

NASA CR- 139038

R-827

**A USERS MANUAL FOR A COMPUTER PROGRAM
WHICH CALCULATES TIME OPTIMAL GEOCENTRIC
TRANSFERS USING SOLAR OR NUCLEAR ELECTRIC
AND HIGH THRUST PROPULSION**

by
**Lester L. Sackett, Theodore N. Edelbaum,
Harvey L. Malchow**

June 1974

(NASA-CR-139038) A USERS MANUAL FOR A COMPUTER PROGRAM WHICH CALCULATES TIME OPTICAL GEOCENTRIC TRANSFERS USING SOLAR OR NUCLEAR (Draper (Charles Stark) Lab., Inc.) 169 p HC \$11.50

N74-30284
Unclas
54632

CSCL 22A G3/30



The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02139

R-827

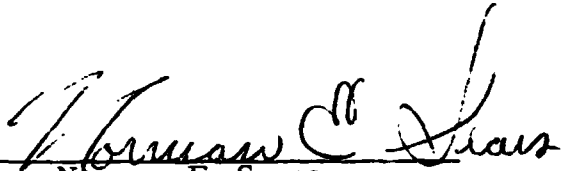
A USERS MANUAL FOR A COMPUTER PROGRAM
WHICH CALCULATES TIME OPTIMAL GEOCENTRIC
TRANSFERS USING SOLAR OR NUCLEAR ELECTRIC
AND HIGH THRUST PROPULSION

by

Lester L. Sackett, Theodore N. Edelbaum,
Harvey L. Malchow

June 1974

Approved:


Norman E. Sears

The Charles Stark Draper Laboratory, Inc.
Cambridge, Massachusetts 02129

ACKNOWLEDGEMENT

This report was prepared under Project 55-51300 sponsored by the Goddard Space Flight Center, National Aeronautics and Space Administration through Contract NAS 5-21791. The Technical Monitor was Mr. Kenneth I. Duck.

The publications of this report does not constitute approval by the National Aeronautics and Space Administration of the findings or the conclusions contained therein. It is published only for the exchange and stimulation of ideas.

A number of people have contributed to the success of this program. We would particularly like to thank Kenneth Duck of GSFC for his active support and encouragement, Huntington Small of Lockheed for his high thrust optimization program, Saul Serben of CSDL for several subroutines and for advice and assistance in programming. Alan Stanley of the Lincoln Laboratory and James Cake of NASA Lewis for their advice on radiation degradation and Michael Teague of GSFC for providing models of the radiation belts.

Abstract

This manual is a guide for using a computer program which calculates time optimal trajectories for high-and low-thrust geocentric transfers. Either SEP or NEP may be assumed and a one or two impulse, fixed total ΔV , initial high thrust phase may be included. Also a single impulse of specified ΔV may be included after the low thrust state. The low thrust phase utilizes equinoctial orbital elements to avoid the classical singularities and Kryloff-Bogoliuboff averaging to help insure more rapid computation time.

The program is written in Fortran IV in double precision for use on an IBM 360 computer. The manual includes a description of the problem treated, input/output information, examples of runs, and source code listings.

The NEP computer program is called NECKSPOT (Nuclear Electric Control Knob Setting Program for Optimal Trajectories) and the SEP program is called SECKSPOT.

TABLE OF CONTENTS

ABSTRACT

SECTION		PAGE
I.	Discussion	1
	I.1 Objective	1
	I.2 Solution Method	2
	I.3 Equinoctial Orbital Elements	?
	I.4 Options	6
II.	The Program Deck	8
III.	Constants and Conversions	1?
IV.	Input	14
	IV.1 Introduction	14
	IV.2 Input for NEP Deck	14
	IV.3 Input for SEP Deck	16
	IV.4 Comments	19
V.	Output	20
VI.	Initial Guesses and Parameters	2?
	References	31
APPENDIX		
A	Flow Charts	33
B	Thrust Direction On An Orbit	37
C	Example of SEP Run	38
D	Example of NEP Run	44
E	Source Code Listings	50

SECTION I

DISCUSSION

I.1 Objective

This computer program is designed to rapidly compute minimum time geocentric transfers for combinations of high thrust and low thrust stages. The most general configuration treated by the program contains three stages. The first stage is a high thrust stage such as a space tug, the second stage is a nuclear electric (NEP) or solar electric propulsion (SEP) low thrust stage and the final stage is a high thrust stage which accelerates a fixed payload. Either or both of the high thrust stages may be eliminated.

For a three stage case, the program operates in the following manner. The vehicle is assumed to start in a low altitude circular orbit. This orbit has a specified altitude and inclination and an optimized line of nodes. The first stage then performs a minimum-fuel time-open transfer to an optimized changeover orbit. This transfer is assumed to use a fixed ΔV and may involve either one or two impulses. All perturbations to the inverse square field are neglected in this phase. The second stage then performs a minimum-time low thrust transfer to a second optimized changeover orbit. This transfer is calculated by Kryloff-Bogoliuboff averaging and includes the effects of oblateness. If solar electric propulsion is used, the program may also calculate the effects of shadowing, radiation degradation, and the varying solar distance. The third stage then makes a minimum fuel, single impulse transfer through a fixed ΔV to the final orbit. This final orbit has a specified major axis, eccentricity and inclination. Its line of nodes and line of apsides are optimized. The overall trajectory is a minimum time trajectory for specified ΔV increments in the high thrust phases.

