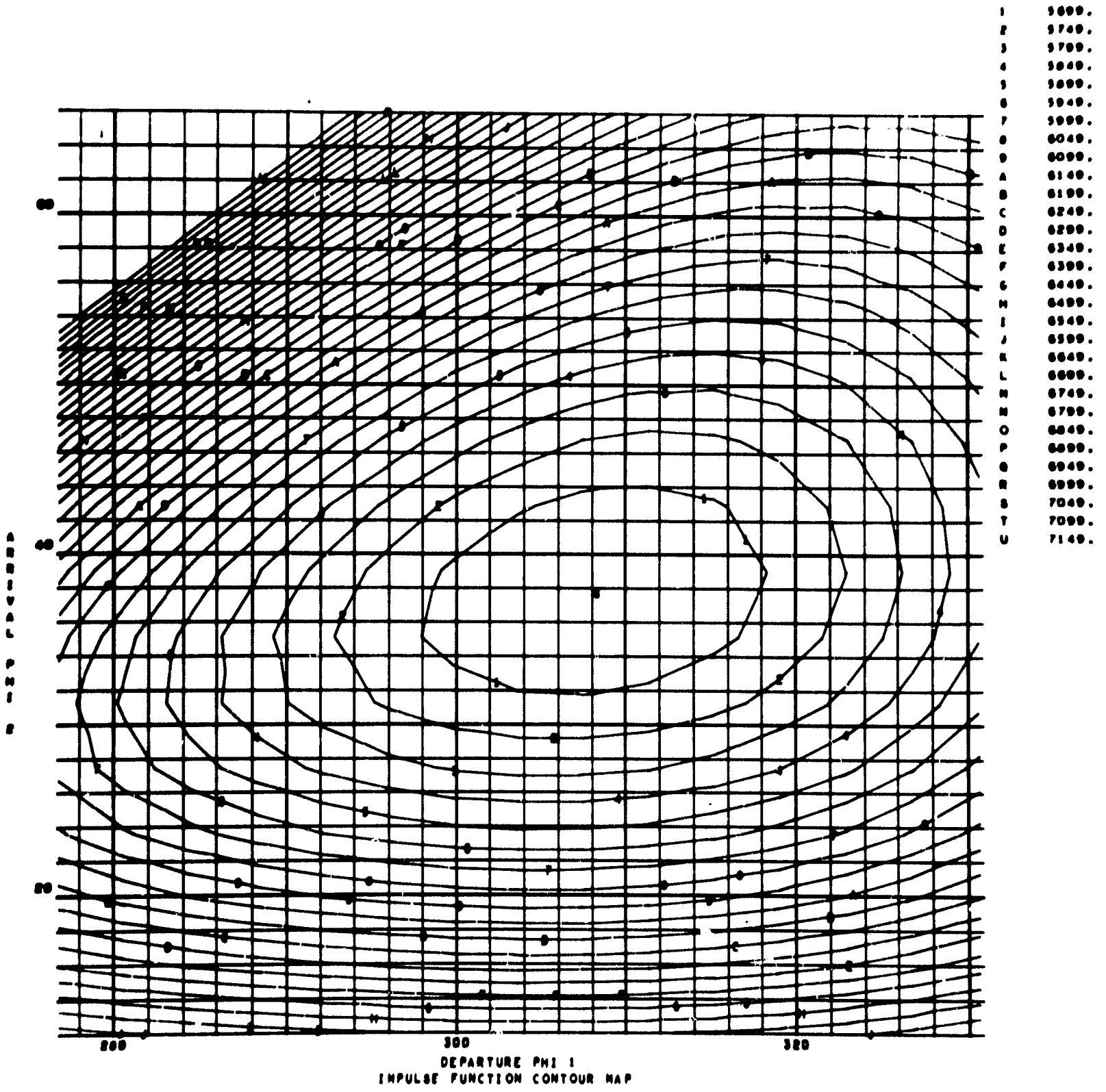

RANK OF OPTIMUM = 4
1 5690.
2 5740.
3 5790.
4 5840.
5 5890.
6 5940.
7 5990.
8 6040.
9 6090.
A 6140.
B 6190.
C 6240.
D 6290.
E 6340.
F 6390.
G 6440.
H 6490.
I 6540.
J 6590.
K 6640.
L 6690.
M 6740.
N 6790.
O 6840.
P 6890.
Q 6940.
R 6990.
S 7040.
T 7090.
U 7140.
60
40
20
280
300
320
DEPARTURE PHI 1
IMPULSE FUNCTION CONTOUR MAP

Figure 31.

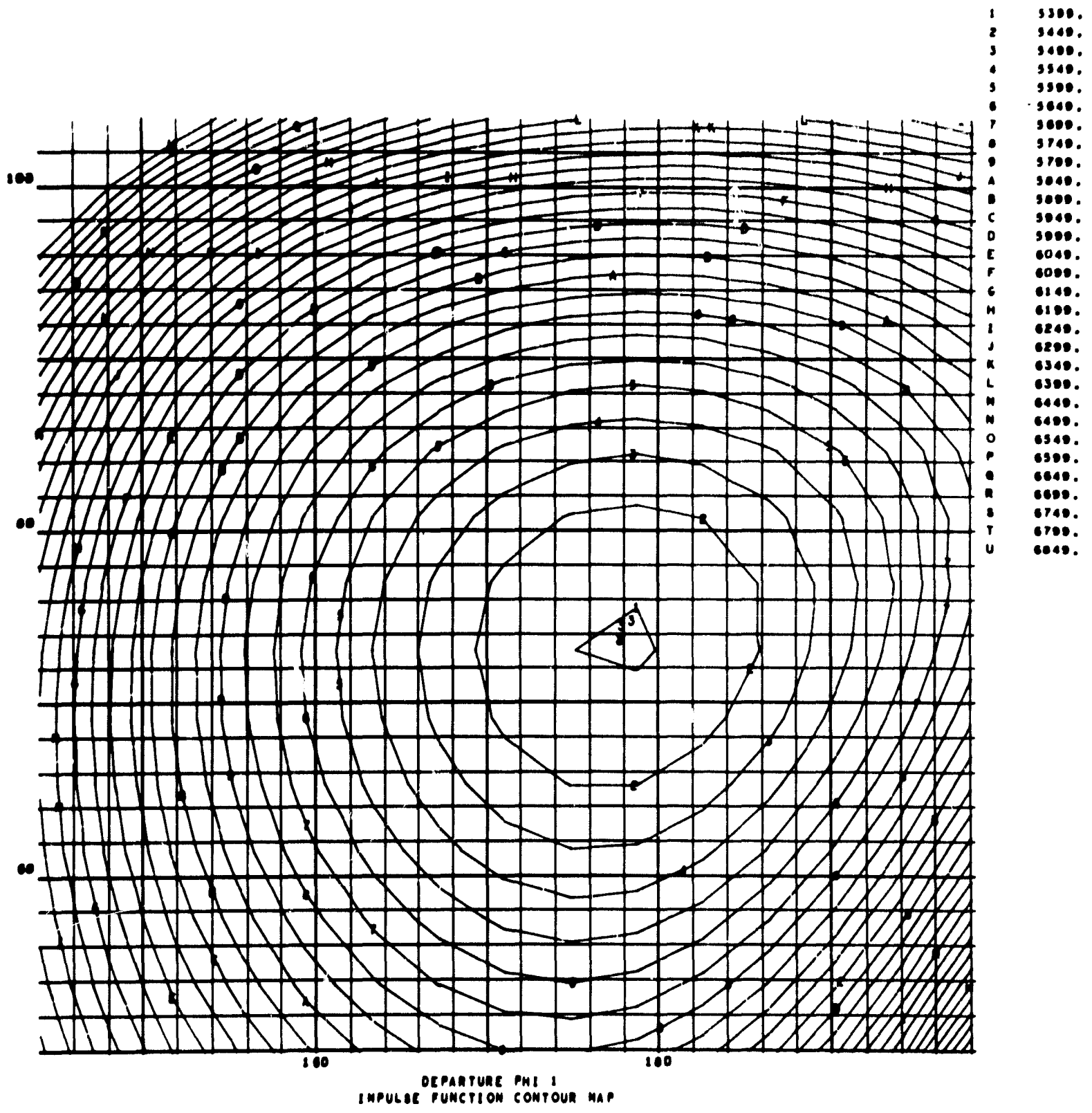

RANK OF OPTIMUM = 3
1 5300.
2 5450.
3 5400.
4 5550.
5 5500.
6 5650.
7 5600.
8 5750.
9 5700.
A 5850.
B 5800.
C 5950.
D 5900.
E 6050.
F 6000.
G 6150.
H 6100.
I 6250.
J 6200.
K 6350.
L 6300.
M 6450.
N 6400.
O 6550.
P 6500.
Q 6650.
R 6600.
S 6750.
T 6700.
U 6850.
ARRIVAL PHI 2
100
60
100
100
DEPARTURE PHI 1
IMPULSE FUNCTION CONTOUR MAP

Figure 32.

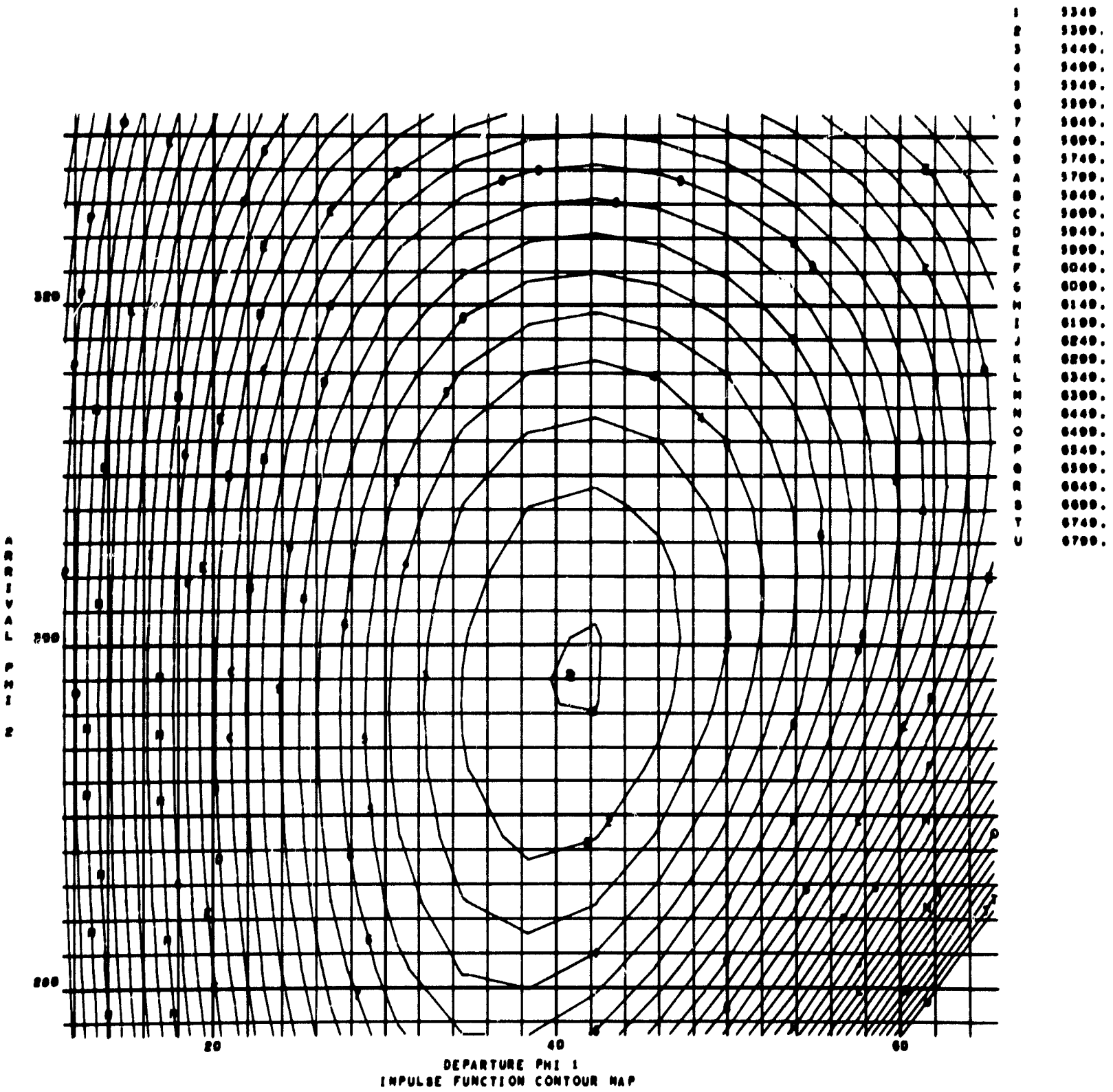

RANK OF OPTIMUM = 2
1 5340
2 5380.
3 5440.
4 5480.
5 5540.
6 5580.
7 5640.
8 5680.
9 5740.
A 5780.
B 5840.
C 5880.
D 5940.
E 5980.
F 6040.
G 6080.
H 6140.
I 6180.
J 6240.
K 6280.
L 6340.
M 6380.
N 6440.
O 6480.
P 6540.
Q 6580.
R 6640.
S 6680.
T 6740.
U 6780.
ARRIVAL PHI 2
300
200
100
20
40
60
DEPARTURE PHI 1
IMPULSE FUNCTION CONTOUR MAP

Figure 33.

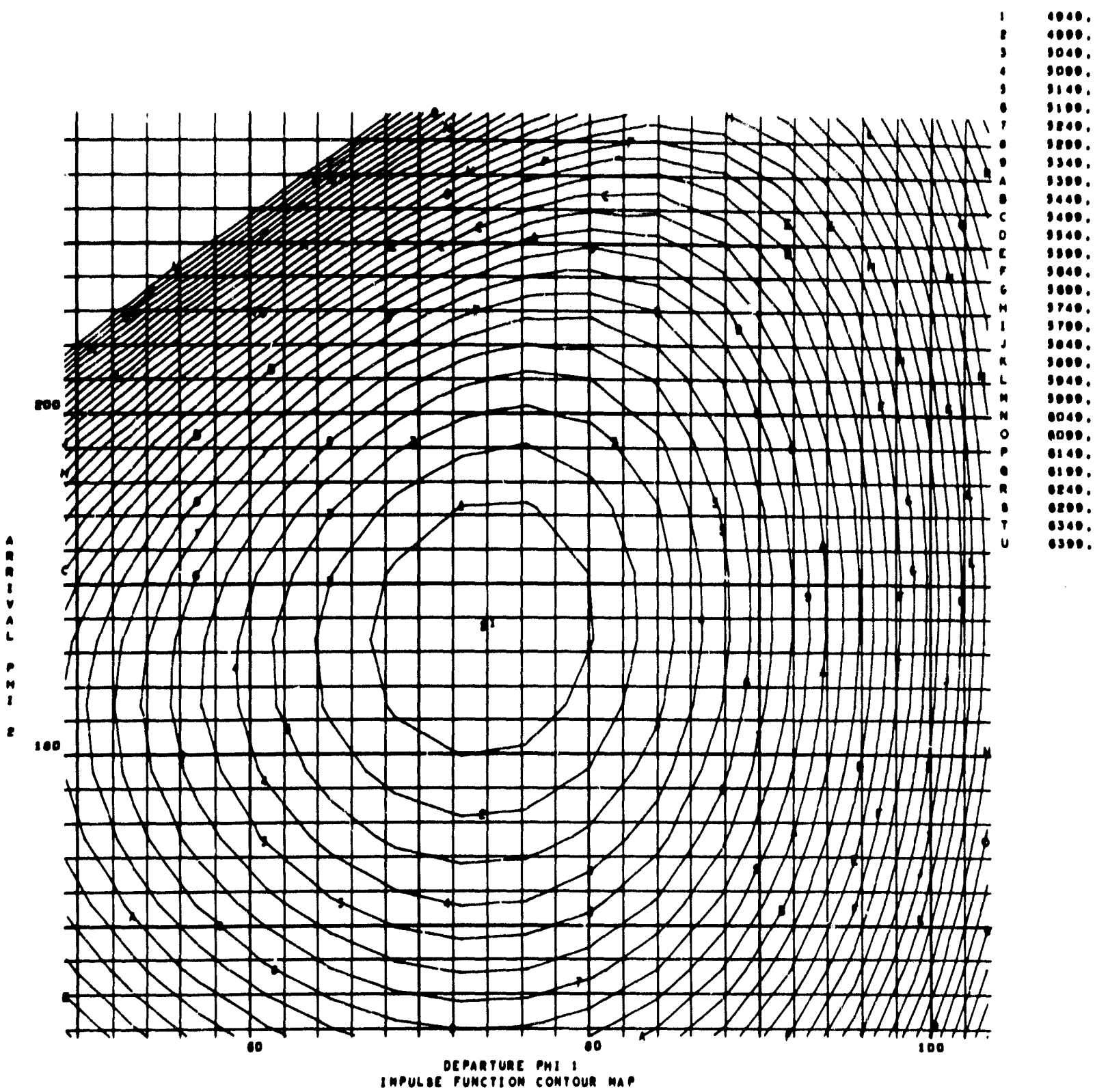

RANK OF OPTIMUM = 1
1 4040.
2 4000.
3 3940.
4 3900.
5 3840.
6 3800.
7 3740.
8 3700.
9 3640.
A 3600.
B 3540.
C 3500.
D 3440.
E 3400.
F 3340.
G 3300.
H 3240.
I 3200.
J 3140.
K 3100.
L 3040.
M 3000.
N 2940.
O 2900.
P 2840.
Q 2800.
R 2740.
S 2700.
T 2640.
U 2600.
200
160
ARRIVAL PHI 2
60
80
100
DEPARTURE PHI 1
IMPULSE FUNCTION CONTOUR MAP

Figure 34.

ADAPTIVE STEEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSFER

PHI1	PHI2	P	DI/DP1	DI/DP2	DI/DP	IMPULSE	GAINS	ITNO
303.6575	38.9947	4.5979400 03	-46.385457	48.806144	0.008408	5.667559973441687OD 03	1.00 00 1.00 00 1.00 02	0 0
309.3666	37.6787	4.5969660 03	1.945115	-4.496654	1.075723	5.65444290825379300 03	1.00 0C 7.20-01 1.20 02	5 2
308.2078	37.7401	4.5929110 03	0.044743	-0.052931	0.013951	5.65419125729003000 03	1.0C 0C 5.20-01 8.50 01	10 8
308.2034	37.7404	4.5928560 03	-0.000297	0.001833	-0.000267	5.65419120969668600 03	1.00 00 3.80-01 6.10 01	15 13
308.2034	37.7403	4.5928570 03	0.000006	0.000089	0.000008	5.65419120967948600 03	1.00 CC 3.80-01 6.10 01	18 17

AMP= 0.15258789D-C4 IFAIL= 17

CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC

	*** FINAL TRANSFER CONDITIONS *** X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
IMPULSE FIRST VECTOR	-0.142468837580 04	0.740707798370 02	-0.136500579550 04	0.157445250CD 04
SECOND VECTOR	-0.273135271550 04	0.194663624700 04	-0.151353680630 04	0.367973830570 04
TOTAL				0.565419120570 04
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT	
P	0.500000000000 04	0.6C0000C0000D 04	0.459285744590 04	
E	0.2CC000000000 0C	C.200000000C0D 00	0.104524209810 00	
I (DEG)	0.5CC0CCC00000 01	0.0	0.392738125590 01	
R A NODE (DEG)	0.0	0.0	0.377403101810 02	
ARG PER (DEG)	-0.9CC00C000C0D 02	C.300000C00000D 02	-0.142408949310 03	
A	0.52C833333330 04	0.625000000000 04	0.464359010940 04	
PERIOD	0.763696370720 04	C.100390495040 05	0.642914745410 04	
TRAN PT RAC	0.432050889370 C4	C.500760430660 04		
PHI-1, PHI-2, P	0.3C8203441530 03	C.377403101810 J2	0.459285744590 04	

RANK OF OPTIMUM = 4

Figure 35.

ADAPTIVE STEEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSFER

PHI1	PHI2	P	DI/DP1	DI/DP2	DI/DP	----IMPULSE----	-------GAINS-------	ITNO
171.0262	76.8894	4.5957450 03	-103.266712	60.110114	0.007332	5.439352998424380D 03	1.00 0C 1.00 00 1.00 02	0 0
178.5048	74.7263	4.5959C40 03	R.679544	-C.097591	-2.949527	5.396313716357944CD 03	1.CC CC 1.00 00 1.00 02	5 0
177.9176	74.4165	4.603344D 03	0.121163	5.658692	-1.156085	5.394486366220816CD 03	1.CD 0C 1.6D 00 1.4D 02	10 2
177.8123	73.7024	4.6115395D 03	-0.059560	1.047350	0.022190	5.393783777344704CD 03	1.CD CC 1.00 00 1.00 02	15 8
177.8123	73.65C4	4.611789D 03	0.010107	0.081385	0.003127	5.393781159008685CD 03	1.00 0C 6.1D-01 6.1D C1	20 12
177.8115	73.6469	4.6117953D 03	0.CCC101	-0.007276	-0.002143	5.393781145612127CD C3	1.00 CC 5.2D-01 7.2D 01	25 13
177.8114	73.6463	4.6118C3D 03	-0.000360	-0.004364	-0.000148	5.393781144799084CD 03	1.CD 0C 4.4D-01 8.5D 01	30 15
177.8114	73.6465	4.6118020 03	0.000007	0.000083	0.CC0009	5.393781144757363CD 03	1.00 00 4.4D-01 4.4D 01	35 20

AMP= 0.15258789D-04 IFAIL= 20

*** FINAL TRANSFER CONDITIONS *** CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC

	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
IMPULSE				
FIRST VECTOR	-0.638985716960 C4	0.451118256110 05	0.206123313480 C4	0.241662228330 04
SECOND VECTOR	-0.41E378209450 05	0.182579943540 05	0.726248032680 02	0.297655866150 04
TOTAL				0.539378114480 04

	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT
P	0.5CC000000000 04	0.600000000000 04	0.461180236260 04
E	0.2CC00C0000CD 0C	0.200000C000C0D 00	0.168164465510 00
I (DEG)	0.5CC000000000 01	0.0	0.196690320810 00
R A NODE (DEG)	0.0	0.0	0.736465275200 02
ARG PER (DEG)	-0.9CCCCC0000CD C2	C.3C0000000000 02	-0.135589725280 03
A	0.520833333330 04	0.625000000C0D 04	0.474601632250 04
PERIOD	0.763696370720 C4	C.10039C495040 05	0.664303302590 04
TRAN PT RAD	0.503848293040 C4	0.524144579280 04	
PHI-1, PHI-2, P	0.177011398970 03	0.736465275200 02	0.461180236260 04

RANK OF OPTIMUM = 3

Figure 36.

ADAPTIVE STEEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSFER

PHI1	PHI2	P	DI/DP1	DI/DP2	DI/DP	----IMPULSE----	-------GAINS------	ITNO
38.3947	304.2573	6.618967D 03	-76.305578	58.963774	0.001905	5.37104812866789800 03	1.00 0C 1.00 00 1.00 02	0 0
40.9671	298.1485	6.618876D 03	4.274738	-1.521588	0.200847	5.34319661468645300 03	1.00 0C 1.00 00 1.20 02	5 1
40.8331	298.2615	6.617795D 03	-0.027793	-0.014357	-0.000363	5.34314869653839000 03	1.00 CC 1.60 00 1.60 C2	10 8
40.8343	298.2634	6.617791D 03	0.000498	-0.000127	0.000021	5.34314869347744200 03	1.00 CC 1.90 00 1.60 02	15 15
40.8343	298.2634	6.617790D 03	0.0CCC45	-0.000011	0.000002	5.34314869347687000 03	1.CO CC 1.90 00 1.60 02	17 17

AMP= 0.15258789C-04 IFAIL= 17

*** FINAL TRANSFER CONDITIONS *** CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC

	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
IMPULSE				
FIRST VECTOR	0.235317208C6D 05	-C.366372306940 05	-0.138312126280 04	0.3565C91C558D 04
SECOND VECTOR	0.341400704200 05	0.271343914640 05	0.128415680500 04	0.177805763770 04
TOTAL				0.534214869350 04
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT	
P	0.5C0000000000 04	0.600000000000 04	0.661779047170 04	
E	0.2CC00CC00000 CC	0.200000000000 00	0.200028036550 00	
I (DEG)	0.5CC000000000 01	0.0	0.334565611C0D 01	
R A NODE (DEG)	C.C	0.0	-0.617366055660 02	
ARG PER (DEG)	-0.90C00C000000 02	0.300000000000 02	0.612275188370 02	
A	G.520833333330 C4	C.625000C00000 04	0.689361227750 C4	
PERIOD	0.763696370720 04	C.10039049504D 05	0.116290128500 05	
TRAN PT RAD	0.575224920930 C4	0.603658756940 04		
PHI-1, PHI-2, P	0.4CE343237C6D C2	0.298263394430 03	0.661779047170 04	

RANK OF OPTIMUM = 2

Figure 37.

ADAPTIVE STEEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSFER

PHI1	PHI2	P	DI/DP1	DI/DP2	DI/DP	——IMPULSE——	——GAINS——	ITNO
76.2894	190.5736	6.6617940 03	48.742046	43.918548	0.004739	4.9152575794116120D 03	1.CD CC 1.0D 00 1.0D C2	0 0
74.4347	187.6877	6.6605955D 03	-1.327932	0.224654	2.076127	4.9042450854971990D 03	1.CD CC 1.2D 00 1.6D 02	5 3
73.8834	187.5837	6.6446970 03	2.217672	0.396805	-0.118189	4.9026602287332530D 03	1.0D CC 1.4D 00 2.3D C2	10 7
73.8187	187.5578	6.6449270 03	0.005476	0.002847	0.009664	4.9026512624172460D 03	1.0D 00 1.9D 00 2.7D 02	15 10
73.8153	187.5569	6.6448560 03	-0.005887	0.001034	0.001067	4.9026512241788520D C3	1.0D 0C 1.6D 00 3.1D 02	20 14
73.8152	187.5568	6.6448500 03	0.000794	0.000119	0.000006	4.9026512238537260D 03	1.CD CC 1.9D 00 1.9D 02	25 19
73.8152	187.5568	6.6448500 03	-0.000015	0.000003	0.000033	4.9026512238523590D 03	1.0D 00 1.9D 00 1.9D 02	27 19

AMP= 0.152587890D-04 IFAIL= 19

CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC

	*** FINAL TRANSFER CONDITIONS *** X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
IMPULSE			0.353476794390 03	0.28682982133D 04
FIRST VECTOR	-0.268284320280 04	C.951073680300 03	0.165312469870 04	0.20343533C1C6D 04
SECOND VECTOR	-0.953402494030 C3	0.704836567310 03		0.4902651223900 04
TOTAL				

	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT
P	0.50C0000000000 04	C.6C0000000000 04	0.664484985440 04
E	0.2CC00C000CC0 CC	C.2000000000000 00	0.104330516440 00
I (DEG)	0.5CC0000000000 01	0.0	0.524526585740 01
R A NODE (DEG)	0.0	0.0	0.755679689320 01
ARG PER (DEG)	-0.9CC0CC000000D 02	C.3000000000000 02	0.212397268780 02
A	0.52C833333333D C4	0.625000000000D 04	0.671797403890 04
PERIOD	0.7E3696370720 C4	C.100390495040 05	0.111874228170 05
TRAN PT RAC	0.E12868186930 04	0.736062370090 04	
PHI-1, PHI-2, P	0.738151796520 02	0.187556796890 03	0.664484985440 04

RANK OF OPTIMUM = 1

Figure 38.

	••• FINAL TRANSFER CONDITIONS •••		CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC
IMPULSE	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
FIRST VECTOR	-0.148408837988D 04	0.740707790337D 02	-0.136500379955D 04	0.197445290400 04
SECOND VECTOR	-0.273135271955D 04	0.194663624700 04	-0.191353660635D 04	0.367973830970 04
TOTAL				0.565419120970 04
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT	
P	0.500000000000D 04	0.600000000000D 04	0.459285744590D 04	
E	0.200000000000D 00	0.200000000000D 00	0.104524209810D 00	
I (DEG)	0.900000000000D 01	0.0	0.302738125590D 01	
R A NODE (DEG)	0.0	0.0	0.377403101810 02	
ARG PER (DEG)	-0.900000000000D 02	0.300000000000D 02	-0.142408949310 03	
A	0.520833333330 04	0.625000000000D 04	0.484359010940 04	
PERIOD	0.763696370720 04	0.100390495040 05	0.642914745410 04	
TRAN PT RAD	0.432090889370 04	0.500760430880 04		
PHI-1, PHI-2	0.308203441530 03	0.377403101810 02	0.459285744590 04	

RANK OF OPTIMUM = 4

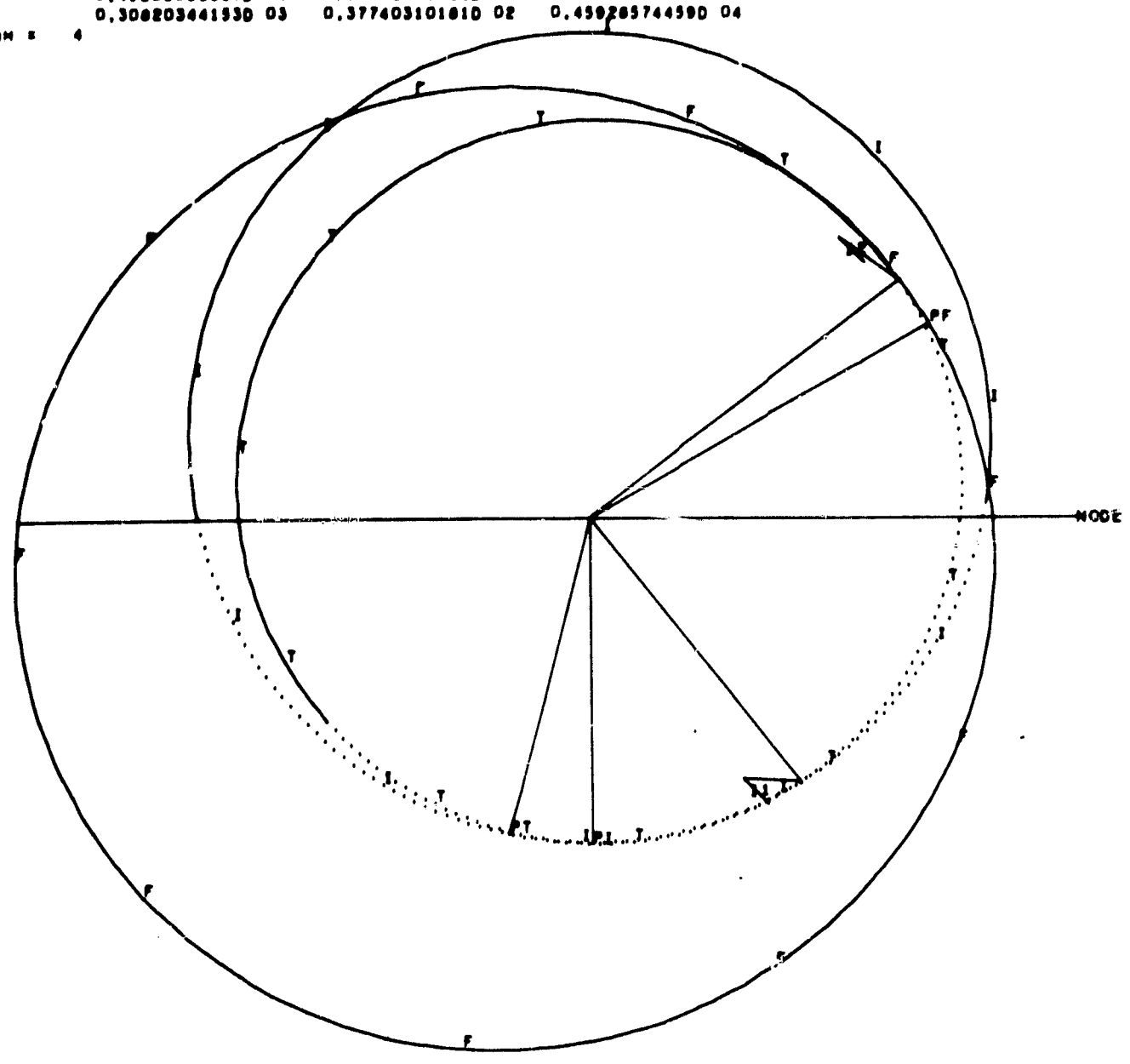


Figure 39.

	••• FINAL TRANSFER CONDITIONS •••		CENTER IS EARTH , UNIT SYSTEM...MILES, FT...	
IMPULSE	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
FIRST VECTOR	-0.638065716060 04	0.491118256110 05	0.204123313460 04	0.241682228330 04
SECOND VECTOR	-0.416378209450 05	0.102979043540 05	0.726246032660 02	0.297695866150 04
TOTAL				0.539378114480 04
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT	
P	0.900000000000D 04	0.600000000000D 04	0.461180236260 04	
E	0.200000000000D 00	0.200000000000D 00	0.168164465510 00	
I (DEG)	0.900000000000D 01	0.0	0.196690320610 00	
R A NODE (DEG)	0.0	0.0	0.736465275200 02	
ARG PER (DEG)	-0.900000000000D 02	0.300000000000D 02	-0.135980725280 03	
A	0.920633333330 04	0.825000000000D 04	0.474601632250 04	
PERIOD	0.763696370720 04	0.100390469040 05	0.664303302590 04	
TRAN PT RAD	0.903648293040 04	0.524144579280 04		
PHI-1, PHI-2, P	0.177811398970 03	0.736465275200 02	0.461180236260 04	

RANK OF OPTIMUM = 3

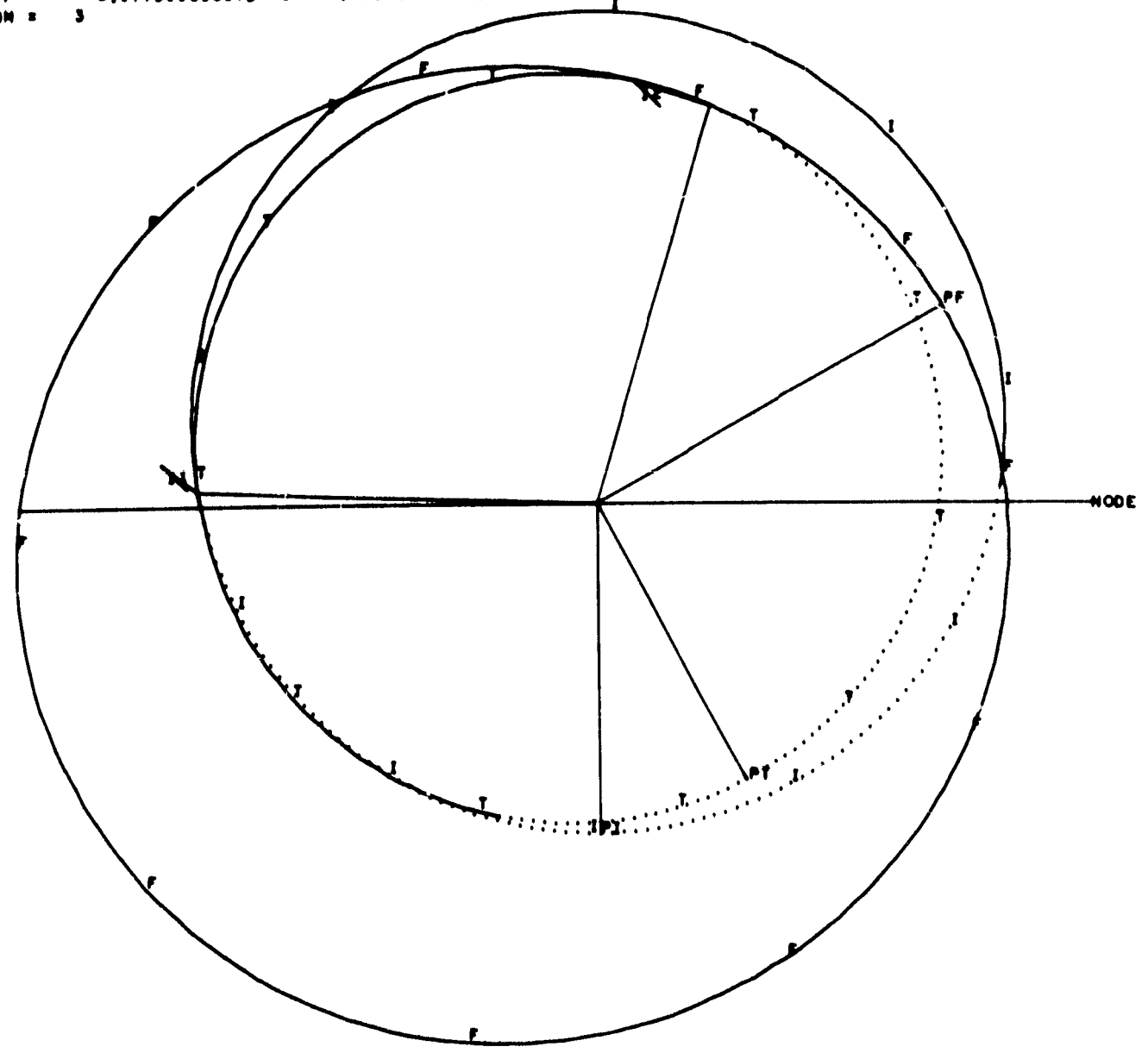


Figure 40.

	*** FINAL TRANSFER CONDITIONS ***		CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC	
IMPULSE	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
FIRST VECTOR	0.235317200060 05	-0.366372306940 05	-0.130312126280 04	0.356509105980 04
SECOND VECTOR	0.341400704200 05	0.271343914640 05	0.128415660500 04	0.177605763770 04
TOTAL				0.534314069350 04
	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT	
P	0.500000000000D 04	0.600000000000D 04	0.661779047170 04	
E	0.200000000000D 00	0.200000000000D 00	0.200028036550 00	
I (DEG)	0.500000000000D 01	0.0	0.334965811000 01	
R A NODE (DEG)	0.0	0.0	-0.617366055660 02	
ARG PER (DEG)	-0.900000000000D 02	0.300000000000D 02	0.612275188370 02	
A	0.520833333330 04	0.625000000000D 04	0.689361227750 04	
PERIOD	0.763696370720 04	0.100390495040 05	0.116290128500 05	
TRAN PT RAD	0.575224920930 04	0.603658756940 04		
PHI-1, PHI-2, P	0.408343237060 02	0.288265394430 03	0.661779047170 04	

RANK OF OPTIMUM = 2

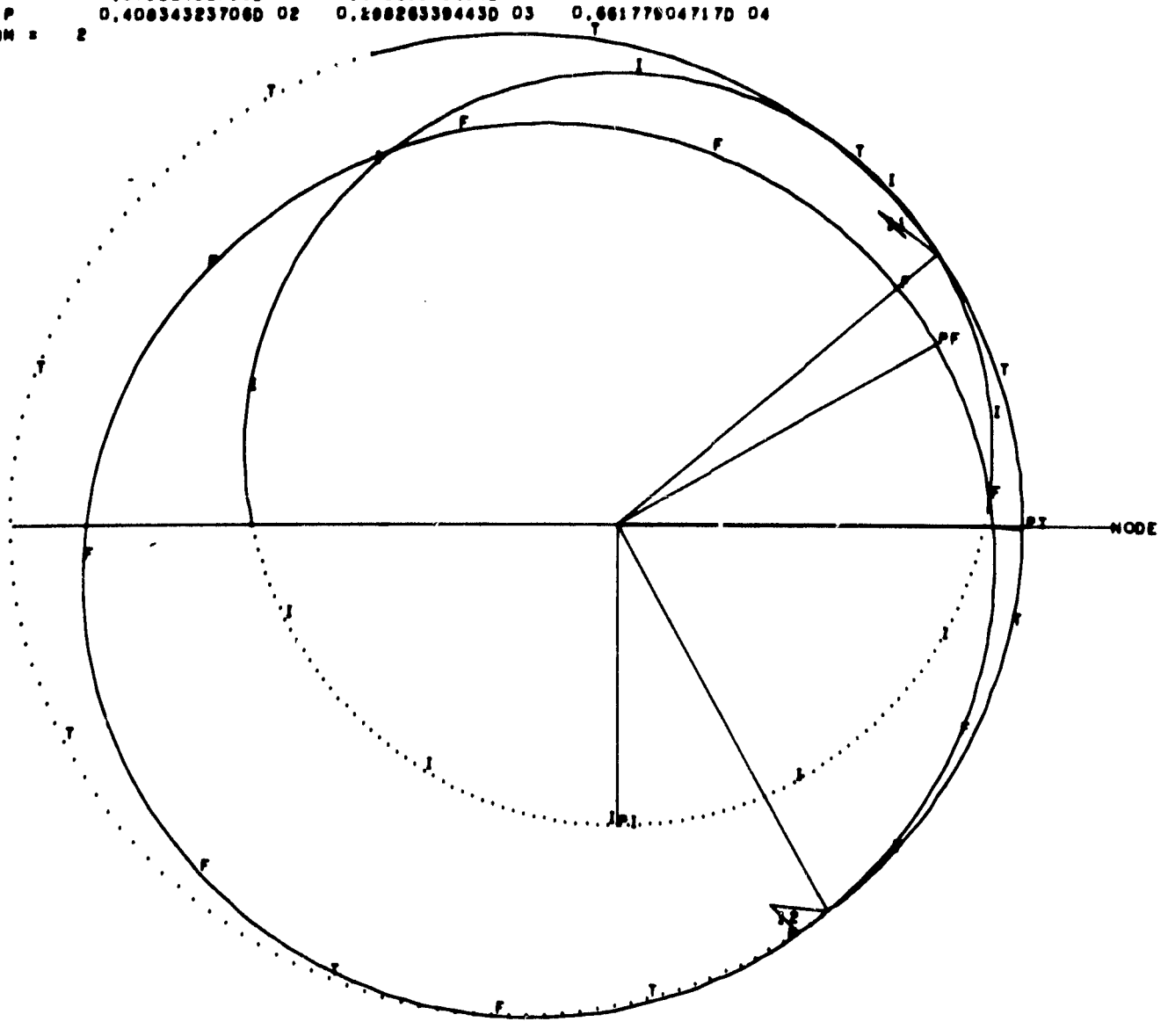


Figure 41.

```
                    *** FINAL TRANSFER CONDITIONS ***        CENTER IS  EARTH  , UNIT SYSTEM...MILES, FT/SEC
                    X COMPONENT          Y COMPONENT         Z COMPONENT          MAGNITUDE
IMPULSE             -0.268204320280D 04  0.951073660300D 03  0.353478794390D 03   0.286829821330D 04
FIRST VECTOR        -0.953402494030D 03  0.704036567310D 03  0.165312469070D 04   0.203433501060D 04
SECOND VECTOR                                                                    0.490265122390D 04
TOTAL

                    INITIAL ORBIT        FINAL ORBIT         TRANSFER ORBIT
P                   0.500000000000D 04   0.600000000000D 04  0.664484985440D 04
E                   0.200000000000D 00   0.200000000000D 00  0.104330516440D 00
I (DEG)             0.300000000000D 01   0.0                 0.324524565740D 01
R A NODE (DEG)      0.0                  0.0                 0.759679889320 01
ARG PER (DEG)       -0.900000000000D 02  0.300000000000D 02  0.212397268780D 02
A                   0.520833333330D 04   0.625000000000D 04  0.871797403890D 04
PERIOD              0.763696370720D 04   0.100390495040D 05  0.111874228170 05
TRAN PT RAD         0.618666166930D 04   0.736062370080D 04
PHI-1, PHI-2, P     0.730151796920D 02   0.187558796890D 03  0.664484985440D 04
RANK OF OPTIMUM =   1
```

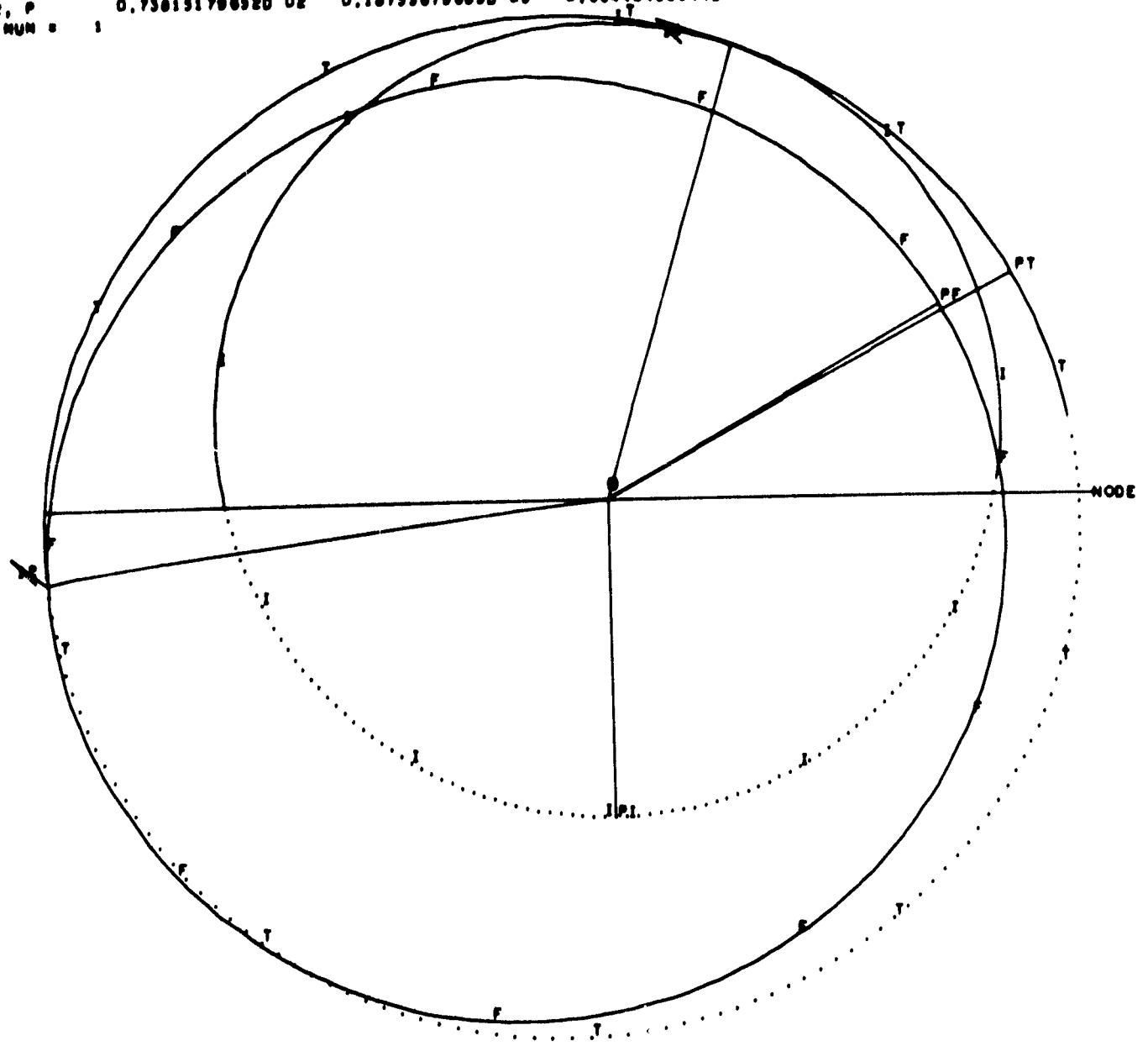


Figure 42.