In addition to the three stage case, the program can be run using purely low thrust or using low thrust with either initial or

final impulses. For pure low thrust, the final line of nodes and line of apsides may also be specified as well as major axis, eccentricity and inclination.

1.2 Solution Method

A more thorough discussion will be found in the final report (Ref. 1). In addition the low thrust optimization technique has been previously described (Ref. 2). The basic technique is Kryloff-Bogoliuboff averaging (Ref. 3) of both the state and the costate. The averaged rates of change of the mean values of the state and costate are found by numerical quadrature. The differential equations for the mean state and costate may then be integrated in large time steps (typically days). A set of nonsingular orbit elements, the equinoctial elements (Ref. 4), is used to avoid numerical difficulties.

The effect of oblateness is included by analytically adding its rate of change of the mean state and costate to that due to thrust. The effects of shadowing are calculated by assuming that thrust is turned off in shadow. The shadow entrance and exit times are calculated analytically by solving a quartic equation. The effects of radiation degradation are calculated by fitting an equivalent 1 MEV electron flux as a function of radius and geomagnetic latitude. The power is then expressed as a function of the total accumulated particle fluence. As for all perturbations, the effect of radiation degradation on the costate as well as the state is calculated.

The first stage high thrust optimization is based on a very efficient computer program developed by Huntington Small (Ref. 5, 6). This program uses a special set of variables and form of the switching conditions developed by Small. Because the initial orbit is circular, it was possible to use an existing analysis (Ref. 7) to constrain the initial costate to the region that yields solutions. This program rapidly calculates either one or two impulse minimum-fuel time-optimal trajectories.

Because these transfers always require less than a full revolution, their time is negligible compared to the low thrust phase and is not considered.

The third stage high thrust program applies a single impulse in the direction of the primer vector at the location of its maximum value on the final orbit of the low thrust phase. This produces an optimum single impulse transfer to the terminal state.

The overall trajectory is optimized by a shooting method. Initial values of the unspecified states and costates or functions thereof are chosen at the initial time. An optimum high and low thrust trajectory is then generated by integrating the state and costate through all three stages. This will generate an optimal trajectory to the wrong terminal state. A sensitivity matrix is then generated by varying the initial conditions and running a set of neighboring trajectories. A Newton iteration on the initial conditions is then used to drive the terminal errors to within specified bounds. The final converged trajectory is then a minimum time trajectory for the specified velocity increments in the high thrust phases.

The NEP program and the SEP program have separate blocks of certain subprograms, while also sharing several others. The state for NEP includes the orbital elements while for SEP, mass and fluence are also included in the state.

I.3 Equinoctial Orbital Elements

The low thrust trajectory calculations are done in equinoctial coordinates (Ref. 4). When trajectory information is printed both classical and equinoctial elements are included. The costate includes the adjoints to the equinoctial orbital elements. Adjoints to the classical elements are generally not calculated or printed.

The equinoctial orbital elements are defined in terms of the classical elements by the following equations.

$$\begin{aligned}
a &= a \\
h &= e \sin (\omega + \Omega) \\
k &= e \cos (\omega + \Omega) \\
p &= \tan \frac{i}{2} \sin \Omega \\
q &= \tan \frac{i}{2} \cos \Omega
\end{aligned}$$

where a is the semimajor axis (in the program output the equinoctial a is usually given in earth radii and the classical a in kilometers), e is the eccentricity, i is the inclination, Ω is the longitude of the ascending node, and ω is the argument of perigee. The classical elements are in terms of an earth equatorial coordinate system with the x axis toward the vernal equinox and the z axis through the N pole. The equinoctial coordinate frame is defined by unit vectors \hat{f} , \hat{g} , \hat{w} illustrated in Fig. 1. and defined by

$$\begin{aligned}
\hat{f} &= \frac{1}{1 + p^2 + q^2} \begin{bmatrix} 1 - p^2 + q^2 \\ 2 p q \\ - 2 p \end{bmatrix} \\
\hat{g} &= \frac{1}{1 + p^2 + q^2} \begin{bmatrix} 2 p q \\ 1 + p^2 - q^2 \\ 2 q \end{bmatrix} \\
\hat{w} &= \frac{1}{1 + p^2 + q^2} \begin{bmatrix} 2 p \\ - 2 q \\ 1 - p^2 - q^2 \end{bmatrix}
\end{aligned}$$

The position in an orbit can be indicated by the eccentric longitude, F , where

$$F = E + \omega + \Omega,$$

and where E is the classical eccentric anomaly.