ADAPTIVE STEEP DESCENT OPTIMIZATION OF TWO-IMPULSE TRANSFER

PHI1	PHI2	P	DI/DP1	DI/DP2	DI/DP	----IMPULSE----	-------GAINS-------	ITNO
70.0000	180.0000	6.6030490 03	-64.661030	-110.320751	0.003941	4.95751944175005400 03	1.00 00 1.00 00 1.00 02	0 0
71.8537	187.8272	6.6031290 03	-3.655584	16.983793	-5.435149	4.91447009111938900 03	1.00 00 1.20 00 1.20 02	5 0
72.5993	186.9274	6.6153580 03	2.031992	-2.697901	-3.737616	4.90818570755662000 03	1.00 00 1.20 00 2.70 02	10 2
73.4433	187.3876	6.6431460 03	-8.876129	-2.202308	0.187799	4.90281892502034200 03	1.00 00 1.40 00 3.70 02	15 7
73.6710	187.5104	6.6436010 03	-2.698855	-0.458387	-0.039309	4.90267421255904600 03	1.00 00 2.40 00 4.30 02	20 7
73.7972	187.5490	6.6447230 03	-0.372094	-0.094194	0.000661	4.90265159073225600 03	1.00 00 1.40 00 3.70 02	25 12
73.8141	187.5556	6.6447950 03	0.038082	-0.010383	-0.008548	4.90265124585072600 03	1.00 00 1.60 00 5.10 02	30 12
73.8148	187.5568	6.6448470 03	-0.008350	0.000165	0.000031	4.90265122400890300 03	1.00 00 2.30 00 3.70 02	35 17
73.8151	187.5568	6.6448490 03	-0.000334	-0.000024	-0.000020	4.90265122385260600 03	1.00 00 1.90 00 3.10 02	40 21
73.8152	187.5568	6.6448490 03	0.000074	-0.000008	-0.000033	4.90265122385230100 03	1.00 00 1.90 00 3.10 02	41 21

AMP= 0.152587890-04 IFAIL= 21

*** FINAL TRANSFER CONDITIONS *** CENTER IS EARTH , UNIT SYSTEM...MILES, FT/SEC

	X COMPONENT	Y COMPONENT	Z COMPONENT	MAGNITUDE
IMPULSE FIRST VECTOR	-0.268284212940 04	0.951074569290 03	0.353476692410 03	0.286829749150 04
SECOND VECTOR	-0.953404750060 03	0.704835692580 03	0.165312465880 04	0.203435373240 04
TOTAL				0.490265122390 04

	INITIAL ORBIT	FINAL ORBIT	TRANSFER ORBIT
P	0.5CC000000000 04	C.600000000000 04	0.664484936610 04
E	0.2CC000000000 00	0.200000000000 00	0.104330532960 00
I (DEG)	0.5CCCCC000000 C1	0.0	0.524524587510 01
R A NODE (DEG)	0.0	0.0	0.755679123860 01
ARG PER (DEG)	-0.9CC000000000 02	0.300000000000 02	0.212396622870 02
A	0.52C833333330 04	0.625000000000 04	0.671797356860 04
PERIOD	0.762696370720 C4	C.100390495040 05	0.111874216420 05
TRAN PT RAD	0.618868173590 C4	C.736062363290 04	
PHI-1, PHI-2, P	0.738151617510 02	0.187556791240 03	0.664484936610 04

ELAPSED EXECUTION TIME = 79.233

Figure 43.

```
INITIAL CONDITIONS FOR QUASILINEARIZATION FROM IMPULSIVE TRANSFER

MIN. NO. OF STEPS                          6.000 00        6.000 00        6.000 00        6.000 00        6.000 00
MAX. STEP SIZE ALLOWED      2.000000D 00   2.000000-01     2.0000000 00    2.0000000-01    2.0000000 00
THE NUMBER OF STEPS IN THE FIVE INTERVALS IS          6          11          70           6           6

THUS KSUB =          1              7              18             88             94            100
 AND TSUB =      0.0          2.651402D 02   4.647805D 02   3.943991D 03   4.060927D 03   4.326067D 03
 AND MASS =      1.000000D 03 1.000000D 03   8.003597D 02   8.003597D 02   6.834232D 02   6.834232D 02
THE PERIGEE TIMES ARE (1, 2, TRANSFER)       -2.952227D 03  -1.087233D 02  -7.861356D 02 SECONDS

                      *** FINAL TRANSFER CONDITIONS ***
C =                                    2.439392D 00
FUEL SPECIFIC IMPULSE =                4.000000D 02 SEC
BURN INCREMENT OVER IMPULSIVE =        0.0          PERCENT
DURATION OF FIRST COAST ARC =          3.000000D 00 SEC
KONTAU =   3

      CENTER IS  EARTH  , UNIT SYSTEM...MILES, FT/SEC
MASS RATE/INITIAL MASS =          1.000000-03 PER SEC
THRUST TO WEIGHT AT START =       4.000000-01
FINAL TIME PERTURBATION =         1.000000-01 SEC
LOS =   1
MAXSTP =    100
```

Figure 44.

THE PRIMER COMPUTATION

IMPULSE CATA ON TRANSFER ORBIT

	IMPULSE COMPONENTS RADIAL	CIRCUM.	BINORMAL	RADIUS	TRUE ANOMALY	HEADING
FIRST	1.91683970 02	2.85233970 03	2.33551170 02	6.18868170 03	7.8624965D-01	3.69003680-02
SECOND	8.52432310 02	-9.71771410 02	1.57086430 03	7.36062360 03	1.58760340 02	-9.15468110-02

A2	B2	A4	A6	C1
-5.72473240 05	-7.11755820 05	1.17125380 05	1.15004920 08	8.90166310 04

PRIMER VECTOR AT THE BURNS

	LENGTH	LENGTH RATE	DERIVATIVE
FIRST	2.99273590 05	2.09486560-05	1.78525690 02
SECOND	2.99273060 05	-1.34415930-05	9.26828350 01

PRIMER ON INITIAL AND FINAL ARCS

	TRUE ANOMALY	HEADING	DLAM1	DLAM2
INITIAL	7.8624965D-01	2.43814940-02	2.61527040 01	1.34045380 04
FINAL	2.77089060 00	0.0	1.29150420 01	0.0

	A2	B2	A4	A6	C1	C1(CHECK)
INITIAL	-3.48278100 05	8.43856860 05	-1.11794870 05	9.61400660 07	5.32726970 04	5.32727350 04
FINAL	-5.59355670 05	-7.18881270 05	9.56940300 04	-2.94528910 07	9.61507610 04	9.61507730 04

Figure 45.

NUMBERS AND VARIABLES ARE
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
XLST R, TH, PH, RH, X, Y, M, K, L1, L2, L4, L5, L6, L3
XKS2 R, PH, RH, Y, M, L1, K, L4, L6, L3
XKS3 R, PH, RH, Y, M, L1, K, L4, L6, TH, X, L2, L5, L3

TIME	R	THETA	PHI	RHO	X	Y	MASS
0.0	6.C68172E 03	7.6785C9E-02	1.074573E 00	4.152178E-01	2.464208E-05	5.99C909E-04	1.CC000CE 03
44.19	6.C86074E C3	7.784438E-02	1.100794E 00	3.949353E-01	2.330259E-05	5.916925E-04	1.C0000CE 03
88.38	6.103074E 03	7.884479E-02	1.126869E 00	3.745008E-01	2.197568E-05	5.8E4930E-04	1.C000CCE 03
132.57	6.119172E C3	7.978678E-02	1.152808E 00	3.539253E-01	2.066075E-05	5.8548SCE-C4	1.C00000E 03
176.76	6.134352E 03	8.067095E-02	1.178617E 00	3.332196E-01	1.935723E-05	5.826771E-04	1.C00000E 03
220.95	6.148617E 03	8.149773E-02	1.204308E 00	3.123939E-01	1.806454E-05	5.8C0545E-C4	1.C00000E 03
265.14	6.161961E 03	8.226764E-02	1.229886E 00	2.914578E-01	1.678204E-05	5.776184E-C4	1.C0900CE 03
283.29	6.167328E 03	8.258742E-02	1.240454E 00	2.920332E-01	1.711366E-05	5.845800E-04	9.818508E 02
301.44	6.172508E 03	8.291382E-02	1.251.46E 00	2.926086E-01	1.744530E-05	5.915419E-04	9.837017E 02
319.59	6.177539E 03	8.324617E-02	1.261554E 00	2.931840E-01	1.777694E-05	5.985035E-04	9.855529E 02
337.74	6.182465E 03	8.358377E-02	1.272904E 00	2.937594E-01	1.810858E-05	6.C54654E-04	9.874033E 02
355.89	6.187328E 03	8.392578E-02	1.283968E 00	2.943348E-01	1.844022E-05	6.124270E-04	9.C92542E 02
374.03	6.192168E 03	8.427155E-02	1.295155E CC	2.949103E-01	1.877185E-05	6.193885E-04	8.511C5CE 02
392.18	6.197027E 03	8.462036E-02	1.306461E 00	2.954857E-01	1.910349E-05	6.263505E-04	8.729550E 02
410.33	6.2C1953E C3	8.497149E-02	1.317890E 00	2.960610E-01	1.943513E-05	6.333124E-04	8.548066E 02
428.48	6.206984E C3	8.532411E-02	1.329438E 00	2.966365E-01	1.976677E-05	6.402740E-04	8.366575E 02
446.63	6.212164E C3	8.567756E-02	1.341106E 00	2.972119E-01	2.009841E-05	6.472359E-04	8.185083E 02
464.78	6.217535E 03	8.603120E-02	1.352893E 00	2.977873E-01	2.043004E-05	6.541978E-04	8.C03591E 02
514.48	6.232543E 03	8.699900E-02	1.385333E 00	3.060606E-01	1.851334E-05	6.5116C3E-04	8.C03591E 02
564.19	6.247953E C3	8.787161E-02	1.417621E 00	3.139759E-01	1.660192E-05	6.4S0513E-04	8.C03591E 02
613.89	6.263746E 03	8.864945E-02	1.449752E 00	3.215292E-01	1.469892E-05	6.448759E-04	8.C03591E 02
663.59	6.279910E C3	8.933294E-02	1.481724E 00	3.287171E-01	1.280733E-05	6.416393E-04	8.C03591E 02
713.30	6.296418E C3	8.992279E-02	1.513534E CC	3.355368E-01	1.C92997E-05	6.383467E-04	8.C03591E 02
763.00	6.313258E C3	9.041977E-02	1.545178E 00	3.419862E-01	9.069528E-06	6.350030E-04	8.C03591E 02
812.70	6.330406E C3	9.082472E-02	1.576656E 00	3.480641E-01	7.228501E-06	6.316134E-04	8.C03591E 02
862.40	6.347852E C3	9.113866E-02	1.607965E 00	3.537695E-01	5.409242E-06	6.281829E-04	8.C03591E 02
912.11	6.365566E C3	9.136277E-02	1.639101E 00	3.591023E-01	3.613936E-06	6.247165E-04	8.C03591E 02
961.81	6.383543E C3	9.149832E-02	1.670064E 00	3.640630E-01	1.844603E-06	6.212189E-04	8.C03591E 02
1011.51	6.401750E C3	9.154665E-02	1.700853E 00	3.686525E-01	1.031035E-07	6.176953E-04	8.C03591E 02
1061.22	6.420180E C3	9.150910E-02	1.731466E 00	3.728722E-01	-1.608862E-06	6.141500E-04	8.C03591E 02
1110.92	6.438812E 03	9.138721E-02	1.761903E 00	3.767241E-01	-3.289748E-06	6.1C5874E-04	8.C03591E 02
1160.62	6.457625E 03	9.118259E-02	1.792162E 00	3.802106E-01	-4.938164E-06	6.070123E-04	8.C03591E 02
1210.33	6.476602E 03	9.089690E-02	1.822244E 00	3.833346E-01	-6.552867E-06	6.C34288E-04	8.C03591E 02
1260.03	6.495723E C3	9.053177E-02	1.852147E 00	3.860993E-01	-8.132758E-06	5.998414E-04	8.C03591E 02
1309.73	6.514977E C3	9.008902E-02	1.881871E 00	3.885083E-01	-9.676874E-06	5.962537E-04	8.C03591E 02
1359.43	6.534336E 03	8.957046E-02	1.911418E 00	3.905655E-01	-1.118439E-05	5.926699E-04	8.C03591E 02
1409.14	6.553793E 03	8.897787E-02	1.940786E 00	3.922752E-01	-1.265460E-05	5.890937E-04	8.C03591E 02
1458.84	6.573328E 03	8.831316E-02	1.969977E 00	3.936421E-01	-1.408692E-05	5.855280E-04	8.C03591E 02
1508.54	6.592918E 03	8.757818E-02	1.998992E 00	3.946708E-01	-1.548089E-05	5.819786E-04	8.C03591E 02
1558.25	6.612555E 03	8.677489E-02	2.027830E 00	3.953664E-01	-1.683616E-05	5.784461E-04	8.C03591E 02
1607.95	6.632215E 03	8.590519E-02	2.056494E 00	3.957343E-01	-1.815247E-05	5.749350E-04	8.C03591E 02

Figure 46.

1657.65	6.651887E 03	8.497107E-02	2.084983E 00	3.957800E-01	-1.942964E-05	5.714481E-04	8.C03591E 02
1707.36	6.671555E 03	8.397442E-02	2.113299E 00	3.955089E-01	-2.066763E-05	5.679883E-04	8.C03591E 02
1757.06	6.691199E 03	8.291721E-02	2.141444E 00	3.949269E-01	-2.186645E-05	5.645582E-04	8.C03591E 02
1806.76	6.710805E 03	8.180141E-02	2.169420E 00	3.940400E-01	-2.302616E-05	5.611605E-04	8.C03591E 02
1856.46	6.730363E 03	8.062893E-02	2.197227E 00	3.928542E-01	-2.414694E-05	5.577975E-04	8.C03591E 02
1906.17	6.749852E 03	7.940173E-02	2.224869E 00	3.913755E-01	-2.522900E-05	5.544717E-04	8.C03591E 02
1955.87	6.769262E C3	7.812166E-02	2.252346E 00	3.896102E-01	-2.627264E-05	5.511851E-04	8.C03591E 02
2005.57	6.788578E 03	7.6790695-02	2.279661E 00	3.875645E-01	-2.727819E-05	5.479399E-04	8.C03591E 02
2055.28	6.807785E 03	7.541066E-02	2.306815E 00	3.852447E-01	-2.824602E-05	5.447378E-04	8.C03591E 02
2104.98	6.826871E 03	7.398349E-02	2.333812E 00	3.826572E-01	-2.917658E-05	5.415808E-04	8.C03591E 02
2154.68	6.845820E C3	7.251096E-02	2.360653E 00	3.798082E-01	-3.007030E-05	5.384702E-04	8.C03591E 02
2204.39	6.864621E C3	7.099491E-02	2.387340E 00	3.767043E-01	-3.092771E-05	5.354080E-04	8.C03591E 02
2254.09	6.883262E 03	6.943715E-02	2.413877E 00	3.733516E-01	-3.174934E-05	5.323954E-04	8.C03591E 02
2303.79	6.901730E 03	6.783938E-02	2.440264E 00	3.697565E-01	-3.253571E-05	5.294336E-04	8.C03591E 02
2353.49	6.920012E 03	6.620347E-02	2.466506E 00	3.659254E-01	-3.328743E-05	5.265239E-04	8.C03591E 02
2403.20	6.938102E C3	6.453103E-02	2.492604E 00	3.618646E-01	-3.400508E-05	5.236675E-04	8.C03591E 02
2452.90	6.955980E C3	6.282371E-02	2.518562E 00	3.575801E-01	-3.469926E-05	5.208654E-04	8.CC3591E 02
2502.60	6.973645E 03	6.108326E-02	2.544383E 00	3.530784E-01	-3.536059E-05	5.181185E-04	8.C03591E 02
2552.31	6.991074E 03	5.931120E-02	2.570067E 00	3.483655E-01	-3.599970E-05	5.154277E-04	8.CC3591E 02
2602.01	7.008270E 03	5.750917E-02	2.595620E 00	3.434475E-01	-3.6547225-05	5.127937E-04	8.C03591E 02
2651.71	7.025215E 03	5.567870E-02	2.621043E 00	3.383305E-01	-3.710379E-05	5.102172E-04	8.C03591E 02
2701.42	7.041898E C3	5.382133E-02	2.646340E 00	3.330203E-01	-3.763003E-05	5.076989E-04	8.C03591E 02
2751.12	7.058312E 03	5.193854E-02	2.671513E 00	3.275230E-01	-3.812659E-05	5.052395E-04	8.C03591E 02
2800.82	7.074453E 03	5.003180E-02	2.696565E 00	3.218442E-01	-3.859408E-05	5.028390E-04	8.C03591E 02
2850.52	7.090305E 03	4.810253E-02	2.721498E 00	3.159899E-01	-3.903314E-05	5.004983E-04	8.C03591E 02
2900.23	7.105863E 03	4.615213E-02	2.746318E 00	3.099656E-01	-3.944439E-05	4.982175E-04	8.C03591E 02
2949.93	7.121113E 03	4.418197E-02	2.771026E 00	3.037768E-01	-3.982845E-05	4.959973E-04	8.C03591E 02
2999.63	7.136055E 03	4.219339E-02	2.795625E 00	2.974292E-01	-4.018589E-05	4.938375E-04	8.C03591E 02
3049.34	7.150680E C3	4.018768E-02	2.820117E 00	2.909281E-01	-4.051731E-05	4.917386E-04	8.C03591E 02
3099.04	7.164973E C3	3.816614E-02	2.844507E 00	2.842790E-01	-4.082329E-05	4.897006E-04	8.C03591E 02
3148.74	7.178934E 03	3.613061E-02	2.868797E 00	2.774870E-01	-4.110440E-05	4.877241E-04	8.C03591E 02
3198.45	7.192555E 03	3.408052E-02	2.892991E 00	2.705574E-01	-4.136119E-05	4.858086E-04	8.C03591E 02
3248.15	7.205828E 03	3.201886E-02	2.917090E 00	2.634952E-01	-4.159419E-05	4.839546E-04	8.C03591E 02
3297.85	7.218746E 03	2.994619E-02	2.941100E 00	2.563055E-01	-4.180393E-05	4.821622E-04	8.C03591E 02
3347.55	7.231305E 03	2.786367E-02	2.965021E 00	2.489932E-01	-4.199090E-05	4.804311E-04	8.C03591E 02
3397.26	7.243496E 03	2.577242E-02	2.988858E 00	2.415632E-01	-4.215562E-05	4.787615E-04	8.C03591E 02
3446.96	7.255316E 03	2.367351E-02	3.012614E 00	2.340202E-01	-4.229853E-05	4.771533E-04	8.C03591E 02
3496.66	7.266758E 03	2.156804E-02	3.036291E 00	2.263691E-01	-4.242011E-05	4.756066E-04	8.CC3591E 02
3546.37	7.277816E 03	1.945704E-02	3.059894E 00	2.186143E-01	-4.252078E-05	4.741212E-04	8.C03591E 02
3596.07	7.288488E C3	1.734156E-02	3.083423E 00	2.107605E-01	-4.260098E-05	4.726972E-04	8.C03591E 02
3645.77	7.298766E 03	1.522258E-02	3.106883E 00	2.028121E-01	-4.266111E-05	4.713342E-04	8.C03591E 02
3695.48	7.308645E C3	1.310111E-02	3.130278E 00	1.947737E-01	-4.270155E-05	4.700324E-04	8.C03591E 02
3745.18	7.318125E 03	1.097811E-02	3.153608E 00	1.866494E-01	-4.272268E-05	4.687915E-04	8.C03591E 02
3794.88	7.327199E C3	8.854534E-03	3.176879E 00	1.784438E-01	-4.272483E-05	4.676115E-04	8.C03591E 02
3844.58	7.335863E 03	6.731313E-03	3.200093E 00	1.701608E-01	-4.270837E-05	4.664923E-04	8.C03591E 02
3894.29	7.344113E C3	4.609369E-03	3.223252E 00	1.618049E-01	-4.267359E-05	4.654336E-04	8.C03591E 02
3943.99	7.351945E 03	2.489603E-03	3.246361E 00	1.533799E-01	-4.262081E-05	4.644352E-04	8.C03591E 02
3963.48	7.355242E 03	1.727968E-03	3.255383E 00	1.754332E-01	-3.551739E-05	4.6C3737E-04	7.808696E 02
3982.97	7.358988E C3	1.105309E-03	3.264327E 00	1.974866E-01	-2.841387E-05	4.563124E-04	7.8138C1E 02

Figure 46. (Continued)

4002.46	7.363148E C3	6.214036E-04	3.273190E 00	2.195399E-01	-2.131041E-05	4.522589E-04	7.418986E 02
4021.95	7.367699E C3	2.760317E-04	3.281971E 00	2.415933E-01	-1.420694E-05	4.481897E-04	7.224011E 02
4041.44	7.372621E 03	6.897101E-05	3.290666E 00	2.636467E-01	-7.103468E-06	4.441282E-04	7.029116E 02
4060.93	7.377887E C3	0.0	3.299273E 00	2.857001E-01	0.C	4.400669E-04	6.834221E 02
4105.12	7.390191E C3	0.0	3.318686E 00	2.711719E-01	0.0	4.386C26E-04	6.834221E 02
4149.31	7.401852E C3	0.0	3.338037E 00	2.565890E-01	0.0	4.372220E-04	6.834221E 02
4193.50	7.412867E 03	0.0	3.357329E 00	2.419551E-01	0.0	4.359235E-04	6.834221E 02
4237.69	7.423234E 03	0.0	3.376566E 00	2.272737E-01	0.0	4.347067E-04	6.834221E 02
4281.88	7.432953E 03	0.0	3.395750E 00	2.125482E-01	0.0	4.335707E-04	6.834221E 02
4326.C7	7.442020E 03	0.0	3.414886E 00	1.977819E-01	0.0	4.325148E-04	6.834221E 02

TIME	K	LAM 1	LAM 2	LAM 4	LAM 5	LAM 6	LAM 3
0.0	-2.287844E 0C	2.794487E 02	-9.878644E 05	-6.181973E 03	-8.341690E 07	1.8C2713E 09	7.677381E 04
44.19	-1.741115E 0C	2.786533E 02	-9.912226E 05	-2.944023E 03	-4.735402E 07	1.810851E 09	7.677387E 04
88.38	-1.276411E CC	2.777441E 02	-9.938415E 05	2.772844E 02	-1.106809E 07	1.817557E C9	7.677387E 04
132.5?	-8.880908E-01	2.767249E 02	-9.957336E 05	3.481051E 03	2.540317E 07	1.822961E 09	7.677394E 04
176.76	-5.758488E-01	2.755996E 02	-9.969047E 05	6.666211E 03	6.201954E 07	1.826936E 09	7.677387E 04
220.95	-3.332818E-C1	2.743708E 02	-9.973678E 05	9.831652E 03	9.873974E C7	1.829539E 09	7.677387E 04
265.14	-1.581030E-01	2.730420E 02	-9.971309E 05	1.297654E 04	1.355256E 08	1.830773E 09	7.677394E 04
283.29	-9.264410E-C2	2.730947E 02	-9.972877E 05	1.428829E 04	1.506249E 08	1.831132E 09	7.677394E 04
301.44	-4.026943E-02	2.741846E 02	-9.978666E 05	1.563981E 04	1.657053E 08	1.831203E 09	7.677394E 04
319.59	-1.140833E-03	2.760713E 02	-9.987381E 05	1.703045E 04	1.807648E 08	1.830992E 09	7.677400E 04
337.74	2.457982E-02	2.785139E 02	-9.997728E 05	1.845952E 04	1.958017E 08	1.830506E 09	7.677400E 04
355.89	3.673064E-C2	2.812725E 02	-1.000841E 06	1.992634E 04	2.108141E 08	1.829753E 09	7.677400E 04
374.03	3.514974E-C2	2.841062E 02	-1.001814E 06	2.143024E 04	2.258001E 08	1.828741E 09	7.677400E 04
392.18	1.967526E-02	2.867751E 02	-1.002562E 06	2.297054E 04	2.407581E 08	1.827477E 09	7.677400E 04
410.33	-9.854689E-C3	2.890383E 02	-1.002956E 06	2.456656E 04	2.556860E 08	1.825967E 09	7.677400E 04
428.48	-5.360199E-02	2.906558E 02	-1.002866E 06	2.615763E 04	2.705820E 08	1.824221E 09	7.677400E 04
446.63	-1.117285E-C1	2.913867E 02	-1.002163E 06	2.780307E 04	2.854444E 08	1.822244E 09	7.677400E 04
464.78	-1.843961E-C1	2.909912E 02	-1.000717E 06	2.948221E 04	3.002714E 08	1.820046E 09	7.677400E 04
514.48	-4.034618E-01	2.881453E 02	-9.956227E 05	3.412618E 04	3.408259E 08	1.813725E 09	7.677394E 04
564.19	-6.987243E-C1	2.850261E 02	-9.894041E 05	3.871214E 04	3.813023E 08	1.8C6C35E 09	7.677394E 04
613.89	-1.C64087E CC	2.816448E 02	-9.820851E 05	4.323548E 04	4.216840E 08	1.796987E C9	7.677394E 04
663.59	-1.490027E CC	2.780125E 02	-9.736934E 05	4.769184E 04	4.619566E 08	1.788601E C9	7.677400E 04
713.30	-1.972733E 00	2.741409E 02	-9.642544E 05	5.207695E 04	5.020995E 08	1.7748B2E 09	7.677387E 04
763.00	-2.502872E 00	2.700410E 02	-9.537991E 05	5.638724E 04	5.421000E 08	1.761851E 09	7.677400E 04
812.70	-3.074920E CC	2.657249E 02	-9.423570E 05	6.061875E 04	5.819369E 08	1.747519E C9	7.677394E 04
862.40	-3.679160E 0C	2.612039E 02	-9.299623E 05	6.476801E 04	6.215936E 08	1.731908E 09	7.677394E 04
912.11	-4.311784E CC	2.564897E 02	-9.166452E 05	6.883194E 04	6.610501E 08	1.715026E 09	7.677387E 04
961.81	-4.962123E 00	2.515937E 02	-9.024422E 05	7.280769E 04	7.002921E 08	1.696897E 09	7.677394E 04
1011.51	-5.627702E CC	2.465276E 02	-8.873859E 05	7.669219E 04	7.392922E 08	1.677529E 09	7.677394E 04
1061.22	-6.298424E CC	2.413020E 02	-8.715133E C5	8.048312E 04	7.780365E 08	1.656947E 09	7.677394E 04
1110.92	-6.968765E 0C	2.359282E 02	-8.548600E 05	8.417806E 04	8.165038E 08	1.635167E 09	7.677394E 04
1160.62	-7.633963E CC	2.304171E C2	-8.374606E 05	8.777500E 04	8.546706E 08	1.6122C2E C9	7.677394E 04
1210.33	-8.287731E 00	2.247794E 02	-8.193527E 05	9.127187E 04	8.925156E 08	1.588073E 09	7.677394E 04
1260.03	-8.924926E C0	2.190255E C2	-8.005717E 05	9.466706E 04	9.300173E 08	1.562797E 09	7.677387E 04
1309.73	-9.538692E 00	2.131651E 02	-7.811556E 05	9.795906E 04	9.671555E 08	1.536397E 09	7.677387E 04
1359.43	-1.012788E C1	2.072084E 02	-7.611381E 05	1.011465E 05	1.003902E C9	1.5C8884E C9	7.677387E 04

Figure 46. (Continued)

1409.14	-1.C68469E C1	2.011647E 02	-7.405575E 05	1.042282E 05	1.040240E 09	1.4E0286E 09	7.E77394E 04
1458.84	-1.120607E 01	1.950431E 02	-7.194493E 05	1.072033E 05	1.076143E 09	1.450621E 09	7.E77400E 04
1508.54	-1.169087E 01	1.888530E 02	-6.978474E 05	1.100707E 05	1.111583E 09	1.419905E 09	7.E77387E 04
1558.25	-1.213338E C1	1.826023E 02	-6.757874E 05	1.128301E 05	1.146542E 09	1.388163E 09	7.E77387E 04
1607.95	-1.253284E C1	1.762994E 02	-6.533023E 05	1.154807E 05	1.180991E 09	1.355411E 09	7.E77394E 04
1657.65	-1.288582E C1	1.699525E 02	-6.304278E 05	1.180222E 05	1.214907E 09	1.321677E 09	7.E77394E 04
1707.36	-1.319061E 01	1.635686E 02	-6.071955E 05	1.204544E 05	1.248267E 09	1.286979E C9	7.E77394E 04
1757.06	-1.344682E C1	1.571556E 02	-5.836382E 05	1.227771E 05	1.281040E 09	1.251341E 09	7.E77387E 04
1806.76	-1.365389E C1	1.507199E 02	-5.597858E 05	1.249904E 05	1.313204E 09	1.214781E 09	7.E77387E 04
1856.46	-1.380933E C1	1.442681E 02	-5.356714E 05	1.270944E 05	1.344736E 09	1.177329E 09	7.E77387E 04
1906.17	-1.391486E C1	1.378066E 02	-5.113229E 05	1.290892E 05	1.375606E 09	1.139004E 09	7.E77394E 04
1955.87	-1.396953E C1	1.313411E 02	-4.867704E 05	1.309752E 05	1.405792E 09	1.059831E 09	7.E77394E 04
2005.57	-1.397372E C1	1.248773E 02	-4.620419E 05	1.327529E 05	1.435269E 09	1.059836E 09	7.E77387E 04
2055.28	-1.392896E C1	1.184207E 02	-4.371646E 05	1.344226E 05	1.464010E 09	1.019043E 09	7.E77387E 04
2104.98	-1.383619E 01	1.119760E 02	-4.121648E 05	1.359850E 05	1.491991E 09	9.774766E 08	7.E77394E 04
2154.68	-1.369732E C1	1.055483E 02	-3.870686E 05	1.374407E 05	1.519185E 09	9.351631E 08	7.E77387E 04
2204.39	-1.351388E C1	9.914169E 01	-3.619001E 05	1.387904E 05	1.545569E 09	8.921203E C8	7.E77394E 04
2254.09	-1.328719E C1	9.276047E 01	-3.366831E 05	1.400350E 05	1.571119E 09	8.483986E 08	7.E77394E 04
2303.79	-1.302012E 01	8.640866E 01	-3.114414E 05	1.411752E 05	1.595810E 09	8.040031E 08	7.E77394E 04
2353.49	-1.271572E C1	8.008989E 01	-2.861961E 05	1.422119E 05	1.619617E 09	7.589665E 08	7.E77387E 04
2403.20	-1.237435E 01	7.380721E 01	-2.609682E 05	1.431461E 05	1.642524E 09	7.133181E 08	7.E77394E 04
2452.90	-1.200099E C1	6.756451E 01	-2.357797E 05	1.439787E 05	1.664500E 09	6.670879E 08	7.E77387E 04
2502.60	-1.159657E C1	6.136418E 01	-2.106487E 05	1.447108E 05	1.685528E 09	6.203026E 08	7.E77394E 04
2552.31	-1.116663E 01	5.520929E 01	-1.855939E 05	1.453434E 05	1.705580E 09	5.729905E 08	7.E77394E 04
2602.01	-1.071117E 01	4.910222E 01	-1.606339E 05	1.458776E 05	1.724644E 09	5.251837E 08	7.E77394E 04
2651.71	-1.023494E C1	4.304562E 01	-1.357861E 05	1.463146E 05	1.742693E 09	4.769124E 08	7.E77387E 04
2701.42	-9.741565E CC	3.704144E 01	-1.110660E 05	1.466553E 05	1.759707E 09	4.282043E 08	7.E77387E 04
2751.12	-9.233713E CC	3.109167E 01	-8.648919E 04	1.469011E 05	1.775667E 09	3.790904E 08	7.E77394E 04
2800.82	-8.713480E CC	2.519835E 01	-6.207159E 04	1.470530E 05	1.790560E 09	3.296049E 08	7.E77394E 04
2850.52	-8.185627E CC	1.936322E 01	-3.782720E 04	1.471122E 05	1.804361E 09	2.797775E 08	7.E77394E 04
2900.23	-7.652060E CC	1.358780E 01	-1.376964E 04	1.470801E 05	1.817058E 09	2.296410E 08	7.E77394E 04
2949.93	-7.117920E CC	7.873325E CC	1.008911E 04	1.469577E 05	1.828627E 09	1.792244E 08	7.E77394E 04
2999.63	-6.584162E CC	2.221296E CC	3.373583E 04	1.467463E 05	1.839061E 09	1.285631E 08	7.E77394E 04
3049.34	-6.053832E CC	-3.367088E 00	5.715887E 04	1.464471E 05	1.848344E 09	7.768986E 07	7.E77400E 04
3099.04	-5.532075E CC	-8.898552E 00	8.034606E 04	1.460614E 05	1.856455E 09	2.663931E 07	7.E77387E 04
3148.74	-5.018889E CC	-1.434554E 01	1.032889E 05	1.455904E 05	1.863389E 09	-2.455891E 07	7.E77394E 04
3198.45	-4.518848E CC	-1.973994E 01	1.259762E 05	1.450353E 05	1.869131E 09	-7.586917E 07	7.E77394E 04
3248.15	-4.033856E CC	-2.506400E 01	1.483987E 05	1.443974E 05	1.873670E 09	-1.272576E 08	7.E77394E 04
3297.85	-3.566770E 00	-3.031998E 01	1.705473E 05	1.436778E 05	1.876995E 09	-1.786903E 08	7.E77394E 04
3347.55	-3.119876E CC	-3.550745E 01	1.924143E 05	1.428778E 05	1.879097E 09	-2.301347E 08	7.E77400E 04
3397.26	-2.695841E CC	-4.062543E 01	2.139897E 05	1.419987E 05	1.879967E 09	-2.815519E 08	7.E77400E 04
3446.96	-2.295999E CC	-4.567361E 01	2.352671E 05	1.410417E 05	1.879600E 09	-3.329098E 08	7.E77400E 04
3496.66	-1.923016E CC	-5.065158E 01	2.562389E 05	1.400079E 05	1.877985E 09	-3.841733E 08	7.E77400E 04

3546.37	-1.578606E CC	-5.555890E 01	2.768984E 05	1.388987E 05	1.875120E 05	-4.353073E 08	7.677394E 04
3596.07	-1.264104E 0C	-6.039522E 01	2.972392E 05	1.377151E 05	1.871002E 05	-4.862771E 08	7.677408E 04
3645.77	-9.821762E-01	-6.516025E 01	3.172551E 05	1.364584E 05	1.865622E 05	-5.370468E 08	7.677408E 04
3695.48	-7.335843E-01	-6.985353E 01	3.369394E 05	1.351299E 05	1.858981E 05	-5.875791E 08	7.677387E 04
3745.18	-5.188999E-01	-7.447514E 01	3.562888E 05	1.337306E 05	1.851080E 05	-6.378437E 08	7.677394E 04
3794.88	-3.400280E-01	-7.902452E 01	3.752962E 05	1.322617E 05	1.841917E 05	-6.878006E 08	7.677394E 04
3844.58	-1.979210E-01	-8.350166E 01	3.939579E 05	1.307244E 05	1.831492E 05	-7.374167E 08	7.677394E 04
3894.29	-9.372199E-02	-8.790642E 01	4.122689E 05	1.291199E 05	1.819806E 05	-7.866560E 08	7.677394E 04
3943.99	-2.857378E-02	-9.223860E 01	4.302250E 05	1.274492E 05	1.806861E 05	-8.354829E 08	7.677394E 04
3963.48	-7.267609E-03	-9.296477E 01	4.372241E 05	1.268074E 05	1.801646E 05	-8.545795E 08	7.677394E 04
3982.97	5.471058E-03	-9.222479E 01	4.441438E 05	1.261929E 05	1.796155E 05	-8.735772E 08	7.677394E 04
4002.46	9.509627E-03	-9.073713E 01	4.509899E 05	1.256064E 05	1.790386E 05	-8.924746E 08	7.677394E 04
4021.95	4.715491E-03	-8.922015E 01	4.577684E 05	1.250491E 05	1.784341E 05	-9.112701E 08	7.677394E 04
4041.44	-9.043947E-03	-8.839233E 01	4.644851E 05	1.245216E 05	1.778020E 05	-9.299620E 08	7.677394E 04
4060.93	-3.190130E-02	-8.897211E 01	4.711461E 05	1.240251E 05	1.771421E 05	-9.485509E 08	7.677394E 04
4105.12	-9.414226E-02	-9.242542E 01	4.861904E 05	1.229349E 05	1.756146E 05	-9.905779E 08	7.677394E 04
4149.31	-1.862769E-01	-9.583012E 01	5.010036E 05	1.218022E 05	1.739846E 05	-1.032188E 09	7.677408E 04
4193.50	-3.060740E-01	-9.918735E 01	5.155848E 05	1.206274E 05	1.722533E 05	-1.073356E 09	7.677394E 04
4237.69	-4.501876E-01	-1.024982E 02	5.299331E 05	1.194114E 05	1.704221E 05	-1.114059E 09	7.677394E 04
4281.88	-6.154944E-01	-1.057637E 02	5.440471E 05	1.181547E 05	1.684925E 05	-1.154274E 09	7.677408E 04
4326.07	-7.997635E-01	-1.089848E 02	5.579256E 05	1.168578E 05	1.664660E 05	-1.193977E 09	7.677394E 04

EXEC.TIME (MAIN PROG) 1.500

Figure 46. (Continued)

OUTPUT IIT = 5

TIME	PRIMER MAG	SWITCH	STEER-PSI	STEER-NU	HAMILTONIAN	TOT. ENERGY	ANGULAR MOM.
0.0	2.9834D 05	-2.2878D 00	-2.6410D 00	9.1189D 01	3.1235D-04	-9.1809126966 CO	2.1867155859 04
4.42238D 01	2.9856D 05	-1.7411D 00	-1.4933D 00	9.0565D 01	7.5328D-05	-9.1809225346 CO	2.1867138678 04
8.84576D 01	2.9875D 05	-1.2764D 00	-3.4781D-01	8.9947D 01	1.7703D-04	-9.1809295856 CO	2.1867128240 04
1.32686D 02	2.9891D 05	-8.8809D-01	7.9578D-01	8.9333D 01	-7.4068D-05	-9.1809159476 00	2.1867149360 04
1.76915D 02	2.9904D 05	-5.7585D-01	1.9375D 00	8.8722D 01	1.6903D-04	-9.1809324846 CO	2.1867123546 04
2.21144D 02	2.9914D 05	-3.3328D-01	3.0773D 00	8.8114D 01	1.1273D-04	-9.1809200596 CO	2.1867127318 04
2.65373D 02	2.9921D 05	-1.5810D-01	4.2154D 00	8.7508D 01	-4.2221D-05	-9.1809265446 00	2.1867142668 04
2.83538D 02	2.9926D 05	-9.2644D-02	4.6811D 00	8.7254D 01	-1.2424D 00	-9.0035676656 00	2.2168820366 04
3.01703D 02	2.9931D 05	-4.0269D-02	5.1459D 00	8.6993D 01	-2.0644D 00	-8.8247516176 CO	2.2470065286 04
3.19868D 02	2.9935D 05	-1.1408D-03	5.6097D 00	8.6723D 01	-2.5231D 00	-8.6443156176 CO	2.2771097736 04
3.38033D 02	2.9937D 05	2.4580D-02	6.0727D 00	8.6445D 01	-2.6513D 00	-8.4620572186 00	2.3072217056 04
3.56198D 02	2.9938D 05	3.6731D-02	6.5348D 00	8.6159D 01	-2.5433D 00	-8.2777867176 CO	2.3373703156 04
3.74363D 02	2.9939D 05	3.5150D-02	6.9960D 00	8.5864D 01	-2.2579D 00	-8.0913036656 CC	2.3675856876 04
3.92528D 02	2.9938D 05	1.9675D-02	7.4564D 00	8.5562D 01	-1.8526D 00	-7.9024036126 CO	2.3978988406 04
4.10693D 02	2.9935D 05	-9.8547D-03	7.9159D 00	8.5251D 01	-1.3751D 00	-7.7108282716 CO	2.4283489946 04
4.28858D 02	2.9932D 05	-5.3602D-02	8.3744D 00	8.4932D 01	-8.5915D-01	-7.5163746986 CO	2.4589673816 04
4.47023D 02	2.9927D 05	-1.1173D-01	8.8319D 00	8.4605D 01	-3.8172D-01	-7.3187874376 CO	2.4897935706 04
4.65188D 02	2.9921D 05	-1.8440D-01	9.2884D 00	8.4270D 01	-1.0438D-04	-7.1178176C36 CO	2.5208652226 04
5.14935D 02	2.9914D 05	-4.0346D-01	1.0533D 01	8.3337D 01	-1.2335D-04	-7.1178186C586 CO	2.5208651376 04
5.64682D 02	2.9904D 05	-6.9872D-01	1.1776D 01	8.2401D 01	-1.3017D-04	-7.1178179886C CO	2.5208651426 04
6.14428D 02	2.9892D 05	-1.0641D 00	1.3015D 01	8.1463D 01	-1.2375D-04	-7.1178260196D CC	2.5208638466 04
6.64175D 02	2.9878D 05	-1.4900D 00	1.4253D 01	8.0521D 01	-1.7230D-04	-7.1178108776D CO	2.5208663016 04
7.13921D 02	2.9863D 05	-1.9727D CC	1.5488D 01	7.9574D 01	-1.1323D-04	-7.1178186C5CD CO	2.5208651126 04
7.63668D 02	2.9845D 05	-2.5029D 00	1.6721D 01	7.8622D 01	-1.4732D-04	-7.1178146856D CO	2.5208656576 04
8.13415D 02	2.9826D 05	-3.0749D 00	1.7951D 01	7.7664D 01	-1.7539D-04	-7.1178237516D CO	2.5208641726 04
8.63161D 02	2.9807D 05	-3.6792D 00	1.9179D 01	7.6699D 01	-1.9191D-04	-7.1178152196D CO	2.5208655646 04
9.12908D 02	2.9786D 05	-4.3118D CC	2.0405D 01	7.5726D 01	-1.9880D-04	-7.1178277C66D 00	2.5208635056 C4
9.62654D 02	2.9765D 05	-4.9621D 00	2.1627D 01	7.4744D 01	-1.6603D-04	-7.1178147336C CO	2.5208656456 04
1.01240D 03	2.9743D 05	-5.6277D CC	2.2847D 01	7.3751D 01	-1.8158D-04	-7.1178285346 CC	2.5208632776 C4
1.06215D 03	2.9721D 05	-6.2984D 00	2.4064D 01	7.2748D 01	-1.5837D-04	-7.1178256546 CO	2.5208638016 04
1.11189D 03	2.9699D 05	-6.9688D 00	2.5277D 01	7.1733D 01	-1.6051D-04	-7.1178144116D CO	2.5208656696 04
1.16164D 03	2.9677D 05	-7.6340D 00	2.6486D C1	7.0704D 01	-1.5662D-04	-7.1178144536C CO	2.5208656556 C4
1.21139D 03	2.9655D 05	-8.2877D 00	2.7690D 01	6.9660D 01	-1.9936D-04	-7.1178172956D CO	2.5208651756 04
1.26113D 03	2.9635D 05	-8.9249D 00	2.8890D 01	6.8601D 01	-2.1759D-04	-7.1178241416D 00	2.5208640116 04
1.31088D 03	2.9614D 05	-9.5387D CC	3.0085D C1	6.7525D 01	-1.9625D-04	-7.1178154886D CO	2.5208654726 04
1.36063D 03	2.9595D 05	-1.0128D 01	3.1273D 01	6.6430D 01	-2.8307D-04	-7.1178287516D CO	2.5208631966 04
1.41037D 03	2.9577D 05	-1.0685D C1	3.2456C 01	6.5315D 01	-3.0133D-04	-7.1178272686D CO	2.5208634436 C4
1.46012D 03	2.9560D 05	-1.1206D 01	3.3631C 01	6.4178D 01	-1.9236-04	-7.1178134046D CO	2.5208658106 04
1.50987D 03	2.9544D 05	-1.1691D C1	3.4798D 01	6.3018D 01	-2.3[illegible]-04	-7.1178213366D CC	2.5208644316 04
1.55961D 03	2.9529D 05	-1.2133D 01	3.5957D 01	6.1833D 01	-2.5660-04	-7.1178156C76D CO	2.5208648136 04

Figure 47.

1.60936D C3	2.95160 05	-1.2533D C1	3.7106D 01	6.0621D 01	-3.1725D-04	-7.1178235700 C0	2.5208640200 04
1.65911D 03	2.9505D 05	-1.2886D C1	3.8245D 01	5.9381D 01	-3.1098D-04	-7.1178210030 C0	2.5208644590 C4
1.70885D 03	2.9495D 05	-1.3191D 01	3.9373D 01	5.8109D 01	-1.9021D-04	-7.1178123800 C0	2.5208659700 04
1.75860D C3	2.9486D 05	-1.3447D 01	4.0488D 01	5.6805D 01	-1.8820D-04	-7.1178128760 C0	2.5208658810 04
1.80835D C3	2.9479D 05	-1.3654D 01	4.1591D 01	5.5465D 01	-3.6402D-04	-7.1178241060 C0	2.5208638780 04
1.85809D C3	2.9474D 05	-1.3809D 01	4.2679D 01	5.4088D 01	-2.9459D-04	-7.1178184320 C0	2.5208648750 C4
1.90784D 03	2.9471D 05	-1.3915D 01	4.3751D 01	5.2671D 01	-4.2230D-04	-7.1178260350 C0	2.5208635080 04
1.95759D C3	2.9469D 05	-1.3970D 01	4.4807D 01	5.1212D 01	-4.6081D-04	-7.1178267820 C0	2.5208633560 04
2.00733D C3	2.9469D 05	-1.3974D 01	4.5844D 01	4.9708D 01	-3.3337D-04	-7.1178157640 00	2.5208644130 04
2.05708D 03	2.9470D 05	-1.3929D 01	4.6862D 01	4.8157D 01	-3.4940D-04	-7.1178193840 C0	2.5208646660 04
2.10683D 03	2.9473D 05	-1.3836D 01	4.7860D 01	4.6555D 01	-1.7046D-04	-7.1178105510 C0	2.5208662800 04
2.15657D C3	2.9478D 05	-1.3697D 01	4.8835D 01	4.4900D 01	-2.9869D-04	-7.1178155680 C0	2.5208653480 04
2.20632D 03	2.9484D 05	-1.3514D 01	4.9786D 01	4.3190D 01	-2.9529D-04	-7.1178155050 00	2.5208652860 04
2.25607D 03	2.9491D 05	-1.3287D 01	5.0711D 01	4.1422D 01	-2.4554D-04	-7.1178145840 C0	2.5208654590 C4
2.30581D C3	2.9500D 05	-1.3020D 01	5.1609D 01	3.9593D 01	-2.5115D-04	-7.1178148440 C0	2.5208654750 04
2.35556D C3	2.9510D 05	-1.2716D 01	5.2477D 01	3.7701D 01	-5.0916D-04	-7.1178257160 C0	2.5208634410 C4
2.40531D 03	2.9521D C5	-1.2374D 01	5.3315D 01	3.5744D 01	-3.1857D-04	-7.1178163900 C0	2.5208651740 04
2.45505D 03	2.9534D 05	-1.2001D 01	5.4119D 01	3.3719D 01	-4.2439D-04	-7.1178214880 C0	2.5208642120 04
2.50480D 03	2.9547D 05	-1.1597D 01	5.4887D 01	3.1626D 01	-2.6698D-04	-7.1178138950 C0	2.5208656250 04
2.55455D 03	2.9561D 05	-1.1167D 01	5.5619D 01	2.9462D 01	-5.5995D-04	-7.1178256560 00	2.5208633900 04
2.60429D C3	2.9576D 05	-1.0711D 01	5.6310D 01	2.7228D 01	-4.1436D-04	-7.1178191630 C0	2.5208644610 04
2.65404D C3	2.9592D 05	-1.0235D 01	5.6960D 01	2.4924D 01	-3.3304D-04	-7.1178155340 C0	2.5208652880 04
2.70379D 03	2.9608D 05	-9.7416D 00	5.7566D 01	2.2550D 01	-4.6398D-04	-7.1178203690 00	2.5208643600 04
2.75353D C3	2.9624D 05	-9.2337D 00	5.8125D 01	2.0108D 01	-5.0742D-04	-7.1178237590 C0	2.5208637200 04
2.80328D 03	2.9641D 05	-8.7135D 00	5.8636D 01	1.7600D 01	-4.1479D-04	-7.1178185880 00	2.5208646980 04
2.85303D C3	2.9659D 05	-8.1856D 00	5.9097D 01	1.5031D 01	-3.5719D-04	-7.1178174710 C0	2.5208649200 04
2.90277D 03	2.9676D 05	-7.6521D 00	5.9505D 01	1.2405D 01	-2.5244D-04	-7.1178125530 C0	2.5208658470 04
2.95252D C3	2.9694D 05	-7.1179D 00	5.9859D 01	9.7275D 00	-4.7543D-04	-7.1178212580 C0	2.5208641640 04
3.00227D 03	2.9711D 05	-6.5842D 00	6.0157D 01	7.0053D 00	-4.6968D-04	-7.1178212360 00	2.5208641720 04
3.05201D 03	2.9729D 05	-6.0538D 00	6.0398D 01	4.2463D 00	-2.4699D-04	-7.1178126940 C0	2.5208658220 04
3.10176D 03	2.9746D 05	-5.5321D 00	6.0581D 01	1.4592D 00	-5.8970D-04	-7.1178235200 C0	2.5208635990 C4
3.15151D C3	2.9763D 05	-5.0189D 00	6.0705D 01	-1.3469D 00	-4.6133D-04	-7.1178205110 C0	2.5208642130 04
3.20125D C3	2.9779D 05	-4.5188D 00	6.0769D 01	-4.1622D 00	-5.1151D-04	-7.1178214070 C0	2.5208640810 04
3.25100D C3	2.9795D 05	-4.0339D 00	6.0774D 01	-6.9764D 00	-4.8522D-04	-7.1178157110 C0	2.5208643970 04
3.30075D 03	2.9810D 05	-3.5668D 00	6.0720D 01	-9.7796D 00	-3.1899D-04	-7.1178160850 C0	2.5208651500 04
3.35049D C3	2.9825D 05	-3.1199D 00	6.0607D 01	-1.2562D 01	-2.9957D-04	-7.1178147170 C0	2.5208654050 04
3.40024D C3	2.9839D 05	-2.6958D 00	6.0435D 01	-1.5314D 01	-3.5616D-04	-7.1178162540 C0	2.5208650900 04
3.44998D C3	2.9852D 05	-2.2960D 00	6.0208D 01	-1.8026D 01	-2.9691D-04	-7.1178137990 C0	2.5208655710 04
3.49973D 03	2.9864D 05	-1.9230D 00	5.9925D 01	-2.0691D 01	-3.0278D-04	-7.1178143560 C0	2.5208654620 04
3.54948D 03	2.9876D 05	-1.5786D 00	5.9588D 01	-2.3302D 01	-4.2288D-04	-7.1178176C80 C0	2.5208647880 04
3.59922D C3	2.9886D 05	-1.2641D 00	5.9200D 01	-2.5852D 01	-2.2369D-04	-7.1178118910 C0	2.5208659580 04
3.64897D 03	2.9895D 05	-9.8218D-01	5.8762D 01	-2.8337D 01	-3.8740D-04	-7.1178165920 C0	2.5208649100 04
3.69872D 03	2.9903D 05	-7.3358D-01	5.8276D 01	-3.0753D 01	-5.0076D-04	-7.1178221510 C0	2.5208638830 04

Figure 47. (Continued)

```
3.74846D 03  2.9910D 05    -5.1890D-01   5.7745D 01  -3.3096D 01  -5.2316D-04 -7.1178212980 C0  2.5208640160 04
3.79821D 03  2.9916D 05    -3.4003D-01   5.7170D 01  -3.5365D 01  -4.1319D-04 -7.1178181080 C0  2.5208646620 04
3.84796D 03  2.9921D 05    -1.9792D-01   5.6555D 01  -3.7560D 01  -3.0321D-04 -7.1178150140 00  2.5208653080 04
3.89770D 03  2.9924D 05    -9.3722D-02   5.5900D 01  -3.9678D 01  -3.0350D-04 -7.1178150860 C0  2.5208652930 04
3.94745D 03  2.9926D 05    -2.8574D-02   5.5209D 01  -4.1722D 01  -5.2954D-04 -7.1178218860 C0  2.5208630710 04
3.96696D 03  2.9930D 05    -7.2676D-03   5.4926D 01  -4.2497D 01  -5.9273D-01 -7.2196890900 C0  2.4980002580 04
3.98646D 03  2.9932D 05     5.4711D-03   5.4632D 01  -4.3250D 01  -7.8471D-01 -7.3161751360 00  2.4759313300 04
4.00597D 03  2.9932D 05     9.5096D-03   5.4327D 01  -4.3979D 01  -6.5541D-01 -7.4074352130 C0  2.4546417600 04
4.02548D 03  2.9931D 05     4.7155D-03   5.4011D 01  -4.4686D 01  -3.4531D-01 -7.4935577200 00  2.4341295390 04
4.04498D 03  2.9929D 05    -9.0439D-03   5.3686D 01  -4.5369D 01  -4.2061D-02 -7.5746370760 C0  2.4143915270 04
4.06449D 03  2.9926D 05    -3.1901D-02   5.3350D 01  -4.6030D 01  -3.3072D-04 -7.6507662980 00  2.3954255730 04
4.10872D 03  2.9925D 05    -9.4142D-02   5.2570D 01  -4.7474D 01  -4.1249D-04 -7.6507677580 C0  2.3954252120 04
4.15295D 03  2.9922D 05    -1.8628D-01   5.1772D 01  -4.8864D 01  -3.0056D-04 -7.6507658940 00  2.3954257020 04
4.19718D 03  2.9919D 05    -3.0607D-01   5.0957D 01  -5.0203D 01  -3.1516D-04 -7.6507664660 C0  2.3954255600 04
4.24141D 03  2.9915D 05    -4.5019D-01   5.0125D 01  -5.1492D 01  -2.9480D-04 -7.6507669760 C0  2.3954254840 04
4.28563D 03  2.9910D 05    -6.1549D-01   4.9278D 01  -5.2734D 01  -2.8023D-04 -7.6507654460 C0  2.3954257750 04
4.32986D 03  2.9905D 05    -7.9976D-01   4.8416D 01  -5.3931D 01  -1.5488D-04 -7.6507600800 C0  2.3954269060 04

HBAR      -0.20194348D 00  VCHAR      0.490265120 04
```

Figure 47. (Continued)

```
SUBROUTINE QASLIN ENTERED...BEGIN NEW ITERATION SET


BEGIN A NEW QUASILINEARIZATION ITERATION


STORAGE INTERVAL =          44.22881          1          7
STOP POINT NO.     1  TIME =    0.26537284D 03  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    0.100D 01
      K            DELTA TIME          KDOT
 -0.157979D 00  -0.398863D 02    0.396074D-02


STORAGE INTERVAL =          18.16505          7         18
STOP POINT NO.     2  TIME =    0.46518835D 03  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    0.0
      K            DELTA TIME          KDOT
 -0.187122D 00  -0.424927D 02    0.440363D-02


STORAGE INTERVAL =          49.74661         18         88
STOP POINT NO.     3  TIME =    0.394745110 04  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    0.100D 01
      K            DELTA TIME          KDOT
 -0.959389D 02  -0.732582D 05    0.130960D-02
```

Figure 48.

```
STORAGE INTERVAL =          19.50650          88          94
STOP POINT NO.     4  TIME =    0.406449010 04  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    0.0
      K            DELTA TIME          KOOT
 -0.102299D 03  -0.7269420 05   0.140725D-02


STORAGE INTERVAL =          44.22881          94          100
STOP POINT NO.     5  TIME =    0.432986290 04  NULL BURN CONTROL =          0.0


     NUMBER OF ORTHOGONALIZATIONS =      0

 COMBINATION COEFF

 4.66D-01 2.27D 03-1.46D 07 7.90D 03 4.53D 06-7.12D 03

 CONVERGENCE PARAMETERS----RHO

 4.44D 00 4.01D-04 1.81D-02 2.16D-06 5.64D 00 3.27D 00 2.59D 00 2.27D 03 1.97D 07 2.27D-04 2.18D-06 1.06D 04
 1.83D 07 7.12D 03

           10.817 SEC
```

Figure 48. (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

OUTPUT IIT = 1

TIME	PRIMER MAG	SWITCH	STEER-PSI	STEER-NU	HAMILTONIAN	TOT. ENERGY	ANGULAR MOM.
0.0	2.9585D 05	-1.8222D 00	-2.5186D C0	9.0758D 01	-4.0556D 00	-9.18C912C56D C0	2.1867155850 04
4.42288D 01	2.9604D C5	-1.3431D 0C	-1.3704D 0C	9.0154D 01	-4.0556D 00	-9.18C9120960 C0	2.1867155850 04
8.84576C 01	2.9621D 05	-9.3737D-01	-2.2401D-01	8.9554D J1	-4.0556D 00	-9.18C9120130 C0	2.1867155950 04
1.32686C 02	2.9634D 05	-6.0133D-01	9.2062D-01	8.8957D 01	-4.0556D 00	-9.18C9117440 00	2.1867156450 04
1.76915C 02	2.9645D 05	-3.3152D-01	2.0635D 00	8.8363D 01	-4.0556D 00	-9.1809114750 C0	2.1867156910 04
2.21144D 02	2.9654D C5	-1.2479D-01	3.2047D 0C	8.7772D 01	-4.0556D 00	-9.1809112180 C0	2.1867157360 04
2.65373D 02	2.966CD 05	2.1727D-02	4.3443D 00	8.7181D 01	-4.0373D 00	-9.1332566660 C0	2.1949457020 04
2.83538D C2	2.9662D 05	6.5940D-02	4.8113D 00	8.6933D 01	-4.036CD 00	-8.9718940020 C0	2.2225949060 04
3.01703C 02	2.9663D 05	1.0123D-01	5.2773D 00	8.6677D 01	-4.0346D 00	-8.8056435340 C0	2.2507672980 04
3.19868C C2	2.9664D 05	1.2736D-01	5.7424D 0C	8.6413D 01	-4.0329D 00	-8.6342537C90 C0	2.2794800710 04
3.38033D 02	2.9664D 05	1.4392D-01	6.2064D 00	8.6141D 01	-4.0314D 00	-8.4576196550 C0	2.3087515560 C4
3.56198D 02	2.9665D 05	1.5061D-01	6.6694D 00	8.5861D 01	-4.0301D 00	-8.2753786050 C0	2.3386014960 04
3.74363C C2	2.9664D C5	1.4719D-01	7.1314D 00	8.5573D 01	-4.0291D 00	-8.0873190400 C0	2.3690503070 04
3.92528C 02	2.9664D C5	1.3350D-01	7.5922D 00	8.5276D 01	-4.0284D 00	-7.8931671830 C0	2.4001202900 04
4.10693D 02	2.9663D 05	1.0939D-01	8.0518D 00	8.4970D 01	-4.0279D 00	-7.6926337850 C0	2.4318348260 04
4.28858D 02	2.9662D 05	7.4819D-02	8.5103D 00	8.4656D 01	-4.0274D 00	-7.4854044780 C0	2.4642196830 04
4.47023C 02	2.966CD 05	2.9775D-02	8.9675D 00	8.4332D 01	-4.0267D 00	-7.2711443130 C0	2.4973021350 04
4.65198D 02	2.9659D 05	-2.5665D-02	9.4235D 0C	8.3999D 01	-4.0350D 00	-7.1212111360 C0	2.5202113350 04
5.14935D C2	2.9652D 05	-2.2513D-01	1.0670D 01	8.3081D 01	-4.0352D 00	-7.1212121610 C0	2.5202111760 04
5.64682D 02	2.9643D 05	-4.9023D-01	1.1913D 01	8.2160D 01	-4.0353D 00	-7.1212119440 C0	2.5202111950 04
6.14428C C2	2.9633D 05	-8.1430D-01	1.3154D 01	8.1236D 01	-4.0355D 00	-7.1212161100 C0	2.5202105670 04
6.641750 02	2.9620D 05	-1.1910D 00	1.4392D 01	8.0308D 01	-4.0356D 00	-7.1212176960 C0	2.5202103140 04
7.13921D C2	2.96060 05	-1.6140D 00	1.5628D 01	7.9375D 01	-4.0357D 00	-7.1212166560 C0	2.5202104740 04
7.63668D C2	2.9591D 05	-2.0765D 0C	1.6861D 01	7.8437D 01	-4.0359D 00	-7.1212186930 00	2.5202101460 04
8.13415C C2	2.9575D 05	-2.5723D 00	1.8091D 01	7.7492D 01	-4.0360D 00	-7.1212157730 C0	2.5202099050 04
8.63161C C2	2.9558D C5	-3.0948D 00	1.9318D 01	7.6540D 01	-4.0361D 00	-7.1212208440 C0	2.5202098150 04
9.12908D 02	2.9540D 05	-3.6379D 0C	2.0542D 01	7.5580D 01	-4.0362D 00	-7.1212209690 C0	2.5202097870 04
9.626540 C2	2.9521D 05	-4.1953D 00	2.1763D 01	7.4610D 01	-4.0364D 00	-7.1212228630 C0	2.5202094660 04
1.01240C C3	2.9503D C5	-4.7612D 00	2.2981D 01	7.3631D 01	-4.0364D 00	-7.1212205980 C0	2.5202097800 04
1.06215C 03	2.9484D 05	-5.3298D 00	2.4194D 01	7.2640D 01	-4.0366D 00	-7.1212258840 CC	2.5202089750 04
1.11199D 03	2.9466D 05	-5.8957D CC	2.5404D 01	7.1637D 01	-4.0366D 00	-7.1212232820 C0	2.5202094020 04
1.16164D 03	2.9447D 05	-6.4538D 00	2.6609D 01	7.0621D 01	-4.0367D 00	-7.1212262280 C0	2.5202089240 04
1.21139C 03	2.94290 05	-6.9991D 00	2.7810D 01	6.9590D 01	-4.0369D 00	-7.1212292C70 C0	2.5202083920 04
1.26113C 03	2.9412D 05	-7.5273D 00	2.9005D C1	6.8544D 01	-4.0369D 00	-7.1212263930 CC	2.5202088660 04
1.31088D 03	2.9395D 05	-8.034CD CC	3.0194D 01	6.7480D 01	-4.037CD 00	-7.1212305540 C0	2.5202081070 04
1.36063D 03	2.9380D 05	-8.5155D CC	3.1377D 01	6.6398D 01	-4.0369D 00	-7.1212320360 C0	2.5202080950 C4
1.41037C 03	2.9365D 05	-8.9682D 00	3.2552D 01	6.5296D 01	-4.0369D 00	-7.1212315260 C0	2.5202081000 04
1.46012C C3	2.9351D 05	-9.3892D 00	3.3721D 01	6.4172D 01	-4.0370D 00	-7.1212325490 C0	2.5202079020 04
1.509870 03	2.9338D 05	-9.7757D CC	3.4881D 01	6.3026D 01	-4.037CD 00	-7.1212317330 CC	2.5202080040 04
1.559610 03	2.9327D 05	-1.0125D 01	3.6032D 01	6.1854D 01	-4.037CD 00	-7.1212320620 00	2.5202079860 C4

Figure 49.

1.609360	03	2.93160	05	-1.04360	01	3.71730	01	6.06560	01	-4.03700	00	-7.12123101400	00	2.52020803E0	04
1.659110	03	2.93080	05	-1.07070	01	3.83030	01	5.94300	01	-4.03490	00	-7.12122722700	CO	2.52020877C0	C4
1.708850	C3	2.93000	05	-1.09360	C1	3.94220	01	5.81730	01	-4.03680	00	-7.12122616700	CO	2.52020894	04
1.758600	03	2.92940	05	-1.11230	01	4.05280	01	5.68830	01	-4.03700	00	-7.12123556900	00	2.52020722	04
1.808350	C3	2.92890	05	-1.12670	01	4.16210	01	5.55590	01	-4.03680	00	-7.12123066500	CC	2.52020814	04
1.858090	C3	2.92860	05	-1.13680	01	4.27000	01	5.41980	01	-4.03670	00	-7.12122678100	CO	2.52020850	04
1.907840	03	2.92840	05	-1.14260	C1	4.37630	01	5.27970	01	-4.03670	00	-7.12122894800	CO	2.52020842	04
1.957590	C3	2.92830	05	-1.14420	01	4.48090	01	5.13550	01	-4.03680	00	-7.12123304100	CC	2.52020770	04
2.007330	C3	2.92840	05	-1.14170	01	4.58370	01	4.98690	01	-4.03660	00	-7.12122552300	CC	2.52020833	04
2.057080	C3	2.92870	05	-1.13510	C1	4.68450	01	4.83350	01	-4.03670	00	-7.12123429800	CO	2.52020747	C4
2.106830	03	2.92900	05	-1.12450	C1	4.78330	01	4.67520	01	-4.03660	00	-7.12123063500	CC	2.52020813	04
2.156570	C3	2.92950	05	-1.11020	01	4.87990	01	4.51170	01	-4.03640	00	-7.12122665C0	CC	2.52020880	04
2.206320	C3	2.93010	05	-1.09240	01	4.97400	01	4.34280	01	-4.03660	00	-7.12123303800	CC	2.52020765	04
2.256070	C3	2.93080	05	-1.07100	01	5.06570	01	4.16810	01	-4.03640	00	-7.12122957800	CO	2.52020836	04
2.305810	C3	2.93160	05	-1.04650	C1	5.15460	01	3.98740	01	-4.03630	00	-7.12122650900	CO	2.52020852	04
2.355560	C3	2.93250	05	-1.01910	01	5.24070	01	3.80040	01	-4.03620	00	-7.12122667800	CO	2.52020883	04
2.405310	C3	2.93350	05	-9.88850	00	5.32360	01	3.60710	01	-4.03620	00	-7.12122737100	CC	2.5202C874	04
2.455050	C3	2.93450	05	-9.56130	00	5.40330	01	3.40710	01	-4.03640	00	-7.12123381500	CC	2.52020753	04
2.504800	03	2.93570	05	-9.21170	CC	5.47950	01	3.20030	01	-4.03610	00	-7.12122659C0	CO	2.52020890	04
2.554550	C3	2.93690	05	-8.84230	00	5.55210	01	2.98660	01	-4.03620	00	-7.12122929400	CO	2.52020838	04
2.604290	C3	2.93820	05	-8.45550	CC	5.62070	01	2.76590	01	-4.03600	00	-7.12122466500	CC	2.52020922	04
2.654040	C3	2.93950	05	-8.05430	00	5.68520	01	2.53820	01	-4.03610	00	-7.12122814600	CO	2.52020855	C4
2.703790	03	2.94090	05	-7.64110	CC	5.74540	01	2.30360	01	-4.03590	00	-7.12122200C0	CO	2.5202C974	04
2.753530	C3	2.94220	05	-7.21890	CC	5.80110	01	2.06220	01	-4.03620	00	-7.12123142900	CC	2.52020792	04
2.803280	C3	2.94370	05	-6.79000	00	5.85190	01	1.81430	01	-4.03600	00	-7.12122374600	CC	2.5202C942	04
2.853030	03	2.94510	05	-6.35720	CO	5.89780	01	1.56030	01	-4.03580	00	-7.12121766500	CO	2.52021059	04
2.902770	03	2.94650	05	-5.92300	00	5.93860	01	1.30050	01	-4.03570	00	-7.12121475400	CO	2.52021121	04
2.952520	C3	2.94790	05	-5.49000	00	5.97400	01	1.03550	01	-4.03580	00	-7.12121772600	CO	2.52021060	04
3.002270	03	2.94940	05	-5.06030	00	6.00380	01	7.66040	00	-4.03600	00	-7.12122C68CC	CC	2.52021001	04
3.052010	03	2.95070	05	-4.63670	CO	6.02810	01	4.92750	00	-4.03550	00	-7.12121001900	CO	2.52021212	04
3.101760	03	2.95210	05	-4.22120	00	6.04650	01	2.16510	00	-4.03590	00	-7.12121941200	CO	2.52021029	04
3.151510	C3	2.95340	05	-3.81590	CC	6.05910	01	-6.17690-01		-4.03600	00	-7.12122166500	CO	2.52020990	04
3.201250	C3	2.95470	05	-3.42300	CO	6.06580	01	-3.41130	00	-4.03570	00	-7.12121353700	CO	2.52021149	04
3.251000	03	2.95600	05	-3.04430	00	6.06660	01	-6.20590	00	-4.03580	00	-7.12121366100	CO	2.52021140	C4
3.300750	03	2.95720	05	-2.68150	CC	6.06150	01	-8.99130	00	-4.03570	00	-7.12121C59800	CO	2.52021199	C4
3.350490	C3	2.95830	05	-2.33660	00	6.05050	01	-1.17580	01	-4.03600	00	-7.12121425800	CO	2.52021131	C0
3.400240	C3	2.95940	05	-2.01130	00	6.03360	01	-1.44960	01	-4.03580	00	-7.12120597460	CO	2.52021226	10
3.449980	03	2.96040	05	-1.70600	00	6.01120	01	-1.71970	01	-4.03600	00	-7.12121326400	CC	2.52021157	C0
3.499730	03	2.96130	05	-1.42310	CC	5.98310	01	-1.98530	01	-4.03600	00	-7.12121254900	CO	2.52021171	20
3.549480	C3	2.96220	05	-1.16300	00	5.94970	01	-2.24560	01	-4.03610	00	-7.12121272100	CO	2.52021170	10
3.599220	C3	2.96300	05	-9.26890-01		5.91100	01	-2.50000	01	-4.03620	00	-7.12121655100	CC	2.52021080	50
3.648970	03	2.96370	05	-7.15700-01		5.86730	01	-2.74810	01	-4.036 0	00	-7.12121559600	CO	2.5202102940	C4
3.698720	03	2.96430	05	-5.30210-01		5.81890	01	-2.98940	01	-4.036 0	00	-7.12123827800	CO	2.5202085440	C4

Figure 49. (Continued)

```
3.74846D 03  2.9648D 05    -3.7155D-01   5.7657D 01  -3.2236D 01  -4.0360D 00 -7.1212056300 C0  2.5202132080 04
3.79821D 03  2.9652D 05    -2.3887D-01   5.7082D 01  -3.4504D 01  -4.0362D 00 -7.1212120120 C0  2.5202119240 04
3.84796D 03  2.9656D 05    -1.3372D-01   5.6466D 01  -3.6698D 01  -4.0366D 00 -7.1212216710 C0  2.5202098600 04
3.89770D 03  2.9658D 05    -5.5719D-02   5.5809D 01  -3.8818D 01  -4.0366D 00 -7.1212236450 C0  2.5202C95150 04
3.94745D 03  2.9660D 05    -5.2474D-03   5.5115D 01  -4.0863D 01  -4.0448D 00 -7.1025233430 C0  2.5243934420 04
3.96696D 03  2.9660D 05     7.1522D-03   5.4832D 01  -4.1641D 01  -4.0176D 00 -7.1901550860 C0  2.5046711120 04
3.98646D 03  2.9661D 05     1.5470D-02   5.4537D 01  -4.2398D 01  -4.0297D 00 -7.2783688150 C0  2.4845037910 04
4.00597D 03  2.9661D 05     1.9860D-02   5.4232D 01  -4.3132D 01  -4.0381D 00 -7.3668126840 C0  2.4639659400 04
4.02548D 03  2.9661D 05     1.9265D-02   5.3917D 01  -4.3844D 01  -4.0724D 00 -7.4555582C70 00  2.4429777170 04
4.04498D 03  2.9661D 05     1.4577D-02   5.3591D 01  -4.4532D 01  -4.0728D 00 -7.5438358210 00  2.4217238660 04
4.06449D 03  2.9661D 05     5.4436D-03   5.3255D 01  -4.5197D 01  -4.0072D 00 -7.6296624010 C0  2.3957001070 04
4.10872D 03  2.9660D 05    -2.9702D-02   5.2469D 01  -4.6644D 01  -3.7934D 00 -7.6480855900 00  2.3960584710 04
4.15295D 03  2.9658D 05    -8.3284D-02   5.1665D 01  -4.8038D 01  -3.9532D 00 -7.6505547210 C0  2.3954798000 04
4.19718D 03  2.9657D 05    -1.5123D-01   5.C842D 01  -4.9380D 01  -3.9772D 00 -7.6488655350 C0  2.3958538670 04
4.24141D 03  2.9654D 05    -2.3093D-01   5.C004D 01  -5.0672D 01  -4.0105D 00 -7.6492258700 C0  2.3958091460 04
4.28563D 03  2.9650D 05    -3.1815D-01   4.9148D 01  -5.1915D 01  -3.7582D 00 -7.6470038620 00  2.3962755860 04
4.32986D 03  2.9649D 05    -4.0929D-01   4.8279D 01  -5.3117D 01  -3.9603D 00 -7.6452685580 C0  2.3957947200 04

H3AR      -0.40294995D 01   VCHAR      C.49041524D 04

DOUBLE PRECISION INITIAL CONDITIONS FOR OPTIMUM TRAJECTORY

   6.0681747591E781D 03     1.074572563171139D 00     4.152178425013220-01     5.950911175155180-04
   9.999999995999990 02     2.7944873046E750D 02    -1.822230591887790 00    -3.910992769761330 03
   1.788C6597434C09D 09     7.678512433020C2D-02     2.464208297799390-05    -9.799692776918710 05
  -7.88895400580C960 07     6.965877133290000 C4     1.058802620047550 03

END OF ITERATION NUMBER         1
```

Figure 49. (Continued)

```
ENTER KHIST --- IGO =        2

STOPPING CONDITIONS
            ON                    OFF                    ON                    OFF                   END                   HBAR                  VCHAR
   0  2.653728380 02  4.651883500 02  3.947451060 03  4.064490070 03  4.329862910 03 -2.019434760-01  4.902651220 03
   1  2.613765280 02  4.611920400 02  3.949791840 03  4.066830850 03  4.329862910 03 -4.029499530 00  4.904152400 03
STOP POINTS FOUND FOR LAST ITERATION
      2.588142060 02  4.567790760 02  3.955705990 03  4.071340510 03
```

Switch times to be used during the computation of the next iteration

Switch times found from last iteration

Figure 50.

```
BEGIN A NEW QUASILINEARIZATION ITERATION

STORAGE INTERVAL =          43.59032                1          7
STOP POINT NO.    1  TIME =    0.2615419CD 03  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    C.100D 01
      K              DELTA TIME            KDOT
 -0.218805D-08  -0.6702190-06    0.3264680-02

STORAGE INTERVAL =          18.16065                7         18
STOP POINT NO.    2  TIME =    0.4613090D 03  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    0.0
      K              DELTA TIME            KDOT
  0.243834D-08   0.6908870-06    0.3529290-02

STORAGE INTERVAL =          49.76129               18         88
STOP POINT NO.    3  TIME =    0.3944595D 04  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    C.100D 01
      K              DELTA TIME            KDOT
 -0.141411D-06  -0.1506440-03    0.9387140-03

STORAGE INTERVAL =          19.47019               88         94
STOP POINT NO.    4  TIME =    0.4061420SD C4  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    C.0
      K              DELTA TIME            KDOT
 -0.142591D-06  -0.1407270-03    0.1013250-02

STORAGE INTERVAL =          44.17891               94        100
STOP POINT NO.    5  TIME =    0.4326494D3 C4  NULL BURN CONTROL =          0.0
```

Figure 51.

```
     NUMBER OF ORTHOGONALIZATIONS =      0

 COMBINATION COEFF

 2.36D-08 2.20D-05-5.40D-01 1.79D-04-7.75D-01-3.20D-04

 CONVERGENCE PARAMETERS-----RHO

 5.19D-05 7.84D-08 1.51D-07 2.98D-11 1.27D-04 1.32D-05 8.14D-07 4.45D-03 1.72D 02 4.44D-09 4.64D-11 6.14D-02
 5.93D 01 3.20D-04

           139.83 SEC

         PROCESS CONVERGED, ITER.NO.=     4

********** LAST ITERATION SET COMPLETE **********
```

Figure 51. (Continued)

OUTPUT IIT = 2

TIME	PRIMER MAG	SWITCH	STEER-PSI	STEER-NU	HAMILTONIAN	TOT. ENERGY	ANGULAR MOM.
0.0	2.9831D 05	-2.1041D 00	-2.6461D 00	9.1186D 01	-9.0162D-03	-9.18C912C96D C0	2.186715587D 04
4.35903D 01	2.9853D 05	-1.5666D 00	-1.5144D 00	9.0577D 01	-9.0162D-03	-9.18C912C96D C0	2.186715585D 04
8.71806D C1	2.9872D 05	-1.1084D 00	-3.8475D-C1	8.9961D 01	-9.0194D-03	-9.18C912C13D C0	2.186715599D 04
1.30771D C2	2.9887D 05	-7.2582D-01	7.4304D-01	8.9383D 01	-9.0223D-03	-9.1809117598 C0	2.186715643D 04
1.74361D 02	2.9900D 05	-4.1560D-01	1.8690D 00	8.8753D 01	-9.0245D-03	-9.18C9115C9D C0	2.186715680D 04
2.17952D C2	2.9910D 05	-1.7460D-C1	2.9932D 00	8.8154D 01	-9.0262D-03	-9.18C9112630 00	2.186715720D 04
2.61542D C2	2.9917D 05	1.1532D-09	4.1156D C0	8.7556D 01	-9.0272D-03	-9.18C911C19C C0	2.186715770D 04
2.79703D C2	2.9919D 05	5.4261D-02	4.5825D 00	8.7303D 01	-9.0272D-03	-9.0210533050 C0	2.214214069D 04
2.97863D C2	2.9921D 05	9.8339D-02	5.0483D CC	8.7043D 01	-9.C224D-03	-8.8563618560 C0	2.242231592D 04
3.16024D C2	2.9922D 05	1.3203D-01	5.5132D 0G	8.6778D 01	-9.0339D-03	-8.6866290020 C0	2.270785149D C4
3.34185D C2	2.9923D 05	1.5511D-01	5.9771D 0C	8.6500D 01	-9.0459D-03	-8.5116346C5D C0	2.299892540D 04
3.52345D C2	2.9924D 05	1.6738D-01	6.4400D 00	8.6217D 01	-9.0587D-03	-8.331144C22C C0	2.329572820D 04
3.70506D C2	2.9924D 05	1.6858D-C1	6.9018D 00	8.5925D 01	-9.0722D-03	-8.1449071170 C0	2.359846312D 04
3.88666D C2	2.9923D 05	1.5848D-01	7.3625D 0C	8.5625D 01	-9.0866D-03	-7.9526566750 0C	2.390734713D 04
4.06827D C2	2.9923D 05	1.3682D-01	7.8221D 0C	8.5316D 01	-9.102CD-03	-7.7541078670 C0	2.422261251D 04
4.24988D C2	2.9921D 05	1.0337D-01	8.2805D 00	8.4999D 01	-9.1183D-03	-7.5489545380 C0	2.454465078D 04
4.43148D C2	2.9920D 05	5.7845D-02	8.7378D 00	8.4672D 01	-9.1357D-03	-7.3368692840 00	2.487329964D 04
4.61309D C2	2.9918D 05	-1.1065D-09	9.1938D 00	8.4336D 01	-9.1544D-03	-7.1175CC32CC 00	2.520927363D 04
5.11070D C2	2.9911D 05	-2.1629D-C1	1.0440D 01	8.3402D 01	-9.1544D-03	-7.1175CC320D C0	2.520927363D 04
5.60832D 02	2.9901D 05	-5.0948D-01	1.1684D 01	8.2466D 01	-9.1540D-03	-7.1175CC350C C0	2.520927359D C4
6.10593D C2	2.9889D 05	-8.7273D-C1	1.2925D 01	8.1527D 01	-9.1508D-03	-7.1175OC384D C0	2.520927351D 04
6.60354D C2	2.9875D 05	-1.2990D 00	1.4164D 01	8.0585D 01	-9.1480D-03	-7.1175CC426C C0	2.520927343D 04
7.10116D C2	2.9860D 05	-1.7813D 0C	1.5401D 01	7.9638D 01	-9.1457D-03	-7.1175CC477C C0	2.520927333D 04
7.59877D 02	2.9842D 05	-2.3123D CC	1.6635D 01	7.8685D 01	-9.1439D-03	-7.1175OC9360 C0	2.520927322D 04
8.09638D C2	2.9823D 05	-2.8848D CC	1.7867D 01	7.7727D 01	-9.1428D-03	-7.1175CC603C 00	2.520927309D 04
8.59399D 02	2.9803D 05	-3.4917D 0C	1.9096D 01	7.6761D 01	-9.1423D-03	-7.1175CCE77C C0	2.520927296D 04
9.09161D C2	2.9783D 05	-4.1259D C0	2.0323D 01	7.5785D 01	-9.1425D-03	-7.1175OC7570 C0	2.520927281D 04
9.58922D C2	2.9761D 05	-4.7803D 0C	2.1548D 01	7.4805D 01	-9.1434D-03	-7.1175CC8420 C0	2.520927266D 04
1.00868D 03	2.9739D 05	-5.4482D GC	2.2769D 01	7.3813D 01	-9.145CD-03	-7.1175CC9330 C0	2.520927245D 04
1.05844D C3	2.9717D 05	-6.1231D 00	2.3987D C1	7.2809D 01	-9.1474D-03	-7.1175C1C27C C0	2.520927232D 04
1.10821D C3	2.9695D 05	-6.7985D 00	2.5201D C1	7.1793D 01	-9.1504D-03	-7.1175OC126C C0	2.520927215D 04
1.15797D 03	2.9673D 05	-7.4683D C0	2.6412D 01	7.0764D 01	-9.1542D-03	-7.1175012270 C0	2.520927190D 04
1.20773D G3	2.9651D 05	-8.1269D 0C	2.7618D 01	6.9721D 01	-9.1586D-03	-7.1175C1331C C0	2.520927178D 04
1.25749D C3	2.9630D 05	-8.7688D 00	2.8820D 01	6.8662D 01	-9.1637D-03	-7.1175C1436C C0	2.520927159D 04
1.30725D C3	2.9610D 05	-9.3889D 00	3.0016D C1	6.7585D 01	-9.1694D-03	-7.1175C15420 C1	2.520927139D 04
1.357010 G3	2.9591D 05	-9.9825D 0C	3.1206D 01	6.6490D 01	-9.1756D-03	-7.1175016490 C0	2.520927120D 04
1.40677D 03	2.9572D 05	-1.0545D G1	3.2390D C1	6.5375D 01	-9.1824D-03	-7.1175017560 C0	2.520927101D 04
1.45653D C3	2.9555D 05	-1.1074D C1	3.3567D 01	6.4239D 01	-9.1896D-03	-7.1175C1E63C C0	2.520927081D 04
1.50630D C3	2.9539D 05	-1.1564D 01	3.4736D 01	6.3079D 01	-9.1972D-03	-7.1175C1F68D C0	2.520927062D 04
1.55606D G3	2.9524D 05	-1.2012D C1	3.5897D 01	6.1894D 01	-9.2052D-03	-7.1175C2C79D C0	2.520927043D 04

Figure 52.

1.60582D C3	2.95110 05	-1.24180 C1	3.70480 01	6.06820 01	-9.21350-03	-7.1175021750 C0	2.5209270240 04
1.65558D C3	2.94990 05	-1.27770 01	3.81880 C1	5.94410 01	-9.22210-03	-7.1175022150 C0	2.5209270090 04
1.70534D 03	2.94890 05	-1.30880 C1	3.93180 01	5.81700 01	-9.23090-03	-7.1175023730 C0	2.5209269830 04
1.75510D C3	2.94800 05	-1.33500 C1	4.04350 01	5.68660 01	-9.23980-03	-7.1175024680 00	2.5209269650 04
1.80486C C3	2.94730 05	-1.35620 01	4.15400 01	5.55260 01	-9.24880-03	-7.1175025600 C0	2.5209269520 C4
1.85463D C3	2.94680 05	-1.37230 01	4.26290 01	5.41500 01	-9.25790-03	-7.1175026490 C0	2.5209269360 04
1.90439C C3	2.94640 05	-1.38340 01	4.37040 01	5.27330 01	-9.26700-03	-7.1175027340 C0	2.5209269190 04
1.95415D 03	2.94620 05	-1.38930 C1	4.47610 01	5.12740 01	-9.27610-03	-7.1175028160 C0	2.5209269040 04
2.00391C 03	2.94620 05	-1.39030 01	4.58010 01	4.97700 01	-9.28500-03	-7.1175028940 C0	2.5209268890 04
2.05367D C3	2.94630 05	-1.38630 01	4.68210 01	4.82190 01	-9.29390-03	-7.1175029680 C0	2.5209268750 04
2.10343D 03	2.94660 05	-1.37750 01	4.78200 01	4.66170 01	-9.30260-03	-7.1175030390 C0	2.5209268610 04
2.15319D 03	2.94710 05	-1.36400 C1	4.87970 01	4.49630 01	-9.31110-03	-7.1175031050 C0	2.5209268490 04
2.20295C C3	2.94760 05	-1.34600 01	4.97500 01	4.32530 01	-9.31940-03	-7.1175031680 C0	2.5209268360 04
2.25272D C3	2.94840 05	-1.32380 01	5.06770 01	4.14850 01	-9.32750-03	-7.1175032260 C0	2.5209268250 04
2.30248D C3	2.94920 05	-1.29740 01	5.15770 01	3.96560 01	-9.33530-03	-7.1175032810 C0	2.5209268140 04
2.35224D 03	2.95020 05	-1.26720 C1	5.24470 01	3.77640 01	-9.34280-03	-7.1175033320 C0	2.5209268040 04
2.40200D 03	2.95130 05	-1.23340 01	5.32870 01	3.58060 01	-9.35010-03	-7.1175033790 C0	2.5209267950 04
2.45176D C3	2.95260 05	-1.19630 01	5.40920 01	3.37820 01	-9.35700-03	-7.1175034220 C0	2.5209267860 04
2.50152D 03	2.95390 05	-1.15620 01	5.48630 01	3.16880 01	-9.36360-03	-7.1175034610 C0	2.5209267770 04
2.55128D 03	2.95530 05	-1.11330 01	5.55960 01	2.95240 01	-9.36990-03	-7.1175034970 C0	2.5209267710 04
2.60104C C3	2.95680 05	-1.06800 01	5.62900 01	2.72900 01	-9.37590-03	-7.1175035290 C0	2.5209267650 04
2.65081D 03	2.95830 05	-1.02060 01	5.69410 01	2.49850 01	-9.38150-03	-7.1175035580 C0	2.5209267590 04
2.70057D C3	2.95990 05	-9.7141D 00	5.75490 01	2.26100 01	-9.38680-03	-7.1175035840 C0	2.5209267540 04
2.75033D 03	2.96160 05	-9.2072D 00	5.81100 01	2.01670 01	-9.39180-03	-7.1175036060 C0	2.5209267490 04
2.80009C C3	2.96330 05	-8.6886D 00	5.86230 01	1.76580 01	-9.39640-03	-7.1175036250 C0	2.5209267450 04
2.84985D C3	2.96500 05	-8.1616D 00	5.90850 01	1.50870 01	-9.40060-03	-7.1175036410 C0	2.5209267420 04
2.89961D 03	2.96680 05	-7.6293D 00	5.94950 01	1.24590 01	-9.40460-03	-7.1175036550 C0	2.5209267390 04
2.94937D C3	2.96850 05	-7.095CD 00	5.98500 01	9.77990 00	-9.40820-03	-7.1175036650 C0	2.5209267370 04
2.99914C C3	2.97030 05	-6.5616D 00	6.01490 01	7.05550 00	-9.41150-03	-7.1175036730 C0	2.5209267350 04
3.04890D C3	2.97200 05	-6.0322D 00	6.03920 01	4.29420 00	-9.41450-03	-7.1175036780 C0	2.5209267340 04
3.09866D C3	2.97370 05	-5.5097D 00	6.05760 01	1.50440 00	-9.41720-03	-7.1175036810 C0	2.5209267330 04
3.14842D C3	2.97540 05	-4.9971D 00	6.07010 01	-1.30440 00	-9.41970-03	-7.1175036820 C0	2.5209267330 04
3.19818C 03	2.97700 05	-4.4969D 00	6.07660 01	-4.12250 00	-9.42180-03	-7.1175036800 C0	2.5209267330 04
3.24794C C3	2.97860 05	-4.0120D 00	6.07710 01	-6.93980 00	-9.42370-03	-7.1175036760 C0	2.5209267330 04
3.29770C C3	2.98020 05	-3.5448D 00	6.07180 01	-9.74610 00	-9.42530-03	-7.1175036710 C0	2.5209267340 04
3.34746D C3	2.98160 05	-3.0977D 00	6.06050 01	-1.25310 01	-9.42670-03	-7.1175036630 C0	2.5209267360 04
3.39723C 03	2.98300 05	-2.6730D 00	6.04340 01	-1.52860 01	-9.42780-03	-7.1175036540 C0	2.5209267380 04
3.44699D C3	2.98430 05	-2.2730D 00	6.02060 01	-1.80020 01	-9.42870-03	-7.1175036430 C0	2.5209267410 04
3.49675C C3	2.98560 05	-1.8995D 00	5.99230 01	-2.06700 01	-9.42940-03	-7.1175036300 C0	2.5209267420 04
3.54651D C3	2.98670 05	-1.5545D 00	5.95870 01	-2.32840 01	-9.43000-03	-7.1175036160 C0	2.5209267450 04
3.59627C C3	2.98770 05	-1.2398D 00	5.91980 01	-2.58370 01	-9.43030-03	-7.1175036010 00	2.5209267480 04
3.64603C C3	2.98870 05	-9.5692D-01	5.87600 01	-2.83250 01	-9.43040-03	-7.1175035840 C0	2.5209267510 04
3.69579C C3	2.98950 05	-7.0734D-01	5.82740 01	-3.07430 01	-9.43040-03	-7.1175035670 C0	2.5209267590 04

Figure 52. (Continued)

```
3.74555D C3   2.9902D 05    -4.9236D-01   5.7742D 01  -3.3089D 01  -9.4303D-03 -7.1175035480 C0   2.5209267590 04
3.79532D C3   2.9908D 05    -3.1318D-01   5.7167D 01  -3.5361D 01  -9.4300D-03 -7.1175C3520C C0   2.5209267620 04
3.84508D C3   2.9912D 05    -1.7082D-01   5.6551D 01  -3.7557D 01  -9.4295D-03 -7.1175035C7D C0   2.52C9267670 04
3.89484D C3   2.9916D 05    -6.6171D-02   5.5896D 01  -3.9678D 01  -9.4290D-03 -7.1175C3485D C0   2.5209267710 04
3.94460D C3   2.9918D 05     1.3000D-09   5.5204D 01  -4.1724D 01  -9.4283D-03 -7.1175C3469D C0   2.5209267750 04
3.96407D 03   2.9919D 05     1.5455D-02   5.4920D 01  -4.2499D 01  -9.4284D-03 -7.206G15C82D C0   2.5009602540 04
3.98354D C3   2.9919D 05     2.5073D-02   5.4627D 01  -4.3251D 01  -9.4248D-03 -7.294817238D C0   2.4806094650 04
4.00301D C3   2.9919D 05     2.8581D-02   5.4322D 01  -4.3981D 01  -9.4307D-03 -7.3838C4722D C0   2.4598759600 04
4.02248D C3   2.9919D 05     2.5721D-02   5.4008D 01  -4.4688D 01  -9.4373D-03 -7.4728743580 C0   2.4387634250 04
4.04195D C3   2.9919D 05     1.6262D-02   5.3683D 01  -4.5372D 01  -9.4450D-03 -7.5619C53880 C0   2.4172775800 04
4.06142D C3   2.9918D 05    -4.1485D-C9   5.3349D 01  -4.6033D 01  -9.4538D-03 -7.65C76C273D C0   2.3954268350 04
4.10560D 03   2.9916D 05    -6.0811D-02   5.2570D 01  -4.7476D 01  -9.4538D-03 -7.65C76C273D C0   2.3954268350 04
4.14978D 03   2.9914D 05    -1.5185D-01   5.1772D 01  -4.8866D 01  -9.4511D-03 -7.65C76C251D C0   2.3954268440 04
4.19396D C3   2.9911D 05    -2.7024D-C1   5.C957D 01  -5.0204D 01  -9.4481D-03 -7.65C76C282D C0   2.3954268600 04
4.23814D C3   2.9907D 05    -4.1313D-01   5.0126D 01  -5.1493D 01  -9.4451D-03 -7.65C7602140 C0   2.3954268750 04
4.28232D 03   2.9902D 05    -5.7761D-01   4.9279D 01  -5.2735D 01  -9.4423D-03 -7.65C76C147D C0   2.3954268900 04
4.326490 C3   2.9897D 05    -7.6079D-01   4.8417D 01  -5.3932D 01  -9.4395D-03 -7.65C76CC80D C0   2.3954269060 04

HBAR      -0.927390370-02  VCHAR      C.490286540 04
```

Figure 52. (Continued)

TIME HISTORIES OF ALL VARIABLES

TIME	R	PHI	RHO	V	MASS	LAM 1	K
0.0	6.068175D C3	1.07[illegible]73D C0	4.152178D-01	5.950911D-04	1.0000000D 03	2.7944870 02	-2.1040980 00
4.359030 01	6.085839D 03	1.100439D 00	3.952117D-01	5.917375D-04	1.0000000 03	2.7866440 02	-1.9666140 00
8.718060 01	6.102627D C3	1.126163D 00	3.750574D-01	5.885774D-04	1.0000000D 03	2.7776930 02	-1.1083810 00
1.307710 02	6.118535D 03	1.151754D 00	3.547657D-01	5.856077C-04	1.0000000D 03	2.7676720 02	-7.2582430-01
1.743610 02	6.133554D 03	1.177220D 00	3.343468D-01	5.828251C-04	1.0000000D 03	2.7566150 02	-4.1559720-01
2.179520 02	6.147681D 03	1.202568D 00	3.138106D-01	5.8022670-04	1.0000000D 03	2.7445530 02	-1.7460470-01
2.615420 02	6.160911D 03	1.227807D 00	2.931665D-01	5.778099D-04	1.0000000D 03	2.7315170 02	1.1531810-09
2.797030 02	6.166117D 03	1.238357D 00	2.8700170-01	5.840966C-04	9.8183930 02	2.7474200 02	5.4260680-02
2.978630 02	6.171342D 03	1.249023D 00	2.8200880-01	5.905160C-04	9.6367870 02	2.7634400 02	9.8338600-02
3.160240 02	6.176428D 03	1.259806D 00	2.782284D-01	5.970684D-04	9.4551800 02	2.7795670 02	1.3202690-01
3.341850 02	6.181456D 03	1.270710D 00	2.757032D-01	6.037540C-04	9.2735740 02	2.7957910 02	1.5511430-01
3.523450 02	6.186449D 03	1.281736D 00	2.744780D-01	6.105730C-04	9.0919670 02	2.8121010 02	1.6737600-01
3.705060 02	6.191433D 03	1.292888D 00	2.7460010-01	6.175253D-04	8.9103610 02	2.8284870 02	1.6857770-01
3.886660 02	6.196432D 03	1.304166D 00	2.761190C-01	6.246112C-04	8.7287540 02	2.8449370 02	1.5847710-01
4.068270 02	6.201471D 03	1.315575D 00	2.790867C-01	6.318304C-04	8.5471480 02	2.8614380 02	1.3682490-01
4.249880 02	6.206578D 03	1.327116D 00	2.835582C-01	6.391831C-04	8.3655410 02	2.8779790 02	1.0336660-01
4.431480 02	6.211780D 03	1.338792D 00	2.8959110-01	6.466689C-04	8.1839350 02	2.8945450 02	5.7844570-02
4.613090 02	6.217106D 03	1.350605D 00	2.972462C-01	6.542878C-04	8.0023280 02	2.9111230 02	-1.1065210-09
5.110700 02	6.232105D 03	1.380308D 00	3.0554570-01	6.512521C-04	8.0023280 02	2.8828380 02	-2.1628730-01
5.608320 02	6.247509D 03	1.415418D 00	3.1348700-01	6.481445D-04	8.0023280 02	2.8518090 02	-5.0548390-01
6.105930 02	6.263298D 03	1.447592D 00	3.2106580-01	6.449702C-04	8.0023280 02	2.8181460 02	-8.1273310-01
6.603540 02	6.279456D 03	1.479606D 00	3.2827860-01	6.417343C-04	8.0023280 02	2.7819630 02	-1.2990470 00
7.101160 02	6.295963D 03	1.511458D 00	3.3512270-01	6.384420C-04	8.0023280 02	2.7433730 02	-1.7813000 00
7.598770 02	6.312802D 03	1.543145D 00	3.4159590-01	6.350984C-04	8.0023280 02	2.7024930 02	-2.3122970 00
8.096380 02	6.329954D 03	1.574664D 00	3.476968C-01	6.317086C-04	8.0023280 02	2.6594380 02	-2.8848290 00
8.593990 02	6.347400D 03	1.606013D 00	3.5342460-01	6.282776C-04	8.0023280 02	2.6143240 02	-3.4917180 00
9.091610 02	6.365121D 03	1.637191D 00	3.5877920-01	6.248104C-04	8.0023280 02	2.5672680 02	-4.1258760 00
9.589220 02	6.383100D 03	1.668195D 00	3.637608C-01	6.213119D-04	8.0023280 02	2.5183E40 02	-4.7803140 00
1.008680 03	6.401317D 03	1.699025D 00	3.683705D-01	6.177868D-04	8.0023280 02	2.4677860 02	-5.4482490 00
1.058440 03	6.419755D 03	1.729679D 00	3.726096C-01	6.142398C-04	8.0023280 02	2.4155880 02	-6.1230820 00
1.108210 03	6.438394D 03	1.760156D 00	3.7648010-01	6.106756C-04	8.0023280 02	2.3618990 02	-6.7984640 00
1.157970 03	6.457217D 03	1.790455D 00	3.799845C-01	6.070985C-04	8.0023280 02	2.3068300 02	-7.4663240 00
1.207730 03	6.476205D 03	1.820575D 00	3.8312540-01	6.035130C-04	8.0023280 02	2.2504850 02	-8.1269060 00
1.257490 03	6.495341D 03	1.850518D 00	3.8590630-01	5.999231C-04	8.0023280 02	2.1929700 02	-8.7687660 00
1.307250 03	6.514606D 03	1.880281D 00	3.8833060-01	5.963331C-04	8.0023280 02	2.1343840 02	-9.3888520 00
1.357010 03	6.533983D 03	1.909866D 00	3.904023C-01	5.927467C-04	8.0023280 02	2.0748280 02	-9.9824670 00
1.406770 03	6.553454D 03	1.939273D 00	3.921257C-01	5.891678C-04	8.0023280 02	2.0143960 02	-1.0545320 01
1.456530 03	6.573002D 03	1.968502D 00	3.935053C-01	5.856000D-04	8.0023280 02	1.9531810 02	-1.1073510 01
1.506300 03	6.592611D 03	1.997554D 00	3.945460C-01	5.820468C-04	8.0023280 02	1.8912710 02	-1.1563570 01

Figure 53.