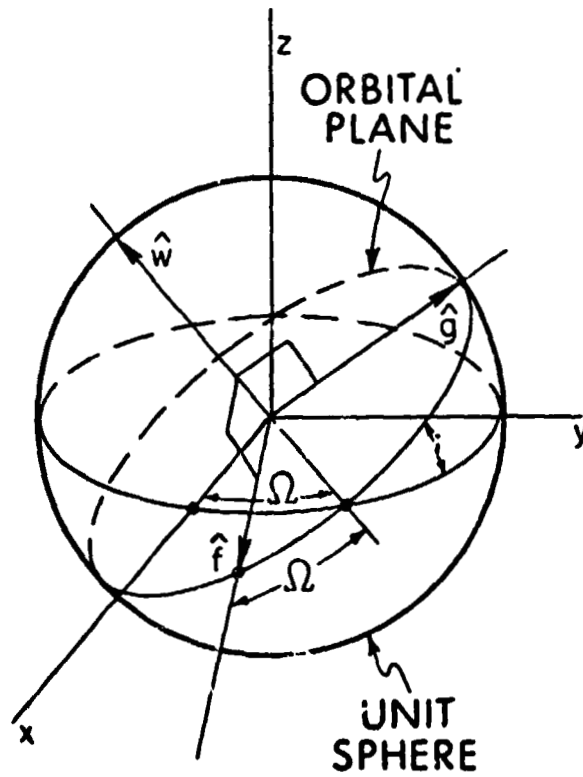


Figure 1 Direct Equinoctial Coordinate Frame

I. 4 Options

The options listed here involve those associated with the physical problem considered, i. e. the nature of the trajectory and orbits, the propulsion system, and physical effects which may be included, and not the options involving variations in the numerical computation techniques. The latter are indicated by the listing of the input variables (Sec. IV) and the comments on their selection in Sec. VI.

- 1) Low thrust may be SEP or NEP. The decks are slightly different.
- 2) Initial high thrust may be used with SEP or NEP. Total ΔV for the initial high thrust is an input.
- 3) A final impulse may be included; the ΔV is an input.
- 4) Either a, e, i, Ω , ω or just a, e, i may be specified at the final time. If high thrust is included the final Ω and ω are free.
- 5) If there is no initial high thrust, the a, e, i, Ω , ω for the initial orbit are input. If initial high thrust is included, the initial orbit must be circular; the initial Ω may be selected (with SEP, NEP) or left free (for SEP).
- 6) Initial power may be selected.
- 7) Specific impulse may be selected.
- 8) Initial mass may be selected.
- 9) Oblateness effect may be included or not included.
- 10) For NEP, power may be constant or may degrade exponentially with a user supplied time constant.
- 11) For SEP, the effect of Van Allen radiation on power output may be included or power may be constant (the latter was included mainly for test purposes).

- 12) A value for housekeeping power may be included.
- 13) For SEP, the shadow effect may be included or not.
- 14) For SEP, power may vary with spacecraft distance from the sun, or else the distance may be assumed to be constant at 1 A. U.

SECTION II THE PROGRAM DECK

There are three blocks of subprograms, one for NEP, one for SEP and a block of subprograms which both NEP and SEP have in common. High thrust may be used in combination with either NEP or SEP. The SEP program includes additional state and costate variables and has several more subprograms than the NEP deck, because of the shadowing and degradation effects. Thus size and run time is considerably greater for the SEP deck.

The program is coded in Fortran IV in double precision for use on an IBM 360 computer and has been compiled using a G compiler. Following is a list of subprograms with a brief description for which the SEP and the NEP decks have separate versions, but which perform essentially the same function.

INPUT, reads and prints input data, sets initial conditions
OUTPC, prints summary of converged trajectory characteristics
ITER, either the Newton-Raphson or Modified Newton-Raphson
iterator
PRTN, prints information at each iteration of ITER
DCROUT, essentially inverts a matrix
TRAJ, calculates a single trajectory by calling the high thrust
subroutines and the low thrust differential equation
integrator, then calculates error vector
OUTP, prints information at each time step of low thrust integrator
FUNCT, calculates low thrust averaged derivative for integrator

The following subprograms are shared by the NEP and SEP decks.

OBLATE, calculates single averaged effect of oblateness
on state and costate
QUAD, either 4, 8, 16 or 32 point vector gaussian quadrature

FCT, EVALMP, calculate the derivative due to thrust for state and costate before averaging

MAINE, the main calling program for the initial high thrust calculation

START, called by MAINE, sets initial values of S array

TIME, iterates on ΔV to satisfy Small's optimality condition

SWITCH, computes the coast angle

DTDU, updates the S array after an impulse or a coast

OUTH1, prints initial high thrust orbit characteristics, also an interface with following low thrust phase

IMPLS, calculates effect of a single final impulse

YF, calculates primer vector for IMPLS

CONTL, main controlling and calling program for SEP and NEP decks

MAINE, START, TIME, DTDU are taken directly from Small's thesis (Ref. 5). MAINE was altered slightly, the other subprograms have not been changed at all. For further information and discussion concerning these subprograms see Ref. 5.

The following subprograms are only part of the SEP deck.

EARTH, sets certain constants associated with the earth rotation, revolution and magnetic field

SUN, calculates sun's direction and magnetic field orientation

SHADOW, calculates shadow entrance and exit angles (if any) and certain associated partial derivatives

DQRTIC, solves a quartic equation

DCUBIC, solves a cubic equation

FLUX, calculates flux effect on state and costate

In addition to the above subprograms, the IBM Scientific Sub-routine Package Runga-Kutta (DRKGS) or Predictor-Corrector (DHPCG) integrators are required (Ref. 8) A chart showing the relation between the various subprograms is shown in Fig. II-1 for the NEP deck and in Fig. II-2 for the SEP decks. An overall flowchart and flowcharts of some of the main calculations are given in Appendix A.

Additional detail on the above subprograms is given in the final report.