1.556060 03	6.612263D 03	2.026429D 00	3.952529C-01	5.785116C-04	8.0023280 02	1.6287520 02	-1.2012440 01
1.605820 03	6.6319420 03	2.0551290 00	3.956311C-01	5.749974D-04	8.0023280 02	1.7657080 02	-1.2417510 01
1.655580 03	6.6516320 03	2.0836550 00	3.956863C-01	5.715074C-04	8.0023280 02	1.7022170 02	-1.2776560 01
1.705340 03	6.6713170 03	2.1120080 00	3.9542400-01	5.680444D-04	8.0023280 02	1.6383540 02	-1.3087820 01
1.755100 03	6.6909800 03	2.1401890 00	3.9485C10-01	5.646111C-04	8.0023280 02	1.5741940 02	-1.3349920 01
1.804860 03	6.7106080 03	2.1682C00 00	3.939704C-01	5.612102C-04	8.0023280 02	1.5C98030 02	-1.3561880 01
1.854630 03	6.7301840 03	2.1960420 00	3.9279100-01	5.578441C-04	8.0023280 02	1.4452500 02	-1.3723150 01
1.904390 03	6.7496950 03	2.2237180 00	3.9131810-01	5.5451500-04	8.0023280 02	1.3805950 02	-1.3833530 01
1.954150 03	6.7691250 03	2.2512300 00	3.8955780-01	5.5122530-04	8.0023280 02	1.3159C00 02	-1.3893280 01
2.003910 03	6.7884600 03	2.2785780 00	3.8751650-01	5.479768C-04	8.0023280 02	1.2512190 02	-1.3902700 01
2.053670 03	6.8076870 03	2.3057670 00	3.8520C30-01	5.4477160-04	8.0023280 02	1.1866070 02	-1.3862860 01
2.103430 03	6.8267920 03	2.3327960 00	3.8261580-01	5.4161140-04	8.0023280 02	1.1221130 02	-1.3774860 01
2.153190 03	6.8457610 03	2.3596700 00	3.7976920-01	5.3849790-04	8.0023280 02	1.0577850 02	-1.3640160 01
2.202950 03	6.8645830 03	2.3863900 00	3.7666690-01	5.3543250-04	8.0023280 02	9.9366820 01	-1.3460460 01
2.252720 03	6.8832440 03	2.4129580 00	3.7331530-01	5.3241680-04	8.0023280 02	9.2980400 01	-1.3237750 01
2.302480 03	6.9017320 03	2.4393780 00	3.6972080-01	5.2945210-04	8.0023280 02	8.6623190 01	-1.2974210 01
2.352240 03	6.9200360 03	2.4656520 00	3.6588960-01	5.2653960-04	8.0023280 02	8.0298860 01	-1.2672230 01
2.402000 03	6.9381430 03	2.4917820 00	3.6182820-01	5.2368040-04	8.0023280 02	7.4010870 01	-1.2334370 01
2.451760 03	6.9560420 03	2.5177710 00	3.5754260-01	5.2087560-04	8.0023280 02	6.7762420 01	-1.1963350 01
2.501520 03	6.9737230 03	2.5436210 00	3.5303930-01	5.1812600-04	8.0023280 02	6.1556510 01	-1.1562040 01
2.551280 03	6.9911740 03	2.5693370 00	3.4832420-01	5.1543260-04	8.0023280 02	5.5395910 01	-1.1133370 01
2.601040 03	7.0086860 03	2.5949200 00	3.4340360-01	5.1279610-04	8.0023280 02	4.9283220 01	-1.0680420 01
2.650810 03	7.0253470 03	2.6203730 00	3.3828350-01	5.1021730-04	8.0023280 02	4.3220820 01	-1.0206290 01
2.700570 03	7.0420490 03	2.6456990 00	3.3296990-01	5.0769660-04	8.0023280 02	3.7210910 01	-9.7141420 00
2.750330 03	7.0584820 03	2.6709C10 00	3.2746860-01	5.0523480-04	8.0023280 02	3.1255540 01	-9.2071770 00
2.800090 03	7.0746370 03	2.6959820 00	3.2178560-01	5.0283230-04	8.0023280 02	2.5356580 01	-8.6885960 00
2.849850 03	7.0905040 03	2.7209450 00	3.1592650-01	5.0048950-04	8.0023280 02	1.9515750 01	-8.1615960 00
2.899610 03	7.1060760 03	2.7457930 00	3.0989710-01	4.9820690-04	8.0023280 02	1.3734630 01	-7.6293490 00
2.949370 03	7.1213430 03	2.7705290 00	3.0370300-01	4.9598470-04	8.0023280 02	8.0146680 00	-7.0949910 00
2.999140 03	7.1362990 03	2.7951560 00	2.9734960-01	4.9382320-04	8.0023280 02	2.3571730 00	-6.5616060 00
3.048900 03	7.1509340 03	2.8196760 00	2.9084260-01	4.9172260-04	8.0023280 02	-3.2366550 00	-6.0322110 00
3.098660 03	7.1652420 03	2.8440940 00	2.8418710-01	4.8968320-04	8.0023280 02	-8.7657290 00	-5.5097480 00
3.148420 03	7.1792140 03	2.8684120 00	2.7738850-01	4.8770510-04	8.0023280 02	-1.4229070 01	-4.9970650 00
3.198180 03	7.1928460 03	2.8926330 00	2.7045200-01	4.8578840-04	8.0023280 02	-1.9625780 01	-4.4969310 00
3.247940 03	7.2061280 03	2.9167600 00	2.6338270-01	4.8393320-04	8.0023280 02	-2.4955060 01	-4.0119830 00
3.297700 03	7.2190560 03	2.9407960 00	2.5618560-01	4.8213950-04	8.0023280 02	-3.0216200 01	-3.5447600 00
3.347460 03	7.2316230 03	2.9647450 00	2.4886570-01	4.8040750-04	8.0023280 02	-3.5408560 01	-3.0976780 00
3.397230 03	7.2438220 03	2.9886090 00	2.4142800-01	4.7873700-04	8.0023280 02	-4.0531550 01	-2.6730260 00
3.446990 03	7.2556480 03	3.0123910 00	2.3387700-01	4.7712800-04	8.0023280 02	-4.5584660 01	-2.2729590 00
3.496750 03	7.2670960 03	3.0360950 00	2.2621770-01	4.7558060-04	8.0023280 02	-5.0567450 01	-1.8995020 00
3.546510 03	7.2781600 03	3.0597230 00	2.1845460-01	4.7409460-04	8.0023280 02	-5.5479500 01	-1.5545380 00
3.596270 03	7.2888360 03	3.0832790 00	2.1059240-01	4.7267000-04	8.0023280 02	-6.0320460 01	-1.2398120 00
3.646030 03	7.2991180 03	3.1067650 00	2.0263550-01	4.7130670-04	8.0023280 02	-6.5090C00 01	-9.5692410-01

Figure 53. (Continued)

3.69579D C3	7.3090010 C3	3.130186D 00	1.945883C-01	4.7000460-04	8.0023280 02	-6.5787840 01	-7.6733520-01
3.74555D C3	7.3184820 C3	3.153542D 00	1.864553C-01	4.6876360-04	8.0023280 02	-7.4413740 01	-4.9236130-01
3.79532D C3	7.3275560 03	3.176839D 00	1.782407C-01	4.6758360-04	8.0023280 02	-7.8567450 01	-3.1317700-01
3.84508D C3	7.3362200 C3	3.200079D 00	1.699488C-01	4.6646430-04	8.0023280 02	-8.3448790 C1	-1.7081540-01
3.89484D 03	7.3444690 C3	3.223264D 00	1.6158380-01	4.6540580-04	8.0023280 02	-8.7857550 01	-6.6170780-02
3.94460D C3	7.3523000 C3	3.246398D 00	1.531497C-01	4.6440780-04	8.0023280 02	-9.2193500 01	1.2599550-09
3.96407D 03	7.3554960 C3	3.2554C6D CC	1.751121C-01	4.6086630-04	7.8076260 02	-9.1859340 01	1.5454650-02
3.98354D C3	7.3591190 C3	3.264343D 00	1.970834C-01	4.5712700-04	7.6129240 02	-9.1444830 C1	2.5073270-02
4.00301D 03	7.3631700 C3	3.273205D 00	2.1909210-01	4.5318400-04	7.4182220 02	-9.0548040 01	2.6580510-02
4.02248D C3	7.3676510 C3	3.2819850 00	2.4116970-01	4.4903080-04	7.2235210 02	-9.0366500 01	2.5720720-02
4.04195D 03	7.3725620 03	3.2906890 00	2.0335180-01	4.4466020-04	7.0288190 02	-8.9699120 01	1.6262130-02
4.06142D C3	7.3779070 C3	3.2993030 CO	2.8567820-01	4.4006470-04	6.8341170 02	-8.8942280 01	-4.1494840-09
4.10560D 03	7.3902070 C3	3.3187110 00	2.7115360-01	4.3860110-04	6.8341170 02	-9.2393650 01	-6.0810630-02
4.14978D 03	7.4018650 03	3.3380580 CC	2.5657440-01	4.3722060-04	6.8341170 02	-9.5796440 01	-1.5184600-01
4.19396D C3	7.4128770 C3	3.3573450 00	2.4194410-01	4.3592260-04	6.8341170 02	-9.9151780 01	-2.7024340-01
4.23814D C3	7.4232420 03	3.3765760 00	2.2726630-01	4.3470610-04	6.8341170 02	-1.0246080 02	-4.1313060-01
4.28232D 03	7.4329570 03	3.3957550 00	2.1254450-01	4.3357050-04	6.8341170 02	-1.0572440 02	-5.7761460-01
4.32649D 03	7.4420210 03	3.4148860 00	1.9778190-01	4.3251500-04	6.8341170 02	-1.0894380 02	-7.6079480-01

TIME	LAM 4	LAM 6	THETA	X	LAM 2	LAM 5	LAM 3
0.0	-6.1679490 03	1.8025400 09	7.6785120-02	2.4642080-05	-9.8752210 05	-8.3571870 07	7.6723770 04
4.35903D C1	-2.9754740 C3	1.8105720 09	7.7830440-02	2.3320700-05	-9.9084120 05	-4.8016120 07	7.6723770 04
8.71806D C1	2.0095340 C2	1.8172520 09	7.8818420-02	2.2011540-05	-9.9344210 05	-1.2241260 C7	7.6723770 04
1.30771D 02	3.3602820 C3	1.8225840 09	7.9749580-02	2.0714060-05	-9.9533420 05	2.3714360 C7	7.6723770 04
1.74361D C2	6.5015160 C3	1.8265710 09	8.0624440-02	1.9427720-05	-9.9652690 05	5.9812480 07	7.6723770 04
2.17952D C2	9.6237030 03	1.8292190 09	8.1443460-02	1.8151940-05	-9.9702890 05	9.6014870 07	7.6723770 04
2.61542D 02	1.2725930 04	1.8305340 09	8.2207080-02	1.6886110-05	-9.9684890 05	1.3228330 08	7.6723770 04
2.79703D C2	1.4031870 04	1.8306900 09	8.2513830-02	1.6904880-05	-9.9767280 05	1.4739410 08	7.6723770 04
2.97863D 02	1.5373680 C4	1.8306200 09	8.2821430-02	1.6980890-05	-9.9840620 05	1.6248730 08	7.6723770 04
3.16024D C2	1.6751870 C4	1.8303270 09	8.3130950-02	1.7115890-05	-9.9904670 05	1.7756040 08	7.6723770 04
3.34185D 02	1.8166910 C4	1.8298150 09	8.3443480-02	1.7311710-05	-9.9959190 05	1.9261100 08	7.6723770 04
3.52345D C2	1.9619330 C4	1.8290880 09	8.3760120-02	1.7570310-05	-1.0000400 C6	2.0763690 08	7.6723770 04
3.70506D C2	2.1109610 04	1.8281510 09	8.4082050-02	1.7893760-05	-1.0003870 06	2.2263630 08	7.6723770 04
3.88666D 02	2.2638260 04	1.8270070 09	8.4410450-02	1.8284270-05	-1.0006330 C6	2.3760750 08	7.6723770 04
4.06827D 02	2.4205790 C4	1.8256600 09	8.4746580-02	1.8744150-05	-1.0007730 06	2.5254890 08	7.6723770 04
4.24988D 02	2.5812690 C4	1.8241160 09	8.5091700-02	1.9275900-05	-1.0008060 06	2.6745950 C8	7.6723770 04
4.43148D C2	2.7459470 C4	1.8223780 09	8.5447160-02	1.9882130-05	-1.0007290 06	2.8233810 C8	7.6723770 04
4.61309D C2	2.9146610 C4	1.8204500 09	8.5814320-02	2.0566560-05	-1.0005410 06	2.9718410 08	7.6723770 04
5.11070D 02	3.3797970 C4	1.8142220 09	8.6790020-02	1.8650120-05	-9.9552810 05	3.3778480 08	7.6723770 04
5.60832D C2	3.8391540 04	1.8066200 09	8.7670510-02	1.6739630-05	-9.8938810 05	3.7830840 08	7.6723770 04
6.10593D C2	4.2922710 C4	1.7976570 09	8.8456130-02	1.4837320-05	-9.8214490 05	4.1873890 08	7.6723770 04
6.60354D C2	4.7387060 C4	1.7873470 09	8.9147360-02	1.2946190-05	-9.7382440 05	4.5905570 08	7.6723770 04

Figure 53. (Continued)

7.10116D 02	5.178043D 04	1.775705D 09	8.974482D-02	1.106908D-05	-9.644543D 05	4.5525390 08	7.6723770 04
7.59877D 02	5.609888D 04	1.762745D 09	9.024928D-02	9.208676D-06	-9.540640D 05	5.393[illegible]20 08	7.6723770 04
8.09638D 02	6.033873D 04	1.748483D 09	9.066162D-02	7.367509D-06	-9.426843D 05	5.791929[illegible] 08	7.6723770 04
8.59399D 02	6.449651D 04	1.732933D 09	9.098288D-02	5.547938D-06	-9.303471D 05	6.189C180 08	7.6723770 04
9.09161D 02	6.856898D 04	1.716113D 09	9.121417D-02	3.752160D-06	-9.170855D 05	6.5841240 08	7.6723770 04
9.58922D 02	7.255315D 04	1.698039D 09	9.135674D-02	1.982210D-06	-9.029334D 05	6.9770550 08	7.6723770 04
1.00868D 03	7.644622D 04	1.678726D 09	9.141192D-02	2.399619D-07	-8.879256D 05	7.3676150 08	7.6723770 04
1.05844D 03	8.024564D 04	1.658192D 09	9.138112D-02	-1.472872D-06	-8.720973D 05	7.7556030 08	7.6723770 04
1.10821D 03	8.394906D 04	1.636454D 09	9.126586D-02	-3.154736D-06	-8.554844D 05	8.1408110 08	7.6723770 04
1.15797D 03	8.755437D 04	1.613531D 09	9.106770D-02	-4.804226D-06	-8.381231D 05	8.5230290 08	7.6723770 04
1.20773D 03	9.105962D 04	1.589440D 09	9.078829D-02	-6.420089D-06	-8.200497D 05	8.9020410 08	7.6723770 04
1.25749D 03	9.446310D 04	1.564199D 09	9.042934D-02	-8.001215D-06	-8.013006D 05	9.2776260 08	7.6723770 04
1.30725D 03	9.776327D 04	1.537828D 09	8.999259D-02	-9.546633D-06	-7.819123D 05	9.6495580 08	7.6723770 04
1.35701D 03	1.0095880 05	1.510346D 09	8.947985D-02	-1.105551D-05	-7.619212D 05	1.0017610 09	7.6723770 04
1.40677D 03	1.0404840 05	1.481771D 09	8.889294D-02	-1.252713D-05	-7.413631D 05	1.0381540 09	7.6723770 04
1.45653D 03	1.0703130 05	1.452125D 09	8.823375D-02	-1.396090D-05	-7.202740D 05	1.0741120 09	7.6723770 04
1.50630D 03	1.0990650 05	1.421427D 09	8.750416D-02	-1.535636D-05	-6.986891D 05	1.1096100 09	7.6723770 04
1.55606D 03	1.1267330 05	1.389698D 09	8.670609D-02	-1.671314D-05	-6.766433D 05	1.1446250 09	7.6723770 04
1.60582D 03	1.1533130 05	1.356960D 09	8.584147D-02	-1.803098D-05	-6.541709D 05	1.1791310 09	7.6723770 04
1.65558D 03	1.1788000 05	1.323233D 09	8.491225D-02	-1.930971D-05	-6.313055D 05	1.2131030 09	7.6723770 04
1.70534D 03	1.2031920 05	1.288540D 09	8.392037D-02	-2.054926D-05	-6.080803D 05	1.2465170 09	7.6723770 04
1.75510D 03	1.2264880 05	1.252903D 09	8.286778D-02	-2.174963D-05	-5.845276D 05	1.2793470 09	7.6723770 04
1.80486D 03	1.2486870 05	1.216345D 09	8.175644D-02	-2.291091D-05	-5.606789D 05	1.3115670 09	7.6723770 04
1.85463D 03	1.2697910 05	1.178889D 09	8.058828D-02	-2.403325D-05	-5.365650D 05	1.3431530 09	7.6723770 04
1.90439D 03	1.2898020 05	1.140558D 09	7.936523D-02	-2.511687D-05	-5.122159D 05	1.3740780 09	7.6723770 04
1.95415D 03	1.3087230 05	1.101378D 09	7.808922D-02	-2.616205D-05	-4.876607D 05	1.4043180 09	7.6723770 04
2.00391D 03	1.3265570 05	1.061372D 09	7.676215D-02	-2.716913D-05	-4.629278D 05	1.4338470 09	7.6723770 04
2.05367D 03	1.3433110 05	1.020565D 09	7.538590D-02	-2.813848D-05	-4.380445D 05	1.4626400 09	7.6723770 04
2.10343D 03	1.3589890 05	9.789826D 08	7.396235D-02	-2.907053D-05	-4.130375D 05	1.4906710 09	7.6723770 04
2.15319D 03	1.3735990 05	9.366509D 08	7.249334D-02	-2.996575D-05	-3.879322D 05	1.5179160 09	7.6723770 04
2.20295D 03	1.3871460 05	8.935960D 08	7.098068D-02	-3.082463D-05	-3.627536D 05	1.5443490 09	7.6723770 04
2.25272D 03	1.3996400 05	8.498447D 08	6.942618D-02	-3.164769D-05	-3.375256D 05	1.5699480 09	7.6723770 04
2.30248D 03	1.4110880 05	8.054241D 08	6.783161D-02	-3.243550D-05	-3.122711D 05	1.5946870 09	7.6723770 04
2.35224D 03	1.4214990 05	7.603618D 08	6.619869D-02	-3.318861D-05	-2.870123D 05	1.6185430 09	7.6723770 04
2.40200D 03	1.4308830 05	7.146858D 08	6.452915D-02	-3.390764D-05	-2.617705D 05	1.6414930 09	7.6723770 04
2.45176D 03	1.4392490 05	6.684246D 08	6.282467D-02	-3.459317D-05	-2.365661D 05	1.6635130 09	7.6723770 04
2.50152D 03	1.4466070 05	6.216072D 08	6.108689D-02	-3.524583D-05	-2.114189D 05	1.6845830 09	7.6723770 04
2.55128D 03	1.4529690 05	5.742627D 08	5.931745D-02	-3.586624D-05	-1.863475D 05	1.7046780 09	7.6723770 04
2.60104D 03	1.4583440 05	5.264210D 08	5.751791D-02	-3.645504D-05	-1.613700D 05	1.7237800 09	7.6723770 04
2.65081D 03	1.4627450 05	4.781121D 08	5.568986D-02	-3.701286D-05	-1.365036D 05	1.7418660 09	7.6723770 04
2.70057D 03	1.4661820 05	4.293665D 08	5.383480D-02	-3.755033D-05	-1.117649D 05	1.7589170 09	7.6723770 04
2.75033D 03	1.4686670 05	3.802150D 08	5.195424D-02	-3.[illegible]03809D-05	-8.716948D 04	1.7749140 09	7.6723770 04
2.80009D 03	1.4702110 05	3.306891D 08	5.004963D-02	-3.850677D-05	-6.273244D 04	1.7898370 09	7.6723770 04

Figure 53. (Continued)

```
2.868850 03   1.470828D C5    2.8082020 08   4.812241C-02  -3.8947000-05  -3.8468090 04   1.8C36700 C9   7.E723770 04
2.89-S1D 03   1.4705270 05    2.3064040 08   4.6173980-02  -3.9359380-05  -1.4390070 04   1.8163940 09   7.E723770 04
2.949370 C3   1.4693230 C5    1.8018190 C8   4.4205710-02  -3.9744530-05   9.4885950 03   1.8279930 09   7.E723770 04
2.999140 C3   1.4672270 05    1.2947740 08   4.2218940-02  -4.0103060-05   3.8155550 04   1.8384530 09   7.E723770 04
3.048900 03   1.4642510 C5    7.8559780 07   4.021498C-02  -4.0435550-05   5.6599020 04   1.8477570 09   7.E723770 04
3.098660 03   1.4604080 C5    2.7462280 07   3.8195110-02  -4.0742580-05   7.9807830 04   1.8558520 09   7.E723770 04
3.148420 03   1.4557100 C5   -2.3781580 07   3.6160580-02  -4.1024710-05   1.0277140 05   1.8628450 09   7.E723770 04
3.198180 03   1.4501690 05   -7.5138C40 07   3.4112630-02  -4.1282500-05   1.2547950 05   1.8686C40 09   7.E723770 04
3.247940 C3   1.4437980 C5   -1.2657310 08   3.2052480-02  -4.1516490-05   1.4792270 05   1.8731570 09   7.E723770 04
3.297700 03   1.4366100 05   -1.7805240 08   2.9981180-02  -4.1727180-05   1.7009200 05   1.8764950 09   7.E723770 04
3.347460 C3   1.4286150 05   -2.2954160 08   2.7900010-02  -4.1915100-05   1.9197860 05   1.8796C80 09   7.E723770 04
3.397230 C3   1.4198280 05   -2.8100610 08   2.5810050-02  -4.2080730-05   2.1357460 05   1.8794860 09   7.E723770 04
3.446990 C3   1.4102600 C5   -3.3241080 08   2.3712390-02  -4.2224550-05   2.3487210 05   1.8791240 09   7.E723770 04
3.496750 03   1.3999230 C5   -3.8372100 08   2.1608110-02  -4.2347010-05   2.5586390 05   1.8775140 09   7.E723770 04
3.546510 C3   1.3888290 C5   -4.3490130 08   1.9498250-02  -4.2448550-05   2.7654310 05   1.8746520 09   7.E723770 04
3.596270 C3   1.3769910 05   -4.8591670 08   1.7383850-02  -4.2529590-05   2.9690330 05   1.8705320 09   7.E723770 04
3.646030 C3   1.3644200 C5   -5.3673160 08   1.5265930-02  -4.2590550-05   3.1693820 05   1.8651510 09   7.E723770 04
3.695790 03   1.3511280 C5   -5.8731070 08   1.3145460-02  -4.2631800-05   3.3664210 05   1.8585060 09   7.E723770 04
3.745550 03   1.3371280 C5   -6.3761850 08   1.1023420-02  -4.2653720-05   3.5600950 05   1.8505980 09   7.E723770 04
3.795320 03   1.3224310 C5   -6.8761950 08   8.9007620-03  -4.2656660-05   3.7503530 05   1.8414240 09   7.E723770 04
3.845080 03   1.3070480 05   -7.3727600 08   6.7784240-03  -4.2640960-05   3.9371450 05   1.8309870 09   7.E723770 04
3.894840 03   1.2909920 C5   -7.8655850 08   4.6573260-03  -4.2606930-05   4.1204260 05   1.8192870 09   7.E723770 04
3.944600 C3   1.2742730 C5   -8.3542550 08   2.5383700-03  -4.2554870-05   4.3001530 05   1.8063280 09   7.E723770 04
3.964070 C3   1.2677480 05   -8.5442570 08   1.7750410-03  -3.5831710-05   4.3694650 05   1.8009150 09   7.E723770 04
3.983540 C3   1.2615220 C5   -8.7335550 C8   1.1439950-03  -2.8965960-05   4.4382950 05   1.7953080 09   7.E723770 04
4.003010 C3   1.2556080 C5   -8.9221480 08   6.4805490-04  -2.1954090-05   4.5066540 05   1.7895060 09   7.E723770 04
4.022480 C3   1.2500200 C5   -9.1100120 08   2.9007150-04  -1.4792180-05   4.5745530 C5   1.7835C70 09   7.E723770 04
4.041950 C3   1.2447720 C5   -9.2971380 08   7.3038200-05  -7.4758000-06   4.6419980 05   1.7773090 09   7.E723770 04
4.061420 03   1.2398800 C5   -9.4835130 08  -9.2091500-13   2.0295080-15   4.7089940 05   1.7709110 05   7.E723770 04
4.105600 03   1.2289780 C5   -9.9035410 08  -1.0449440-12  -7.5064280-17   4.8593550 05   1.7556490 09   7.E723770 04
4.149780 03   1.2176520 C5   -1.0319350 09   6.5662100-13   2.8182310-16   5.0074070 05   1.7393620 09   7.E723770 04
4.193960 03   1.2059060 05   -1.0730840 09   9.3611520-13   1.1591840-15   5.1531410 05   1.7220640 09   7.E723770 04
4.238140 03   1.1937470 C5   -1.1137630 09  -3.1245890-13   5.2557240-16   5.2965470 05   1.7037680 09   7.E723770 04
4.282320 03   1.1811820 C5   -1.1539540 C9  -1.3684780-12   1.4804170-17   5.4376140 05   1.6844880 09   7.E723770 04
4.326490 C3   1.1682160 05   -1.1936340 09   4.8042580-13  -3.0739340-15   5.5763330 05   1.6642390 09   7.E723770 04

DOUBLE PRECISION INITIAL CONDITIONS FCR CPTIPUM TRAJECTCRY

   6.06817475918781D 03      1.0745725631713SD 00      4.152178425013220-01      5.9509111751551800-04
   9.999999999999999D 02     2.7944873046875D 02      -2.104095787903170 00     -6.16794911C593160 C3
   1.80253997C131290 09      7.67851243302CC2D-02      2.464208297799390-05      -9.875221411054640 05
  -8.35718723130163D 07      7.6723765211444 2D 04     1.0679064147137 8D 03

   END OF ITERATICN NUMBER          4
```

Figure 53. (Continued)

```
ENTER KHIST --- IGO =        2

STOPPING CONDITIONS
          ON                    OFF                    ON                    OFF                   END                  HBAR.                VCHAR
    0  2.653728380 02  4.651883500 02  3.947451060 03  4.064490070 03  4.329862910 03 -2.019434760-01  4.902651220 03
    1  2.613765280 02  4.611920400 02  3.949751840 03  4.066830850 03  4.329862910 03 -4.029459530 00  4.904152400 03
    2  2.595440010 02  4.591938850 02  3.950962230 03  4.068001240 03  4.329862910 03 -3.973980100 00  4.902782100 03
    3  2.598018180 02  4.592767930 02  3.951468960 03  4.068879760 03  4.329862910 03 -3.967243820 00  4.903010400 03
    4  2.598079700 02  4.592833500 02  3.951549670 03  4.068671910 03  4.329862910 03 -3.966335130 00  4.902594150 03
    5  2.598078670 02  4.592832750 02  3.951549260 03  4.068668550 03  4.329862910 03 -3.966150510 00  4.902588780 03

NEW QASLIN ITERATION SET -- FINAL TIME CHANGED.
    7  2.595801060 02  4.588806430 02  3.948085120 03  4.065101740 03  4.326067120 03 -3.966150750 00  4.902589050 03
    8  2.615782610 02  4.612881080 02  3.945744340 03  4.062750960 03  4.326067120 03  4.398822230-01  4.922602910 03
    9  2.611746358 02  4.611578330 02  3.944573950 03  4.061590570 03  4.326067120 03  5.015341970-01  4.902664520 03
   10  2.611755986 02  4.611607220 02  3.943394620 03  4.060676100 03  4.326067120 03  5.031425200-01  4.902635740 03
   11  2.611549960 02  4.611556770 02  3.943717100 03  4.060505260 03  4.326067120 03  5.034218990-01  4.902680510 03
   12  2.611755069 02  4.611597180 02  3.943722130 03  4.060505820 03  4.326067120 03  5.033000530-01  4.902671190 03
   13  2.611755069 02  4.611597180 02  3.943722080 03  4.060505860 03  4.326067120 03  5.033033320-01  4.902671200 03

NEW QASLIN ITERATION SET -- FINAL TIME CHANGED.
   15  2.617800180 02  4.616052990 02  3.944111540 03  4.060906850 03  4.326494340 03  5.030033320-01  4.902671200 03
   16  2.615382480 02  4.613058740 02  3.944612290 03  4.061425290 03  4.326494340 03 -9.885446400-03  4.903217390 03
   17  2.615418850 02  4.613091020 02  3.944599510 03  4.061421050 03  4.326494340 03 -9.304491630-03  4.902688550 03
   18  2.615418590 02  4.613090750 02  3.944599720 03  4.061420870 03  4.326494340 03 -9.273718730-03  4.902669470 03
STOP POINTS FOUND FOR LAST ITERATION
       2.615419000 02  4.613090720 02  3.944599730 03  4.061420830 03
```

Figure 54.

```
 SOLUTION CHECK BY INTEGRATION OF INITIAL CONDITIONS
STOP POINT NO.     1  TIME =    0.2615419CD 03  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    C.100D 01
STOP POINT NO.     2  TIME =    0.46130SC7D 03  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    C.0
STOP POINT NO.     3  TIME =    0.39445S57D 04  NULL BURN CONTROL =          0.0
THRUST CHANGE---BETA =    C.100D 01
STOP POINT NO.     4  TIME =    0.406142C5D C4  NULL BURN CONTROL =          C.0
THRUST CHANGE---BETA =    0.0
STOP POINT NO.     5  TIME =    0.43264943D 04  NULL BURN CONTROL =          0.0

 CONVERGENCE PARAMETERS----RHO

1.57D-04 2.24C-08 1.00D-C7 1.90D-11 2.62D-06 4.22D-05 5.53D-05 9.17C-03 3.37C 02 7.80C-C9 1.10D-11 4.C0D-C2
1.33D 02
```

Figure 55.

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

Since the data parameters indicated that further investigation of the suspected minima was desired, the program generated four detailed contour maps (Figures 31-34). The program then automatically conducted a steep descent search for a local minimum within each suspected optimum region. The results of the searches are tabulated in Figures 35-38.

The printout for the steep-descent optimization is rather straightforward. It includes the three independent variables (ϕ_1, ϕ_2, p_t), their partial derivatives and the impulse required. The "GAINS" are the current values of the s_i, "AMP" is the final value of α, and "IFAIL" is the number of times Equation 50 of Ref. 11 is not satisfied. The last two columns of each iteration's printout contain the iteration number and the current value of IFAIL.

Figures 39-42 give the initial, final, and transfer orbits associated with each local minimum. The descent path followed by the program is noted by a sequence of characters plotted near the center of each of the previously noted local area contour maps. Note that in these cases the search initiation point was quite near the search termination point. The four transfer orbits corresponding to the four locally minimum transfers between the initial and final orbits are quite interesting to examine. Note that two of these transfers are internal—that is, they tend to be inside both the initial and final orbits—and that two of the transfer orbits are external. Note also that the transfer angle for two of the cases is greater than 180°.

At this point in the computations the programmer has the option (by using IUSE on card 20) of utilizing the two-impulse transfer circumstances from the optimum with least impulse. By specifying zero value of IUSE he can direct the program to perform an additional steep descent using the initial values given on card 03. Either of these procedures will yield an optimum two-impulse transfer that will serve as the basis for further computations. Note that the latter procedure was used here and that an additional steep descent resulted (Figure 43).

Control is then transferred to subroutine XFIRST which produces the output concerning the vehicle specification, storage table specification, primer vector computations, and complete time histories of the state variables and Lagrange multipliers. This output is shown in Figures 44 through 46. The program next enters subroutine QUTPUT, as it does many times during the computations, and produces a listing of the time histories of the primer vector, switching function, steering angles, Hamiltonian, energy and angular momentum (Figure 47). These particular parameters are especially useful for watching the iteration-by-iteration progress of the quasilinearization solution procedure. The program next enters subroutine KHIST and completes initial computations. At this point subroutine QASLN is entered for the first time. Each iteration of quasilinearization produces a sequence of output similar to that shown in Figures 48, 49, and 50. The output of Figure 48 consists of five lines generated at each thrust initiation or termination point which allows the user to note the convergence of the

switching function and the quantity Δt specified in Equation 114. At the end of the iteration the combination coefficients of Equation 93 are printed. Note that there are six of these, corresponding to the six unknown boundary conditions. The convergence parameters printed next are obtained by an evaluation of Equation 95 and indicate how closely the n+1'st iteration corresponds to the n'th iteration. These parameters should approach zero as the quasilinearization process converges. The first of these, the difference in radius (r) is examined for convergence and the process is terminated if it is less than RHØS1 (data card 16). The call to ØUTPUT is then repeated and the control parameters and problem constants are printed for the current iteration (Figure 49). The program next enters subroutine KHIST and does an analysis of the switching points indicated by the current time history of the switching function. These currently determined switching times are represented by the last line appearing in Figure 50. Based upon this evaluation it may allow changes to be made prior to executing the next iteration. The magnitude of these changes are constrained according to the tolerances entered on data card 13. These constraints are usually necessary during the first iterations wherein the program has not stabilized to the point where it is providing information more accurate than that of the initial iterate. Once the changes indicated by the current switching function time history become smaller than the tolerances, the switching function is used to provide the estimate of the burn initiation and termination times to be used for the next iteration. Figure 50 also contains two columns

entitled HBAR and VCHAR which represent the average Hamiltonian and the characteristic velocity determined for that particular iteration. As will be seen later, the stopping conditions table includes one line for each iteration. That is, a new line will be added at the end of each iteration and the final table generated will contain a complete summary of all of the changes utilized during the computations.

The above process continues for "ITSQL" iterations and the printout of Figures 48 through 50 is repeated for each iteration. As described in Appendix I, the program then changes the total time and performs another set of iterations. A number of such sets are computed depending upon the magnitude of data parameter "IQCYCL." At the end of IQCYCL sets, the last iteration will provide output similar to that of Figures 51 through 54. Inspection of the combination coefficients and convergence parameters in Figure 51 indicates that they are approaching zero as expected. The values of k and the small time corrections at each switch point are also seen to be very small. Note that the switching time summary table (Figure 54) has now expanded, and also that the switching times, HBAR, and characteristic velocity have all converged. One may also observe that the switching times computed from the last iteration, the last line in Figure 54, closely agree with the switching times used as a basis for computing the last iteration. This feature is more easily recognized in Figure 52 wherein the value of the switching function is seen to be near 10^{-9} at each of the switch points. Note also, in Figure 52, that the Hamiltonian is rather small and nearly

constant. A better idea of the numerical accuracy of the calculations can be obtained by inspection of total energy and angular momentum data in the last columns of Figure 52. Figure 53 represents the converged solutions for all 14 variables. These data are also plotted by subroutine QPLØT if desired. These plots, comparing the converged solutions with the initial iterates, have already been included in the body of the report as Figures 6 through 21.

One last feature of the program is worth noting. Figure 55 was produced when the program was requested to integrate the complete set of initial conditions appearing at the bottom of Figure 53. This reintegration was performed without the use of quasilinearization algorithm and therefore provides a check of the original solution. The convergence parameters for the 14 variables appearing at the bottom of Figure 55 represent the maximum point-by-point difference between the two tabulated solutions. They were obtained by applying an algorithm similar to Equation 95.

The total IBM 360 Model 65 computer time required to produce the entire orbit analysis appearing in this appendix was about 3.5 minutes.

APPENDIX III

COMPUTER PROGRAM LISTINGS

```
C     ----------------------------------------------------------------MAIN0030
C     EXECUTIVE PROGRAM FOR THE FINITE-THRUST OPTIMIZING PACKAGE.      MAIN0040
C     PERFORMS COMMON ASSIGNMENTS AND HANDLES THE LOGICAL OPERATIONS   MAIN0050
C     ASSOCIATED WITH SOLVING A PARTICULAR OPTIMIZATION PROBLEM.       MAIN0060
C     ----------------------------------------------------------------MAIN0070
      IMPLICIT REAL*8(A-H , O-Z)                                       MAIN0080
      LOGICAL  RATES ,          PRINT , CRTOUT                         MAIN0090
      REAL*8 IA                                                        MAIN0100
      REAL*4 XLST,HI(13,6,100),EPSP,EPSPI,RHOS,XERR,DYN,RESET,TRU,TWOBRNMAIN0110
     *    , STO                                                        MAIN0120
      COMMON/JACSET/ ID, JACPRN                                        MAIN0130
      COMMON / ZI / TX, H ,NI,NIC,NSC, NMIN,NMAX,NRST, TNSTOR, NDIF,NBP MAIN0140
     A , DTPRNT , XK(20)                                               MAIN0150
      COMMON /ZX/ XT(10),EPSP,EPSPI,IA(20)                             MAIN0160
      COMMON /ZSUB/          TSUB(10),VARIND(100),XMNSTP(9),STPSIZ(9), MAIN0170
     *KSUB(10) , MAXSTP                                                MAIN0180
      COMMON/ZOO/N(5), MX, XERR,DYN,B(10),L(10),M(10),MT(10),RHOS(20)  MAIN0190
      COMMON /ZIC/BETA,  C , GG ,XMU,TW,XXK(20)                        MAIN0200
      COMMON /SOLN/ XLST(15,100), TRU(100), HBAR, VCHAR, STO(100,20,2) MAIN0210
      DIMENSION              XJAC(13,14),P(13,100),CTP(6,1), T(6,6,1)  MAIN0220
     A, DXKP1(14,13), XKP1(14,13), ACUM(14,13)                         MAIN0230
     X                      ,XKS2(9,100),XKS3(13,100)                 MAIN0240
       COMMON/ZCONST/ PI,RAD,GXM,BETAS,CONV,II,JJ                      MAIN0250
C     INPUT DATA COMMON BLOCK                                          MAIN0260
C     COMMON FOR STEEP DESCENT(FIRST SIX CARDS)                        MAIN0270
      COMMON /INDATA/          XORB1(6),  XORB2(6),   AEST(6)          MAIN0280
     A, AMPINC   , AMPDEC   , FREEZE   , XAMP     , SAVE     , BL46    MAIN0290
     B, SCLINC   , SCLDEC   , ISCL     , IAMP     , BL55     , BL56    MAIN0300
     C, SCALE(3)                       , TOL      , MAXIT    , IPRIN   MAIN0310
C     COMMON FOR LASLIN ORB. TRAN. PROG.--( TWO DIM.CASE)              MAIN0320
     D, XIN(14),  XEN(14),  RATES, IDIMEN                              MAIN0330
     I, BETASV  ,  XISP     ,  TEPERT ,  PERTX    ,  COAST1    ,  BL126  MAIN0340
     J, BSNINC  ,  BMAX1    ,  BMAX2  ,  BSH1     ,  BSH2      ,  TWOBRN MAIN0350
     K, RESET   ,  KMULT    ,   NOPRT ,  PRNJ     ,  BL145     ,  BL146  MAIN0360
     L, INTSTP  ,  PRINT    ,  CRTOUT ,  ITSOL    , IOPRIN     , IOCYCL  MAIN0370
     M, HBARTL  ,  RHOSI , BL163    ,  BL164  ,   BL165      ,  BL166   MAIN0380
     N, XNB         ,  XNC  ,     DTB     ,  DTC      ,  JMAX    ,IBL176 MAIN0390
      COMMON /STOP1/ XKDOT(10), TWOBR , XNOCOP, SAVK(7,6),IOCYC        MAIN0400
      EQUIVALENCE (INT,XERR)                                           MAIN0410
```

Figure 56. Main Program Listing

```
C     ------------------------------------------------------------------MAIN0500
C        FIRST DATA READ IS IN SUBROUTINE STPCON                        MAIN0510
C     ------------------------------------------------------------------MAIN0520
  100 CALL STPCON                                                       MAIN0530
      JMAX = 100                                                        MAIN0540
      IDIMEN = 3                                                        MAIN0550
      IF ( XORB1(3) .EQ. 0.0 .AND. XORB2(3) .EQ. 0.0 ) IDIMEN = 2       MAIN0560
      ID = IDIMEN                                                       MAIN0570
      TWOBP = TWOBRN                                                    MAIN0580
      IQCYC = IQCYCL                                                    MAIN0590
      INT=INTSTP                                                        MAIN0600
      JACPRN = PRNJ                                                     MAIN0610
      C = GO * XISP                                                     MAIN0620
      BETA   = BETASV                                                   MAIN0630
      BETAS = BETASV                                                    MAIN0640
      TW =  XISP * BETA / XIN(7)                                        MAIN0650
      CALL XFIRST                                                       MAIN0660
      CALL TIMEV(EXTIME)                                                MAIN0670
      WRITE(6,530) EXTIME                                               MAIN0680
      CALL JACOB( TX,XK,XK,XJAC,0)                                      MAIN0690
      HSAV = 0.                                                         MAIN0700
      N1 = 9                                                            MAIN0710
      N(1) = N1                                                         MAIN0720
      N2 = 4                                                            MAIN0730
      N(2) = N2                                                         MAIN0740
      N(3) = ITSOL                                                      MAIN0750
      N(4) = IOPRIN                                                     MAIN0760
      N(5) = 0                                                          MAIN0770
      NDIF = 2                                                          MAIN0780
      NU = 10                                                           MAIN0790
      NTM= 1                                                            MAIN0800
      NS              = N2 + 1                                          MAIN0810
      RHOS(1) = RHOS1                                                   MAIN0820
      DO 10 I=2,14                                                      MAIN0830
   10 RHOS(I) =  1.E+20                                                 MAIN0835
      XTEP = TEPERT * ( TSUB(5) - TSUB(2) ) / 100.00                    MAIN0838
      PERT = PERTX * ( TSUB(5) - TSUB(2) ) / 100.00                     MAIN0840
```

Figure 56. Main Program Listing (Continued)

```
      TCHANG = ( TSUB(6) + XTFP    ) / TSUB(6)                                  MAIN0841
      DO 1115  I = 1,100                                                        MAIN0842
 1115 VARIND(I) = VARIND(I) * TCHANG                                            MAIN0843
      DO 1116  I = 1,6                                                          MAIN0844
 1116 TSUB(I) = TSUB(I) * TCHANG                                                MAIN0845
C     ***************************************************************          MAIN0850
      IF (ID.EQ.3) GO TO 300                                                    MAIN0860
      DO 130 I=1,4                                                              MAIN0870
      XI(I)= 1.D0                                                               MAIN0880
      M(I) = I                                                                  MAIN0890
      MI(I)= I     +6                                                           MAIN0900
  130 L(I) =       JMAX                                                         MAIN0910
       EPSP = -1.                                                               MAIN0920
      DO 135 I = 1,JMAX                                                         MAIN0930
      XKS2(1,I) = XLST(1,I)                                                     MAIN0940
      XKS2(2,I) = XLST(3,I)                                                     MAIN0950
      XKS2(3,I) = XLST(4,I)                                                     MAIN0960
      XKS2(4,I) = XLST(6,I)                                                     MAIN0970
      XKS2(5,I) = XLST(7,I)                                                     MAIN0980
      XKS2(6,I) = XLST(9,I)                                                     MAIN0990
      XKS2(7,I) = XLST(8,I)                                                     MAIN1000
      XKS2(8,I) = XLST(11,I)                                                    MAIN1010
      XKS2(9,I) = XLST(13,I)                                                    MAIN1020
      DO 135  J=1,9                                                             MAIN1030
  135 XLST(J,I) = SNGL( XKS2(J,I) )                                             MAIN1040
      DO 138  I = 1,7                                                           MAIN1050
      J = I + 1/2                                                               MAIN1060
C        J = 1, 3, 4, 6, 7                                                      MAIN1070
      XKS2(I,1) = XIN(J)                                                        MAIN1080
  138 XKS2(I,JMAX) = XFN(J)                                                     MAIN1090
      DO 140 I=1,4                                                              MAIN1100
  140  B(I) = XKS2(I,JMAX)                                                      MAIN1110
      XK(10) = XLST(14,1)                                                       MAIN1120
      CALL OUTPUT (XKS2,N1,JMAX,5)                                              MAIN1130
      CALL KHIST (1,XKS2,N1,JMAX)                                               MAIN1140
  142 CALL QASLN (XKS2,P,H1,XJAC,T CTP,       NTM,N1,N2,NU,JMAX                 MAIN1150
     * , DXKP1 , XKP1 , ACUM ,      NS)                                         MAIN1160
```

Figure 56. Main Program Listing (Continued)

```
      CALL TIMEV(EXTIME )                                                    MAIN1170
      WRITE(6,530) EXTIME                                                    MAIN1180
      IF( HSAV .NE. 0. ) GO TO 109                                           MAIN1190
      PERT =DSIGN( PERT , HBAR )                                             MAIN1200
      GO TO 110                                                              MAIN1210
  109 SLOPE =-( HSAV - HBAR )/ PERT1                                         MAIN1220
      PERT1= - HBAR / SLOPE                                                  MAIN1230
      PERT = PERT1                                                           MAIN1240
  110 PERT1 = PERT                                                           MAIN1250
      HSAV  = HBAR                                                           MAIN1260
      TCHANG  =  (  TSUB(6) +       PERT )/ TSUB(6)                          MAIN1270
      DO 115 I=1,100                                                         MAIN1280
  115 VARIND(I) = VARIND(I) * TCHANG                                         MAIN1290
      DO 116 I=1,6                                                           MAIN1300
  116 TSUB(I) = TSUB(I) * TCHANG                                             MAIN1310
      CALL KHIST(3,XKS2,N1,JMAX)                                             MAIN1320
      IQCYC = IQCYC - 1                                                      MAIN1330
      IF (IQCYC  .NE. 0) GO TO 142                                           MAIN1340
      IF ( CRTOUT ) CALL QPLOT(XKS2,N1,JMAX,VARIND)                          MAIN1350
      GO TO 520                                                              MAIN1360
C     ***************************************************************      MAIN1370
  300 DO 330  I=1,3                                                          MAIN1380
       XI(I) = 1.00                                                          MAIN1390
      XI(I+3) = 1.00                                                         MAIN1400
       M(I) = I                                                              MAIN1410
       MI(I) = I + 6                                                         MAIN1420
  330 MI(I+3) = I + 11                                                       MAIN1430
      DO 331 I = 1,6                                                         MAIN1440
  331 L(I) = JMAX                                                            MAIN1450
      M(4) = 4                                                               MAIN1460
      M(5) = 10                                                              MAIN1470
      M(6) = 11                                                              MAIN1480
C     M  VARIABLE NUMBERS ARE  1,  2,  3,  4, 10, 11                         MAIN1490
C     MI VARIABLE NUMBERS ARE  7,  8,  9, 12, 13, 14                         MAIN1500
      N1 = 13                                                                MAIN1510
       N(1) = N1                                                             MAIN1520
      N2 = 6                                                                 MAIN1530
      N(2) = N2                                                              MAIN1540
```