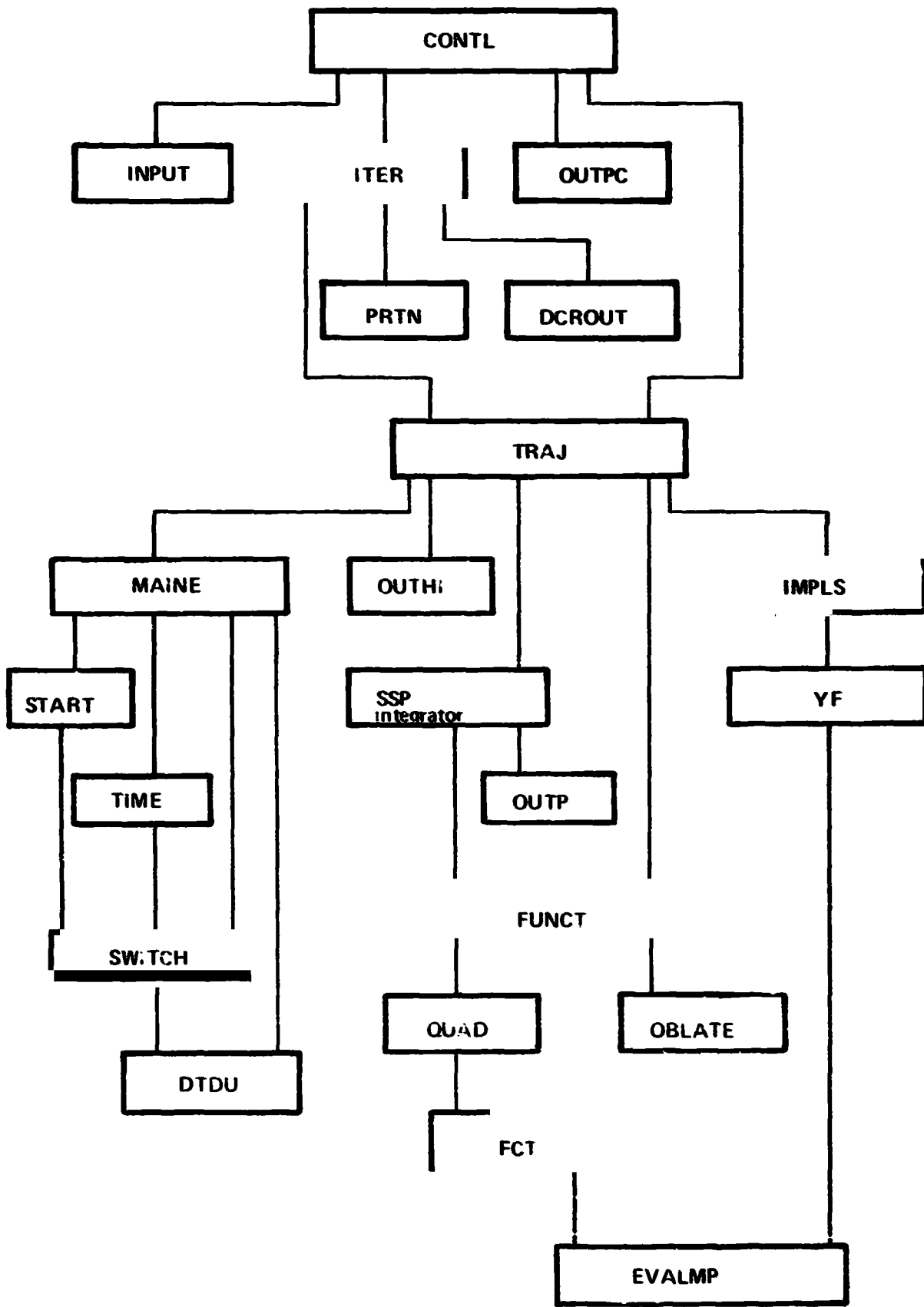


Fig. II-1 NEP System Diagram

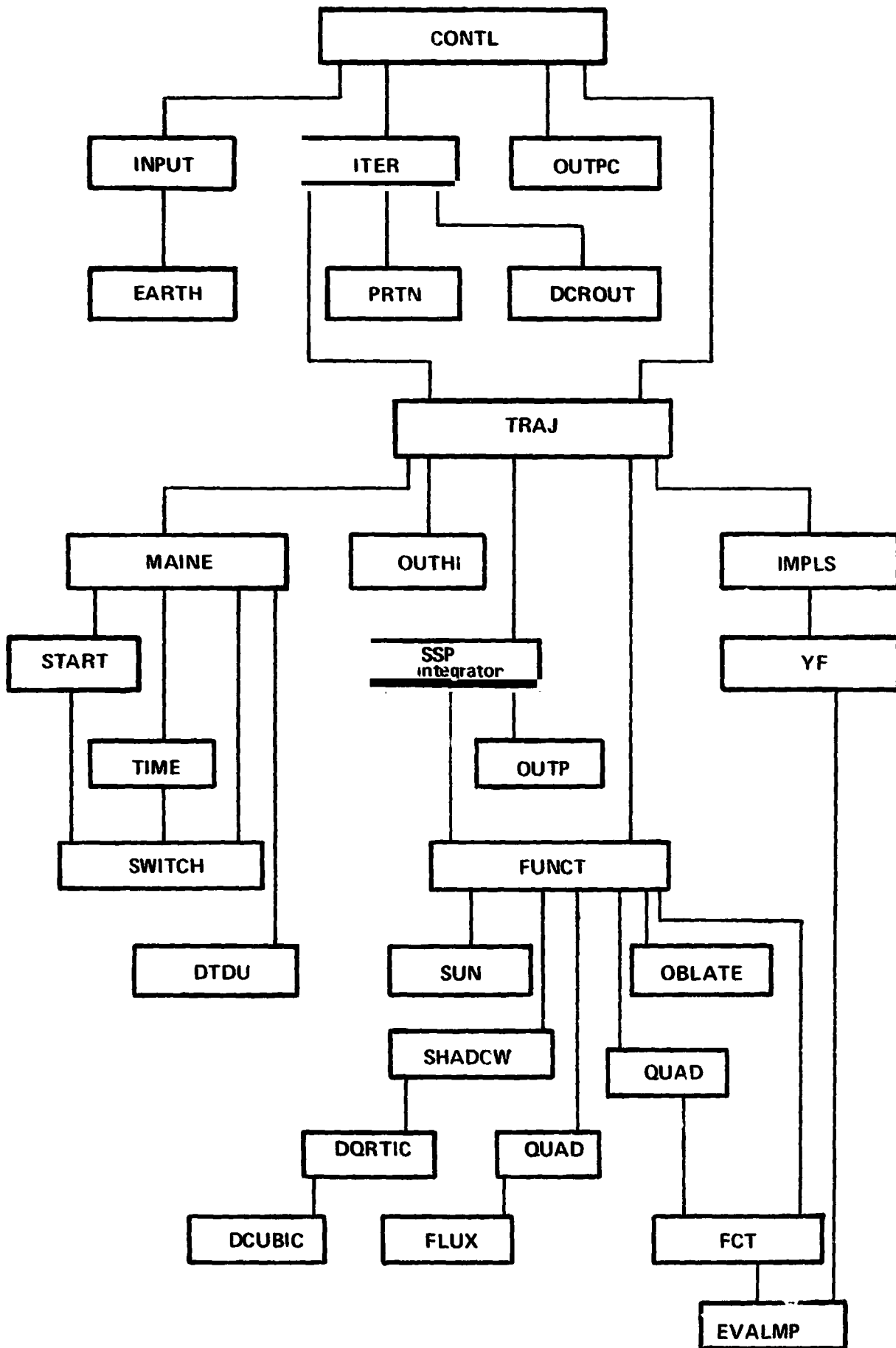


Fig. II-2 SEP System Diagram

SECTION III

CONSTANTS AND CONVERSIONS

The following are some constants and initial values which are assumed.

equatorial earth radius = 6378.16 km (Ref. 9)
earth gravitation coefficient = $398601.2 \text{ km}^3/\text{sec}^2$ (Ref. 9)
oblateness $J_2 = .0010827$ (Ref. 9)

Earth's orbital elements (epoch JD= 2436935.0) (Ref. 10)

$a = 1$ A. U.

$e = .016726$

$\omega = 102^\circ 25253$

mean orbital motion = $.985609^\circ/\text{day}$ (Ref. 10)

obliquity of ecliptic = $23^\circ 45$ (Ref. 10)

earth rotational frequency = $359^\circ 0170416/\text{day}$ (Ref. 10)

Latitude of north magnetic pole = $78^\circ 6$ (Ref. 11)

Longitude of north magnetic pole = $289^\circ 9$ (Ref. 11)

Internal units are in equatorial earth radii, 10^3 kg. , and internal time units calculated such that a circular orbit at 1 earth radii would have a period of 2π internal time units. In this system of units the gravitational coefficient, $\mu = 1$. Other conversions can be derived from these basic equivalences. For example,

units to seconds = 806.8147206095579

units to days = .0093381333403884

units to kilowatts = 77458.55283702227

SECTION IV

INPUT

IV.1 Introduction

The quantities discussed in this section are all read by the subprogram INPUT. Unless otherwise indicated each value is read on a separate line, real variables in fixed format (F25.15), integer variables, beginning with i, j, k, l, m, n are read in format, I2. The input for the NEP deck and the SEP deck are listed in this section along with a brief description and nominal values, if any.

IV.2 Input for NEP Deck

Initial orbit characteristics,

- W(1) km, semimajor axis
- W(2) eccentricity
- W(3) (degrees), inclination
- W(4) (degrees), longitude of ascending node, not used if initial high thrust
- W(5) (degrees), argument of perigee, not used if initial high thrust

Initial guesses for iteration parameters

- ZL0(1) λ_a , adjoint to semimajor axis or if initial high thrust the transformed T (see Ref. 6)
- ZL0(2) λ_h , adjoint to orbital element h or if initial high thrust the transformed Small variable k (see Eq. 6 of Ref. 6)
- ZL0(3) λ_k , adjoint to orbital element k, or if initial high thrust, the transformed Small variable j (see Eq. 6 of Ref. 6)
- ZL0(4) λ_p , adjoint to orbital element p, or if initial high thrust the scale factor relating high and low thrust costates

ZL0(5) λ_q , adjoint to orbital element q, if initial high thrust, not used

The desired final orbit

WF(1) (km), semimajor axis

WF(2) eccentricity

WF(3) (degrees), inclination

WF(4) (degrees), longitude of ascending node, not used if NOP=2

WF(5) (degrees), argument of perigee, not used if NOP=2

TF2 (days), guess for final time

PKW (kw), electrical power including efficiency factor

SPIM (sec), specific impulse of NEP

AM0 (kg), initial mass (NEP)

The following input may be read or, optionally, left at nominal values. IRDFLG is read followed by the additional input or operations and then IRDFLG is again read, until IRDFLG = 01 and input is ended.