Figure 56. Main Program Listing (Continued)

```
      NS = N2 + 1                                                        MAIN1550
      NU = 14                                                            MAIN1560
      EPSP = -1.                                                         MAIN1570
      HSAV = 0.                                                          MAIN1580
      DO 350  I=1,JMAX                                                   MAIN1590
      XKS3(1,I) = XLST(1,I)                                              MAIN1600
      XKS3(2,I) = XLST(3,I)                                              MAIN1610
      XKS3(3,I) = XLST(4,I)                                              MAIN1620
      XKS3(4,I) = XLST(6,I)                                              MAIN1630
      XKS3(5,I) = XLST(7,I)                                              MAIN1640
      XKS3(6,I) = XLST(9,I)                                              MAIN1650
      XKS3(7,I) = XLST(8,I)                                              MAIN1660
      XKS3(8,I) = XLST(11,I)                                             MAIN1670
      XKS3(9,I) = XLST(13,I)                                             MAIN1680
      XKS3(10,I) = XLST(2,I)                                             MAIN1690
      XKS3(11,I) = XLST(5,I)                                             MAIN1700
      XKS3(12,I) = XLST(10,I)                                            MAIN1710
      XKS3(13,I) = XLST(12,I)                                            MAIN1720
      DO 350  J=1,13                                                     MAIN1730
  350 XLST(J,I) = SNGL( XKS3(J,I) )                                      MAIN1740
      DO 338  I = 1,5                                                    MAIN1750
      J = I + I/2                                                        MAIN1760
      XKS3(I,1) = XIN(J)                                                 MAIN1770
  338 XKS3(I,JMAX) = XFN(J)                                              MAIN1780
      XKS3(10,    1) = XIN(2)                                            MAIN1790
      XKS3(10,JMAX) = XFN(2)                                             MAIN1800
      XKS3(11,    1) = XIN(5)                                            MAIN1810
      XKS3(11,JMAX) = XFN(5)                                             MAIN1820
       DO 340  I = 1,4                                                   MAIN1830
  340  B(I) = XKS3(I,JMAX)                                               MAIN1840
      B(5) = XKS3(10,JMAX)                                               MAIN1850
      B(6) = XKS3(11,JMAX)                                               MAIN1860
      XK(14) = XLST(14,1)                                                MAIN1870
C     PERFORM QUASILINEARIZATION                                         MAIN1880
      CALL OUTPUT (XKS3,N1,JMAX,5)                                       MAIN1890
      CALL KHIST (1,XKS3,N1,JMAX)                                        MAIN1900
  342 CALL QASLN (XKS3,P,H1,XJAC,T,CTP,NTM,N1,N3,NU,JMAX,DXKP1,XKP1,     MAIN1910
     *    ACUM,NS)                                                       MAIN1920
```

Figure 56. Main Program Listing (Continued)

```
      CALL TIMEV (EXTIME)                                                MAIN1930
      WRITE (6,530) EXTIME                                               MAIN1940
      IF( HSAV .NE. 0. ) GO TO 309                                       MAIN1950
      PERT =DSIGN( PERT , HBAR )                                         MAIN1960
      GO TO 310                                                          MAIN1970
  309 SLOPE =-( HSAV - HBAR )/ PERT1                                     MAIN1980
      PERT1= - HBAR / SLOPE                                              MAIN1990
      PERT = PERT1                                                       MAIN2000
  310 PERT1 = PERT                                                       MAIN2010
      HSAV  = HBAR                                                       MAIN2020
      TCHANG  =  (  TSUB(6) +       PERT )/ TSUB(6)                      MAIN2030
      DO 315 I=1,100                                                     MAIN2040
  315 VARIND(I) = VARIND(I) * TCHANG                                     MAIN2050
      DO 316 I=1,6                                                       MAIN2060
  316 TSUB(I) = TSUB(I) * TCHANG                                         MAIN2070
      CALL KHIST(3,XKS3,N1,JMAX)                                         MAIN2080
      IQCYC = IQCYC - 1                                                  MAIN2090
      IF (IQCYC  .NE. 0) GO TO 342                                       MAIN2100
      IF ( CRTOUT ) CALL QPLOT(XKS3,N1,JMAX,VARIND)                      MAIN2110
  520 CALL TIMEV(EXTIME )                                                MAIN2120
      WRITE(6,530) EXTIME                                                MAIN2130
  530 FORMAT(23H0 EXEC.TIME (MAIN PROG)   F8.3    )                      MAIN2140
      GO TO 100                                                          MAIN2150
      END                                                                MAIN2160
```

Figure 56. Main Program Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
      SUBROUTINE STPCON                                                 STPC0000
C     ------------------------------------------------------------------STPC0010
C     THIS SUBROUTINE IS THE MAIN PROGRAM FOR ANALYZING TWO-IMPULSE     STPC0020
C     TRANSFERS BY CONTOURING AND STEEP DESCENT.  IN THE PRESENT        STPC0030
C     APPLICATION IT HAS BEEN CONVERTED TO AN EXECUTIVE SUBROUTINE.     STPC0040
C     ------------------------------------------------------------------STPC0050
C     THREE DIM. VERSION--IBM 360--JANUARY 1969                         STPC0060
      IMPLICIT REAL*8(A-H , O-Z)                                        STPC0070
      REAL*8 IA                                                         STPC0080
      REAL*4  EPSP, EPSP1, TITEST                                       STPC0090
      REAL*4 PAD, SPAR, SCXY, SCIMP, S2XY, S3XY, S6IMP,TITLE,BODY,UNSYS STPC0100
      DIMENSION TITLE(18), TITEST(10), UNSYS(3), BODY(3)                STPC0110
      DIMENSION A(3), XGRAD(3), VN(3), XB(3),GRAD(3), STEP1(3), GSAV    STPC0120
     1(3),SX(3),UTAM(3), SSV(3)                                         STPC0130
      DIMENSION  DATA(150),MASK(6,25)                                   STPC0140
      DIMENSION  THX1(2,3), THX2(2,3)                                   STPC0150
      DIMENSION  SRAD1(3), SRAD2(3)                                     STPC0160
      DIMENSION Z(36,36,2), XORD(36), YORD(36)                          STPC0170
        DIMENSION G(6,2),CONVR(4), GR(4)                                STPC0180
      COMMON /ZX/ XI(10),EPSP,EPSP1,IA(20)                              STPC0190
      COMMON /ZIC/BETA,  C , GO ,XMU,TW,XXK(20)                         STPC0200
      COMMON /OPTS/ ZMINS(10), XMINS(10), YMINS(10), PMINS(10),IRANK(10)STPC0210
C     INPUT DATA COMMON BLOCK                                           STPC0220
      COMMON /INDATA/          XORB1(6),  XORB2(6),   AEST(6)           STPC0230
     A, AMPINC   , AMPDEC   , FREEZE   , XAMP     , SAVE     , BL46     STPC0240
     B, SCLINC   , SCLDEC   , ISCI     , IAMP     , BL55     , BL56     STPC0250
     C, SCALE(3)                       , TOL      , MAXIT    , IPRIN    STPC0260
     D, PAD(130) , IMAPS, NSTEEP, IUSE, SCXY,SCIMP, IPAD3, SPAR(6,5)    STPC0270
      COMMON /ORBCON/ P1,E1,W1, XI1,P2,E2,W2,P,E,W,ETRN(6)              STPC0280
      COMMON /STCOM/           Q1(3), Q2(3),XE(3),XU(3),XV(3),AUX(3), CSTPC0290
     1XI,DVT(5),RIE,R2E,SXI,TAM(3),UN1(3),UN2(3),VE1(3),VE2(3          STPC0300
     2),XL1,XL2,XS1,XS2,PERT,PER1,PER2,PAD1(3),PAD2(3),UMAG,XIM1(3,2),XISTPC0310
     3M2(3,2),DTHETA,EPSI,ZETEST,                                       STPC0320
     4JN , LOS , KONTAU, IBOX                                           STPC0330
        COMMON/ZCONST/ PI,RAD,GXM,BETAS,CONV,II,JJ                      STPC0340
C     DEFINE DATA MASK FOR PRODUCTION PROGRAM                           STPC0350
      DATA MASK / 26*2, 3, 3, 6*2, 3, 3, 28*2, 5, 3, 11*2, 1, 1, 3, 3,  STPC0360
     *2, 2, 2, 3, 5, 5, 3, 3, 3, 10*2, 3, 13*1, 3, 3, 3, 1,  1, 3, 30*1/STPC0370
```

Figure 57. Subroutine STPCØN Listing

```
      DATA TITEST /'BRITMETEKILOBODYEARTMOONSUN VENUMARSJUPI'/                  STPC0380
        DATA CONVR / 5280.D0, 1000.D0, 1.D0, 1.D0 /                             STPC0390
       DATA G/ 95634.501D0, 1176.235D0, 3.1787279D+10, 77787.1D0,               STPC0400
     * 10293.7D0, 3.0348570+07,   398601.5D0, 4902.75D0, 13.24948D+10,          STPC0410
     * 324230.D0, 42906.D0, 12.6498D+07  /                                      STPC0420
       DATA GR / .0060984800 ,   9.811776800 , .009811776800, 1.D0 /            STPC0430
C     GR IS WEIGHT/MASS OR ACCEL OF GRAV AT EARTH (BRITISH,METRIC,METRIC)       STPC0440
C      G(I,J) ARE GXM IN (J=1)MI**3/SEC**2, OR (J=2)K**3/SEC**2                 STPC0450
C         EARTH,     MOON,      SUN,       VENUS,     MARS,      JUPITER       STPC0460
C      FOR  I=1       I=2        I=3        I=4        I=5        I=6           STPC0470
       EQUIVALENCE( DATA,XORB1 )                                                STPC0480
C     ***************************************************************************STPC0490
C     CONSTANTS                                                                 STPC0500
       PI = 3.14159 26535 89793 24                                              STPC0510
       RAD  = 180.D0/PI                                                         STPC0520
       EPSP = -1.                                                               STPC0530
       ZETEST = 1.0                                                             STPC0540
       KONTAU=3                                                                 STPC0550
       EPST = 1.E-05                                                            STPC0560
       CALL CLOCK( TIME )                                                       STPC0570
       CALL CDATEV( DATE )                                                      STPC0580
       CALL COUNTV                                                              STPC0590
       WRITE(6,10) DATE , TIME                                                  STPC0600
    10 FORMAT('1 DATE ',A8, ',TIME =',  F8.3 // )                               STPC0610
C      TWO CARDS A, B ARE REQUIRED.  ON CARD A THE TITLE AT USER'S              STPC0620
C       DISCRETION (72 COLUMNS).                                                STPC0630
C      THE ATTRACTING CENTER CAN BE EARTH, MOON, SUN, VENUS, MARS,              STPC0640
C       JUPITER, OR ANY BODY SPECIFIED (12 COLUMNS STARTING AT COLUMN 1,        STPC0650
C       CARD B)                                                                 STPC0660
C      IF BODY IS USED, GXM IS REQUIRED IN COLUMNS 25-36, CARD B.               STPC0670
C       THIS IS AN OPTIONAL OVERRIDE OTHERWISE.                                 STPC0680
C      THE UNIT SYSTEM CAN BE BRITISH (MILES, FT/SEC),                          STPC0690
C       METRIC (KM, METERS/SEC), KMETRIC (KM, KM/SEC), OR BODY (STARTING        STPC0700
C       AT COLUMN 13 CARD B)                                                    STPC0710
C      WHEN BODY UNITS ARE USED, THE ACCELERATION OF GRAVITY AT EARTH'S         STPC0720
C       SURFACE IS PUT = 1.00.  THUS, XISP (ITEM 2, CARD 12) MUST BE            STPC0730
C       FUEL EXHAUST VELOCITY IN BODY SYSTEM UNITS.                             STPC0740
C      TIME IS IN SECONDS FOR ALL SYSTEMS EXCEPT BODY SYSTEM UNITS.             STPC0750
```

Figure 57. Subroutine STPCØN Listing (Continued)

```
C     READ A TITLE AND UNIT DEFINITION CARD.                                  STPC0760
      READ (5,5) TITLE, BODY, UNSYS, GXM1                                     STPC0770
    5 FORMAT (18A4/ 6A4, D12.6)                                               STPC0780
      JJ = 2                                                                  STPC0790
      IF ( UNSYS(1) .EQ. TITEST( 1) ) JJ = 1                                  STPC0800
      IF ( UNSYS(1) .EQ. TITEST( 2) ) JJ = 2                                  STPC0810
      IF ( UNSYS(1) .EQ. TITEST( 3) ) JJ = 3                                  STPC0820
      IF ( UNSYS(1) .EQ. TITEST( 4) ) JJ = 4                                  STPC0830
C     JJ = 2 IS CHOSEN IF NO MATCH IS FOUND  ( CONV = 1000. )                 STPC0840
      DO 8  I = 1,6                                                           STPC0850
      II = I                                                                  STPC0860
      IF ( BODY(1) .EQ. TITEST(I+4) ) GO TO 16                                STPC0870
    8 CONTINUE                                                                STPC0880
C     IF NO MATCH IN DO 8 LOOP PROGRAM ASSUMES BODY WITH GXM = GXM1.          STPC0890
      GXM = GXM1                                                              STPC0900
      JJ = 7                                                                  STPC0910
      IF ( GXM1 .NE. 0.00 ) GO TO 18                                          STPC0920
      WRITE (6,9)                                                             STPC0930
    9 FORMAT ('0PROGRAM UNABLE TO DETERMINE ATTRACTING CENTER, EXIT CALL      STPC0940
     *ED. ')                                                                  STPC0950
      WRITE (6,7) TITLE, BODY, UNSYS, GXM                                     STPC0960
      CALL EXIT                                                               STPC0970
   16 JG = JJ                                                                 STPC0980
      IF ( JJ .EQ. 3 ) JG = 2                                                 STPC0990
      GXM = G(II,JG)                                                          STPC1000
      IF ( GXM1 .NE. 0.00 ) GXM = GXM1                                        STPC1010
      CONV = CONVR(JJ)                                                        STPC1020
      GO = GR(JJ)                                                             STPC1030
      XMU = GXM                                                               STPC1040
   18 WRITE (6,7) TITLE, BODY, UNSYS, GXM                                     STPC1050
    7 FORMAT ( '0CASE IDENTIFICATION' / 1H , 18A4 / '0BODY IS ', 3A4 /        STPC1060
     *     ' UNIT SYSTEM IS ', 3A4 /  ' GXM = ', D12.6 )                      STPC1070
      INIT = 0                                                                STPC1080
      NCARDS = 25                                                             STPC1090
      CALL INPUT(      MASK,INIT,NCARDS)                                      STPC1100
      CALL DATSET                                                             STPC1110
      INIT = 3                                                                STPC1120
  100 WRITE(6,10) DATE , TIME                                                 STPC1130
      CALL INPUT(      MASK,INIT,NCARDS)                                      STPC1140
```

Figure 57. Subroutine STPCØN Listing (Continued)

```
      WRITE(6,10) DATE , TIME                                                  STPC1150
    6 FORMAT (6E16.6)                                                          STPC1160
  105 CALL DEX( XORB1, XORB2,P1,E1,W1, XI1,P2,E2,W2,PER1,PER2 )                STPC1170
      XI1 = XI1 * RAD                                                          STPC1180
      W1 = W1 *RAD                                                             STPC1190
      W2 = W2 *RAD                                                             STPC1200
      WRITE(6,7002)                                                            STPC1210
 7002 FORMAT(T1,'0 TRANSFORMED ELEMENTS'/' ',T10,'P1',T30,'E1',T50,'W1',STPC1220
     *T67,'INCL')                                                              STPC1230
      WRITE (6,303)  P1, E1, W1, XI1, P2, E2, W2                               STPC1240
  303 FORMAT (1H 4D19.11/' ',T10,'P2',T30,'E2',T50,'W2'/1H 3D19.11)            STPC1250
      AEST4 = (P1 + P2) / 2.D0                                                 STPC1260
      XI1 = XI1 / RAD                                                          STPC1270
      W1 = W1 /RAD                                                             STPC1280
      W2 = W2 /RAD                                                             STPC1290
C     ------------------------------------------------------------------------STPC1300
      DO 3000   JCARD = 1,5                                                    STPC1310
      DO 33  I1 = 1,10                                                         STPC1320
   33 IRANK(I1) = 0                                                            STPC1330
      NPT = SPAR(1,JCARD)                                                      STPC1340
      IF ( NPT .EQ. 0 ) GO TO 3000                                             STPC1350
      WRITE (6,41)                                                             STPC1360
   41 FORMAT ('1GENERATE CONTOUR MAP')                                         STPC1370
      WRITE (6,46) (SPAR(I,JCARD),I=1,6)                                       STPC1380
   46 FORMAT (6E16.6)                                                          STPC1390
      CALL SURVEY ( Z, XORD, YORD, NPT, SPAR(2,JCARD), SPAR(3,JCARD),          STPC1400
     *     SPAR(4,JCARD), SPAR(5,JCARD), AEST4  , SPAR(6,JCARD), 1 )           STPC1410
      DO 2020  I = 1,10                                                        STPC1420
      IF ( IRANK(I) .EQ. 0 ) GO TO 2040                                        STPC1430
      NNXV = NXV ( SNGL (XMINS(I) ) )                                          STPC1440
      NNYV = NYV ( SNGL (YMINS(I) ) )                                          STPC1450
      IXSP = 24                                                                STPC1460
      DO 2020  J = 1,5                                                         STPC1470
      IXSP = IXSP - 8                                                          STPC1480
 2020 CALL PLOTV ( (NNXV+IXSP), NNYV, IRANK(I) )                               STPC1490
 2040 IF ( JCARD .GT. 1 ) GO TO 3000                                           STPC1500
```

Figure 57. Subroutine STPCØN Listing (Continued)

```
      DO 2900  IST = 1,10                                                    STPC1510
      IST1 = 11 - IST                                                        STPC1520
      IRKSV  =  0                                                            STPC1530
      DO 2050  IRK = 1,10                                                    STPC1540
 2050 IF ( IST1 .EQ. IRANK(IRK) ) IRKSV = IRK                                STPC1550
      IF  (  IRKSV  .EQ.  0  )  GO TO 2900                                   STPC1560
      IF ( IST1.GT. IMAPS ) GO TO 2100                                       STPC1570
      S2XY = SPAR(2,1) * SCXY                                                STPC1580
      S3XY = SPAR(3,1) * SCXY                                                STPC1590
      S6IMP = SPAR(6,1) * SCIMP                                              STPC1600
      NPT1  =  15                                                            STPC1610
      CALL SURVEY ( Z, XORD, YORD, NPT1,S2XY, S3XY, XMINS(IRKSV),            STPC1620
     *    YMINS(IRKSV), PMINS(IRKSV), S6IMP, 0 )                             STPC1630
      WRITE (16,2090) IRANK(  IRKSV  )                                       STPC1640
 2090 FORMAT (' RANK OF OPTIMUM =', I4 )                                     STPC1650
 2100 IF(  IST1  .GT.  NSTEEP  )    GO TO 2900                              STPC1660
      A(1) = XMINS(IRKSV)                                                    STPC1670
      A(2) = YMINS(IRKSV)                                                    STPC1680
      A(3) = PMINS(IRKSV)                                                    STPC1690
      CALL STEEP ( A  ,  IRKSV  )                                            STPC1700
      WRITE ( 6,2090) IRANK(  IRKSV  )                                       STPC1710
      WRITE (16,2090) IRANK(  IRKSV  )                                       STPC1720
 2900 CONTINUE                                                               STPC1730
 3000 CONTINUE                                                               STPC1740
      IF  (  IUSE  .EQ.  0  )  CALL STEEP(  AEST  ,  1  )                    STPC1750
C     ------------------------------------------------------------------     STPC1760
      XI1 = XI1 * RAD                                                        STPC1770
      W1 = W1 *RAD                                                           STPC1780
      W2 = W2 *RAD                                                           STPC1790
      CALL TIMEV( TIME1)                                                     STPC1800
      WRITE(6,301) TIME1                                                     STPC1810
  301 FORMAT('0ELAPSED EXECUTION TIME = ',F12.3 )                            STPC1820
      CALL COUNTV                                                            STPC1830
      RETURN                                                                 STPC1840
      END                                                                    STPC1850
```

Figure 57. Subroutine STPCØN Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
      SUBROUTINE DATSET                                                 DATS0000
C     ------------------------------------------------------------------DATSC010
C        THIS SUBROUTINE SUPPLIES REASONABLE VALUES FOR ALL PARAMETERS  DATS0020
C                     REQUIRED TO RUN A COMPLETE CASE.                  DATS0030
C     ------------------------------------------------------------------DATS0040
      IMPLICIT REAL*8(A-H , O-Z)                                        DATS0050
      LOGICAL RATES, REINT, CRTOUT                                      DATS0060
      REAL*4 RESET, TWOBRN, SCXY, SCIMP, SPAR,PAD                       DATS0070
      COMMON /INDATA/          XORB1(6),  XORB2(6),   AEST(6)           DATS0080
     A, AMPINC    , AMPDEC    , FREEZE    , XAMP      , SAVE      , BL46 DATS0090
     B, SCLINC    , SCLDEC    , ISCL      , IAMP      , BL55      , BL56 DATS0100
     C, SCALE(3)                          , TOL       , MAXIT     , IPRIN DATS0110
     D, XIN(14),  XFN(14),  RATES, IDIMEN                               DATS0120
     I, BETASV  ,  XTSP     ,  TFPERT ,  PERT     ,  COAST1   ,  BL126 DATS0130
     J, BRNINC  ,  BMAX1    ,  BMAX2  ,  BSH1     ,  BSH2     ,  TWOBRN DATS0140
     K, RESET   ,  KMULT    ,   NOPRT ,  PRNJ     ,  BL145    ,  BL146 DATS0150
     L, INTSTP  ,  REINT    ,  CRTOUT ,  ITSQL    , IQPRIN    , IQCYCL DATS0160
     M, HBARTL  ,  RHOS1 , BL163    ,  BL164  ,   BL165     ,  BL166    DATS0170
     N, XNB         ,  XNC  ,     DTB     ,   DTC      ,  JMAX   ,IBL176 DATS0180
     O , PAD(12)                                                        DATS0190
     Q , IMAPS    ,   NSTEEP  ,  IUSE,  SCXY  ,  SCIMP  ,  IPAD3       DATS0200
     R , SPAR(6,5)                                                      DATS0210
C     ------------------------------------------------------------------DATS0220
C     CARD 3                                                            DATS0230
C           AEST(3)    = 5000.D0                                        DATS0240
C     CARD 4                                                            DATS0250
C           AMPINC     = 2.D0                                           DATS0260
C           AMPDEC     = .500                                           DATS0270
C           FREEZE     = 1.D-08                                         DATS0280
C           XAMP       = 2.D0                                           DATS0290
C           SAVE       = 2.D0                                           DATS0300
C           BL46       = 1.D0                                           DATS0310
C     CARD 5                                                            DATS0320
C           SCLINC     = 0.D0                                           DATS0330
C           SCLDEC     = .8500                                          DATS0340
C           ISCL       = 1                                              DATS0350
C           IAMP       = 5                                              DATS0360
C           BL55       = 0.D0                                           DATS0370
C           BL56       = 0.D0                                           DATS0380
```

Figure 58. Subroutine DATSET Listing

```
C      CARD 6                                                                  DATS0390
                SCALE(1)   = 1.00                                              DATS0400
                SCALE(2)   = 1.00                                              DATS0410
                SCALE(3)   = 100.00                                            DATS0420
                TOL        = 1.0-05                                            DATS0430
                MAXIT      = 200                                               DATS0440
                IPRIN      = 5                                                 DATS0450
C      ------------------------------------------------------------------------DATS0460
C     CARDS 7-19 ARE REQUIRED BY THE FINITE THRUST OPTIMIZATION PROGRAM        DATS0470
C                                                                              DATS0480
C      CARD 7                                                                  DATS0490
                XIN(1)     = 0.00                                              DATS0500
                XIN(2)     = 0.00                                              DATS0510
                XIN(3)     = 0.00                                              DATS0520
                XIN(4)     = 0.00                                              DATS0530
                XIN(5)     = 0.00                                              DATS0540
                XIN(6)     = 0.00                                              DATS0550
C      CARD 8                                                                  DATS0560
                XIN(7)     = 1000.00                                           DATS0570
                XIN(8)     = 0.00                                              DATS0580
                XIN(9)     = 0.00                                              DATS0590
                XIN(10)    = 0.00                                              DATS0600
                XIN(11)    = 2.04                                              DATS0610
                XIN(12)    = 0.00                                              DATS0620
C      CARD 9                                                                  DATS0630
                XIN(13)    = 0.00                                              DATS0640
                XIN(14)    = 0.00                                              DATS0650
                XFN(1)     = 0.00                                              DATS0660
                XFN(2)     = 0.00                                              DATS0670
                XFN(3)     = 0.00                                              DATS0680
                XFN(4)     = 0.00                                              DATS0690
C      CARD 10                                                                 DATS0700
                XFN(5)     = 0.00                                              DATS0710
                XFN(6)     = 0.00                                              DATS0720
                XFN(7)     = 0.00                                              DATS0730
                XFN(8)     = 0.00                                              DATS0740
                XFN(9)     = 0.00                                              DATS0750
                XFN(10)    = 0.00
```

Figure 58. Subroutine DATSET Listing (Continued)

```
C       CARD 11                                                          DATS0760
                XFN(11)    = 0.D0                                        DATS0770
                XFN(12)    = 0.D0                                        DATS0780
                XFN(13)    = 0.D0                                        DATS0790
                XFN(14)    = 0.D0                                        DATS0800
                RATES      = .TRUE.                                      DATS0810
                IDIMEN     = 3                                           DATS0820
C       CARD 12                                                          DATS0830
                BETASV     = 1.D0                                        DATS0840
                XISP       = 400.D0                                      DATS0850
                TFPERT     = 0.D0                                        DATS0860
                PERT       = .1D0                                        DATS0870
                COAST1     = 3.D0                                        DATS0880
                BL126      = 0.000                                       DATS0890
C       CARD 13                                                          DATS0900
                BRNINC     = 0.D0                                        DATS0910
                BMAX1      = 1.D0                                        DATS0920
                BMAX2      = 1.D0                                        DATS0930
                BSH1       = 1.D0                                        DATS0940
                BSH2       = 1.D0                                        DATS0950
                TWOBRN     = 0.0                                         DATS0960
C       CARD 14                                                          DATS0970
                RESET      = 0.0                                         DATS0980
                KMULT      = 0                                           DATS0990
                NOPRT      = 0                                           DATS1000
                PRNJ       = 0.D0                                        DATS1010
                BL145      = 0.D0                                        DATS1020
                BL146      = 0.D0                                        DATS1030
C       CARD 15                                                          DATS1040
                INTSTP     = 1                                           DATS1050
                REINT      = .TRUE.                                      DATS1060
                CRTOUT     = .TRUE.                                      DATS1070
                ITSOL      = 4                                           DATS1080
                IQPRIN     = 1                                           DATS1090
                IOCYCL     = 3                                           DATS1100
```

Figure 58. Subroutine DATSET Listing (Continued)

```
C       CARD 16                                                                 DATS1110
                HBARTL      = 0.D0                                              DATS1120
                RHOS1       = .00100                                            DATS1130
                BL163       = 0.D0                                              DATS1140
                BL164       = 0.D0                                              DATS1150
                BL165       = 0.D0                                              DATS1160
                BL166       = 0.D0                                              DATS1170
C       CARD 17                                                                 DATS1180
                XNB         = 6.D0                                              DATS1190
                XNC         = 6.D0                                              DATS1200
                DTB         = .2D0                                              DATS1210
                DTC         = 2.D0                                              DATS1220
                JMAX        = 100                                               DATS1230
                IBL176      = 0                                                 DATS1240
C       ----------------------------------------------------------------------  DATS1250
C                                                                               DATS1260
C       CARD 20                                                                 DATS1270
                IMAPS       = 4                                                 DATS1280
                NSTEEP      = 4                                                 DATS1290
                IUSE        = 0                                                 DATS1300
                SCXY        = .2                                                DATS1310
                SCIMP       = .1                                                DATS1320
                IPAD3       = 0                                                 DATS1330
C       CARD 21                                                                 DATS1340
                SPAR(1,1) = 24.                                                 DATS1350
                SPAR(2,1) = 15.                                                 DATS1360
                SPAR(3,1) = 15.                                                 DATS1370
                SPAR(4,1) = 180.5                                               DATS1380
                SPAR(5,1) = 181.1                                               DATS1390
                SPAR(6,1) = 500.0                                               DATS1400
C       CARD 22                                                                 DATS1410
                SPAR(1,2) = 0.0                                                 DATS1420
        RETURN                                                                  DATS1430
        END
```

Figure 58. Subroutine DATSET Listing (Continued)

```
      SUBROUTINE XFIRST                                                 XFIR0000
C     ------------------------------------------------------------------XFIR0010
C     EXECUTIVE SUBROUTINE FOR HANDLING THE COMPUTATION OF THE INITIAL  XFIR0015
C     ITERATE ( X(T) ).                                                 XFIR0020
C     ------------------------------------------------------------------XFIR0030
      IMPLICIT REAL*8 ( A-H , O-Z)                                      XFIR0040
      REAL*4  XLST(15,100), RESET, TRU(100), TWOBRN                     XFIR0050
      LOGICAL  RATES ,          REINT , CRTOUT                          XFIR0060
        DIMENSION TP(3)                                                 XFIR0070
      DIMENSION FORM(7), FRM2(8)                                        XFIR0072
      DATA FORM/' EARTH   MOON    SUN     VENUS   MARS    JUPITER X BODYXFIR0074
     * '   /                                                            XFIR0076
      DATA FRM2 /'MILES, FT/SEC   KM, METERS/SEC  KM, KM/SEC      BODY UXFIR0078
     *NIT SYSTEM' /                                                     XFIR0079
      COMMON /ZIC/BETA,  C , GO ,XMU,TW,XXK(20)                         XFIR0080
      COMMON /INDATA/          XORB1(6),  XORB2(6),   AEST(6)           XFIR0090
     A, AMPINC   , AMPDEC   , FREEZE   , XAMP     , SAVF     , BL46     XFIR0100
     B, SCLINC   , SCLDEC   , ISCL     , IAMP     , BL55     , BL56     XFIR0110
     C, SCALE(3)                       , TOL      , MAXIT    , IPRIN    XFIR0120
     D, XIN(14),  XFN(14),  RATES, IDIMEN                               XFIR0130
     I, BETASV   ,  XISP     ,  TFPERT ,  PERT     ,  COAST1    ,  BL126   XFIR0140
     J, BRNINC   ,  BMAX1    ,  BMAX2  ,  BSH1     ,  BSH2      ,  TWOBRN  XFIR0150
     K, RESET    ,  KMULT    ,   NOPRT ,  BL144    ,  BL145     ,  BL146   XFIR0160
     L, INTSTP   ,  REINT    ,  CRTOUT ,  ITSOL    ,  IOPRIN    ,  IBL156  XFIR0170
     M, HBARTL   ,  PHOS1 , BL163    ,  BL164  ,   BL165     ,  BL166     XFIR0180
     N, XNB          ,  XNC  ,     DTB     ,   DTC      ,  JMAX    ,IBL176  XFIR0190
      COMMON /ZLGR/ ZL(2,6), THET(2), ALPH(2), FT(2), PA(2), AVH(3),    XFIR0200
     *       XTRA(10)                                                   XFIR0210
        COMMON/ZCONST/ PI,RAD,GXM,BETAS,CONV,II,JJ                      XFIR0220
      COMMON /ORBCON/ P1,E1,W1,XI1,P2, E2,W2,P,F,W,ETRN(6)              XFIR0230
      COMMON /STCOM/             Q1(3), Q2(3),XF(3),XU(3),XV(3),AUX(3), CXFIR0240
     1XI,DVT(5),R1E,R2E,SXI,TAM(3),UN1(3),UN2(3),VE1(3),VE2(3           XFIR0250
     2),XL1,XL2,XS1,XS2,PRT ,PER1,PER2,RAD1(3),RAD2(3),UMAG,XIM1(3,2),XIXFIR0260
     3M2(3,2),DTHETA,EPSI,ZETEST,                                       XFIR0270
     4JN , LOS , KONTAU                                                 XFIR0280
      COMMON /ZSUB/          TSUB(10),VARIND(100),XMNSTP(9),STPSIZ(9),  XFIR0290
     *KSUB(10) , MAXSTP                                                 XFIR0300
        COMMON /SOLN/ XLST, TRU, HBAR                                   XFIR0310
```

Figure 59. Subroutine XFIRST Listing

```
      WRITE (6,999)                                                     XFIR0315
  999 FORMAT ('1INITIAL CONDITIONS FOR QUASILINEARIZATION FROM IMPULSIVEXFIR0320
     * TRANSFER')                                                       XFIR0330
      DO 10  I = 1,100                                                  XFIR0340
      DO 10  J = 1,15                                                   XFIR0350
   10 XLST(J,I) =0.0                                                    XFIR0360
      XMNSTP(1) =       XNC                                           XFIR0370
      XMNSTP(2) =       XNB                                           XFIR0380
      XMNSTP(3) =       XNC                                           XFIR0390
      XMNSTP(4) =       XNB                                           XFIR0400
      XMNSTP(5) =       XNC                                           XFIR0410
      STPSIZ(1) =    DTC                                              XFIR0420
      STPSIZ(2) =    DTB                                              XFIR0430
      STPSIZ(3) =    DTC                                              XFIR0440
      STPSIZ(4) =    DTB                                              XFIR0450
      STPSIZ(5) =    DTC                                              XFIR0460
      MAXSTP = JMAX                                                     XFIR0470
      WRITE (6,50) (XMNSTP(I), I=1,5), (STPSIZ(I), I=1,5)               XFIR0480
   50 FORMAT ('0MIN. NO. OF STEPS     ', 1P5D16.2/                      XFIR0490
     *        ' MAX. STEP SIZE ALLOWED', 5D16.6)                       XFIR0500
      CALL HMASS ( C,COAST1,XIN(7),XFN(7),BETA,BRNINC,TPERT,TP,JMAX )   XFIR0510
      J = 2 * (JJ - 1 ) + 1                                             XFIR0515
      WRITE (6,504) FORM(II), FRM2(J), FRM2(J+1)                        XFIR0520
  504 FORMAT('0',22X,'*** FINAL TRANSFER CONDITIONS ***', 10X,'CENTER ISXFIR0525
     * ', A8, ', ', 'UNIT SYSTEM...', 2A8 )                            XFIR0526
      BETM = BETA / XIN(7)                                              XFIR0528
      WRITE (6,52)C,BETM,XISP,TW,BRNINC,TPERT,COAST1,LOS,KONTAU,MAXSTP XFIR0530
   52 FORMAT (                                                          '  XFIR0540
     *C =                              ',1P1D16.6,'          MASS RATE/INXFIR0550
     *ITIAL MASS =    ',1D16.6,' PER SEC  '/                           '  XFIR0560
     *FUEL SPECIFIC IMPULSE =          ',1D16.6,  ' SEC        THRUST TO WEXFIR0570
     *IGHT AT START =',1D16.6/                                          '  XFIR0580
     *BURN INCREMENT OVER IMPULSIVE =',1D16.6,  ' PERCENT   FINAL TIME PXFIR0590
     *ERTURBATION =  ',1D16.6,' SEC'/                                   '  XFIR0600
     *DURATION OF FIRST COAST ARC =  ',1D16.6,' SEC       LOS = ',I3/'  XFIR0610
     *KONTAU = ',I3,46X,'MAXSTP = ',I6)                                 XFIR0620
```

Figure 59. Subroutine XFIRST Listing (Continued)

```
      CALL COAST(P1,E1,W1,0.D0,XI1, TP(1), 1, 1 )                       XFIR0630
      CALL COAST(P, E, W,ETRN(4),ETRN(3), TP(3) , 3 ,1 )                XFIR0640
      CALL COAST(P2,E2,W2,0.D0,0.D0, TP(2), 5, 1 )                      XFIR0650
      XTRA(1) = XIN(11)                                                 XFIR0670
      CALL LGRANG                                                       XFIR0680
      DL4 = 200.                                                        XFIR0690
      CALL BURN( 2,3,TP)                                                XFIR0730
      CALL BURN( 4,5,TP )                                               XFIR0740
      WRITE (6,95)                                                      XFIR0750
   95 FORMAT (     'ONUMBERS AND VARIABLES ARE '                        XFIR0760
     */    '       1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14'XFIR0770
     */    ' XLST R, TH, PH, RH,  X,  Y,  M,  K, L1, L2, L4, L5, L6, L3'XFIR0780
     */    ' XKS2 R, PH, RH,  Y,  M, L1,  K, L4, L6, L3'                XFIR0790
     */    ' XKS3 R, PH, RH,  Y,  M, L1,  K, L4, L6, TH,  X, L2, L5, L3'XFIR0800
     *)                                                                 XFIR0810
      WRITE (6,555)                                                     XFIR0820
  555 FORMAT ('0', '     TIME             R             THETA          PHIXFIR0830
     *             RHO              X              Y              MASS  */ )XFIR0840
      DO 101 K=1,100                                                    XFIR0850
  101 WRITE (6,110) (VARIND(K),(XLST(L,K), L=1,7))                      XFIR0860
      WRITE (6,556)                                                     XFIR0870
  556 FORMAT ('0', '     TIME             K             LAM 1          LAM XFIR0880
     *2          LAM 4          LAM 5          LAM 6          LAM 3 */ )XFIR0890
      DO 102 K=1,100                                                    XFIR0900
  102 WRITE (6,110) (VARIND(K),(XLST(L,K), L=8,14))                     XFIR0910
  110 FORMAT (' ', F12.2, 1P7E15.6)                                     XFIR0920
      RETURN                                                            XFIR0930
      END                                                               XFIR0940
```

Figure 59. Subroutine XFIRST Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
      SUBROUTINE HMASS ( C,COAST,XIN7,XM2,BETA,BRNINC,TFPERT,TP,IMAX )  HMAS0010
C     ----------------------------------------------------------------HMAS0020
C     COMPUTES THE BURNING TIME HISTORIES FROM THE IMPULSIVE           HMAS0030
C     APPROXIMATION AND DETERMINES THE SIZES AND ORGANIZATION OF THE   HMAS0040
C     STORAGE TABLE FOR THE INITIAL ITERATE.  IT ALSO DETERMINES THE   HMAS0050
C     PERIGEE PASSAGE TIMES FOR ALL THREE ORBITS.                      HMAS0060
C     ----------------------------------------------------------------HMAS0070
        IMPLICIT REAL*8 (A-H,O-Z)                                      HMAS0080
        REAL*4 XLST(15,100), FE, FI , RF                               HMAS0090
         COMMON /SCLN/ XLST                                            HMAS0100
      DIMENSION NI(5) ,B(5), TP(3),XI(2) ,QX(3), NJ(5)                 HMAS0110
       COMMON/ZCONST/ PI,RAD,GXM,BETAS,CONV,II,JJ                      HMAS0120
      COMMON /ZSUB/          TSUB(10),VAR   (100),XNM   (9),DET   (9), HMAS0130
     *KSUB(10) , JMAX                                                  HMAS0140
C     DET(I) ARE MAXIMUM TIME STEP INTERVALS FOR UP TO 9 SECTIONS      HMAS0150
C      IN PERCENT OF THE AVERAGE PERIOD.                               HMAS0155
C     XNM(I) ARE MINIMUM NUMBER OF STEPS.  JMAX IS MAXIMUM ALLOWABLE   HMAS0160
C     TOTAL NUMBER OF STEPS                                            HMAS0170
C     COAST IS LENGTH OF COAST IN PERCENT OF A REVOLUTION--AVERAGE     HMAS0172
C     PERIOD IS USED.                                                  HMAS0174
      COMMON /ORBCON/ P1,E1,W1,XI1,P2, E2,W2,P,F,W,ETRN(6)             HMAC0180
      COMMON /STCOM/             Q1(3), Q2(3),XF(3),XU(3),XV(3),AUX(3), CHMAS0190
     1XI,DVT(5),R1E,R2E,SXI,TAM(3),UN1(3),UN2(3),VE1(3),VE2(3          HMAS0200
     2),XL1,XL2,XS1,XS2,PERT,PER1,PER2,RAD1(3),RAD2(3),UMAG,XIM1(3,2),XIHMAS0210
     3M2(3,2),DTHETA,EPSI,ZETEST,                                      HMAS0220
     4JN , LOS , KONTAU                                                HMAS0230
         DATA  B/ 5*0.D0 /, HNORD /100.D0/                             HMAS0240
         KSUB(7) = 0                                                   HMAS0270
C     FIRST BURNING TIMES FROM THE IMPULSES. C = EXHAUST VEL. B =-MDOT. HMAS0280
         XI(1) = XS1                                                   HMAS0290
         XI(2) = XS2                                                   HMAS0300
         LOS = LOS                                                     HMAS0310
         GO TO (23,22) , LOS                                           HMAS0320
   22    XI(1) = XL1                                                   HMAS0330
         XI(2) = XL2                                                   HMAS0340
   23    B(2) = BETA                                                   HMAS0350
      PERAV = ( PER1 + PER2 ) * .5D0                                   HMAS0355
         B(4) = BETA                                                   HMAS0360
```

Figure 60. Subroutine HMASS Listing

```
      XM1 = XIN7/DEXP (XI(1)/C/CONV     )                          HMAS0370
      XM2 = XM1/ DEXP (XI(2)/C/CONV   )                            HMAS0380
      DT1 = (XIN7 - XM1) / B(2)                                    HMAS0390
      DT2 = (XM1 - XM2)/ B(4)                                      HMAS0400
    TBF = ( HNDRD + BRINC ) / HNDRD                                HMAS0410
      BURN1 = DT1 * TBF                                            HMAS0420
      BURN2 = DT2 * TBF                                            HMAS0430
410 CALL DBLCRS(TAM,XE,QX)                                         HMAS0440
    THY=DBLUNT( QX, QX,3)                                          HMAS0450
    THY=DBLDOT(UN2, QX,3)                                          HMAS0460
    THX=DBLDOT(UN2,XE,3)                                           HMAS0470
    PH3=DATAN2(THY,THX)                                            HMAS0480
    DTHET1=PH3-DTHETA                                              HMAS0490
    CALL DORBIT(E,DTHET1,TP1)                                      HMAS0500
    CALL DORBIT(E,PH3,TP2)                                         HMAS0510
    TDT = TP2 - TP1 + 1.D0                                         HMAS0520
    TDT = TDT - DBLE(AINT(SNGL(TDT)))                              HMAS0530
    TDT = TDT * PERT                                               HMAS0540
 10 FORMAT  (6D16.8)                                               HMAS0570
    TSUB(1)=0.                                                     HMAS0580
    TSUB(2)=COAST*PERAV/HNDRD                                      HMAS0590
    TSUB(3)=TSUB(2)+BURN1                                          HMAS0600
    TSUB(4)=TSUB(2)+TDT+ (BURN1-BURN2)/2.D0                        HMAS0610
    TSUB(5)=TSUB(4)+BURN2                                          HMAS0620
    TSUB(6)=TSUB(5)+TSUB(2)                                        HMAS0630
      E1 =(AUX(1)-W1)/RAD                                          HMAS0640
      E2 = (AUX(2) - W2)/RAD                                       HMAS0650
    CALL DORBIT(E1,E1,DTP1)                                        HMAS0660
    CALL DORBIT(E2,E2,DTP2)                                        HMAS0680
    TX1=TSUB(2) + BURN1/2.D+00                                     HMAS0690
    TX2=TX1+TDT                                                    HMAS0700
    TP(1)=TX1-DTP1*PER1                                            HMAS0710
    TP(2)=TX2-DTP2*PER2                                            HMAS0720
    TP(3)=TX1-TP1 *PERT                                            HMAS0730
    DO 585  I=1,5                                                  HMAS0740
    NI(I) = IDINT( XNM(I) )                                        HMAS0750
    NJ(I) = NI(I)                                                  HMAS0755
    D=(TSUB(I+1)-TSUB(I))/(DET(I)*PERAV) * HNDRD                   HMAS0760
    IF(D.GT.XNM(I))   NI(I)=IDINT(D)                               HMAS0770
585 CONTINUE                                                       HMAS0780
```

Figure 60. Subroutine HMASS Listing (Continued)

```
C     FIND SUM OF NI(I)  (=JMX)   COMPARE WITH IMAX                          HMAS0790
  590 JMX=NI(1)                                                              HMAS0800
      DO 595  I=2,5                                                          HMAS0810
  595    JMX = JMX + NI(I)                                                   HMAS0820
      IF(JMX.LT.IMAX)   GO TO 605                                            HMAS0825
      KR= JMX-IMAX + 1                                                       HMAS0830
        WRITE(6,603) KR                                                      HMAS0835
  603  FORMAT(' PROGRAM CALLED FOR TOO MANY STEPS--EXCESS WAS', I4)          HMAS0840
      FE = FLOAT(KR) / FLOAT(IMAX)                                           HMAS0845
C     IF THE EXCESS IS LESS THAN .1, THE INTERVAL WITH THE LARGEST           HMAS0850
C      NUMBER OF STEPS IS REDUCED BY KR, OTHERWISE ALL ARE REDUCED IN        HMAS0855
C      PROPORTION.                                                           HMAS0860
      FET = FE - .1                                                          HMAS0865
      IF ( FET ) 550,550,570                                                 HMAS0870
  550 NB = NI(1)                                                             HMAS0875
      IB = 1                                                                 HMAS0880
      DO 560  I = 2,5                                                        HMAS0885
      IF ( NB .GT. NI(I) ) GO TO 560                                         HMAS0890
      IB = I                                                                 HMAS0895
      NB = NI(I)                                                             HMAS0900
  560 CONTINUE                                                               HMAS0905
      NI(IB) = NI(IB) - KR                                                   HMAS0910
      GO TO 610                                                              HMAS0915
  570 RF = 1.0 / (1. + FE )                                                  HMAS0920
      DO 580  I = 1,5                                                        HMAS0925
      NI(I) = INT( FLOAT( NI(I) ) * RF )                                     HMAS0930
      IF ( NI(I) .LT. NJ(I) ) NI(I) = NJ(I)                                  HMAS0935
  580 CONTINUE                                                               HMAS0940
      GO TO 590                                                              HMAS0945
  605 NI(3)=IMAX-NI(1)-NI(2)-NI(4)-NI(5) -1                                  HMAS1000
  610   IL = 1                                                               HMAS1010
        VAR(1) = 0.D0                                                        HMAS1020
      XLST(7,1)=XIN7                                                         HMAS1030
      KSUB(1) = 1                                                            HMAS1040
```

Figure 60. Subroutine HMASS Listing (Continued)

```
      DO 300 K=1,5                                                          HMAS1060
    KSUB(K+1) = KSUB(K) + NI(K)                                             HMAS1070
      I1 =NI(K)                                                             HMAS1080
      DI1 = DFLOAT(I1)                                                      HMAS1090
      DELT = (TSUB(K+1)- TSUB(K))/DI1                                       HMAS1100
      DO 250 I=1,I1                                                         HMAS1110
      VAR(IL+1) = VAR(IL) + DELT                                            HMAS1120
    XLST(7,IL+1) = XLST(7,IL) - DELT*B(K)                                   HMAS1130
250   IL = IL+1                                                             HMAS1140
300   CONTINUE                                                              HMAS1150
    WRITE (6,12) NI,          (KSUB(I),I=1,7),(TSUB(J),J=1,6)               HMAS1160
 12 FORMAT (' THE NUMBER OF STEPS IN THE FIVE INTERVALS IS',               HMAS1170
   *      5I9//' THUS KSUB = ', I10,6I16/'  AND TSUB = ',1P6D16.6)         HMAS1180
    WRITE (6,14) XIN7, XIN7, XM1, XM1, XM2, XM2                             HMAS1190
 14 FORMAT ('  AND MASS = ',1P6D16.6)                                       HMAS1200
    WRITE (6,24) TP                                                         HMAS1210
 24 FORMAT (' THE PERIGEE TIMES ARE (1, 2, TRANSFER)       '     ,1P3D16HMAS1220
   *.6,' SECONDS')                                                          HMAS1230
C                                                                           HMAS1240
900   RETURN                                                                HMAS1250
      END                                                                   HMAS1260
```

Figure 60. Subroutine HMASS Listing (Continued)

```
      SUBROUTINE COAST(PL,EE,WPR,WNR,XINCR,TPER,ICA,N)                  CST00100
C     ------------------------------------------------------------------CST00300
C     COMPUTES THE STATE VARIABLES ON THE THREE COASTING ARCS, IN       CST00400
C     SPHERICAL COORDINATES, FROM ORBITAL ELEMENTS AND TIME.            CST00500
C     ------------------------------------------------------------------CST00600
C     INPUT IN DEGREES, BUT LAT. AND LONG. ARE FOUND IN RADIANS         CST00700
C     MUST BE CALLED IN SEQUENCE -- FIRST COAST, TRANSFER, FINAL COAST  CST00800
C           TO KEEP PHI INCREASING                                      CST00900
C                                                                       CST01000
      IMPLICIT REAL*8 (A-H,O-Z)                                         CST01100
      REAL*8 U(3),E(3),NV(3),V(3),W(3),Q(3)                             CST01200
     REAL*4  XLST(15,100), TRU(100)                                     CST01300
     DIMENSION XSAV(6)                                                  CST01400
      COMMON/ZCONST/ PI,RAD,GXM                                         CST01500
       COMMON /SOLN/ XLST, TRU                                          CST01600
     COMMON /ZSUB/          TSUB(10),VAR    (100),XMNSTP(9),STPSIZ(9),  CST01700
    *KSUB(10) , MAXSTP                                                  CST01800
     COMMON /INDATA/          XORB1(6),  XORB2(6),   AEST(6)           CST01900
    A, AMPINC    , AMPDEC    , FREEZE    , XAMP      , SAVE      , BL46  CST02000
    B, SCLINC    , SCLDEC    , ISCL      , IAMP      , BL55      , BL56  CST02100
    C, SCALE(3)                          , TOL       , MAXIT     , IPRIN CST02200
C     COMMON FOR QASLIN ORB. TRAN. PROG.--( TWO DIM.CASE)               CST02300
    D, XIN(14),  XFN(14)                                                CST02400
C     ICA=ARC NUMBER                                                    CS;02500
C     N(POS)= NO. OF SUCCESSIVE ARCS OVER WHICH                         CST02600
C     TO COMPUTE COAST TRAJECTORY.                                      CST02700
C     N(NEG)=CALLED BY BURN TO COMPUTE ONE POINT FOR I=KSUB(ICA)-N      CST02800
C     UNIT VECTOR SETS--UI,UJ,UK REFERENCE -- U,V,W LOCAL(ORBIT  PLANE) CST02900
C           -- U,E,NV LOCAL(UP,EAST,NORTH)-- P,Q,W  PERIGEE(ORB. PLANE)CST03000
C     FIRST GET CONSTANT VECTORS W AND Q                                CST03100
      IF ( ICA .EQ. 1 ) PHISAV = 0.D0                                   CST03400
      NFLAG=1                                                           CST03500
      IF(N.LE.0)  NFLAG=2                                               CST03600
       WN = WNR/RAD                                                     CST03700
       WP = WPR/RAD                                                     CST03800
       XINC = XINCR/RAD                                                 CST03900
        SN = DSIN(WN)                                                   CST04000
        CN = DCOS(WN)                                                   CST04100
```

Figure 61. Subroutine CØAST Listing