IRDFLG	NOMINAL
1 End of input	
2 IPR, print flag	0
3 NIMAX, Max. no. of iterations (if 0, by pass iteration to print time history)	20
4 TFMAX2, (days), max. TF	190.
5 DT2, (days), time step for integrator	1.
6 UEB, upper error bound for integrator	1.D10
7 EW, Error weights (Format 5D6.1)	1., 1., 1., 1., 1., 0.,
8 UTKM, Equatorial earth radius	6378.16
9 GM, (km ³ /sec ²), earth grav. coefficient	398601.2
10 NOP = 1, five orbital elements specified at TF, use only if IHI= 1 = 2, three orbital elements specified at TF	1

11	Sets oblateness coeff, $AJ2, = 1.0827D-3$	0.
12	STEP, Step size for numerical differentiation in ITER	1. D-6
	KSTEP = 0, STEP as fraction in ITER	
	= 1, STEP as constant in ITER	
13	IPOW = 0, constant power	0
14	= 1, exponential degradation	
	BB, (sec), time constant for degradation	
	PH, (kw), housekeeping power if IPOW=1	
14	EMPTY	
15	IHI = 1, low thrust only	1
	= 2, high/low	
	= 3, high/low/high	
	= 4, low/high	
	DVI1, (m/s), total initial high thrust ΔV	0.
	DVI2, (m/s), ΔV for final impulse	0.
16	EMPTY	
17	FLIM, Norm limit in ITER routine	1. D-6
18	SGN = -1, if initial λ_i is negative	SIGN (WF(3) - W(3))
	= +1, if initial λ_i is positive	

Additional information concerning the meaning of these variables and suggestions in setting their values is given in Sec. VI.

IV.3 Input for SEP Deck

Initial orbit

W(1)	(km) semi major axis
W(2)	eccentricity
W(3)	(degrees) inclination
W(4)	(degrees) longitude of ascending node, not used if any high thrust and NOD Σ =1
W(5)	(degrees) argument of perigee, not used if any high thrust
W(6)	(kg) mass at beginning of low thrust stage
W(7)	(10^{14} equivalent 1MEV electrons/cm ²) initial fluence

Initial guesses

- ZL0(1) λ_a , adjoint to semi major axis or if initial high thrust the transformed \mathcal{T} (see Ref. 6)
ZL0(2) λ_h , adjoint to orbital element h or if initial high thrust the transformed Small variable k (see Eq. 6 of Ref. 6)
ZL0(3) λ_k , adjoint to orbital element k, or if initial high thrust the transformed Small variable j (see Eq. 6 and Ref. 6)
ZL0(4) λ_p , adjoint to orbital element p, or if initial high thrust the scale factor relating high and low thrust costates
ZL0(5) λ_q , adjoint to orbital element q, or if initial high thrust, the long. of node (radians) if NODE=0, or adjoint to long. of node if NODE=1
ZL0(6) λ_m , adjoint to mass
ZL0(7) λ_N , adjoint to fluence

The desired final orbit

- WF(1) (km) semimajor axis
WF(2) eccentricity
WF(3) (deg) inclination
WF(4) (deg), longitude of ascending node, not used if NOP=2
WF(5) (deg), argument of perigee, not used if NOP=2
TF2 (days), guess for final time
PKW (kw), electrical power at 1 A. U. including efficiency factor
SPIM (sec), specific impulse of SEP
TL Julian date at initial time

The following input may be read or, optionally, left at nominal values. IRDFLG is read followed by the addition input or operations and then IRDFLG is read again until IRDFLG = 01 and input is ended.

IRDFLG		NOMINAL
1	End of Input	
2	IPR print flag	0
3	NIMAX max. no. of iterations (if 0, bypass iteration to print time history)	20
4	TFMAX2 (days), max. TF	190.

5	DT2	(days), time step for integrator	1.
6	UEB	upper error bound for integrator	1.D10
7	EW	error weights for integrator (7D6, 1)	1., 1, 1, 1, 1, 0, ...
8	UTKM	equatorial earth radius (km)	6378.16
9	GM	(km ³ /sec ²) earth grav. const.	398601.2
10	NOP	= 1, five orbital elements specified at TF use only if IHI= 1 = 2, three orbital elements specified at TF	1
11	Sets oblateness, AJ2, = 1.0827D-3		0.
12	STEP	step size for numerical differentiation in ITER, 8 dim., eighth element for time variation of Hamiltonian	1.D-6
	KSTEP	= 0, step as fraction in ITER = 1, step as constant in ITER (except STEP (8))	0
13	ISON	= 0, shadow effect off = 1, shadow effect on	0
14	ISUN	= 0, sun distance effect on power off = 1, effect on	0
	PH	(kw) housekeeping power	0.
15	IHI	= 1, low thrust only = 2, high/low = 3, high/low/high = 4, low/high	1
	DVI1	(m/s) total initial high thrust ΔV	0.
	DVI2	(m/s) ΔV for final impulse	1.
	NODE	= 0, initial line of nodes free, λ_{Ω} fixed = 1, initial line of nodes fixed, λ_{Ω} free	
16	IPOW	= 0, constant power = 1, degradation effect	1
17	FLIM	norm limit in iteration routine	1.D-6
18	SGN	= -1, if initial λ_i is negative = +1, if initial λ_i is positive	SIGN (WF(3)-W(3))

IV.4 Comments

As coded at the time of this writing the following limitations exist. For NEP and NOP=2 (final ω and Ω free), the final eccentricity and inclination should not be set to zero; they may be set to small numbers (e.g. $e_f = .0001$, $i_f = .01^\circ$). For NEP and initial high thrust, the final inclination should not be set equal to the initial inclination. This may also cause difficulties for SEP and initial high thrust. When initial high thrust is included, ZL0(1) and ZL0(2) (i. e. T and j) should not both be set to zero. Final inclination should not be set to 180° . At this writing there has been little experience with inclination greater than 90° . Also the coded equations are not valid for eccentricities greater than or equal to one.

SECTION V OUTPUT

Most of the output is self-explanatory and a look at an example will familiarize the user with it. There are certain basic groups of output. The first is the printing of the read-in initial data and a few internally set constants. Normally this will be followed by output from the iterator. After convergence, a summary of characteristics of the converged trajectory is printed. Finally, a time history of the converged trajectory will be printed. Usually, even if convergence was unsuccessful, a time history of the last trajectory to be calculated will be printed.

The printing of the initial data should be understandable. There are a few abbreviations used.

A,	semi-major axis
E,	eccentricity
I,	inclination
LON ASC NODE,	longitude of ascending node
ARG PERIG,	argument of perigee
SPEC IMP,	specific impulse
EXH VEL,	exhaust velocity
M/S,	meters/second
E. R. /T. U.,	earth radii/time unit
UTKM,	internal units to kilometers
UTS,	internal units to seconds
UTD,	internal units to days
UTKG,	internal units to kilograms
UTKW,	internal units to kilowatts
UTMS2,	internal units to meters/sec ²

After the initial input print, the iteration begins. The iteration number (ITER NO.) and the total number of calls to TRAJ are printed followed by X, the iteration parameters (ZL0), then Y, the error in the final conditions. The final conditions are the final values of a, h, k, p, q if

NOP = 1, or $a, e, \tan^i \frac{1}{2}, \lambda_{\Omega}, \lambda_{\omega}$ if NOP = 2, λ_m, λ_n for SEP, and finally the Hamiltonian. Then the final time (TF) is printed in internal units, followed by, F0, the sum of the squares of the errors in the final conditions. For convergence this value must be less than FLIM, the "norm limit in ITER". In order to calculate the partial derivative matrix or sensitivity matrix the nominal values "X" are changed slightly by inputted amounts; these perturbed values of X (X(I) + DX(I)) are next printed followed by the corresponding Y. The partial derivative matrix is printed as well as its determinant. This matrix is inverted and premultiplies the error vector to obtain the changes in the X's, DELX:S, which are next printed.

A new trajectory is calculated and the sum of the squares of the errors in the final conditions is printed (F1). If this is smaller than F0, a new iteration begins; if it is larger than F0, the DELX:S are halved and printed. This continues until $F1 < F0$ or until a certain number of halvings. What follows depends on how well the method converges and on whether the Newton-Raphson or modified Newton-Raphson subprogram is used. Further output is basically permutations of the above, terminating with convergence or a message indicating lack of success.

After exit from the iteration, a summary of characteristics of the last trajectory (the optimal, if convergence was successful) is printed. Included are the actual final orbital elements, the error in the final orbital elements, the values for the iteration parameters, the final time, the equivalent particles (fluence) in units of 10^{14} (for SEP), the final mass, the ratio of final to initial mass, the final power, the ratio of final to initial power and the total low thrust ΔV (DELV).

Next is printed a time history of the final (optimal if convergence was successful) trajectory. If NIMAX = 00, then a time history is printed immediately following printing of the input data, bypassing the iteration routine, and summary print. If the trajectory includes initial high thrust impulses, the orbit number is printed (ORBIT =) followed by "EQUINOCTIAL O. E. AND COSTATE/S. F. 1000", after which the

equinoctial orbital elements (a, h, k, p, q) and the equinoctial costate divided by the scale factor x 1000, which relates the high and low thrust costates and is an iteration parameter. Following "CLASSICAL O. E." are printed the classical orbital elements (a(km), e, i (°), Ω(°), ω(°)). Also printed is the true anomaly at which the last impulse occurred, and ϕ, T and ΔV where the thrust direction is given by

$$\bar{\beta} = \sin \phi \underline{e}_R + \cos \phi \cos T \underline{e}_L + \cos \phi \sin T \underline{e}_h$$

where \underline{e}_R , \underline{e}_L , \underline{e}_h are unit vectors, \underline{e}_R along the radius vector, \underline{e}_h perpendicular to the orbit and $\underline{e}_L = \underline{e}_h \times \underline{e}_R$ (Ref. 1, 5 and 6).