```
      SI = DSIN(XINC)                                                    CST04200
      CI = DCOS(XINC)                                                    CST04300
      SP = DSIN(WP)                                                      CST04400
      CP = DCOS(WP)                                                      CST04500
      W(1) = SI*SN                                                       CST04600
      W(2) =-SI*CN                                                       CST04700
      W(3) = CI                                                          CST04800
      Q(1) = -CN*SP-CI*SN*CP                                             CST04900
      Q(2) = -SN*SP+CI*CN*CP                                             CST05000
      Q(3) =  SI*CP                                                      CST05100
     GO TO (20,25) , NFLAG                                               CST05200
   20 I1=KSUB(ICA)                                                       CST05300
      I2 = KSUB(ICA+N)                                                   CST05400
     GO TO 30                                                            CST05500
   25 I1 = KSUB(ICA)-N                                                   CST05600
     I2=I1                                                               CST05700
   30   A = PL/(1.D0 -EE**2)                                            CST05800
      XN = DSQRT(GXM/ A**3 )                                             CST05900
      F(3) = 0.D0                                                        CST06000
      VM  = DSQRT(GXM/PL)                                                CST06100
C                                                                        CST06200
C     NOW DO LOOP FOR COORDINATES AND VELOCITIES                         CST06300
      DO 200 I=I1,I2                                                     CST06400
      XM = XN*(VAR(I)- TPER)                                             CST06500
      F = DTRUEH(XM,EE,EA,SSF,CSF,IFL,JFL)                               CST06600
      SB = DSIN(F+WP)                                                    CST06700
      CB = DCOS(F+WP)                                                    CST06800
      U(1) = -CI*SN*SB+CN*CB                                             CST06900
      U(2) =  CI*CN*SB+SN*CB                                             CST07000
      U(3) = SI*SB                                                       CST07100
      STH  = U(3)                                                        CST07200
      CTH  = DSQRT(1.D0-STH**2)                                          CST07300
    XSAV(1) = PL / (1.D0+EE*CSF)                                         CST07400
    R = XSAV(1)                                                          CST07500
    XSAV(2) = DATAN2(STH,CTH)                                            CST07600
    XLST(2,I) = XSAV(2)                                                  CST07700
C     THETA MUST LIE BETWEEN +-INCL.   SO CTH IS POSITIVE.               CST07800
```

Figure 61. Subroutine CØAST Listing (Continued)

```
      XSAV(3) = DATAN2( U(2),U(1) )                                      CST07900
      PHI = XSAV(3)                                                      CST08000
      IF ( I .EQ. 1 ) GO TO 65                                           CST08100
      IF( PHI  .LT.  PHISAV  )  PHI  =  PHI  +  2.DO * PI                CST08200
      IF( PHI  .LT.  PHISAV  )  PHI  =  PHI  +  2.DO * PI                CST08300
   65   XLST(3,I)=  PHI                                                  CST08400
C     PHI=XLST(3,I) IS DETERMINED TO PROPER QUADRANT                     CST08500
      PHISAV = PHI                                                       CST08600
      XSAV(3) = PHI                                                      CST08700
C     XLST(3,I) MUST ALWAYS INCREASE                                     CST08800
   70   CALL DBLCRS(W,U,V)                                              CST08900
        SPH  =  DSIN( PHI )                                             CST09000
        CPH  =  DCOS( PHI )                                             CST09100
        E(1) = -SPH                                                     CST09200
        E(2) =  CPH                                                     CST09300
        NV(1) = -STH*CPH                                                CST09400
        NV(2) = -STH*SPH                                                CST09500
        NV(3) =  CTH                                                    CST09600
      XSAV(3)  =  XLST(3,I)                                             CST09700
      XSAV(4) = VM * EE * SSF                                           CST09800
      XLST(4,I) = XSAV(4)                                               CST09900
      XSAV(5) = VM*( DBLDOT(V,NV,3) + EE*DBLDOT(Q,NV,3))/R              CST10000
      XLST(5,I) = XSAV(5)                                               CST10100
      XSAV(6) = VM*(DBLDOT(V,E,3) + EE*DBLDOT(Q,E,3) )/(R*CTH)          CST10200
      XLST(6,I) = XSAV(6)                                               CST10300
       XLST(1,I) = R                                                    CST10400
      TRU(I) = E                                                        CST10500
      PHISAV  = XLST(3,I)                                               CST10600
      IF ( I .NE. 1 ) GO TO 200                                         CST10700
      DO 133  J=1,6                                                     CST10800
  133 XIN(J) = XSAV(J)                                                  CST10900
  200 CONTINUE                                                          CST11000
      IF ( ICA .NE. 5 ) GO TO 900                                       CST11100
      DO 333 J=1,6                                                      CST11200
  333 XEN(J) = XSAV(J)                                                  CST11300
C                                                                       CST11400
  900   RETURN                                                          CST11500
        END                                                             CST11600
```

Figure 61. Subroutine CØAST Listing (Continued)

```
      SUBROUTINE LGRANG                                                  LAGR0000
C     ------------------------------------------------------------------LAGR0010
C     PERFORMS THE COMPUTATIONS FOR ESTIMATING THE VARIABLES OF THE      LAGR0011
C     LAGRANGE MULTIPLIERS PRIOR TO INITIATING QUASILINEARIZATION.       LAGR0012
C     IT ALSO COMPUTES THE PRIMER VECTOR WHICH IS USED TO COMPUTE THE    LAGR0013
C     INITIAL TIME HISTORIES OF THE SWITCHING FUNCTION.                  LAGR0014
C     ------------------------------------------------------------------LAGR0015
      IMPLICIT REAL*8 (A-H, O-Z)                                         LAGR0020
      DIMENSION IAX(10), ANG(10), XI1(3), XI2(3), XJ1(3), XJ2(3), DC1(3)LAGR0030
     *, DC2(3), ALP(2), FG(2), YL(2,6), CSA(2), SNA(2), CSF(2), SNF(2), LAGR0040
     *CSTH(2), SNTH(2), TNTH(2), AL(2,6), SL(2), EC(2), AGM(2),ZL2(2),  LAGR0050
     * ZL3(2),A(6),ELM(6), YY(2),IN(300),DBL(150), QX(3),ZIT(3), C1K(2) LAGR0060
          REAL*4 TRU(100), XLST(15,100), DATA(300)                       LAGR0070
          COMMON /ZCONST/ PI, RAD, GXM, BETAS, CONV, II, JJ              LAGR0080
          COMMON /SOLN/ XLST, TRU                                        LAGR0090
          COMMON /ZLGR/ ZL(2,6),THET(2),ALPH(2),FT(2),RA(2),AVH(3),      LAGR0100
     *                  XTRA(10), ALP                                    LAGR0110
          COMMON /ZSUB/ TSUB(10), VAR(100), XNM(9), DET(9),KSUB(10),JMAX LAGR0120
          COMMON /ORBCON/ P1,E1,W1,AI1,P2,E2,W2,P,E,W,FTRN(6)            LAGR0130
C           FTRN(6) ARE P,E,AINC,ANODE,W,A                               LAGR0140
          COMMON /STCOM/ Q1(3), Q2(3), XE(3), XU(3), XV(3),AUX(3), CXI,  LAGR0150
     1     DVT(5), R1E, R2E, SXI, TAM(3), UN1(3), UN2(3), VE1(3),        LAGR0160
     2     VE2(3), XL1, XL2, XS1, XS2, PERT, PER1, PER2, RAD1(3),        LAGR0170
     3     RAD2(3), UMAG, XIM1(3,2), XIM2(3,2), DTHETA, EPSI, ZETEST,    LAGR0180
     4     JN, LOS, KONTAU                                               LAGR0190
          COMMON /INDATA/ DATA, MPOS(6,25)                               LAGR0200
          EQUIVALENCE (DATA,IN,DBL)                                      LAGR0210
C                                                                        LAGR0220
      DO 50  I = 1,3                                                     LAGR0230
      IAX(I) = 3                                                         LAGR0240
      IAX(2) = 1                                                         LAGR0250
      XI1(I) = XIM1(I,LOS)                                               LAGR0260
   50 XI2(I) = XIM2(I,LOS)                                               LAGR0270
      XTRA(2) = DSQRT( XI1(1)**2 + XI1(2)**2 + XI1(3)**2) / CONV         LAGR0280
      XTRA(3) = DSQRT( XI2(1)**2 + XI2(2)**2 + XI2(3)**2) / CONV         LAGR0290
      CALL DBLCRS (TAM, XE, QX)                                          LAGR0300
      THY = DBLUNT(QX, QX, 3)                                            LAGR0310
      THY = DBLDOT(UN2, QX, 3)                                           LAGR0320
      THX = DBLDOT(UN2, XE, 3)                                           LAGR0330
```

Figure 62. Subroutine LGRANG Listing

```
      PH3 = DATAN2(THY, THX)*RAD                                          LAGR0340
C     NOTE DTHETA IS IN RADIANS                                           LAGR0350
      F1 = PH3 - DTHETA * RAD                                             LAGR0360
      ANG(1) = - ETRN(4)                                                  LAGR0370
      ANG(2) = - ETRN(3)                                                  LAGR0380
      N = 3                                                               LAGR0390
      DO 85  J = 4,10                                                     LAGR0392
      ANG(J) = 0.D0                                                       LAGR0393
   85 IAX(J) = 1                                                          LAGR0394
      ANG(3) = -W - F1                                                    LAGR0400
C     THE SUBROUTINE ROTATD REQUIRES ANGLES IN DEGREES                    LAGR0410
      CALL ROTATD (XJ1, XI1, N, IAX, ANG)                                 LAGR0420
      ANG(3) = -W -PH3                                                    LAGR0430
      CALL ROTATD (XJ2, XI2, N, IAX, ANG)                                 LAGR0440
      AM = DSQRT(GXM*P)                                                   LAGR0470
      F1 = F1/RAD                                                         LAGR0480
      F2 = PH3/RAD                                                        LAGR0490
      SF1 = DSIN(F1)                                                      LAGR0500
      SF2 = DSIN(F2)                                                      LAGR0510
      CF1 = DCOS(F1)                                                      LAGR0520
      CF2 = DCOS(F2)                                                      LAGR0530
      SFD = DSIN(DTHETA)                                                  LAGR0540
      CFD = DCOS(DTHETA)                                                  LAGR0550
      R1 = P/(1. + E*CF1)                                                 LAGR0560
      R2 = P/(1. + E*CF2)                                                 LAGR0570
        IF(DABS(XJ2(1)).LT.1.D-02) XJ2(1) = 1.D-04 * XJ2(2)               LAGR0580
      S = R2*XJ2(2)/XJ2(1)                                                LAGR0590
      R = R2*XJ2(3)/XJ2(1)                                                LAGR0600
      R1PS = (R1/P)**2                                                    LAGR0610
      R2PS = (R2/P)**2                                                    LAGR0620
      R1A = R1*(1. + R1/P)                                                LAGR0630
      R2A = R2*(1. + R2/P)                                                LAGR0640
      IF ( AI1 .NE. 0.D0 ) GO TO 110                                      LAGR0650
      IF ( E1 .NE. 0.D0  .AND.  E2 .NE. 0.D0 ) GO TO 110                  LAGR0660
C     FOR COPLANAR CASE WITH ONE (OR BOTH) OF THE ORBITS CIRCULAR C1 = 0LAGR0670
C      IS REQUIRED FOR TRANSFER ELLIPSE                                   LAGR0680
      C1 = 0.D0                                                           LAGR0690
      A2 = 0.D0                                                           LAGR0700
      B2 = 0.D0                                                           LAGR0710
```

Figure 62. Subroutine LGRANG Listing (Continued)

```
C      IF BOTH IMPULSES FORWARD, BACKWARD  A6 MUST BE POSITIVE,NEGATIVE   LAGR0720
C       FOR THIS CASE WE USE XTRA(1) = A6                                  LAGR0730
       A6 = XTRA(1)                                                       LAGR0740
       IF ( XJ1(2) .LT. 0.D0 ) A6 = -A6                                   LAGR0750
       CTNU = XJ1(1) / XJ1(2)                                             LAGR0760
C        WE BELIEVE A4 MUST HAVE THE SAME SIGN AS A6 AND MUST BE          LAGR0770
C       SMALL ENOUGH TO KEEP STEERING ANGLE NEAR 90 DEG.                   LAGR0780
       A4 = E * A6 / 6000.D0                                              LAGR0790
         XL6 = A6*R1PS + R1A*A4*CSF1                                      LAGR0800
         XL5 = 0.D0                                                       LAGR0810
         XL4 = CTNU*XL6/R1                                                LAGR0820
C      BEFORE PROCEEDING WE CHECK ON SIGN OF L6 AT SECOND BURN.  FOR      LAGR0830
C         NEARLY CIRCULAR COPLANAR ORBITS, IT CAN GET THE WRONG SIGN.     LAGR0840
  180  XL62 = A6 * R2PS + R2A * (A4*CF2 + C1*SF2)                         LAGR0850
       TEST6 = XJ2(2) / XL62                                              LAGR0860
       IF ( TEST6 .GT. 0.D0 ) GO TO 270                                   LAGR0870
         A4 = A4/10.D0                                                    LAGR0880
         WRITE(6,184)                                                     LAGR0890
  184    FORMAT('  A4 DIVIDED BY 10 TO CORRECT SIGN OF SL62 ')            LAGR0900
       GO TO 270                                                          LAGR0910
  110  BRD = -S*R1PS*SFD + R1PS*R2A*CFD - R2PS*R1A                        LAGR0920
        D2 = R2**2 * SFD/AM                                               LAGR0930
       DTR = D2*BRD                                                       LAGR0940
       D3 = R2PS*R*R1A*CFD - R1PS*R*R2A                                   LAGR0950
         RRS = (R2/R1)**2                                                 LAGR0960
       XL4 = XTRA(1)                                                      LAGR0970
         IF(DABS(XJ1(1)).LT.1.D-02) XJ1(1) = 1.D-04 * XJ1(2)              LAGR0980
       IF ( XJ1(1) .LT. 0.D0 ) XL4 = -XL4                                 LAGR0990
       XL5 = XL4*R1*XJ1(3)/XJ1(1)                                         LAGR1000
       XL6 = XL4*R1*XJ1(2)/XJ1(1)                                         LAGR1010
  220  A2 = (XL4*SF1*D3 + XL6*SF1*R*R2PS*SFD + XL5* RRS*SF2*BRD)/DTR      LAGR1020
  230  A4 = (XL4*(R1PS*(S*CF2+R2A*SF2)-R2PS*R1A*SF1) -XL6*R2PS*CF1)/BRD   LAGR1030
       B2 = (XL4*CF1*D3 + XL6*CF1*R*R2PS*SFD + XL5* RRS*CF2*BRD)/DTR      LAGR1040
       A6 = (-XL4*R1A*(S*CFD+R2A*SFD) + XL6*(-S*SFD+R2A*CFD))/BRD         LAGR1050
       C1 = (XL4*(R1PS*(S*SF2-R2A*CF2)+ R2PS*R1A*CF1)-XL6*R2PS*SF1)/BRD   LAGR1060
```

Figure 62. Subroutine LGRANG Listing (Continued)

```
C      SET PROPERTIES OF IMPULSE POINTS FOR COMMON /ZLGR/ AND LCOAST,      LAGR1100
C         AND FOR COMPUTING CONSTANTS ON GIVEN ORBITS.                     LAGR1110
C         ALPH     HEADINGS ON TRANSFER AT IMPULSES                        LAGR1120
C         FT       TRUE ANOMALIES ON TRANSFER AT IMPULSES                  LAGR1130
C         FG       TRUE ANOMALIES ON INITIAL,FINAL AT IMPULSES             LAGR1140
C         ALP      HEADINGS ON INITIAL,FINAL ORBITS AT IMPULSES            LAGR1150
C                                                                          LAGR1160
  270 TL4 = A4*SF1 - C1*CF1                                                LAGR1170
      TL5 = R1**2*(A2*CF1 - B2*SF1) / AM                                   LAGR1180
      TL6 = A6*R1PS + R1A*(A4*CF1 + C1*SF1)                                LAGR1190
      XL42 = A4*SF2 - C1*CF2                                               LAGR1200
      XL52 = R2**2 * (A2*CF2 - B2*SF2) / AM                                LAGR1210
      XL62 = A6*R2PS + R2A*(A4*CF2 + C1*SF2)                               LAGR1220
      TR = XL52 / XL42                                                     LAGR1230
      TS = XL62 / XL42                                                     LAGR1240
  300 RA(1) = R1E                                                          LAGR1310
      RA(2) = R2E                                                          LAGR1320
      THET(2) = 0.D0                                                       LAGR1330
      FT(1) = F1                                                           LAGR1340
      FT(2) = F2                                                           LAGR1350
      FG(1) = (AUX(1) - W1)/RAD                                            LAGR1360
      FG(2) = (AUX(2) - W2)/RAD                                            LAGR1370
      SI = DSIN(AI1/RAD)                                                   LAGR1380
      CI = DCOS(AI1/RAD)                                                   LAGR1390
      SU = DSIN(AUX(1)/RAD)                                                LAGR1400
      CU = DCOS(AUX(1)/RAD)                                                LAGR1410
      STH = SI*SU                                                          LAGR1420
      CTH = DSQRT(1.D0 - STH**2)                                           LAGR1430
      THET(1) = DATAN2(STH, CTH)                                           LAGR1440
      SA1 = SI*CU/CTH                                                      LAGR1450
      CA1 = CI/CTH                                                         LAGR1460
      ALP(1) = DATAN2(SA1,CA1)                                             LAGR1470
      ALP(2) = 0.D0                                                        LAGR1480
      SIT = DSIN(ETRN(3)/RAD)                                              LAGR1490
      CIT = DCOS(ETRN(3)/RAD)                                              LAGR1500
```

Figure 62. Subroutine LGRANG Listing (Continued)

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

```
      SA = SIT*DCOS(F1 + W/RAD)/CTH                                          LAGR1510
      CA = CIT/CTH                                                           LAGR1520
      ALPH(1) = DATAN2(SA, CA)                                               LAGR1530
      ALPH(2) = -ETRN(3)/RAD                                                 LAGR1540
  420  CALL DBLCRS( TAM, UN2,ZIT )                                           LAGR1550
       IF(ZIT(3).GT.0.D0)   ALPH(2) = -ALPH(2)                               LAGR1560
C     DESCENDING NODE ASSUMED.  CHANGE SIGN IF ASCENDING.                    LAGR1570
  425 SL(1) = P1                                                             LAGR1580
      SL(2) = P2                                                             LAGR1590
      EC(1) = F1                                                             LAGR1600
      EC(2) = F2                                                             LAGR1610
C                                                                            LAGR1620
      A(1) = A2                                                              LAGR1630
      A(2) = B2                                                              LAGR1640
      A(3) = A4                                                              LAGR1650
      A(4) = A6                                                              LAGR1660
      A(5) = C1                                                              LAGR1670
      A(6) = 0.                                                              LAGR1680
      ELM(1) = P                                                             LAGR1690
      ELM(2) = E                                                             LAGR1700
      ELM(3) = ETRN(3)/RAD                                                   LAGR1710
      ELM(4) = ETRN(4)/RAD                                                   LAGR1720
      ELM(5) = W                                                             LAGR1730
      WRITE (6,430) XJ1, RA(1), F1, ALPH(1), XJ2, RA(2), PH3, ALPH(2),       LAGR1740
     * (A(I),I=1,5)                                                          LAGR1741
  430 FORMAT (                    '1THE PRIMER COMPUTATION' / '0 IMPULSE DLAGR1742
     *ATA ON TRANSFER ORBIT' / 25X, 'IMPULSE COMPONENTS', 19X, 'RADIUS',LAGR1743
     *  7X, 'TRUE ANOMALY',  6X, 'HEADING' / 14X, 'RADIAL', 10X, 'CIRCUMLAGR1744
     *.',  8X, 'BINORMAL' / '  FIRST  ', 1P6D16.7 /'  SECOND ', 6D16.7 /LAGR1745
     *'0',           16X, 'A2',14X, 'B2', 14X, 'A4', 14X, 'A6', 14X, 'C1'LAGR1746
     * / 9X, 5D16.7 )                                                      LAGR1747
       NK = 3                                                              LAGR1748
  500 CALL LCOAST (A,ELM,NK)                                               LAGR1750
```

Figure 62. Subroutine LGRANG Listing (Continued)

```
C                                                                       LAGR1790
  700 DO 800  J=1,2                                                     LAGR1800
      AGM(J) = DSQRT(GXM*SL(J))                                         LAGR1810
      YY(J) = AGM(J)/RA(J)**2                                           LAGR1820
      YL(J,4) = ZL(J,4)                                                 LAGR1830
      CSA(J) = DCOS(ALP(J))                                             LAGR1840
      SNA(J) = DSIN(ALP(J))                                             LAGR1850
      CSF(J) = DCOS(FG(J))                                              LAGR1860
      SNF(J) = DSIN(FG(J))                                              LAGR1870
      CSTH(J) = DCOS(THET(J))                                           LAGR1880
      SNTH(J) = DSIN(THET(J))                                           LAGR1890
      TNTH(J) = SNTH(J)/CSTH(J)                                         LAGR1900
      YL(J,5) = ZL(J,5)*CSA(J) - ZL(J,6)*SNA(J)/CSTH(J)                 LAGR1910
      YL(J,6) = ZL(J,5)*SNA(J) + ZL(J,6)*CSA(J)/CSTH(J)                 LAGR1920
      YL(J,1) = ZL(J,1)                                                 LAGR1930
      YL(J,2) = ZL(J,2)*CSA(J)-ZL(J,3)/CSTH(J)*SNA(J)+   TNTH(J)*(YL(J,6LAGR1940
     *)*CSA(J)**3+YL(J,5)*SNA(J)**3)*YY(J)                              LAGR1950
  750 YL(J,3) = ZL(J,2)*SNA(J)+ZL(J,3)/CSTH(J)*CSA(J)+   TNTH(J)*CSA(J)*LAGR1960
     *(YL(J,6)*CSA(J)*SNA(J)-YL(J,5)*(1.+SNA(J)**2))*YY(J)              LAGR1970
  760 AL(J,1) = YL(J,2)*SNF(J) + YL(J,5)*AGM(J)*CSF(J)/RA(J)**2         LAGR2010
      AL(J,2) = YL(J,2)*CSF(J) - YL(J,5)*AGM(J)*SNF(J)/RA(J)**2         LAGR2020
      AL(J,3) = 2.*YL(J,6)*CSF(J)/RA(J) - YL(J,1)*CSF(J)*RA(J)**2/AGM(J)LAGR2030
     * + YL(J,4)*SNF(J)                                                 LAGR2040
      AL(J,4) = YL(J,1)*RA(J)*SL(J)**2 * (1. + RA(J)/SL(J))/AGM(J) - YL(LAGR2050
     *J,6)*(1. + 2.*RA(J)/SL(J))*(SL(J)/RA(J))**2                       LAGR2060
  770  C1K(J) = YL(J,3)*SL(J)/(AGM(J)*EC(J))                            LAGR2070
       AL(J,5) = - YL(J,4) * CSF(J)  +                                  LAGR2080
     *  2.DO *YL(J,6)*SNF(J)/RA(J) - YL(J,1)*RA(J)**2 * SNF(J)/AGM(J)    LAGR2090
  800 CONTINUE                                                          LAGR2130
      WRITE (6,740) (AL(1,I),I=1,5), C1K(1), (AL(2,I),I=1,5), C1K(2)    LAGR2100
  740 FORMAT ('0', 16X, 'A2',14X, 'B2', 14X, 'A4', 14X, 'A6', 14X, 'C1',LAGR2110
     *10X, 'C1(CHECK)', / ' INITIAL ', 1P6D16.7 / ' FINAL   ',6D16.7)   LAGR2120
  827 NK = 1                                                            LAGR2140
      DO 850  K=1,5                                                     LAGR2150
  850 A(K) = AL(NK,K)                                                   LAGR2160
```

Figure 62. Subroutine LGRANG Listing (Continued)

```
      ELM(1) = P1                                                    LAGR2170
      ELM(2) = E1                                                    LAGR2180
      ELM(3) = AI1/RAD                                               LAGR2190
      ELM(4) = 0.D0                                                  LAGR2200
      ELM(5) = W1/RAD                                                LAGR2210
      CALL LCOAST (A,ELM,NK)                                         LAGR2220
C                                                                    LAGR2230
      NK = 2                                                         LAGR2240
      DO 900  K=1,5                                                  LAGR2250
  900 A(K) = AL(NK,K)                                                LAGR2260
      ELM(1) = P2                                                    LAGR2270
      ELM(2) = E2                                                    LAGR2280
      ELM(3) = 0.D0                                                  LAGR2290
      ELM(5) = W2/RAD                                                LAGR2300
      CALL LCOAST (A,ELM,NK)                                         LAGR2310
      RETURN                                                         LAGR2320
      END                                                            LAGR2330
```

Figure 62. Subroutine LGRANG Listing (Continued)

```
      SUBROUTINE LCOAST (A, ELM, NK)                                    LCST0000
C     ------------------------------------------------------------------LCST0002
C     COMPUTES THE LAGRANGE MULTIPLIERS OVER THE COAST ARCS FROM        LCST0004
C     INFORMATION SUPPLIED BY SUBROUTINE LGRANG.                        LCST0006
C     ------------------------------------------------------------------LCST0008
C                                                                       LCST0020
C     ELM(6) ARE  P, E, INCL., NODE, PERI., A OR TIME OF PER(NOT USED)  LCST0030
C      A(6) ARE  A2, B2, A4, A6, C1, AND XX(NOT USED)                   LCST0040
       IMPLICIT REAL*8(A-H,O-Z)                                         LCST0050
       REAL*4 XLST(15,100),TRU(100),QRL(100),P0,C1, DP0, P1             LCST0060
        COMMON /ZCONST/ PI, RAD, GXM                                    LCST0070
        COMMON /SOLN/ XLST, TRU                                         LCST0080
        COMMON /ZLGR/ ZL(2,6),THET(2),ALPH(2),FT(2),RA(2),AVH(3),       LCST0090
     *                XTRA(10), ALP(2)                                  LCST0100
C     VARIABLES IN COMMON /ZLGR/ ARE COMPUTED FOR THE TRANSFER ORBIT    LCST0110
C      ZL ARE COMPUTED HERE BELOW FROM THET ETC SUPPLIED BY LGRANG       LCST0120
        COMMON /ZSUB/ TSUB(10), VAR(100), XNM(9), DET(9),KSUB(10),JMAX  LCST0130
       COMMON /ZIC/BETA,  C , GO ,XMU,TW,XXK(20)                        LCST0140
        DIMENSION A(6), ELM(6),XL(5), HML(100)  ,G(3)                   LCST0150
        DIMENSION PDOTV(3), PDOTG(3),PRL(100)                           LCST0160
       DIMENSION  DLAM1(2), DLAM2(2), PL(2)                             LCST0170
C                                                                       LCST0180
C      NK=1  INITIAL ORBIT TO FIRST BURN                                LCST0190
C      NK=2  FINAL ORBIT FROM  SECOND BURN                              LCST0200
C      NK=3  TRANSFER ORBIT MUST BE CALLED FIRST TO SETUP               LCST0210
C              CONSTANTS FOR INITIAL AND FINAL ORBITS                   LCST0220
C                                                                       LCST0230
       GO TO (10,20,30),NK                                              LCST0240
    10 IB1 = KSUB(2)                                                    LCST0270
       GO TO 100                                                        LCST0300
    20 IB2 = KSUB(5)                                                    LCST0310
       IEND = KSUB(6)                                                   LCST0320
       GO TO 100                                                        LCST0330
C                                                                       LCST0340
C      NK=3 BRANCH (TRANSFER ORBIT BRANCH)                              LCST0370
    30 IC1 = KSUB(3)                                                    LCST0380
       IC2 = KSUB(4)                                                    LCST0390
       SUM = 0.00                                                       LCST0400
   100 P = ELM(1)
```

Figure 63. Subroutine LCØAST Listing

```
      AM = DSQRT(GXM*P)                                                    LCST0410
      NNK = NK + 3*(NK-1)*(3-NK)                                           LCST0412
      I1 = KSUB(NNK)                                                       LCST0413
      I2 = KSUB(NNK+1)                                                     LCST0414
      DO 400  I=I1,I2                                                      LCST0420
      F = DBLE(TRU(I) )                                                    LCST0430
      X = DBLE(XLST(5,I) )                                                 LCST0440
      Y = DBLE(XLST(6,I) )                                                 LCST0450
      R = DBLE(XLST(1,I) )                                                 LCST0460
      RH= DBLE(XLST(4,I) )                                                 LCST0470
       CTH = DBLE(COS(XLST(2,I)))                                          LSCT0480
       STH = DBLE(SIN(XLST(2,I)))                                          LSCT0490
      TN = STH/CTH                                                         LCST0500
      CF = DCOS(F)                                                         LCST0510
      SF = DSIN(F)                                                         LCST0520
      YO = AM/R**2                                                         LCST0530
      SA = X/YO                                                            LCST0540
      CA = Y*CTH/YO                                                        LCST0550
      XL(1)=2.DO*A(4)*AM/(R*P**2)+YO*(1.DO+2.DO*R/P)*(A(3)*CF+A(5)*SF)     LCST0560
      XL(2) = A(1)*SF + A(2)*CF                                            LCST0570
      XL(3) = A(5)*AM*ELM(2)/P                                             LCST0580
      XL(4) = A(3)*SF - A(5)*CF                                            LCST0590
      XL(5) = (A(1)*CF - A(2)*SF)/YO                                       LCST0600
      XL(6) = A(4)*R**2/P**2 +R*(1.DO+ R/P)*(A(3)*CF + A(5)*SF)            LCST0610
      PRL(I) = DSQRT( XL(4)**2 + ( XL(5)**2 + XL(6)**2 )/R**2)             LCST0620
      XLST(13,I) =                                                         LCST0630
     *        SNGL((-XL(5)*SA + XL(6)*CA)*CTH)                             LCST0640
      XLST(9,I)  =                                                         LCST0650
     *        SNGL(XL(1))                                                  LCST0660
      XLST(10,I) =                                                         LCST0670
     *        SNGL(XL(3)*SA + XL(2)*CA - Y*TN*XLST(13,I))                  LCST0680
      XLST(14,I) =                                                         LCST0690
     *       SNGL((XL(3)*CA - XL(2)*SA)*CTH + YO*XL(5)*STH)                LCST0700
      XLST(11,I) =                                                         LCST0710
     *        SNGL(XL(4))                                                  LCST0720
      XLST(12,I) =                                                         LCST0730
     *        SNGL(XL(5)*CA + XL(6)*SA)                                    LCST0740
      ORL(I) = SNGL( DSQRT( XLST(11,I)**2 + ( XLST(12,I)**2                LCST0750
     *             + ( XLST(13,I)/CTH )**2) / R**2 ) )                     LCST0760
```

Figure 63. Subroutine LCØAST Listing (Continued)

```
      G(1) = R*(X**2 + (Y*CTH)**2) - GXM/R**2                                  LCST0850
      G(2)=2.DO *RH*X/R - STH*CTH*Y**2                                         LCST0860
      G(3)=2.DO*Y*(X*TN - RH/R)                                                LCST0870
      HML(I) = XLST(4,I)*XLST(9,I) + XLST(5,I)*XLST(10,I) + XLST(6,I)          LCST0880
     *    *XLST(14,I) + XLST(11,I)*G(1) + XLST(12,I)*G(2) + XLST(13,I)*G(LCST0890
     *    3)                                                                   LCST0900
      SUM = SUM + HML(I)                                                       LCST0910
  400 CONTINUE                                                                 LCST0920
      TNO = DFLOAT(I2-I1)                                                      LCST0930
      AVH(NK) = SUM/TNO                                                        LCST0940
      GO TO ( 999,700,420 ), NK                                                LCST0950
C     SECTION FOR MULTIPLIERS AT CORNERS COMPUTED                              LCST0960
C          FROM VALUES ON TRANSFER ARC.                                        LCST0970
  420 WRITE (6,430)                                                            LCST0972
  430 FORMAT ('OPRIMER VECTOR AT THE BURNS' / 15X, 'LENGTH',  7X, 'LENGTLCST0974
     *H RATE',  6X, 'DERIVATIVE' )                                            LCST0976
      DO 500  J=1,2                                                            LCST0980
      F = FT(J)                                                                LCST0990
      R = RA(J)                                                                LCST1000
      CTH = DCOS(THET(J))                                                      LCST1010
      STH = DSIN(THET(J))                                                      LCST1020
      TN = STH/CTH                                                             LCST1030
      CF = DCOS(F)                                                             LCST1040
      SF = DSIN(F)                                                             LCST1050
      RH = ELM(2) * SF * DSQRT(GXM/P)                                          LCST1060
      YO = AM/R**2                                                             LCST1070
         SA = DSIN(ALPH(J))                                                    LCST1080
         CA = DCOS(ALPH(J))                                                    LCST1090
      X = YO * SA                                                              LCST1100
        Y = YO * CA/CTH                                                        LCST1110
      XL(1)=2.DO*A(4)*AM/(R*P**2)+YO*(1.DO+2.DO*R/P)*(A(3)*CF+A(5)*SF)         LCST1120
      XL(2) = A(1)*SF + A(2)*CF                                                LCST1130
      XL(3) = A(5)*AM*ELM(2)/P                                                 LCST1140
      XL(4) = A(3)*SF - A(5)*CF                                                LCST1150
      XL(5) = (A(1)*CF - A(2)*SF)/YO                                           LCST1160
      XL(6) = A(4)*R**2/P**2 +R*(1.DO+ R/P)*(A(3)*CF + A(5)*SF)                LCST1170
```

Figure 63. Subroutine LCØAST Listing (Continued)

```
      ZL(J,6) =     (-XL(5)*SA + XL(6)*CA)*CTH                              LCST1180
      ZL(J,1) = XL(1)                                                       LCST1190
      ZL(J,2) =     XL(3)*SA + XL(2)*CA - Y*TN* ZL(J,6)                     LCST1200
      ZL(J,3) =    (XL(3)*CA - XL(2)*SA)*CTH + YO*XL(5)*STH                 LCST1210
      ZL(J,4) =     XL(4)                                                   LCST1220
      ZL(J,5) =     XL(5)*CA + XL(6)*SA                                     LCST1230
        DP1 = DSQRT(XL(4)**2 + (XL(5)**2+XL(6)**2)/R**2 )                   LCST1240
         DP2 = DSQRT( ZL(J,4)**2 +  (ZL(J,5)**2+(ZL(J,6)/CTH)**2)/R**2)     LCST1250
      PL(J) = DP2                                                           LCST1260
       PDOTV(1) = -XL(1) + XL(6)*YO/R                                       LCST1270
       PDOTV(3) = -XL(3)/R +RH*XL(6)/R**2 - YO*XL(4)                        LCST1280
       PDOTV(2) = -XL(2)/R + XL(5)*RH/R**2                                  LCST1290
         PDOT1 = DSQRT(PDOTV(1)**2 + PDOTV(2)**2 + PDOTV(3)**2)             LCST1300
        PVLD1 = (XL(4)*PDOTV(1) + (XL(5)*PDOTV(2)+XL(6)*PDOTV(3))/R)/DP1LCST1310
       PDOTG(1) = -ZL(J,1) + (ZL(J,5)*X + ZL(J,6)*Y)/R                      LCST1320
       PDOTG(3) = -ZL(J,3)/(R*CTH) - Y*ZL(J,4)*CTH + ZL(J,5)*Y*STH/R -      LCST1330
     *             ZL(J,6)*X*STH/(R*CTH**2) + ZL(J,6)*RH/(R**2*CTH)       LCST1340
       PDOTG(2) = -ZL(J,2)/R - X*ZL(J,4) +ZL(J,5)*RH/R**2-ZL(J,6)*Y*TN/RLCST1350
         PDOT2 = DSQRT(PDOTG(1)**2 + PDOTG(2)**2 + PDOTG(3)**2)             LCST1360
         PVLD2 = (ZL(J,4)*PDOTG(1) + (ZL(J,5)*PDOTG(2) + ZL(J,6)*PDOTG(3LCST1370
    *)/CTH)/R)/DP2                                                          LCST1380
      IF ( J .EQ. 1 ) WRITE (6,440) DP1, PVLD1, PDOT1                       LCST1390
440 FORMAT (' FIRST   ', 1P3D16.7 )                                         LCST1400
      IF ( J .EQ. 2 ) WRITE (6,450) DP2, PVLD2, PDOT2                       LCST1410
450 FORMAT (' SECOND  ', 1P3D16.7 )                                         LCST1420
      DLAM1(J) = (ZL(J,5)**2 + (ZL(J,6)/CTH)**2)*XTRA(J+1)/(R**3 * DP2)     LCST1450
      DL4M2(J) = (ZL(J,6)/(R*CTH))**2*TN*XTRA(J+1)/DP2                      LCST1460
      SGN =  DFLOAT( 2*J - 3 )                                              LCST1490
      ZL(J,1) = ZL(J,1) + SGN*DLAM1(J)                                      LCST1500
      ZL(J,2) = ZL(J,2) - SGN*DLAM2(J)                                      LCST1510
500   CONTINUE                                                              LCST1520
      WRITE (6,605) (ET(I), ALP(I), DLAM1(I), DLAM2(I), I=1,2)              LCST1522
605 FORMAT ('0PRIMER ON INITIAL AND FINAL ARCS' / 11X, 'TRUE ANOMALY',LCST1524
     *  6X, 'HEADING', 10X, 'DLAM1', 11X, 'DLAM2' / ' INITIAL ', 1P4D16.LCST1526
     *7 / ' FINAL   ', 4D16.7 )                                          LCST1528
      GO TO 999                                                         LCST1530
```

Figure 63. Subroutine LCØAST Listing (Continued)

```
  700 PO = SNGL(PL(1))                                                          LCST1540
       C1 = SNGL(C)                                                             LCST1545
      DO 710  I=1,IB1                                                           LCST1550
  710  XLST(8,I) = C1/ XLST(7,I) * (QPL(I) - PO)                                 LCST1560
      DPO = ( SNGL(PL(2))- PO ) / FLOAT( IC2-IC1)                               LCST1565
      DO 720  I=IC1,IC2                                                         LCST1570
      P1 = PO + DPO*FLOAT(I-IC1)                                                LCST1572
C      P1 AND DPO TO GET SMOOTH SWITCHING FUNCTION WHEN MAXIMUM PRIMER          LCST1574
C      VALUES ARE DIFFERENT AT THE TWO IMPULSE POINTS.                          LCST1576
  720  XLST(8,I) = C1/ XLST(7,I) * (QRL(I) - P1)                                LCST1580
      DO 730  I=IB2,IEND                                                        LCST1590
  730  XLST(8,I) = C1/ XLST(7,I) * (QRL(I) - P1)                                LCST1600
  999 RETURN                                                                    LCST1610
      END                                                                       LCST1620
```

Figure 63. Subroutine LCØAST Listing (Continued)

```
      SUBROUTINE BURN(N1,N2,TP)                                          BURN0000
C     ------------------------------------------------------------------BURN0005
C     CONSTRUCTS THE INITIAL TIME HISTORIES OF ALL VARIABLES OVER THE    BURN0010
C     SHORT COAST ARCS.  IT DOES SO BY FITTING A CUBIC OR LINEAR CURVE   BURN0015
C     BETWEEN END POINTS AT THE BEGINNING AND END OF EACH BURN.          BURN0020
C     THREE-DIM. VERSION--IBM 360--JULY 1967                             BURN0025
C     ------------------------------------------------------------------BURN0030
      IMPLICIT REAL*8 (A-H , O-Z )                                       BURN0040
      REAL*4 XLST(15,100)                                                BURN0050
      DIMENSION A(14), B(14), C(14),SL2(14),TP(3)                        BURN0060
      COMMON /SOLN/ XLST                                                 BURN0070
      COMMON /ZSUB/          TSUB(10),VARIND(100),XMNSTP(9),STPSIZ(9),   BURN0080
     *KSUB(10) , MAXSTP                                                  BURN0090
C      THIS VERSION FITS A CUBIC OVER BURN ARCS.  COAST ARC              BURN0100
C     FUNCTIONS ARE NEEDED                                               BURN0110
      M1 = KSUB(N1) +1                                                   BURN0120
      M2 = KSUB(N2) -1                                                   BURN0130
      I1=M1-1                                                            BURN0140
      I2=M2+1                                                            BURN0150
C     CAN BE USED ON ALL VARIABLES BY CHANGING THE 6.                    BURN0160
      X=VARIND(I2)-VARIND(I1)                                            BURN0170
      DO 100  K=1,3                                                      BURN0180
      A(K)=(XLST(K,I1)-XLST(K,I1-1))/(VARIND(I1)-VARIND(I1-1))           BURN0190
      Q=(XLST(K,I2)-XLST(K,I1))/X**2                                     BURN0200
      SL2(K)=(XLST(K,I2+1)-XLST(K,I2))/(VARIND(I2+1)-VARIND(I2))         BURN0210
C     A(K)=SL1(K)  SO SL1(K) IS NOT USED                                 BURN0220
      B(K)=3.*Q- (SL2(K)+2.*A(K))/X                                      BURN0230
      C(K)= -2.*Q/X + (SL2(K)+A(K))/X**2                                 BURN0240
      DO 60 I=M1,M2                                                      BURN0250
      XT=VARIND(I)-VARIND(I1)                                            BURN0260
      XLST(K,I)=XLST(K,I1) + ((C(K)*XT+B(K))*XT+A(K))*XT                 BURN0270
   60 CONTINUE                                                           BURN0280
  100 CONTINUE                                                           BURN0290
```

Figure 64. Subroutine BURN Listing

```
C       WE USE A LINEAR CURVE FOR VELOCITY VARIABLES  RHO, X, Y.          BURN0300
        DO 120  K=4,6                                                     BURN0310
         A(K) = (XLST(K,I2) - XLST(K,I1) )/X                              BURN0320
        DO 130  I=M1,M2                                                   BURN0330
         XT = VARIND(I) - VARIND(I1)                                      BURN0340
        XLST(K,I) = XLST(K,I1) + A(K)*XT                                  BURN0350
  130 CONTINUE                                                            BURN0360
  120 CONTINUE                                                            BURN0370
        DO 200  K=8,14                                                    BURN0380
        A(K)=(XLST(K,I1)-XLST(K,I1-1))/(VARIND(I1)-VARIND(I1-1))          BURN0390
        Q=(XLST(K,I2)-XLST(K,I1))/X**2                                    BURN0400
        SL2(K)=(XLST(K,I2+1)-XLST(K,I2))/(VARIND(I2+1)-VARIND(I2))        BURN0410
C       A(K)=SL1(K)  SO SL1(K) IS NOT USED                                BURN0420
        B(K)=3.*Q- (SL2(K)+2.*A(K))/X                                     BURN0430
        C(K)= -2.*Q/X + (SL2(K)+A(K))/X**2                                BURN0440
          DO 160 I=M1,M2                                                  BURN0450
        XT=VARIND(I)-VARIND(I1)                                           BURN0460
        XLST(K,I)=XLST(K,I1) + ((C(K)*XT+B(K))*XT+A(K))*XT                BURN0470
  160  CONTINUE                                                           BURN0480
  200  CONTINUE                                                           BURN0490
        RETURN                                                            BURN0500
        END                                                               BURN0510
```

Figure 64. Subroutine BURN Listing (Continued)

```
      SUBROUTINE QASLN ( XKS, P, H1, XJAC, T, CTP, NTM, N1, N2, NU,      QASL0000
     A                   IMAX, DXKP1, XKP1, ACUM, NS )                   QASL0010
C     ------------------------------------------------------------------QASL0060
C     PERFORMS LOGICAL OPERATIONS AND COMPUTATIONS TO IMPLEMENT THE      QASL0070
C     QUASILINEARIZATION ALGORITHM                                       QASL0080
C     ------------------------------------------------------------------QASL0090
C       J.R.RADBILL, OCEAN SYSTEMS OPERATIONS, EXT.3934,ST183. 02-03-67  QASL0120
      DOUBLE PRECISION ACUM, TNSTOR, F, IA, DOTD, H2, H3, P, P2, C, CS,  QASL0130
     A  CTP, T, E, SS, SP, DBLUNT, DBLDOT, DSQRT, MU(20), D(20),         QASL0140
     B  MUD2 (20), TX, XKP1, EWL, XK, DOTJX, B, XI, G, XJAC, H, XKS,     QASL0150
     C  DXKP1, TSUB, VARIND, RHO(20), DTPRNT , XMNSTP , STPSIZ          QASL0160
      DIMENSION XKS(N1,IMAX), P(N1,IMAX), H1(N1,N2,IMAX), XJAC(N1,NU),   QASL0170
     A  T(N2,N2,NTM), CTP(N2,NTM), DXKP1(NU,NS), XKP1(NU,NS),ACUM(NU,NS)QASL0180
      DIMENSION B(10), H2( 6, 6), N(5), G(20),                           QASL0190
     A MEMBER(10), LOCATE(10), TEMP( 12,12) ,   C(5), MI(10)            QASL0200
      COMMON /ZOO/ N, MX, XERR, EPS, B, LOCATE, MEMBER, MI, RHOS(20)     QASL0210
      COMMON /ZT/ KTRANS(50), F(6,6), P2(6), CS, H3(6,8)                 QASL0220
      COMMON /ZSUB/          TSUB(10),VARIND(100),XMNSTP(9),STPSIZ(9),   QASL0230
     *KSUB(10) , MAXSTP                                                 QASL0240
      COMMON/ZX/ XI(10), EPSP, EPSP1, IA(20)                             QASL0250
      COMMON / ZI / TX, H ,NI,NIC,NSC, NMIN,NMAX,NRST, TNSTOR, NDIF,NBP  QASL0260
     A , DTPRNT , XK(20)                                                 QASL0270
      LOGICAL STOR                                                       QASL0280
C     ------------------------------------------------------------------QASL0290
      WRITE(6, 1)                                                        QASL0300
    1 FORMAT( '1 SUBROUTINE QASLIN ENTERED...BEGIN NEW ITERATION SET' )  QASL0310
      DO 11  I=1,6                                                       QASL0320
      DO 11  J=1,8                                                       QASL0330
   11 H3(I,J) = 0.                                                       QASL0340
C       INITIALIZE INTEGER CONTROL VARIABLES                             QASL0350
                N3 = N(3)                                                QASL0360
                N1P1 = N1 + 1                                            QASL0370
                NC = NU - N1                                             QASL0380
                N(1) = N1                                                QASL0390
                N(2) = N2                                                QASL0400
```

Figure 65. Subroutine QASLN Listing

```
                N5  = N(5)                                                  QASL0410
                NIC = N1                                                    QASL0420
                NSC = NS                                                    QASL0430
                NU1 = NU                                                    QASL0440
              IF ( IABS( NBP ).LE.NC ) NU1 = NU - NBP                       QASL0450
C     ------------------------------------------------------------------    QASL0460
C       SET-UP CALLS FOR MULTIPLE ENTRY SUBROUTINES                         QASL0470
                CALL DORTH ( H2, E, IA, T, C, N2, 6, 6, 0 )                 QASL0480
C     ------------------------------------------------------------------    QASL0490
C       GENERATE INITIAL GUESSED DISTRIBUTION OF DEPENDANT VARIABLES        QASL0500
                CALL XIN ( XKS, N1, IMAX          )                         QASL0510
C     ------------------------------------------------------------------    QASL0520
C       DO N3 ITERATIONS OF QUASILINEARIZATION PROCESS (MAY CONVERGE 1ST)   QASL0530
  250 DO 950 NITER = 1, N3                                                  QASL0540
      WRITE (6,251)                                                         QASL0550
  251 FORMAT ('1BEGIN A NEW QUASILINEARIZATION ITERATION')                  QASL0560
              IF ( N(4).LT.3 ) GO TO 300                                    QASL0570
              WRITE (6,254)                                                 QASL0580
  254            FORMAT( 1H1 )                                            QASL0590
              DO 257 J = 1, IMAX                                            QASL0600
  257         WRITE (6,258) VARIND(J), (XKS(I,J), I=1,N1)                   QASL0610
  258            FORMAT (1X, F11.3, 1P10D10.3 )                           QASL0620
              IF (NC.NE.0) WRITE (6,396) (XK(I)    , I=N1P1, NU )           QASL0630
C     ------------------------------------------------------------------    QASL0640
C       SET UP INITIAL CONDITION FOR PARTICULAR AND N2 HOMOGENEOUS SOLN     QASL0650
  300         DO 330 K = 2, NS                                              QASL0660
              DO 320 J = 1, NU                                              QASL0670
  320            XKP1( J, K )    = 0. D0                                    QASL0680
                 M = MI(K-1)                                                QASL0690
  330            XKP1( M, K ) = XI( K-1 )                                   QASL0700
              DO 351 J = 1, N1                                              QASL0710
                 G(J) = 0.D0                                                QASL0720
                 K  =  N1  +  J                                             QASL0730
                 XKP1( J, 1 )    = XKS( J, 1 )                              QASL0740
  351          IF ( J.LE.NC ) XKP1(K,1) = XK(K)                           QASL0750
                 MT = 1                                                     QASL0760
                 KTRANS(1) = 0                                              QASL0770
C             ------------------------------------------------------------  QASL0780
```

Figure 65. Subroutine QASLN Listing (Continued)

```
C              COMPUTE ALGEBRAIC INITIAL CONDITIONS                          QASL0790
               CALL ALGBC ( XKP1, G, XJAC )                                  QASL0800
               DO 385 J = 1, N1                                              QASL0810
                  IF ( G(J).NE.0.DO ) XKP1(J,1) = G(J)                       QASL0820
               DO 384 K = 2, NS                                              QASL0830
                  FWL = 0.DO                                                 QASL0840
                  DO 380 M = 1, NU                                           QASL0850
  380               FWL = XJAC(J,M) * XKP1(M,K) + FWL                        QASL0860
                  IF (FWL.NE.0.DO) XKP1(J,K) = FWL                           QASL0870
                  H1(J,K-1,1) = SNGL( XKP1(J,K) )                            QASL0880
  384          CONTINUE                                                      QASL0890
  385          P(J,1) = XKP1(J,1)                                            QASL0900
       IF ( N(4) .LT. 2 ) GO TO 390                                          QASL0910
       WRITE( 6,388 )                                                        QASL0920
  388 FORMAT( '1   INITIAL CONDITIONS FOR NEXT QUASILIN. ITERATION '  )      QASL0930
                  DO 387 J = 1, NS                                           QASL0940
  387               WRITE (6,386) ( XKP1(I,J), I = 1, NU )                   QASL0950
  386              FORMAT ( / ( 1X, 1P1209.2 ))                              QASL0960
C      ---------------------------------------------------------------------QASL0970
C        INTEGRATE THRU UP TO 10 SEGMENTS WITH BOUNDARIES GIVEN BY TSUB      QASL0980
  390 DO 715 KP=1,10                                                         QASL0990
       IF( KSUB(KP+1).EQ.0 ) GO TO 716                                       QASL1000
         I1=KSUB(KP)                                                         QASL1010
       I2 = KSUB(KP+1)-  1                                                   QASL1020
       DTPRNT = VARIND(I1+1) - VARIND(I1)                                    QASL1030
                  NRST = -1                                                  QASL1040
                 NI =  -1                                                    QASL1050
                   WRITE (6,393) DTPRNT, I1, KSUB(KP+1)                      QASL1060
  393              FORMAT(  //19HOSTORAGE INTERVAL = F16.5, 2I10 )           QASL1070
                   NMIN = I1                                                 QASL1080
                   NMAX = KSUB(KP+1)                                         QASL1090
       DO 715 KPRINT =  I1,I2                                                QASL1100
C        INTEGRATE THRU I2-I1 EQUALLY SPACED STORAGE INTERVALS               QASL1110
                  KPRINT = KPRINT                                            QASL1120
                  TX = VARIND( KPRINT )                                      QASL1130
                  TNSTOR = VARIND( KPRINT + 1 )                              QASL1140
C      ---------------------------------------------------------------------QASL1150
```

Figure 65. Subroutine QASLN Listing (Continued)

```
C       INTERPOLATE FOR XK IN XKS                                          QASL1160
C                                                                          QASL1170
  400 IF( NI .EQ. 0 .OR. NI .EQ. 2  )   GO TO 470                          QASL1180
          CALL INTER ( XKS, N1, IMAX ,NU,D,MUD2,VARIND , KPRINT )          QASL1190
C     ---------------------------------------------------------------------QASL1200
C                                                                          QASL1210
C       EVALUATE DERIVATIVES AND JACOBIAN                                  QASL1220
          IF (N(4).GT.2) WRITE (6,386) XK                                  QASL1230
  450     CALL JACOB ( TX, XK, G, XJAC, 1 )                                QASL1240
  470     DO 490 I = 1, N1                                                 QASL1250
            DOTD = G(I)                                                    QASL1260
            DO 476 J = 1, NU1                                              QASL1270
  476         DOTD =      XJAC(I,J)  * ( XKP1(J,1) -  XK(J) ) + DOTD       QASL1280
              DXKP1(I,1) =       DOTD                                      QASL1290
  490       CONTINUE                                                       QASL1300
            DO 550 M = 2, NS                                               QASL1310
            DO 550 I = 1, N1                                               QASL1320
              DOTJX = 0.00                                                 QASL1330
              DO 516 J = 1, NU                                             QASL1340
  516           DOTJX = XJAC(I,J) * XKP1(J,M) + DOTJX                      QASL1350
              DXKP1(I,M) = DOTJX                                           QASL1360
  550     CONTINUE                                                         QASL1370
C     ---------------------------------------------------------------------QASL1380
C                                                                          QASL1390
C       INTEGRATE DXKP1                                                    QASL1400
          CALL INTEG ( XKP1,DXKP1, NU,NS ,ACUM ,TSUB(KP+1) )               QASL1410
          IF (NI.NE.0 ) GO TO 400                                          QASL1420
      IF( DABS( TNSTOR - TX) .GT. 1.D-6 ) GO TO 400                        QASL1430
  586     IF ( N(4) .LT. 3 ) GO TO 625                                     QASL1440
            WRITE (6,386) TX                                               QASL1450
            WRITE (6,386) XKP1                                             QASL1460
C     ---------------------------------------------------------------------QASL1470
C                                                                          QASL1480
C       SAVE PARTICULAR AND HOMOGENEOUS SOLUTIONS                          QASL1490
  625     DO 650 I = 1, N2                                                 QASL1500
            M = MEMBER( I )                                                QASL1510
            STOR =                  KPRINT.EQ.LOCATE(I) -1                 QASL1520
            IF(STOR) H3(I,NS) = B(I) - XKP1(M,1)
            DO 650 J = 1, N2
            IF (STOR) H3(I,J) = XKP1(M,J+1)
  650 CONTINUE
```

Figure 65. Subroutine QASLN Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
            DO 660 I = 1, N1                                                    QASL1530
               P(I,KPRINT+1) = XKP1(I,1)                                        QASL1540
         DO 660 J = 1, N2                                                       QASL1550
  660            H1(I,J,KPRINT+1) = SNGL( XKP1(I,J+1) )                        QASL1560
C        ---------------------------------------------------------------------QASL1570
C          TEST B.C. SUBSPACE OF HOMOG. SOLN. MATRIX. IF VECTORS ARE NEARLY   QASL1580
C          PARALLEL, CONSTRUCT NEW ORTHO-NORMAL SET, STORE TRANS. MATRIX.     QASL1590
C          REDUCE PART.SOLN.TO UNIT VECTOR BY SUB.HOMOG.SOLNS.                QASL1600
         IF(KPRINT.EQ.1.OR.KPRINT.EQ. 12 .OR.MT.GE.NTM )  GO TO 715             QASL1610
            DO 665 I = 1, N2                                                    QASL1620
               M = MEMBER(I)                                                    QASL1630
            DO 664 J = 1, N2                                                    QASL1640
                TEMP(I,J) = SNGL(XKP1(M,J+1) )                                  QASL1650
  664           H2(I,J) = XKP1(M,J+1)                                           QASL1660
  665           P2(I) = XKP1(M,1)                                               QASL1670
               SP = 1.D0                                                        QASL1680
            DO 673 J = 1, N2                                                    QASL1690
               SS = 0.D0                                                        QASL1700
            DO 672 I = 1, N2                                                    QASL1710
  672           SS = H2(I,J)**2 + SS                                            QASL1720
  673           SP = SS * SP                                                    QASL1730
            IF ( SP.EQ.0.D0 ) GO TO 715                                         QASL1740
               SCALE = 1. / SQRT( SNGL( SP ) )                                  QASL1750
               MX = IDETRM ( 12, N2,TEMP ,SCALE)                                QASL1760
            IF (ABS(SCALE).GE. EPSP ) GO TO 715                                 QASL1770
               NRST = 0                                                         QASL1780
            CALL DORTOG ( T(1,1,MT), N2 )                                       QASL1790
               CS = 1.D0 - 1.D0 / DBLUNT( P2, IA, N2 )                          QASL1800
            DO 694 K = 1, N2                                                    QASL1810
               CTP(K,MT) = CS * DBLDOT( P2, E(1,K), N2 )                        QASL1820
            DO 694 I = 1, NU                                                    QASL1830
  694           ACUM(I,K) = XKP1(I,K+1)                                         QASL1840
  695       CALL MATMPY (ACUM,T(1,1,MT),XKP1(1,2),NU,N2,N2,NU,N2,N2)            QASL1850
  696       CALL MATMPY (XKP1(1,2),CTP(1,MT),IA,NU,N2,1,NU,N2,1)                QASL1860
               KTRANS(MT) = KPRINT + 1                                          QASL1870
               MT = MT + 1                                                      QASL1880
```

Figure 65. Subroutine QASLN Listing (Continued)

```
                DO 708 I = 1, NU                                                  QASL1890
                   XKP1(I,1) = XKP1(I,1) - IA(I)                                  QASL1900
                DO 708 J = 1, NS                                                  QASL1910
  708              ACUM(I,J) = 0.DO                                               QASL1920
                IF (N(4).LE.1) GO TO 715                                          QASL1930
                   WRITE (6,386) XKP1                                             QASL1940
  715   CONTINUE                                                                  QASL1950
  716              MTM1 = MT - 1                                                  QASL1960
                WRITE (6,717) MTM1                                                QASL1970
  717              FORMAT( 37H0     NUMBER OF ORTHOGONALIZATIONS =    I5  )      QASL1980
C       ------------------------------------------------------------------------QASL1990
C       SET UP AND SOLVE LINEAR ALGEBRAIC EQS.FOR COEFS.OF HOMOG.SOLNS.           QASL2000
                IF ( N5.LE.N2 ) GO TO 751                                         QASL2010
                DO 749 K = 1, N2                                                  QASL2020
                 DO 749 J = 1, NS                                                 QASL2030
                    ACUM(K,J) = 0.DO                                              QASL2040
                  DO 749 I = 1, N5                                                QASL2050
  749               ACUM(K,J) = H3(I,K) * H3(I,J) + ACUM(K,J)                     QASL2060
                DO 750 J = 1, NS                                                  QASL2070
                 DO 750 I = 1, N2                                                 QASL2080
  750               H3(I,J) = ACUM(I,J)                                           QASL2090
  751           IF ( N2.EQ.1) GO TO 805                                           QASL2100
                   F = 1.DO                                                       QASL2110
                DO 753 J = 1, N2                                                  QASL2120
  753           G(J) =   H3(J,NS)                                                 QASL2130
      IF ( N(4) .LT. 2 ) GO TO 754                                                QASL2140
                WRITE(6,755) H3                                                   QASL2150
  755       FORMAT(1H-,15H THE MATRIX IS    /(1P6D17.7))                        QASL2160
  754           MX = ISIMDD (  6, N2, 1, H3, G, F, IA )                         QASL2170
                   DO 752 I = 1, N2                                               QASL2180
  752                 IA( I ) = H3  (I,1)                                       QASL2190
                GO TO ( 850, 770, 790 ), MX                                       QASL2200
  770           WRITE (6,775)                                                     QASL2210
  775              FORMAT ( 1H1, 10X, 27HOVER-FLOW OCCURED IN DSIMEQ )            QASL2220
                GO TO 850                                                         QASL2230
  790           WRITE (6,795)                                                     QASL2240
  795              FORMAT ( 1H0, 10X, 19HMATRIX WAS SINGULAR          )           QASL2250
      IIT = 7                                                                     QASL2260
                GO TO 980                                                         QASL2270
```

Figure 65. Subroutine QASLN Listing (Continued)

```
805         IF ( H3(1,1)   .EQ. 0.D0 ) GO TO 790                          QASL2280
               IA(1) = H3(1,NS) / H3(1,1)                                 QASL2290
850         WRITE (6,851)                                                 QASL2300
851            FORMAT ( 20HO COMBINATION COEFF    )                       QASL2310
            WRITE (6,386) ( IA(I), I = 1, N2 )                            QASL2320
C     ------------------------------------------------------------------QASL2330
C       COMBINE P AND H SOLNS TO OBTAIN X(K+1)                            QASL2340
            DO 852 I = 1, NU                                              QASL2350
852            RHO(I) = 0.D0                                              QASL2360
            DO 840 I = 1, NC                                              QASL2370
               M = I + N1                                                 QASL2380
            DO 835 K = 1, N2                                              QASL2390
835            XKP1(M,1) = IA(K) * XKP1(M,K+1)+XKP1(M,1)                  QASL2400
               RHO(M) = DABS( XKP1(M,1)   - XK(M) )                       QASL2410
840            XK(M) = XKP1(M,1)                                          QASL2420
            DO 895 J = 1, IMAX                                            QASL2430
               L = IMAX + 1 - J                                           QASL2440
            DO 872 I = 1, N1                                              QASL2450
               DOTD = P(I,L)                                              QASL2460
               DO 866 K = 1, N2                                           QASL2470
866               DOTD = IA(K) * DBLE( H1(I,K,L) ) + DOTD                 QASL2480
               RHO(I) = DMAX1 (RHO(I), DABS( DOTD - XKS(I,L) ) )          QASL2490
872            XKS(I,L) = DOTD                                            QASL2500
C              TRANSFORM COMBINATION COEF VECTOR IF SOLN WAS TRANS        QASL2510
               IF( L-1 .NE. KTRANS(MT-1) .OR. MT.LE.1 ) GO TO 895         QASL2520
                 MT = MT - 1                                              QASL2530
               DO 886 I = 1, N2                                           QASL2540
886            C(I) = IA(I) - CTP(I,MT)                                   QASL2550
            CALL MATMPY ( T(1,1,MT), C, IA, N2, N2, 1,N2, N2, 1 )         QASL2560
895         CONTINUE                                                      QASL2570
C     ------------------------------------------------------------------QASL2580
      WRITE(6,910)                                                        QASL2590
910   FORMAT(40HO CONVERGENCE PARAMETERS----PHO                )          QASL2600
      WRITE(6,386)(RHO(I),I=1,NU)                                         QASL2610
               CALL TIMEV ( EXT )                                         QASL2620
               WRITE (6,898) EXT                                          QASL2630
898               FORMAT ( 1HO, 10X, 1F7.3,1X, 3HSEC )                    QASL2640
```

Figure 65. Subroutine QASLN Listing (Continued)

```
      DO 945 I=1,NU                                                          QASL2650
      IF( RHO(I).GE.DBLE(RHOS(I)) ) GO  TO 949                               QASL2660
  945   CONTINUE                                                             QASL2670
                WRITE (6,946) NITER                                          QASL2680
  946             FORMAT ( / 9X, 29H PROCESS CONVERGED, ITER.NO.= I5 )       QASL2690
      IIT = 2                                                                QASL2700
            GO TO 980                                                        QASL2710
  949       CALL OUTPUT ( XKS, N1, IMAX, 1 )                                 QASL2720
  950 CONTINUE                                                               QASL2730
C     ------------------------------------------------------------------     QASL2740
                IIT = 3                                                      QASL2750
                WRITE (6,965) N3                                             QASL2760
  965           FORMAT (/10X, 17HNO CONVERGENCE IN ,I3,8H  CYCLES )          QASL2770
  980 CALL OUTPUT( XKS,N1,IMAX,IIT)                                          QASL2780
      IF( IIT .EQ. 0 ) GO TO 1200                                            QASL2790
      WRITE(6,981)                                                           QASL2800
  981 FORMAT(54H1 SOLUTION CHECK BY INTEGRATION OF INITIAL CONDITIONS )      QASL2810
      DO 1000 I=1,N1                                                         QASL2820
            RHO(I) = 0.D0                                                    QASL2830
      ACUM(I,1) =  0.D0                                                      QASL2840
 1000           XKP1(I,1) = XKS(I,1)                                         QASL2850
      DO 1001 I = N1P1 , NU                                                  QASL2860
      XKP1(I,1) = XK(I)                                                      QASL2870
 1001 ACUM(I,1) = 0.D0                                                       QASL2880
      DO 1050 I=1,10                                                         QASL2890
        I1=KSUB(I)                                                           QASL2900
      I2 = KSUB(I+1) - 1                                                     QASL2910
      DTPRNT = VARIND(I1+1)  - VARIND(I1)                                    QASL2920
      NRST  = -1                                                             QASL2930
 1020 DO 1050 J=    I1,I2                                                   QASL2940
                TX = VARIND(J)                                               QASL2950
                TNSTOR = VARIND(J+1)                                         QASL2960
 1030         CALL JACOB ( TX, XKP1, DXKP1, XJAC, 2 )                        QASL2970
              CALL INTEG ( XKP1, DXKP1, N1, 1, ACUM, TSUB(I+1) )             QASL2980
              IF ( NI.NE.0 ) GO TO 1030                                      QASL2990
              IF ( DABS( TNSTOR - TX ) .GT. 1.D-6 ) GO TO 1030               QASL3000
```

Figure 65. Subroutine QASLN Listing (Continued)

```
            DO 1040 K = 1, N1                                                    QASL3010
               RHO(K) = DMAX1 ( RHO(K), DABS( XKS(K,J+1) - XKP1(K,1)))           QASL3020
1040  P(K,J+1) = XKP1(K,1)                                                       QASL3030
      IF( J .EQ. IMAX -1 )  GO TO 1100                                           QASL3040
1050     CONTINUE                                                                QASL3050
1100          WRITE (6,910)                                                      QASL3060
              WRITE (6,386) ( RHO(I), I = 1, N1 )                                QASL3070
1110  CALL OUTPUT(  P ,N1 , IMAX, 6 )                                            QASL3080
 990  RETURN                                                                     QASL3090
1200  CALL OUTPUT( XKS, N1 , IMAX , 4  )                                         QASL3100
      RETURN                                                                     QASL3110
      ------------------------------------------------------------------------   QASL3120
      END                                                                        QASL3130
```

Figure 65. Subroutine QASLN Listing (Continued)

```
      SUBROUTINE JACOB (TIME, XOFT, F, AJ, IJUMP)                       JACB0000
C     ------------------------------------------------------------------JACB0001
C     CONTAINS ALL OF THE DIFFERENTIAL EQUATIONS REQUIRED BY THIS       JACB0002
C     PARTICULAR PROBLEM.  WHEN IT IS CALLED, IT RETURNS THE CURRENT    JACB0003
C     ESTIMATE OF THE DERIVATIVE OF EACH VARIABLE AND EVALUATES THE     JACB0004
C     JACOBIAN MATRIX AS WELL.                                          JACB0005
C     ------------------------------------------------------------------JACB0006
      IMPLICIT REAL*8 (A-Z)                                             JACB0010
      INTEGER*4 IJUMP, ID, I, J  ,K,N, JACPRN                           JACB0020
      DIMENSION  XOFT(14),F(14),AJ(13,14)                               JACB0030
     1 , A2(9,10) , A3(13,14)                                           JACB0040
      COMMON /ZIC/BETA,  C , GO ,XMU,TW,XXK(20)                         JACB0050
      COMMON/JACSET/ ID, JACPRN                                         JACB0060
      EQUIVALENCE              (A2,A3)                                  JACB0070
C     THE ORDER OF VARIABLES FOR THREE DIMENSIONS IS                    JACB0080
C     R, PHI, RHO, Y, M, L1, K, L4, L6, THETA, X, L2, L5, L3.           JACB0090
C     FOR TWO DIMENSIONS L3 IS MOVED TO NO. 10 (IN PLACE OF THETA)      JACB0100
      IF( IJUMP .NE. 0 ) GO TO 40                                       JACB0110
      DO 30 I=1,13                                                      JACB0120
      DO 30 J=1,14                                                      JACB0130
      A3(I,J)=0.                                                        JACB0140
   30 AJ(I,J)=0.                                                        JACB0150
      MU = XMU                                                          JACB0160
      RETURN                                                            JACB0170
   40 GO TO (500,50,75),ID                                              JACB0180
   50 X=0.                                                              JACB0190
      THTA=0.                                                           JACB0200
      L5=0.                                                             JACB0210
      L2=0.                                                             JACB0220
      L3=XOFT(10)                                                       JACB0230
      GO TO 100                                                         JACB0240
   75 L2=XOFT(12)                                                       JACB0250
       L3 =XOFT(14)                                                     JACB0260
      X=XOFT(11)                                                        JACB0270
      THTA=XOFT(10)                                                     JACB0280
      L5=XOFT(13)                                                       JACB0290
  100 R=XOFT(1)                                                         JACB0300
```

Figure 66. Subroutine JACØB Listing

```
      PHI=XOFT(2)                                                    JACB0310
      P=XOFT(3)                                                      JACB0320
      Y=XOFT(4)                                                      JACB0330
      M=XOFT(5)                                                      JACB0340
      L1=XOFT(6)                                                     JACB0350
      L4=XOFT(8)                                                     JACB0360
      L6=XOFT(9)                                                     JACB0370
      R2=R**2                                                        JACB0380
      R3=R2*R                                                        JACB0390
      RM=1./R                                                        JACB0400
      RM2=1./R2                                                      JACB0410
       CS=DCOS (THTA)                                                JACB0420
       CS2= CS**2                                                    JACB0430
       TN=DTAN (THTA)                                                JACB0440
       SN=DSIN (THTA)                                                JACB0450
       SN2= SN**2                                                    JACB0460
      SNCS= SN*CS                                                    JACB0470
      CSSN = CS2 - SN2                                               JACB0480
      RC=RM/CS                                                       JACB0490
       XTN=X*TN                                                      JACB0500
      X2=X**2                                                        JACB0510
      Y2=Y**2                                                        JACB0520
      D2=L4**2*R2* CS2+L5**2* CS2+L6**2                              JACB0530
      D=DSQRT(D2)                                                    JACB0540
      Q1=L4*R* CS/D                                                  JACB0550
      Q2=L6/D                                                        JACB0560
      Q3=L5* CS/D                                                    JACB0570
      CBM = C*BETA/M                                                 JACB0580
      Q1S=Q1**2                                                      JACB0590
      Q2S=Q2**2                                                      JACB0600
      Q3S=Q3**2                                                      JACB0610
      MRC=RC/M                                                       JACB0620
      MQ1=1.-Q1S                                                     JACB0630
      AQ1=1.+Q1S                                                     JACB0640
      MQ2=1.-Q2S                                                     JACB0650
      DM=1./D                                                        JACB0660
```

Figure 66. Subroutine JACØB Listing (Continued)

```
      Q12=Q1*Q2                                                                 JACB0670
      Q23=Q2*Q3                                                                 JACB0680
      Q13=Q1*Q3                                                                 JACB0690
      CM=C/M                                                                    JACB0700
      MM=1./M                                                                   JACB0710
      MR=RM*MM                                                                  JACB0720
      IF (IJUMP.EQ.2) GO TO 1000                                                JACB0730
      A2(1,3) =1                                                                JACB0740
      A2(2,4) =1                                                                JACB0750
      A2(3,1) =X2+ CS2*Y2+2*MU/R3+CBM*Q1*MQ1*RM                                 JACB0760
      A2(3,4) =2*R*Y* CS2                                                       JACB0770
      A2(3,5)=  -CBM*Q1*MM                                                      JACB0780
      A2(3,8) =CBM*Q1*MQ1/L4                                                    JACB0790
      A2(3,9) =-CBM*Q12*DM                                                      JACB0800
      A2(4,1) =2.*P*Y*RM2-CBM*RC*Q2*AQ1*RM                                      JACB0810
      A2(4,3) =-2.*Y*RM                                                         JACB0820
      A2(4,4) = 2.*X*TN -2.*P*RM                                                JACB0830
      A2(4,5) = -CBM*MRC*Q2                                                     JACB0840
      A2(4,8) = -CBM*Q12*DM                                                     JACB0850
      A2(4,9)=  CBM*RC*DM*MQ2                                                   JACB0860
      A2(6,1) = 6.*L4*MU*RM2**2+4.*P*(L5*X+L6*Y)*RM*RM2-CBM*D*RC*MQ1*RM2JACB0870
     1*(2.+Q1S)                                                                 JACB0880
      A2(6,3) = -2.*RM2*(L5*X+L6*Y)                                             JACB0890
      A2(6,4) = -2.*(L4*Y* CS2+L6*P*RM2)                                        JACB0900
      A2(6,5) = -CBM*MRC*D*MQ1*RM                                               JACB0910
      A2(6,8) = -(X2+Y2* CS2+2*MU*RM*RM2)-CBM*Q1*MQ1*RM                         JACB0920
      A2(6,9) = -2.*P*Y*RM2+CBM*PM*Q2*AQ1*RC                                    JACB0930
      A2(7,1) = -CM*RM* (L1*Q1*MQ1-Q2*L3*AQ1*RC)-C*P*D*MQ1*(2.+Q1S)*MRC*JACB0940
     1RM2                                                                       JACB0950
      A2(7,3) = C*D*MQ1*MRC*RM                                                  JACB0960
      A2(7,5) =   CM*MM*(L1*Q1+Q2*L3*RC-P*D*MQ1*RC*RM)                          JACB0970
      A2(7,6) =-CM*Q1                                                           JACB0980
      A2(7,8) =   -CM*DM*(L1*R* CS*MQ1-Q12*L3)-C*P*Q1*MR*MQ1                    JACB0990
      A2(7,9) =   CM*DM*(L1*Q12-L3*MQ2*RC)+C*P*MRC*RM*Q2*AQ1                    JACB1000
      A2(7,10) = -CM*Q2*RC                                                      JACB1010
      A2(8,1) = -2.*(L5*X+L6*Y)*RM2                                             JACB1020
      A2(8,4) = 2.*L6*RM                                                        JACB1030
      A2(8,6) = -1.                                                             JACB1040
      A2(8,9) = 2.*Y*PM                                                         JACB1050
```

Figure 66. Subroutine JACØB Listing (Continued)

```
      A2(9,1) = -2.*Y*L4* CS2-2.*L6*P*PM2                                    JACB1060
      A2(9,3) = 2.*L6*RM                                                     JACB1070
      A2(9,4) = -2.* CS*(R*_4* CS-L5* SN)                                    JACB1080
      A2(9,8) = -2.*R*Y* CS2                                                 JACB1090
      A2(9,9) = -2.*(XTN  -P*RM)                                             JACB1100
      A2(9,10)= -1.                                                          JACB1110
      IF(ID . EQ. 2) GO TO 181                                               JACB1120
C     3-D PACKING                                                            JACB1130
      DO 190 K=1,9                                                           JACB1140
      DO 190 N=1,10                                                          JACB1150
  190 AJ(K,N) = A2(K,N)                                                      JACB1160
      GO TO 200                                                              JACB1170
C     2-D PACKING                                                            JACB1180
  181 DO 191 K=1,13                                                          JACB1190
      DO 191 N=1,14                                                          JACB1200
  191 AJ(K,N) = A3(K,N)                                                      JACB1210
      GO TO 1000                                                             JACB1220
  200 AJ(3,10)= -2.*R*Y2*SNCS-CBM*Q1*Q2S*TN                                  JACB1230
      AJ(3,11)=2.*R*X                                                        JACB1240
      AJ(3,13)= -CBM*Q1*Q3S/L5                                               JACB1250
      AJ(4,10)= (2*X*Y+CBM* SN*Q2*RM*(1.+MQ2))/ CS2                          JACB1260
      AJ(4,11)= 2.*Y*TN                                                      JACB1270
      AJ(4,13)= -CBM*Q23*RM*DM                                               JACB1280
      AJ(6,10)= 2.*L4*Y2*SNCS+CBM*D*RC**2* SN*Q2S*AQ1                        JACB1290
      AJ(6,11)= -2.*(L4*X+L5*P*RM2)                                          JACB1300
      AJ(6,13)= -2.*P*X*RM2+CBM*RM2*Q3*AQ1                                   JACB1310
      AJ(7,1) = AJ(7,1)  +CM*RM2*L2*Q3*AQ1+C*MRC*RM*XTN  *L6*Q2*AQ1          JACB1320
      AJ(7,5) = AJ(7,5)  +CM*MR*Q3*L2+ CM*MM*X*L6*Q2*TN* RC                  JACB1330
      AJ(7,8)= AJ(7,8)+DM*CM* CS*Q13*L2+CM*X*Q2S*Q1*TN                       JACB1340
      AJ(7,9) = AJ(7,9)  +CM*DM*L2*Q23*RM-C*MRC*XTN  *Q2*(1.+MQ2)            JACB1350
      AJ(7,10) = CM*TN*Q2*(L1*Q12 + L2*Q23*RM - L3*RC*(1. + MQ2)) - CM*      JACB1360
     1RC*L6*Q2*(X/CS2 + XTN*TN*(1. + MQ2) - P*AQ1*RM*TN)                     JACB1370
      AJ(7,11)=-CM*RC*D*TN *Q2S                                              JACB1380
      AJ(7,12)=-CM*RM*Q3                                                     JACB1390
      AJ(7,13)=CM*RM*DM*(L1*Q13*R* CS-L2*(1.-Q3S)*CS+L3*Q23)+CM*RM*XTN *     JACB1400
     1Q2S*Q3+CM*P*Q3*RM2*AQ1                                                 JACB1410
      AJ(7,14) = -CM*Q2*RC                                                   JACB1420
      AJ(8,11)= 2.*L5*RM                                                     JACB1430
      AJ(8,13)= 2.*X*RM                                                      JACB1440
```

Figure 66. Subroutine JACØB Listing (Continued)

```
      AJ(9,10)=4.*R*Y*L4*SNCS+2.*L5*Y*CSSN-2.*L6*X/ CS2                    JACB1450
      AJ(9,11)=-2.*L6*TN                                                  JACB1460
      AJ(9,13)= 2.*Y*SNCS                                                 JACB1470
      AJ(9,14)= -1.                                                       JACB1480
      AJ(10,11)= 1.                                                       JACB1490
      AJ(11,1)= RM2*2.*P*X-CBM*RM2*Q3*AQ1                                 JACB1500
      AJ(11,3 )= -2.*X*RM                                                 JACB1510
      AJ(11,4)= -2.*Y*SNCS                                                JACB1520
      AJ(11,5)= -CBM*MR*Q3                                                JACB1530
      AJ(11,8)= -CBM*RM*Q1S*Q3/L4                                         JACB1540
      AJ(11,9) =-CBM*RM*DM*Q23                                            JACB1550
      AJ(11,10)=-Y2*CSSN-CBM*RM*DM*Q2S*L5*SN                              JACB1560
      AJ(11,11)= -2.*P*RM                                                 JACB1570
      AJ(11,13)= CBM*RM*Q3*(1.-Q3S)/L5                                    JACB1580
      AJ(12,1) =2 *L4*Y2*SNCS+CBM*D* SN*Q2S*RC**2*AQ1                     JACB1590
      AJ(12,4) = 4.*R*L4*Y*SNCS+2.*L5*Y*CSSN-2.*X*L6/ CS2                 JACB1600
      AJ(12,5) = CBM*MRC*TN *D*Q2S                                        JACB1610
      AJ(12,8) = 2.*R*Y2*SNCS+CBM*TN * Q2S*Q1                             JACB1620
      AJ(12,9) = -2.*X*Y/CS2 -CBM*RC*TN *Q2*(2.-Q2S)                      JACB1630
      AJ(12,10)=2.*P*L4*Y2*CSSN-4.*Y*SN *(L5*Y*CS +L6*X/(CS *CS2 ))-CBM*  JACB1640
     1 D*RC*Q2S*( SN2*(1.+MQ2)+1)/ CS2                                    JACB1650
      AJ(12,11)= -2*L6*Y/ CS2                                             JACB1660
      AJ(12,13)= Y2*CSSN+CBM* TN*RM*Q2S*Q3                                JACB1670
      AJ(13,1) = -2.*(X*L4+L5*P*RM2)                                      JACB1680
      AJ(13,3)= 2.*L5*RM                                                  JACB1690
      AJ(13,4)= -2.*L6*TN                                                 JACB1700
      AJ(13,8) = -2.*R*X                                                  JACB1710
      AJ(13,9) = -2.*Y*TN                                                 JACB1720
      AJ(13,10)= -2.*L6*Y/CS2                                             JACB1730
      AJ(13,11)= -2.*R*L4                                                 JACB1740
      AJ(13,12)= -1.                                                      JACB1750
      AJ(13,13)=  2*P*RM                                                  JACB1760
```

Figure 66. Subroutine JACØB Listing (Continued)

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

```
1000 F(1)=P                                                                    JACB1770
     F(2)=Y                                                                    JACB1780
     F(3)= R*(X2+CS2*Y2)-MU*RM2+CBM*Q1                                         JACB1790
     F(4)=2.*Y*(XTN-P*RM)+CBM*RC*Q2                                            JACB1800
     F(5) = -BETA                                                              JACB1810
     F(6)= -L4*(X2+Y2*CS2+2.*MU/R3     )-L5*RM2*(2.*P*X-CBM*Q3)-L6*RM2*(JACB1820
    12.*P*Y-CBM*Q2/CS)                                                         JACB1830
     F(7)= -CM*(L1*Q1+L2*Q3*RM+Q2 *L3*RC+XTN*RC*D*Q2S-P*D*MQ1*RC*RM)           JACB1840
     F(8)= -L1+2*RM*(L5*X+L6*Y)                                                JACB1850
     F(9)= -L3-2.*Y*(R*L4*CS2-L5*SNCS)-2.*L6*(XTN-P*RM)                        JACB1860
     F(10)=0.                                                                  JACB1870
 499 IF (ID.EQ.2) GO TO 500                                                    JACB1880
     F(10)= X                                                                  JACB1890
     F(11)= -2.*P*X*RM-Y2*SNCS+CBM*Q3*RM                                       JACB1900
     F(12)= L4*2.*R*Y2*SNCS+L5*Y2*CSSN-L6*(2.*X*Y/CS2 +CBM*RC*Q2*TN)           JACB1910
     F(13)=-L2-2.*R*X*L4+2.*L5*P*RM-2.*L6*Y*TN                                 JACB1920
     F(14)= 0.                                                                 JACB1930
 500 RETURN                                                                    JACB1940
     END                                                                       JACB1950
```

Figure 66. Subroutine JACOB Listing (Continued)

```
      SUBROUTINE STOP(XX,DX,NU,NS,INIT)                                 STOP0010
C     ------------------------------------------------------------------STOP0011
C     EACH TIME THE THRUST IS INITIATED OR TERMINATED THE QASLN PROGRAM STOP0012
C     CALLS SUBROUTINE STOP, WHICH PROVIDES THE SMALL CORRECTIONS.  IT  STOP0013
C     ALSO STORES THE DERIVATIVES FOR LATER USE BY SUBROUTINE KHIST.    STOP0014
C     ------------------------------------------------------------------STOP0015
C     THREE DIM. VERSION--IBM 360--MARCH 1967                           STOP0016
      IMPLICIT REAL*8(A-H , O-Z)                                        STOP0020
      REAL*8 IA,KDOT                                                    STOP0030
      REAL*4 XLST                                                       STOP0040
      DIMENSION XX(NU,NS),DX(NU,NS)                                     STOP0050
      COMMON /ZOO/ N(5)                                                 STOP0060
      COMMON /ZI/ TX,H,PAD(5),DTPRNT,XK(20)                             STOP0070
      COMMON /ZSUB/          TSUB(10),VARIND(100),XMNSTP(9),STPSIZ(9),  STOP0080
     *KSUB(10) , MAXSTP                                                 STOP0090
      COMMON /SOLN/ XLST(15,100)                                        STOP0100
      COMMON /ZIC/BETA,  C , GO ,XMU,TW,XXK(20)                         STOP0110
      COMMON /JACSET/ ID                                                STOP0120
        COMMON/ZCONST/ PI,RAD,GXM,BETAS                                 STOP0130
      COMMON /STOP1/ XKDOT(10), TWOBR , XNOCOR, SAVK(7,6),IQCYC         STOP0140
C     ******************************************************************STOP0150
C     DETERMINE WHICH STOP POINT                                        STOP0160
      BETA = 0.DO                                                       STOP0170
      DO 1  K=1,6                                                       STOP0180
      KB = K                                                            STOP0190
      KP = K - 1                                                        STOP0200
      IF (TX - TSUB(K) .LT. 1.D-6) GO TO 5                              STOP0210
    1 CONTINUE                                                          STOP0220
      WRITE (6,2)                                                       STOP0230
    2 FORMAT (30H NO STOP POINTS FOUND                                  )STOP0240
      RETURN                                                            STOP0250
    5 WRITE (6,3) KP, TX, TWOBR                                         STOP0260
    3 FORMAT (15H STOP POINT NO. I5, 9H  TIME =    1D16.8 ,  '  NULL BURNSTOP0270
     * CONTROL =  ' F10.2 )                                             STOP0280
      IF (KB .EQ. 6) RETURN                                             STOP0290
```

Figure 67. Subroutine STØP Listing

```
C       DETERMINE IF NULL BURN                                                  STOP0300
C        TWOBR    = 0  --- TWO BURNS                                           STOP0310
C        TWOBR    = 1  --- FIRST BURN NULL                                     STOP0320
C        TWOBR    = 2  --- SECOND BURN NULL                                    STOP0330
         IF (TWOBR  - 1.) 30, 10, 20                                           STOP0340
      10 IF (KB .NE. 2 .AND.  KB .NE. 3)  GO TO 30                             STOP0350
         WRITE (6,11) KP                                                       STOP0360
      11 FORMAT (34H  NULL BURN PERIOD---STOP POINT =  I5, 20H   NO THRUST     STOP0370
        1CHANGE      )                                                        STOP0380
         RETURN                                                                STOP0390
      20 IF (KB .NE. 4 .AND.  KB .NE. 5) GO TO 30                              STOP0400
         WRITE (6,11) KP                                                       STOP0410
         RETURN                                                                STOP0420
      30 IF (KB .EQ. 2  .OR.  KB .EQ. 4) BETA = BETAS                          STOP0430
         WRITE(6,110) BETA                                                     STOP0440
     110 FORMAT(25H THRUST CHANGE---BETA =   1D10.3 )                          STOP0450
         IF (XNOCOR .EQ. 99.) RETURN                                           STOP0460
         KDOT =      DABS( XKDOT(KP   ))                                       STOP0470
         XK13 = XK(13)                                                         STOP0480
         TH = XK(10)                                                           STOP0490
         IF ( ID .EQ. 2 ) TH = 0.                                              STOP0500
         IF ( ID .EQ. 2 ) XK13 = 0.                                            STOP0510
         CS = DCOS(TH)                                                         STOP0520
         CS2 = CS*CS                                                           STOP0530
         DXK = C * BETAS/(XK(5)*DSQRT(XK(9)**2 + CS2 * ((XK(8) * XK(1))**2     STOP0540
        *    + XK13  **2))*KDOT)                                               STOP0550
         DELT =XX(7,1) / KDOT                                                  STOP0560
         WRITE (6,210) XX(7,1), DELT, KDOT                                     STOP0570
     210 FORMAT (45H      K          DELTA TIME        KDOT      /3D15.6,/)    STOP0580
         IF ( N(4) .LE. 1 ) GO TO 212                                          STOP0590
         WRITE (6,211)                                                         STOP0600
     211 FORMAT (                                                              STOP0610
        *' PARTICULAR SOLN.  PERTURBATIONS IN RHO, Y, MASS, LAM 1, X, LAM 2    STOP0620
        * ARE' )                                                              STOP0630
C        ***************************************************************      STOP0640
```

Figure 67. Subroutine STØP Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
212 IEQ = 7                                                          STOP0650
    IF ( ID .EQ. 2 ) IEQ = 5                                         STOP0660
    DO 150 I=1,IEQ                                                   STOP0670
    DELT = XX(7,I) / KDOT                                            STOP0680
    D =  DXK*XX(7,I)                                                 STOP0690
    DK3 = XK(1)*XK(8) * D * CS                                       STOP0700
    DK4  = D *XK(9) /XK(1) / CS                                      STOP0710
    DK5  =-DELT * BETAS                                              STOP0720
    DK6 = D*(CS2*XK13  **2 + XK(9)**2)/(XK(1)**2 * CS)               STOP0730
    XX( 3,I) = XX( 3,I) + DK3                                        STOP0740
    XX( 4,I) = XX( 4,I) + DK4                                        STOP0750
    XX( 5,I) = XX( 5,I) + DK5                                        STOP0760
    XX( 6,I) = XX( 6,I) + DK6                                        STOP0770
    IF ( ID .EQ. 2 ) GO TO 148                                       STOP0780
    DK11 = D * XK13   / XK(1) * CS                                   STOP0790
    DK12 = -D * XK(9)**2 * DSIN(TH) /(CS2 * XK(1) )                  STOP0800
    XX(11,I) = XX(11,I) + DK11                                       STOP0810
    XX(12,I) = XX(12,I) + DK12                                       STOP0820
148 IF ( N(4) .LE. 1 ) GO TO 150                                     STOP0830
149 WRITE (6,301) DK3,DK4,DK5,DK6,DK11,DK12                          STOP0840
301 FORMAT (1P6D16.8)                                                STOP0850
150 CONTINUE                                                         STOP0860
    DO 200 I=1,IEQ                                                   STOP0870
200 SAVK(I,KP    ) =DX(7,I)                                          STOP0880
500 RETURN                                                           STOP0890
    END                                                              STOP0900
```

Figure 67. Subroutine STØP Listing (Continued)

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

```
      SUBROUTINE OUTPUT (XKS, N1, JMAX, ITER)                          OUTP0000
C     ---------------------------------------------------------------OUTP0001
C     PROVIDES STANDARD OUTPUT AT VARIOUS LOCATIONS IN THE COMPUTATIONS.OUTP0002
C     IT IS USUALLY CALLED BY THE QASLN SUBROUTINE.                    OUTP0003
C     ---------------------------------------------------------------OUTP0004
C        THREE DIM. VERSION--IBM 360--MARCH 1967                       OUTP0010
      IMPLICIT REAL*8 (A-H, O-Z)                                       OUTP0020
      REAL*8     L1, L2, L3, L4, L5, L6, L7                            OUTP0030
      REAL*4 XLST  ,TRU, STO                                           OUTP0040
      DIMENSION XKS(N1,JMAX)                                           OUTP0050
      DIMENSION TEMP(22)                                               OUTP0060
      COMMON /JACSET/ ND                                               OUTP0070
      COMMON /700/ N(5)                                                OUTP0080
      COMMON /ZI/ TX, H, PAD(5), DTPRNT, XK(20)                        OUTP0090
      COMMON /ZX/ XI(10), EPSP, EPSP1, IA(20)                          OUTP0100
      COMMON /ZSUB/ TSUB(10), VARIND(100), XMNSTP(9), STPSIZ(9),       OUTP0110
     *              KSUB(10), MAXSTP                                   OUTP0120
      COMMON /ZIC/BETA,  C , GD ,XMU,TW,XXK(20)                        OUTP0130
      COMMON /SOLN/ XLST(15,100), TRU(100), HBAR, VCHAR, STO(100,20,2) OUTP0140
       COMMON/ZCONST/ PI,RAD,GXM,BETAS,CONV,II,JJ                      OUTP0150
      COMMON /STOP1/ XKDOT(10), TWOBR , XNOCOR, SAVK(7,6),IQCYC        OUTP0160
C     ***************************************************************OUTP0170
C     ITER = 1    --  NORMAL END OF EACH QUASILIN CYCLE               OUTP0180
C     ITER = 2    --  CONVERGENCE OF QASILIN ITERATION SET           OUTP0190
C     ITER = 3    --  NON CONVERGENCE                                OUTP0200
C     ITER = 4    --  FINAL CALL AT END OF ITERATION SET             OUTP0210
C     ITER = 5    --  INITIAL CALL BEFORE QUASILINEARIZATION         OUTP0220
C     ITER = 6    --  CALL AFTER REINTEGRATION COMPLETED             OUTP0230
C     ITER = 7    --  SINGULAR MATRIX FAILURE                        OUTP0240
C     ***************************************************************OUTP0250
   20 FORMAT(1P1D13.5)                                                 OUTP0260
   21 FORMAT (' ', 1PD12.5, 7D15.6)                                    OUTP0270
   22 FORMAT ('0', '     TIME            R               PHI          RHOOUTP0280
     *          Y               MASS            LAM 1            K   '/ )OUTP0290
   23 FORMAT ('0', '     TIME         LAM 4           LAM 6          LAM OUTP0300
     *3   '/ )                                                         OUTP0310
   24 FORMAT ('0', '     TIME         LAM 4           LAM 6          THETOUTP0320
     *A            X              LAM 2           LAM 5          LAM 3 '/ )OUTP0330
```

Figure 68. Subroutine ØUTPUT Listing

```
 25 FORMAT ('0CHECK OF CONSTANTS FOR REINTEGRATED SOLUTION -- NO QUASIOUTP0340
   *LINEARIZATION')                                                  OUTP0350
 26 FORMAT( '0 DOUBLE PRECISION INITIAL CONDITIONS FOR OPTIMUM TRAJECTOUTP0360
   *ORY'  // ( 1P4E25.14 )  )                                        OUTP0370
 27 FORMAT('0********** LAST ITERATION SET COMPLETE **********' )    OUTP0380
 36 FORMAT (20H1OUTPUT   IIT =                 I5)                 OUTP0390
250 FORMAT (10H+          1P2D15.4,1P3D13.4,1P3D17.9)               OUTP0400
290 FORMAT(30H0 END OF ITERATION NUMBER                  I5    )    OUTP0410
502 FORMAT(50H1       TIME HISTORIES OF ALL VARIABLES               )OUTP0420
920 FORMAT(10H0 HBAR    ,1D16.8,10H  VCHAR   ,1D16.8 )              OUTP0430
1100 FORMAT('0     TIME     PRIMER MAG       SWITCH      STEER-PSI    SOUTP0440
   *TEER-NU   HAMILTONIAN    TOT. ENERGY     ANGULAR MOM.          ' ) OUTP0450
     ITSAV = ITER                                                    OUTP0460
     GO TO (35,400,400,35,35,35,35)  ,  ITER                         OUTP0470
400  ITER = 0                                                        OUTP0480
     WRITE(6,27)                                                     OUTP0490
     IF(                  ITSAV.EQ. 3  ) ITNO = ITNO - 1             OUTP0500
     IF (IQCYC .NE. 1) RETURN                                        OUTP0510
     ITER = ITSAV                                                    OUTP0520
     XNOCOR = 99.                                                    OUTP0530
     WRITE (6,36) ITER                                               OUTP0540
     GO TO 200                                                       OUTP0550
  35 WRITE (6,36) ITER                                               OUTP0560
     IF( ITER .EQ.  5  )   XNOCOR = 0.                               OUTP0570
     IF( ITER .EQ.  5  )   ITNO = -1                                 OUTP0580
     IF( ITER .EQ.  6  )   WRITE(6,25)                               OUTP0590
     IF ( ITER .EQ. 6 ) ITNO = ITNO - 1                              OUTP0600
 200 WRITE (6,1100)                                                  OUTP0610
     HBAR = 0.                                                       OUTP0620
       X = 0.                                                        OUTP0630
      TH = 0.                                                        OUTP0640
     STH = 0.                                                        OUTP0650
     CTH = 1.                                                        OUTP0660
      L2 = 0.                                                        OUTP0670
      L5 = 0.                                                        OUTP0680
      TN = 0.                                                        OUTP0690
      L3 = XK(10)                                                    OUTP0700
      PS = 0.                                                        OUTP0710
     CPS = 1.                                                        OUTP0720
     SPS = 0.                                                        OUTP0730
```

Figure 68. Subroutine ØUTPUT Listing (Continued)

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

```
      IF (ND .EQ. 2) GO TO 50                                          OUTP0740
      L3 = XK(14)                                                      OUTP0750
   50 DO 270  I=1,JMAX                                                 OUTP0760
       R = XKS(1,I)                                                    OUTP0770
      PH = XKS(2,I)                                                    OUTP0780
      RH = XKS(3,I)                                                    OUTP0790
       Y = XKS(4,I)                                                    OUTP0800
      XM = XKS(5,I)                                                    OUTP0810
      L1 = XKS(6,I)                                                    OUTP0820
      YK = XKS(7,I)                                                    OUTP0830
      L4 = XKS(8,I)                                                    OUTP0840
      L6 = XKS(9,I)                                                    OUTP0850
      IF (ND .EQ. 2) GO TO 70                                          OUTP0860
       TH = XKS(10,I)                                                  OUTP0870
      CTH = DCOS(TH)                                                   OUTP0880
      STH = DSIN(TH)                                                   OUTP0890
       TN = STH/CTH                                                    OUTP0900
       X = XKS(11,I)                                                   OUTP0910
       L2 = XKS(12,I)                                                  OUTP0920
       L5 = XKS(13,I)                                                  OUTP0930
C     STEERING ANGLE                                                   OUTP0940
   70 DNU = DSQRT((R*L4*CTH)**2 + L6**2)                               OUTP0950
      CNU = R*L4*CTH/DNU                                               OUTP0960
      SNU = L6/DNU                                                     OUTP0970
      GNU = DATAN2(SNU,CNU)*RAD                                        OUTP0980
      XLST(15,I) = GNU                                                 OUTP0990
      D = DNU                                                          OUTP1000
      IF (ND.EQ.2) GO TO 201                                           OUTP1010
      D = DSQRT(DNU**2+ (L5*CTH)**2)                                   OUTP1020
      XLP = D / R / CTH                                                OUTP1030
      SPS = L5*CTH/D                                                   OUTP1040
      CPS = DNU/D                                                      OUTP1050
      PS = DATAN2(SPS,CPS)*RAD                                         OUTP1060
       GO TO 203                                                       OUTP1070
  201   XLP = D/R                                                      OUTP1080
        PS = 0.00                                                      OUTP1090
  203 IF ( ITER .NE. 5 ) GO TO 205                                     OUTP1100
      STO(I, 14 ,1) = SNGL(XLP)                                        OUTP1110
      STO(I, 15 ,1) = SNGL(GNU)                                        OUTP1120
      STO(I, 16 ,1) = SNGL( PS)                                        OUTP1130
```

Figure 68. Subroutine ØUTPUT Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
C     FIRST INTEGRAL                                                          OUTP1140
  205 IF (YK) 215,215,211                                                     OUTP1150
  211 BETA = BETAS                                                            OUTP1160
      GO TO 220                                                               OUTP1170
  215 BETA = 0.D0                                                             OUTP1180
  220 BETAX = BETA * C                                                        OUTP1190
      FDOT = DSQRT(X**2 + (Y*CTH)**2)                                         OUTP1200
      DRHO = R*(FDOT**2) - XMU/R**2 + BETAX*CPS*CNU/XM                        OUTP1210
      DX = -2.*RH*X/R - CTH*STH*Y**2 + BETAX*SPS/(R*XM)                       OUTP1220
      DY =  2.*X*Y*TN - 2.*RH*Y/R + BETAX*CPS*SNU/(R*CTH*XM)                  OUTP1230
      ENER =    (.5D0*((R*FDOT)**2 + RH**2) - XMU/R)                          OUTP1240
      L7 = -YK + D*C/(XM*R*CTH)                                               OUTP1250
      FINT = L1*RH + L2*X + L3*Y + L4*DRHO + L5*DX + L6*DY - L7*BETA          OUTP1260
      ANGM =    FDOT*R**2                                                     OUTP1270
      HBAR = HBAR + FINT * .01                                                OUTP1280
      IF ( ITER .NE. 1 ) GO TO 239                                            OUTP1290
      IF ( N(4) .GT. 0 ) GO TO 239                                            OUTP1300
      DO 236  KS=1,6                                                          OUTP1310
      IF ( DABS(VARIND(I) - TSUB(KS) ) .LT. 1.D-6 ) GO TO 239                 OUTP1320
  236 CONTINUE                                                                OUTP1330
      GO TO 270                                                               OUTP1340
  239 WRITE (6,20) VARIND(I)                                                  OUTP1350
  240 WRITE (6,250) XLP    , YK, PS, GNU, FINT, ENER, ANGM                    OUTP1360
      STO(I, 14 ,2) = SNGL(XLP)                                               OUTP1370
      STO(I, 15 ,2) = SNGL(GNU)                                               OUTP1380
      STO(I, 16 ,2) = SNGL( PS)                                               OUTP1390
  270 CONTINUE                                                                OUTP1400
      VCHAR = C * DLOG(XKS(5,1)/XKS(5,100) ) * CONV                           OUTP1410
      ITNO = ITNO + 1                                                         OUTP1420
      WRITE (6,920) HBAR, VCHAR                                               OUTP1430
C     ************************************************************************OUTP1440
      GO TO (526,501,501,526,501 ,501 , 501  )  ,  ITER                       OUTP1450
  501 WRITE (6,502)                                                           OUTP1460
      WRITE (6,22)                                                            OUTP1470
      DO 505  I=1,JMAX                                                        OUTP1480
  505 WRITE (6,21) (VARIND(I), (XKS(J,I), J=1,7))                             OUTP1490
      IF (ND .EQ. 3) GO TO 520                                                OUTP1500
```

Figure 68. Subroutine ØUTPUT Listing (Continued)

```
      WRITE (6,23)                                                          OUTP1510
      DO 510  I=1,JMAX                                                      OUTP1520
  510 WRITE (6,21) (VARIND(I), (XKS(J,I), J=8,9), XK(10))                   OUTP1530
      GO TO 526                                                             OUTP1540
  520 WRITE (6,24)                                                          OUTP1550
      DO 525  I=1,JMAX                                                      OUTP1560
  525 WRITE (6,21) (VARIND(I), (XKS(J,I), J=8,13), XK(14))                  OUTP1570
  526  WRITE(6,26)  (XKS(K,1), K = 1,N1 ), L3  ,  L7                        OUTP1580
      WRITE (6,290) ITNO                                                    OUTP1590
      IF( ITER .EQ.  1  )   CALL KHIST( 2, XKS , N1 , JMAX )                OUTP1600
      IF( ITER .EQ.  4  )   WRITE(6,27)                                     OUTP1610
      IF( ITER .EQ.  4  )   ITNO = 0                                        OUTP1620
      IF( ITER .EQ.  6  )   ITNO = 0                                        OUTP1630
      IF( ITER .EQ.  6  )   XNOCOR = 0.                                     OUTP1640
      IF( ITER .EQ.  7  )   ITER = 0                                        OUTP1650
      IF( ITER .EQ.  7  )   IQCYC = 1                                       OUTP1660
 1000 RETURN                                                                OUTP1670
      END                                                                   OUTP1680
```

Figure 68. Subroutine ØUTPUT Listing (Continued)

```
      SUBROUTINE KHIST(IGO,XKS, N1, JMAX )                              KHST0000
C     ------------------------------------------------------------------KHST0005
C     AT THE END OF EACH QUASILINEARIZATION ITERATION IT IS NECESSARY   KHST0010
C     TO EXAMINE THE SWITCHING FUNCTION AND MAKE APPROPRIATE ADJUSTMENTSKHST0015
C     TO THE THRUST INITIATION AND TERMINATION TIMES.  THIS SUBROUTINE  KHST0020
C     PERFORMS THESE ADJUSTMENTS THROUGH THE JUDICIOUS USE OF CONTROL   KHST0025
C     PARAMETERS SUPPLIED AS INPUT DATA.                                KHST0026
C     ------------------------------------------------------------------KHST0030
C     THREE DIM. VERSION--IBM 360--MARCH 1967                           KHST0040
      IMPLICIT REAL*8(A-H , O-Z)                                        KHST0050
      LOGICAL  RATES, REINT, CRTOUT                                     KHST0060
      REAL*8 IA                                                         KHST0070
      REAL*4 XLST,EPSP,EPSP1,RHOS,XERR,DYN,RESET,BL156, TWOBRN,TRU      KHST0080
      DIMENSION TZRO(8), TOL(4),DKSAV(4),TSTOP(5)                       KHST0090
     *, STOPS(7,40), XKS(N1,JMAX), XDOT(8)                              KHST0100
      COMMON /JACSET/ ID                                                KHST0110
      COMMON /ZI/ TX,H,PAD(5),DTPRNT,XK(20)                             KHST0120
      COMMON /ZX/ XI(10),EPSP,EPSP1,IA(20)                              KHST0130
      COMMON /ZSUB/          TSUB(10),VARIND(100),XMNSTP(9),STPSIZ(9),  KHST0140
     *KSUB(10) , MAXSTP                                                 KHST0150
      COMMON/ZOO/N(5), MX- XERR,DYN,B(10),L(10),M(10),MI(10),RHOS(20)   KHST0160
      COMMON /SOLN/ XLST(15,100), TRU(100), HBAR   ,VCHAR               KHST0170
      COMMON /ZIC/BETA,  C , GO ,XMU,TW,XXK(20)                         KHST0180
        COMMON/ZCONST/ PI,RAD,GXM,BETAS                                 KHST0190
C     INPUT DATA COMMON BLOCK                                           KHST0200
C     COMMON FOR STEEP DESCENT(FIRST SIX CARDS)                         KHST0210
      COMMON /INDATA/          XORB1(6),  XORB2(6),   AEST(6)           KHST0220
     A, AMPINC   , AMPDEC   , FREEZE   , XAMP     , SAVE     , BL46     KHST0230
     B, SCLINC   , SCLDEC   , ISCL     , IAMP     , BL55     , BL56     KHST0240
     C, SCALE(3)                       , TOLCOM   , MAXIT    , IPRIN    KHST0250
C     COMMON FOR GASLIN ORB. TRAN. PROG.--( TWO DIM.CASE)               KHST0260
     D, XIN(14),  XFN(14),  RATES, IDIMEN                               KHST0270
     I, BETASV  ,  XISP     ,  TFPERT ,  PERT    ,  COASTI   ,  BL126  KHST0280
     J, BRNINC  ,  BMAX1    ,  BMAX2  ,  BSH1X   ,  BSH2X    ,  TWOBRN KHST0290
     K, RESET   ,  KMULT    ,   NOPRT ,  PRNJ    ,  BCHTOL   ,  BL146  KHST0300
     L, INTSTP  ,  REINT    ,  CRTOUT ,  ITSQL   , IQPRIN    ,  BL156  KHST0310
     M, HBARTL  ,  RHOS1                                                KHST0320
      COMMON /STOP1/ XKDOT(8,0), TWOBR , XNOCOR, SAVK(7,6),IQCYC        KHST0330
C     ******************************************************************KHST0340
```

Figure 69. Subroutine KHIST Listing

```
      WRITE (6,2) IGO                                                    KHST0350
    2 FORMAT (25HIENTER KHIST --- IGO =      I5)                         KHST0360
C     IGO = 1 --- INITIALIZATION                                         KHST0370
C     IGO = 2 --- BURN POINT ANALYSIS                                    KHST0380
C     IGO = 3 --- RESET AT THE END OF A QUASLIN CYCLE--CALL FROM MAIN    KHST0390
      GO TO ( 1,100,17 ) , IGO                                           KHST0400
C     ***************************************************************** KHST0410
C     PERFORM INITIALIZATION                                             KHST0420
C     ***************************************************************** KHST0430
    1 DO 3  I=1,4                                                        KHST0440
      IK = KSUB(I+1)                                                     KHST0450
      IF ( I .EQ. 2 .OR. I .EQ. 4 ) IK = IK + 1                          KHST0460
    3 XKDOT(I) = (XKS(7,IK) - XKS(7,IK-1)) / (VARIND(IK) - VARIND(IK-1))KHST0470
      XDOT(2) = .1D0                                                     KHST0480
      DO 4  I=1,7                                                        KHST0490
      DO 4  J=1,6                                                        KHST0500
    4 SAVK(I,J) = 0.                                                     KHST0510
      DO 7  I=1,8                                                        KHST0520
    7  TZRO(I) = 0.                                                     KHST0530
      DO 8  I=1,40                                                       KHST0540
      DO 8  J=1,7                                                        KHST0550
    8 STOPS(J,I) = 0.                                                    KHST0560
      ILST = 0                                                           KHST0570
      IL ST1 = 0                                                         KHST0580
      IS = MAXSTP                                                        KHST0590
C     COMPUTE BURN TOLERANCES                                            KHST0600
      TOL(1)= BMAX1     *     ( TSUB(3)-TSUB(2)) / 100.                  KHST0610
      TOL(2)=  TOL(1)                                                    KHST0620
      TOL(3)= BMAX2     *      (TSUB(5)-TSUB(4) )/100.                   KHST0630
      TOL(4)=  TOL(3)                                                    KHST0640
      BSH1 = BSH1X * ( TSUB(3) - TSUB(2) ) / 100.D0                      KHST0645
      BSH2 = BSH2X * ( TSUB(5) - TSUB(4) ) / 100.D0                      KHST0647
      GO TO 19                                                           KHST0650
```

Figure 69. Subroutine KHIST Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
C     ************************************************************************KHST0660
C     MAIN LOOP TO PERFORM SWITCH POINT ANALYSIS BEGINS HERE                    KHST0670
C     ************************************************************************KHST0680
                                                                                KHST0690
  100 ILST = ILST + 1                                                           KHST0700
      ILST1 = ILST1 + 1                                                         KHST0710
  102 IZRO=  1                                                                  KHST0720
       DO 998 I = 1,8                                                           KHST0730
  998 TZRO(I) = 0.                                                              KHST0740
C     COMPUTE SWITCHING FUNCTION DERIVATIVES --ONE TO FOUR SWITCH POINTS        KHST0750
      DO 98 I=1,4                                                               KHST0760
      SAVK(1,I) = SAVK(1,I)+ SAVK(2,I)*IA(1) +    SAVK(3,I)*IA(2)               KHST0770
     *      +                 SAVK(4,I)*IA(3) +    SAVK(5,I)*IA(4)             KHST0780
      IF ( ID .EQ. 2 ) GO TO 98                                                 KHST0790
      SAVK(1,I) = SAVK(1,I) +SAVK(6,I)*IA(5) +    SAVK(7,I)*IA(6)               KHST0795
   98 CONTINUE                                                                  KHST0800
C     CHECK TO SEE IF  K  IS POSITIVE INITIALLY                                 KHST0810
  110 SK1    =DSIGN(1.D0,       XKS(7,1))                                       KHST0820
      IF(SK1)112,112,111                                                        KHST0830
  111 TZRO(1) = 0.                                                              KHST0840
      XKDOT( 1     ) =         XDOT( 2     )                                    KHST0850
      TSUB(2) = TSUB(2) -          TOL(1)    * 2.D0                             KHST0860
      IZRO    = 2                                                               KHST0870
C     ************************************************************************KHST0880
C     LOOP TO FIND ZEROS OF THE SWITCHING FUNCTION                              KHST0890
C     ************************************************************************KHST0900
  112 DO 119 I=2,IS                                                             KHST0910
      SK2 =DSIGN (1.D0,       XKS(7,I) )                                        KHST0920
C     ASK IF SWITCHING FUNCTION CHANGES SIGN                                    KHST0930
      IF( SK2 + SK1 )119,113,119                                                KHST0940
C     IF SIGN CHANGES COMPUTE INTERCEPT BY LINEAR INTERPOLATION                 KHST0950
  113 X DOT(IZRO) = ( XKS(7,I) - XKS(7,I-1)  ) / (VARIND(I)-VARIND(I-1))        KHST0960
      TZRO(IZRO) =-XKS(7,I-1) / X DOT(IZRO)  + VARIND(I-1)                      KHST0970
      XKDOT( IZRO ) =         XDOT( IZRO )                                      KHST0980
      IZRO = IZRO + 1                                                           KHST0990
      IF ( IZRO .GT. 8 ) GO TO 120                                              KHST1000
  119 SK1 = SK2                                                                 KHST1010
C     CHECK TO SEE IF LAST K  VALUE IS POSITIVE                                 KHST1020
  120 IF( XKS(7,100) .LT. 0. ) GO TO 160                                        KHST1030
      TZRO(IZRO) =        TSUB(6)    - 1.
```

Figure 69. Subroutine KHIST Listing (Continued)

```
C     ************************************************************************KHST1040
C     LOOK AT THE FOUR ORIGINAL BURN TIMES AND MAKE NECESSARY CHANGES   KHST1050
C     ************************************************************************KHST1060
  160 IF ( TWOBRN .NE. 1. ) GO TO 170                                     KHST1062
      IF ( TZRO(3) .NE. 0. ) GO TO 170                                    KHST1064
      TZRO(3) = TZRO(1)                                                   KHST1066
      TZRO(4) = TZRO(2)                                                   KHST1068
      TZRO(1) = 0.                                                        KHST1070
      TZRO(2) = 0.                                                        KHST1072
      XKDOT(3) = XKDOT(1)                                                 KHST1074
      XKDOT(4) = XKDOT(2)                                                 KHST1076
  170 DO 1400 JJ =1,4                                                     KHST1078
      IF( TZRO (JJ) .EQ. 0.) GO TO 1400                                   KHST1080
      JJX = KSUB( JJ + 1 )                                                KHST1090
      TEST =       TZRO(JJ) -       VARIND(JJX)                          KHST1100
      IF(DABS( TEST)  .GT. TOL(JJ) )  GO TO 1399                          KHST1110
      XKDOT(JJ) =       SAVK(1,JJ)                                        KHST1120
 1350 TSUB(JJ+1) = TSUB(JJ+1) - XKS(7,JJX)/     SAVK(1,JJ)               KHST1130
      GO TO 1400                                                          KHST1140
 1399 TSUB(JJ+1)=TSUB(JJ+1)+TOL(JJ) *DSIGN(1.D0, TEST )                   KHST1150
 1400 CONTINUE                                                            KHST1160
      IF( TZRO(2) .EQ. 0. ) GO TO 1550                                    KHST1170
      TEST = ( TZRO(1) + TZRO(2) ) / 2. -      (TSUB(2)+TSUB(3))/ 2.      KHST1180
      IF(DABS( TEST ) .LT. BSH1       ) GO TO 1550                        KHST1190
      TMOVE = BSH1       *DSIGN(1.D0,TEST)                                KHST1200
      TSUB(2) = TSUB(2) + TMOVE                                           KHST1210
      TSUB(3) = TSUB(3) + TMOVE                                           KHST1220
 1550 IF( TZRO(4) .EQ. 0. ) GO TO 1890                                    KHST1230
      TEST = ( TZRO(3) + TZRO(4) ) / 2. -     ( TSUB(4)+TSUB(5))/ 2.      KHST1240
      IF(DABS( TEST ) .LT. BSH2       ) GO TO 1890                        KHST1250
      TMOVE = BSH2       *DSIGN(1.D0,TEST)                                KHST1260
      TSUB(4) = TSUB(4) + TMOVE                                           KHST1270
      TSUB(5) = TSUB(5) + TMOVE                                           KHST1280
 1890 IF( TSUB(3) - TSUB(2) .GE. 0.D0 ) GO TO 1895                        KHST1290
      TSUB(2) =       STOPS(1,JL)                                         KHST1300
      TSUB(3) =       STOPS(2,JL)                                         KHST1310
 1895 IF( TSUB(5) - TSUB(4) .GE. 0.D0 ) GO TO 1900                        KHST1320
```

Figure 69. Subroutine KHIST Listing (Continued)

```
      TSUB(4) =        STOPS(3,JL)                                          KHST1330
      TSUB(5) =        STOPS(4,JL)                                          KHST1340
C     IF TIME ERROR SUFFICIENTLY LARGE USE PREVIOUS VALUE                   KHST1350
 1900 DO 3005 J=2,6                                                         KHST1360
      DTM = (TSUB(J) - TSUB(J-1)) /     ( FLOAT( KSUB(J) - KSUB(J-1)) )    KHST1370
      K1 = KSUB(J-1)+1                                                      KHST1380
      K2 = KSUB(J)                                                          KHST1390
      DO 3005 I =K1,K2                                                      KHST1400
 3005 VARIND(I)  = VARIND(I-1) + DTM                                        KHST1410
      GO TO 19                                                              KHST1420
C     ********************************************************************KHST1430
C     PRINT STOP TIME SUMMARY TABLE AND RESET                               KHST1440
C     ********************************************************************KHST1450
   17 ILST1 = ILST1 + 2                                                     KHST1460
   18 ILST = 0                                                              KHST1470
   19 JL = ILST1+ 1                                                         KHST1480
      DO 14 I=1,5                                                           KHST1490
   14 STOPS(I,JL) = TSUB(I+1)                                               KHST1500
      STOPS(6,JL)  = HBAR                                                   KHST1510
      STOPS(7,JL)  = VCHAR                                                  KHST1520
      WRITE(6,20)                                                           KHST1530
   20 FORMAT ('0STOPPING CONDITIONS'/12X, 2HON, 14X, 3HOFF, 13X, 2HON,      KHST1540
     *  14X, 3HOFF, 13X, 3HEND,12X,  4HHBAR,  12X,  5HVCHAR  )              KHST1550
      DO 23 J=1,JL                                                          KHST1560
      JJ=J - 1                                                              KHST1570
      IF (STOPS(1,J) .NE. 0. ) GO TO 21                                     KHST1580
      WRITE (6,9022)                                                        KHST1590
9022 FORMAT ('0NEW QASLIN ITERATION SET -- FINAL TIME CHANGED.')            KHST1600
      GO TO 23                                                              KHST1610
   21 WRITE(6,9021) JJ,(STOPS(I,J),I=1,7 )                                  KHST1620
9021 FORMAT( I5,1P7D16.8  )                                                 KHST1630
   23 CONTINUE                                                              KHST1640
      WRITE (6,26) (TZRO(I), I=1,4)                                         KHST1650
   26 FORMAT (38H STOP POINTS FOUND FOR LAST ITERATION /5X,1P4D16.8    )    KHST1660
      RETURN                                                                KHST1670
      END                                                                   KHST1680
```

Figure 69. Subroutine KHIST Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
                SUBROUTINE QPLOT ( XKS, N1, JMAX, VARIND )                      QPL00000
C       ------------------------------------------------------------------------QPL00001
C       ALLOWS THE PROGRAMMER TO SELECT PLOTTED OUTPUT OF THE INITIAL           QPL00002
C       ITERATE AND CONVERGED SOLUTION FOR EACH VARIABLE.  TO PLOT THE          QPL00003
C       GRAPHS IT CALLS SUBROUTINE PICTUR.                                      QPL00004
C       ------------------------------------------------------------------------QPL00005
        REAL*8 XKS, VARIND, HBAR, VCHAR                                         QPL00010
        DIMENSION X(100), XKS(N1,JMAX), VARIND(100)                             QPL00020
       DIMENSION XLAB(3), TITLE(6), YLAB(3,20)                                  QPL00030
       COMMON /SOLN/ XLST(15,100), TRU(100), HBAR, VCHAR, STO(100,20,2)         QPL00040
       DATA  XLAB/'    TIME    '/ , TITLE/'QUAS.LIN.ORBIT TRANSFER '/,YLAQPL00050
      *B/   '      R            PHI           RHO           Y             M     QPL00060
      *   LAM1            K           LAM4         LAM6         TH           XQPL00070
      *          LAM2        LAM5          PRIMER   STEER-NU    STEER-PSI     QPL00080
      *                                                '/                      QPL00090
        DO 10  I=1,JMAX                                                         QPL00100
     10 X(I) = SNGL(VARIND(I))                                                  QPL00110
C       COMPARE INITIAL AND FINAL SOLN                                          QPL00120
        DO 100  J=1,N1                                                          QPL00130
        DO 100  I=1,JMAX                                                        QPL00140
        STO(I,J,1) = XLST(J,I)                                                  QPL00150
    100 STO(I,J,2) = SNGL( XKS(J,I) )                                           QPL00160
        DO 200  J=1,16                                                          QPL00170
        IF ( N1 .EQ. 13 ) GO TO 150                                             QPL00180
        IF ( ( J .GT. 9 .AND. J .LT. 14 ) .OR. J .EQ. 16 ) GO TO 200            QPL00190
    150 CALL PICTUR( 1, X, STO(1,J,2), 7., 7., 1, 1, 100, XLAB, YLAB(1,J),QPL00200
       *          1, TITLE, XB, XT, YB, YT,50, 1 )                            QPLC0210
        CALL PICTUR( 3, X, STO(1,J,1), 7., 7., 1, 1, 100, XLAB, YLAB(1,J),QPL00220
       *          1, TITLE, XB, XT, YB, YT,48, 2 )                            QPL00230
        IF(  J .NE.  7 .AND.  J  .NE.  14  )   GO TO 200                        QPL00240
        CALL PICTUR( 1, X, STO(1,J,1), 7., 7., 1, 1, 100, XLAB, YLAB(1,J),QPL00250
       *          1, TITLE, XB, XT, YB, YT,50, 1 )                            QPL00260
    200 CONTINUE                                                                QPL00270
        RETURN                                                                  QPL00280
        END                                                                     QPL00290
```

Figure 70. Subroutine QPLØT Listing

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
      SUBROUTINE PICTUR ( ISET, X, Y, DIMX, DIMY, ISKIP, NI, NF, XLAB,    PICT0000
     *     YLAB  ,  IYG  ,    TITLE, XB, XT, YB, YT, IC   , ITYPE )       PICT0010
C     ------------------------------------------------------------------PICT0012
C                                                                         PICT0014
C     THIS IS A STANDARD SUBROUTINE FOR PLOTTING SEVERAL KINDS OF DATA    PICT0016
C     USING THE SC-4020 CRT OUTPUT DEVICE.                                PICT0018
C     ------------------------------------------------------------------PICT0020
      DIMENSION  X(1  ), Y(1  ),XLAB(3),YLAB(3,72) , TITLE(6)             PICT0030
C     ISET = 1       GENERATE NEW GRID.....SCALE TO DATA                  PICT0040
C     ISET = 2       GENERATE NEW GRID.....USE DIMENSIONS SUPPLIED        PICT0050
C     ISET = 3       PLOT NEW DATA ON EXISTING GRID                       PICT0060
      DIMENSION ICH(50)                                                   PICT0070
      IF(ICH(1).EQ.1) GO TO 4                                             PICT0080
    2 DO 1  K=1,9                                                         PICT0090
      ICH(K)    = K                                                       PICT0100
      ICH(K+9 ) = K + 16                                                  PICT0110
      ICH(K+18) = K + 32                                                  PICT0120
      ICH(  K  +  36  )  =  K  +  9                                       PICT0130
    1 ICH(K+27) = K + 49                                                  PICT0140
      ICH(46)  =  42                                                      PICT0150
      ICH(47)  =  63                                                      PICT0160
      ICH(48)  =  16                                                      PICT0170
      ICH(49)  =  44                                                      PICT0180
      ICH(50)  =  48                                                      PICT0190
    4 NOP  =  NF                                                          PICT0200
      IFIRST  =  NI                                                       PICT0210
    3 GO TO (5,5,75),ISET                                                 PICT0220
    5 ML=16                                                               PICT0230
      MB=32                                                               PICT0240
      MR=0                                                                PICT0250
      MT=22                                                               PICT0260
      MAPH= (134.*DIMX) +.5                                               PICT0270
      MAPV= (134.*DIMY) +.5                                               PICT0280
      MR1= 1024 - MAPH                                                    PICT0290
      MT1= 1024 - MAPV                                                    PICT0300
      MR = MAXO(MR,MR1)                                                   PICT0310
      MT = MAXO(MT,MT1)
```

Figure 71. Subroutine PICTUR Listing

```
      CALL FRAMEV                                                        PICT0320
      GO TO (15,37,37),ISET                                              PICT0330
   15 YL = Y(IFIRST)                                                     PICT0340
      YR= Y(IFIRST)                                                      PICT0350
      XL= X(IFIRST)                                                      PICT0360
      XR= X(IFIRST)                                                      PICT0370
       DO 35 J=IFIRST,NOP,ISKIP                                          PICT0380
      XL= AMIN1(XL,X(J))                                                 PICT0390
      XR= AMAX1(XR,X(J))                                                 PICT0400
      YL=AMIN1(YL,Y(J))                                                  PICT0410
      YR=AMAX1(YR,Y(J))                                                  PICT0420
   35  ITOP = J                                                          PICT0430
       YL = YL -(YR-YL)/10.                                              PICT0440
       YR = YR + (YR-YL)/10.                                             PICT0450
       XL = XL -(XR-XL)/10.                                              PICT0460
       XP = XR + (XR-XL)/10.                                             PICT0470
      GO TO 39                                                           PICT0480
   37 XL=XB                                                              PICT0490
      XR=XT                                                              PICT0500
      YL=YB                                                              PICT0510
      YR=YT                                                              PICT0520
   39 CALL SETMIV(ML,MR,MB,MT)                                           PICT0530
   55 CALL DXDVV (1,XL        ,XR      ,DX,N,I,NX,25.,IERR)              PICT0540
      CALL DXDVV (2,YL        ,YP      ,DY,M,J,NY,25.,IERR)              PICT0550
      CALL GRIDIV(2,XL,XR,YL,YR,DX,DY, N,M,I,J,NX,NY)                    PICT0560
   60 IY=1023-MT-32                                                      PICT0570
      IX=   1023-MR-20                                                   PICT0580
      CALL PRINTV ( 24,TITLE,400,8)                                      PICT0590
      CALL PRINTV ( 12,XLAB ,450,24)                                     PICT0600
      CALL APRNTV (0,-16, 12,YLAB(1,IYG)  ,10  ,500  )                   PICT0610
```

Figure 71. Subroutine PICTUR Listing (Continued)

```
 75 IPSI = NOP - 1                                                          PICT0620
    GO TO (78,95), ITYPE                                                    PICT0630
 78  CONTINUE                                                               PICT0640
    DO 80 J = IFIRST,IPSI,ISKIP                                             PICT0650
    K= J + ISKIP                                                            PICT0660
    DO 80    JX  =  1 , 2                                                  PICT0670
 80 CALL LINEV (NXV(X(J)),NYV(Y(J)),NXV(X(K)),NYV(Y(K)))                    PICT0680
    DO 81  J=IFIRST,IPSI,4                                                  PICT0690
 81 CALL PLOTV ( NXV(X(J)),(NYV(Y(J))  +  5 )  ,ICH(IC)     )               PICT0700
    RETURN                                                                  PICT0710
 95 DO 100 J= IFIRST,NOP,ISKIP                                              PICT0720
    DO 100  JX  =  1 , 2                                                    PICT0730
100 CALL PLOTV ( NXV(X(J)), NYV(Y(J)), ICH(IC)      )                       PICT0740
    RETURN                                                                  PICT0750
       END                                                                  PICT0760
```

Figure 71. Subroutine PICTUR Listing (Continued)

SPACE DIVISION of NORTH AMERICAN ROCKWELL CORPORATION

```
      DOUBLE PRECISION FUNCTION DTRUEH(YM, E, CA, SSF, CSF, IFLAG,JFLAG)DTRU0010
C     ------------------------------------------------------------------DTRU0020
C     THIS PROGRAM CONVERGES FOR ALL VALUES OF ECCENTRICITY.  THE MEAN  DTRU0030
C     ANOMALY AGREEMENT IS TO 1.D-14 RADIANS FOR ELLIPTICAL ORBITS      DTRU0040
C     WITHIN 4 ITERS AND HYPERBOLIC ORBITS  WITH A MAX NO. OF 7 ITERS.  DTRU0060
C     THIS VERSION GIVES IFLAG = -2 IF HYPERBOLIC SECT DOES NOT         DTRU0070
C      CONVERGE IN 15 ITERATIONS                                        DTRU0080
C     ------------------------------------------------------------------DTRU0082
C     WRITTEN MARCH 1966 BY D.F.BENDER - SPACE SCIENCES LABORATORY-SID  DTRU0088
      DOUBLE PRECISION YM,XM,ZM,E,CA,SSF,CSF,DTOR,RTOD,PI,PI2,XMTL,TST, DTRU0090
     *SGN, H,DH,DYDH,TFH, C,S,S2,S4,EC,E2,XM1,EAH,A,B,CC,SE,SIG,EC2,THFHDTRU0100
     *,EXF1,SHF1,CHF1,DFDM,XLF                                          DTRU0110
      DATA XMTL,PI,PI2/1.0D-14, 3.1415926535898D0,6.283185307179600    / DTRU0120
      DATA DTOR,RTOD/1.7453292519943290-02, 57.295779513082400 /        DTRU0130
C                                                                       DTRU0140
        IFLAG = 1                                                       DTRU0150
        JFLAG =1                                                        DTRU0160
        SGN = 1.0D0                                                     DTRU0170
        IF(E.GE.1.0D0) IFLAG=2                                          DTRU0180
        GO TO(15,20),IFLAG                                              DTRU0190
   15     ZM  =  YM   -    DBLE(AINT(SNGL(YM/PI2))) * PI2               DTRU0200
        IF(ZM.LT.-PI)  ZM=ZM+PI2                                        DTRU0210
        IF(ZM.GT. PI)  ZM=ZM-PI2                                        DTRU0220
C     THIS SETS ZM IN RANGE -PI TO +PI FOR ELLIPTIC ORBITS              DTRU0230
        IF(ZM.LT.0.0D0) SGN =-1.0D0                                     DTRU0240
   18   XM = ZM*SGN                                                     DTRU0250
C     XM IS ALWAYS POSITIVE                                             DTRU0260
        GO TO 25                                                        DTRU0270
   20   IF(YM.LT.0.0D0) SGN =-1.0D0                                     DTRU0280
        XM = YM*SGN                                                     DTRU0290
C     FIRST WE CHECK FOR PARABOLIC CASE  IF ABS(E-1) LESS THAN 1.D-10   DTRU0300
   25   IF(DABS(E-1.D0).LT.1.0D-10) JFLAG=2                             DTRU0310
        GO TO(30,40),JFLAG                                              DTRU0320
   30   EC = DSQRT((1.D0+E)/DABS(1.D0-E))                               DTRU0330
        EC2 = DABS(1.D0-E*E)                                            DTRU0340
   35   IF(E.GT.2.0D0) GO TO 400                                        DTRU0350
        IF(E.LT..98D0) GO TO 300                                        DTRU0360
        IF(E.LT.1.0D0.AND.XM.GT..15700) GO TO 300                       DTRU0370
```

Figure 72. Function DTRUEH Listing

```
C     ECCENTRICITY IS NEAR 1.0, USE PARABOLIC APPROXIMATION                  DTRU0380
C                                                                            DTRU0390
        H = 2.DO* XM/ DSQRT(EC2**3)                                          DTRU0400
        GO TO 50                                                             DTRU0410
C     FOR THE PARABOLIC CASE YM IS TAKEN TO BE = T*SQRT(GXM/P**3)            DTRU0420
   40   H = XM * 2.0DO                                                       DTRU0430
C     FIRST APRROX. TO CUBIC DEPENDS ON SIZE OF H.                           DTPU0440
   50   IF(H.GT.3.DO) GO TO 65                                               DTRU0450
   60   TFH = H/(1.DO + H*H/3.DO)                                            DTRU0460
        GO TO 70                                                             DTRU0470
   65   TFH = (3.DO*H)**.33333                                               DTRU0480
   70   NK = 5 * JFLAG                                                       DTRU0490
C     FOR THE PARABOLIC CASE WE DOUBLE THE NUMBER OF ITERATIONS              DTRU0500
        DO 75 K=1,NK                                                         DTRU0510
        DYDH = 1.DO /(1.DO +TFH**2)                                          DTRU0520
        DH = H - TFH*(1.DO + TFH**2/3.DO)                                    DTRU0530
   75   TFH = TFH +DH*DYDH*(1.DO -DH*TFH* DYDH**2 )                          DTRU0540
        GO TO(100,90) ,JFLAG                                                 DTRU0550
C       FOR THE PARABOLIC CASE TFH IS ALL WE NEED                            DTRU0560
   90   DTRUEH = 2.DO*DATAN(TFH)                                             DTRU0570
        SSF = DSIN(DTRUEH)                                                   DTRU0580
        CSF = DCOS(DTRUEH)                                                   DTRU0590
        CA = 0.DO                                                            DTRU0600
        GO TO 1000                                                           DTRU0610
  100   GO TO(110,150), IFLAG                                                DTRU0620
  110   CA = 2.DO*DATAN(TFH/EC)                                              DTRU0630
        GO TO 350                                                            DTRU0640
  150   THFH = EC*TFH                                                        DTRU0650
        IF(THFH.GT..99D0) THFH =.98DO                                        DTRU0660
C     TANH(F/2) CANNOT BE LARGER THAN 1.0.  WE USE .98 ARBITRARILY IF ITDTRU0670
C     COMES OUT LARGER THAN .99                                              DTRU0680
        CA = .50DO * DLOG((1.DO+THFH)/ (1.DO-THFH) )                         DTRU0690
        GO TO 450                                                            DTRU0700
C     THIS COMPLETES FIRST APPROX. BY PARABOLIC FORMULAE                     DTRU0710
C
```

Figure 72. Function DTRUEH Listing (Continued)

```
C                                                                              DTRU0720
C      THE ELLIPTICAL ORBIT CASE                                               DTRU0730
  300    C = DCOS(XM)                                                          DTRU0740
         S = DSIN(XM)                                                          DTRU0750
         S2 = S*S                                                              DTRU0760
         S4 = S2*S2                                                            DTRU0770
         E2 = E*E                                                              DTRU0780
         CA = XM + E*S* (1.D0 + E*C + E2*(1.D0-1.5D0*S2) + E*E2*C*(1.D0-       DTRU0790
     *2.66666667D0*S2) + E2*E2*(1.D0-5.66666667D0*S2 + 5.20833333D0*S4))       DTRU0800
  350    DO 370 I=1,15                                                         DTRU0810
         SE = DSIN(CA)                                                         DTRU0820
         XM1 = CA - E*SE                                                       DTRU0830
         IF(DABS(XM-XM1).LT.XMTL) GO TO 380                                    DTRU0840
         SIG = 1.D0 - E*DCOS(CA)                                               DTRU0850
         A = (XM-XM1)/SIG                                                      DTRU0860
         B = A*E*SE/SIG                                                        DTRU0870
C      THIS B = A* B OF TEXT                                                   DTRU0880
         CC = (1.D0-SIG)/SIG                                                   DTRU0890
         ITN = I                                                               DTRU0900
  370    CA = CA + A* (1.D0-.5D0*B*(1.D0-B) - CC*A*A/6.D0 )                    DTRU0910
         WRITE(6,375) YM,XM,E,CA,XMTL                                          DTRU0920
  375    FORMAT(1HO,88HELLIPTICAL ORBIT SECTION DID NOT CONVERGE IN 15 ITDTRU0930
     *ERATIONS. CASE IS YM,XM,E,CA,XMTL         / 1P5D18.10  )                 DTRU0940
  380    EAH = CA/2.D0                                                         DTRU0950
         DTRUEH = 2.D0* DATAN(EC*DSIN(EAH)/DCOS(EAH)) *SGN                     DTRU0960
         SSF = DSIN(DTRUEH)                                                    DTRU0970
         CSF = DCOS(DTRUEH)                                                    DTRU0980
         GO TO 1000                                                            DTRU0990
C                                                                              DTRU1000
C      THE HYPERBOLIC ORBIT CASE                                               DTRU1010
  400    CA = XM/(E-1.D0 + E*(XM/(E-1.D0))**2/ 6.D0)                           DTRU1020
         XLF = 2.0D0 * XM / E                                                  DTRU1030
         IF(XLF.GE. 3.0D0) CA = DLOG(XLF)*(1.D0 + 1.D0/(XM-1.D0))              DTRU1040
C      THE ABOVE  ADDED FOR CASES WHEN XM IS LARGE AND  400 IS NOT GOOD        DTRU1050
```

Figure 72. Function DTRUEH Listing (Continued)

```
  450   DO 470I=1,15                                                          DTRU1060
        EXF1 = DEXP(CA)                                                       DTRU1070
        SHF1 = .5DO*(EXF1-1.DO/EXF1)                                          DTRU1080
        XM1 = E*SHF1 - CA                                                     DTRU1090
        IF(DABS(XM-XM1).LT.XMTL) GO TO 480                                    DTRU1100
        DFDM = 2.DO/(E*(EXF1+1.DO/EXF1) - 2.DO)                               DTRU1110
        A = (XM-XM1)*DFDM                                                     DTRU1120
        B = A*E*SHF1*DFDM                                                     DTRU1130
C     THIS B = A* B OF TEXT                                                   DTRU1140
        CC = DFDM + 1.0DO                                                     DTRU1150
        ITN= I                                                                DTRU1160
  470   CA = CA + A*(1.DO - .5DO *B* (1.DO-B)- A*A*CC/6.DO )                  DTRU1170
        WRITE(6,475) YM,XM,E,CA,A                                             DTRU1180
  475   FORMAT(1HO,88HHYPERBOLIC ORBIT SECTION DID NOT CONVERGE IN 15         DTRU1190
     *ITERATIONS.  THE CASE IS YM,XM,E,CA,A      / 1P5D18.10  )              DTRU1200
        IFLAG = -2                                                            DTRU1210
  480   CHF1 = .50DO*(EXF1+1.DO/EXF1)                                         DTRU1220
        SSF =  DSQRT(EC2)*SHF1/(E*CHF1 - 1.DO) *SGN                           DTRU1230
        CSF = (CHF1-E) / (1.DO - E*CHF1)                                      DTRU1240
        DTRUEH = DATAN2(SSF,CSF)                                              DTRU1250
C                                                                             DTRU1260
 1000   RETURN                                                                DTRU1270
        END                                                                   DTRU1280
```

Figure 72. Function DTRUEH Listing (Continued)

SPACE DIVISION OF NORTH AMERICAN ROCKWELL CORPORATION

```
      SUBROUTINE ROTATD(V,U,N,IAX,ANG)                                  ROT41010
C     ------------------------------------------------------------------ROT41011
CC    TAKES A GIVEN VECTOR IN CARTESIAN COORDINATES AND ROTATES IT IN A ROT41012
CC    COORDINATE SYSTEM OR ROTATES THE COORDINATE SYSTEM.  AS MANY AS   ROT41013
CC    TEN ROTATIONS ARE ALLOWED AND EACH ONE CALLS ROTMXD               ROT41014
CC    (ANGLE IN DEGREES).                                               ROT41015
C     ------------------------------------------------------------------ROT41016
      DOUBLE PRECISION V(3),U(3),ANG(10),VI(3), A(3,3)                  ROT41020
      DIMENSION IAX(10)                                                 ROT41025
                                                                        POT41030
CC       ROTATD EXPRESSES A VECTOR GIVEN AS U INITIALLY.   THE ANGLES   ROT41035
CC    ARE POSITIVE IF THE VECTOR IS ROTATED IN THE COORDINATE SYSTEM ANDROT41040
CC    THEY ARE NEGATIVE WHEN THE SYSTEM IS ROTATED. THE LIMIT IS TEN    ROT41045
CC    ROTATIONS.  THE ANGLES ARE IN DEGREES.                            ROT41050
C                                                                       ROT41055
        DO 20 I=1,3                                                     ROT41060
   20   VI(I) = U(I)                                                    ROT41065
        DO 100 K=1,N                                                    ROT41070
        CALL ROTMXD(A,IAX(K),ANG(K))                                    ROT41075
        DO 80 I=1,3                                                     ROT41080
   80   V(I) = A(I,1)*VI(1) + A(I,2)*VI(2) + A(I,3)*VI(3)               ROT41085
        DO 90 J=1,3                                                     ROT41090
   90   VI(J) = V(J)                                                    ROT41095
  100   CONTINUE                                                        ROT41100
  200   RETURN                                                          ROT41105
        END                                                             ROT41110
```

Figure 73. Subroutine ROTATD Listing

```
      SUBROUTINE ROTMXD(A,IAX,ANG)                                      ROMX1010
C     ------------------------------------------------------------------ROMX1015
C     ROTMXD LOADS MATRIX FOR SINGLE AXIS ROTATION - ANGLE IN DEGREES   ROMX1020
C     ------------------------------------------------------------------ROMX1021
C       D.F.BENDER, SPACE SCIENCES LAB.  EXT. 5893  DP FORM FEB. 1966   ROMX1025
C                                                                       ROMX1030
      DOUBLE PRECISION A(3,3),CA,SA ,DTOR                               ROMX1035
      DATA DTOR/.0174532925199433D0/                                    ROMX1040
C                                                                       ROMX1045
        DO 40 I=1,9                                                     ROMX1050
   40   A(I,1) = 0.D0                                                   ROMX1055
        CA = DCOS(ANG*DTOR)                                             ROMX1060
        SA = DSIN(ANG*DTOR)                                             ROMX1065
        GO TO(100,200,300), IAX                                         ROMX1070
C                                                                       ROMX1080
C     LOAD MATRIX FOR AXIS 1 ROTATION                                   ROMX1085
C                                                                       ROMX1090
C                                                                       ROMX1095
  100   A(1,1) = 1.D0                                                   ROMX1100
        A(2,2) = CA                                                     ROMX1105
        A(3,2) = SA                                                     ROMX1110
        A(2,3) =-SA                                                     ROMX1115
        A(3,3) = CA                                                     ROMX1120
        GO TO 500                                                       ROMX1125
C                                                                       ROMX1130
C     LOAD MATRIX FOR AXIS 2 ROTATION                                   ROMX1135
C                                                                       ROMX1140
C                                                                       ROMX1145
  200   A(1,1) = CA                                                     ROMX1150
        A(3,1) =-SA                                                     ROMX1155
        A(2,2) = 1.00                                                   ROMX1160
        A(1,3) = SA                                                     ROMX1165
        A(3,3) = CA
        GO TO 500
```

Figure 74. Subroutine RØTMXD Listing

```
C
C       LOAD MATRIX FOR AXIS 3 ROTATION                          ROMX1170
C                                                                ROMX1175
C                                                                ROMX1180
  300     A(1,1) = CA                                            ROMX1185
          A(2,1) = SA                                            ROMX1190
          A(1,2) =-SA                                            ROMX1195
          A(2,2) = CA                                            ROMX1200
          A(3,3) = 1.00                                          ROMX1205
C                                                                ROMX1210
  500     RETURN                                                 ROMX1215
          END                                                    ROMX1220
```

Figure 74. Subroutine RØTMXD Listing (Continued)

```
      SUBROUTINE DORBIT(  E ,  ZCAP , TORB  )                          DORB0000
C     ---------------------------------------------------------------DORB0002
CC    DETERMINES MEAN ANOMALY IN REVOLUTIONS FROM TRUE ANOMALY         DORB0004
C     (IN RADIANS) AND ECCENTRICITY FOR ELLIPTIC ORBITS.               DORB0006
C     ---------------------------------------------------------------DORB0008
      REAL*8 E, ZCAP, PI2, ECAP, TORB, EC, Z2                          DORB0010
      PI2=6.28318530718 D+00                                           DORB0020
    5 IF(ZCAP) 10,20,20                                                DORB0030
   10 ZCAP = ZCAP + PI2                                                DORB0040
      GO TO 5                                                          DORB0050
   20 EC = DSQRT( (1.D0-E) / (1.D0+E) )                                DORB0060
      Z2 = ZCAP / 2.D0                                                 DORB0070
      ECAP = 2.D0 * ( DATAN( EC * DSIN(Z2) / DCOS(Z2) ) )              DORB0080
   24 IF( ECAP ) 26,30,30                                              DORB0090
   26 ECAP = ECAP + PI2                                                DORB0100
   30 TORB = ( ECAP - E*DSIN(ECAP) )/   PI2                            DORB0110
   40 RETURN                                                           DORB0120
      END                                                              DORB0130
```

Figure 75. Subroutine DØRBIT Listing

```
      SUBROUTINE XIN (XKS,NU,IMAX )                                    XIN 0000
      DOUBLE PRECISION XKS(NU,IMAX)                                    XIN 0010
      RETURN                                                           XIN 0020
      END                                                              XIN 0030
```

Figure 76. Subroutine XIN Listing

```
      SUBROUTINE ALGBC ( XKS, G, XJAC )                                ALGB0000
      DOUBLE PRECISION  XKS, G(14), XJAC(13,14)                        ALGB0010
            DO 130 J = 1, 14                                           ALGB0020
            DO 120 I = 1, 13                                           ALGB0030
  120 XJAC(I,J) = 0.00                                                 ALGB0040
  130 G(J)      = 0.00                                                 ALGB0050
      RETURN                                                           ALGB0060
      END                                                              ALGB0070
```

Figure 77. Subroutine ALGBC Listing