Next is printed the low thrust trajectory time history at each time step. First is printed TIME in various units. ΔV (DV(K/S)) in kilometers/sec and the time step number are also printed. Next is printed the equinoctial orbital elements (a, h, k, p, q) and mass (10^3 kg) and fluence (10^{14} particles) if SEP. Then classical orbital elements (a, e, i, Ω, ω) and mass (kg), power (kw), thrust (newtons), and thrust acceleration (meters/sec²). Next is the costate, then the state derivative, then the costate derivative and then the value of the Hamiltonian, the period (hours), perigee and apogee (km) and the divisions of the time step performed by the integrator. For SEP with shadowing the time spent in shadow is printed in hours and as a fraction of the period (if the orbit passed through shadow). This print is repeated at each time step.

Finally, if a final impulse is included, equinoctial and classical orbital elements are printed for the final orbit, as well as the impulse direction and location in the equinoctial coordinate frame.

A number of error messages are scattered through the code. A few will be mentioned here. Several, in INPUT, call attention to bad input data. For bad input data following an IRDFLG value, a message, IRDFLG = (number), is printed. In some cases additional information is given. When shadowing is included, a message, ISHAD=1, indicates that only one shadow crossing was found. This arises from small numerical inaccuracies in solving the quartic equation and can usually be ignored.

SECTION VI

COMMENTS ON INITIAL GUESSES AND OTHER PARAMETERS

Picking initial conditions is very important for running a program such as this. There are no built in values and the actual choices of the user can greatly influence the rapidity of convergence or, in some cases, if convergence occurs at all. As additional perturbations are added to the basic most simple problem, solution becomes more and more sensitive to the initial parameter choices, including those parameters of subprograms which affect numerical accuracy such as the integrator and its parameters (error bound, error weights, time step) and quadrature formula (8, 16, 32 point), and step size in the calculation of the numerically derived sensitivity matrix in the 2PBVP solutions.

The usual difficulty when looking at a new case will be picking the initial values for the costate (when initial high thrust impulses are included, some of the iteration parameters are functions of the actual costate) and the guess for the time of flight. Frequently ball-park values for these parameters will be known from previous similar cases. If nothing is known about the likely values it may be less costly to run a simpler case (e.g. without shadowing or oblateness) and with less accuracy (a lower point quadrature formula or a larger time step). However, if numerical accuracy is too poor, convergence will be affected. The converged values for such an example would then be input guesses for the more complex and more accurate case.

A less accurate solution might utilize a 8-point quadrature and time steps of from 2 to 6 days for low thrust accelerations of 10^{-4} g's with smaller steps for larger accelerations and vice/versa.

Previous experience has shown that convergence can be particularly difficult when shadowing is included. One useful technique in this case is to get convergence for a nominal case without shadowing (or without other perturbations which may be causing trouble) and then to add shadowing, and using the iteration parameter values from the nominal converged case as input to the shadow case, try to converge to a point

along the nominal trajectory (using a corresponding final time). These new values for the iteration parameters can be used (again with a corresponding final time) to converge to a point further down the nominal trajectory. This process is continued until the desired final conditions are met. Three to six steps might be used. This procedure helps insure that the guessed initial trajectory is not too far from the desired extremal.

If the approximate ΔV is known, a final time estimate can be calculated assuming constant thrust acceleration.

$$\Delta V = a \cdot t_f$$

The initial guesses for the iteration parameters must be specified. These are ZL0(I), I = 1, 5 for NEP and ZL0(I), I = 1, 7 for SEP. For SEP (with or without high thrust) ZL0(6) is the initial adjoint of mass and ZL0(7) is the initial adjoint of fluence. Typical values are

$$ZL0(6) = - 5000.$$

$$ZL0(7) = - 100.$$

For SEP or NEP, without initial high thrust, ZL0(I), I = 1, 5 are the initial adjoints to the equinoctial orbital elements (a, h, k, p, q). ZL0(1) should always be non zero and positive for orbit raising. The others may have positive, zero or negative values, ZL0(2) and ZL0(3) with magnitudes usually less than 10^3 . ZL0(4) and ZL0(5) with magnitudes usually less than 10^4 . The signs depend on the values of h, k, p, q. Typical values are:

$$ZL0(1) = 3000.$$

$$ZL0(2) = - 100.$$

$$ZL0(3) = 500.$$

$$ZL0(4) = - 100.$$

$$ZL0(5) = - 5000.$$

When initial high thrust is combined with SEP or NEP, ZL0(1), I = 1, 5 are no longer the adjoints to the orbital elements. Instead the first three elements are related to Small's variables T , k , j (Ref. 1, 5, 6) by

$$T = \frac{\pi}{2} \frac{ZL0(1)}{\sqrt{1 + ZL0(1)^2}}$$

$$k = \cos T \left(.75 + .25 \frac{ZL0(2)}{\sqrt{1 + ZL0(2)^2}} \right)$$

$$j = (1 + k \cos T) \sqrt{\frac{\cos T - k}{\cos T + k} \cdot \frac{ZL0(3)}{\sqrt{1 + ZL0(3)^2}}}$$

The above transformation insures that T , k , j are maintained within valid bounds for the initial circular parking orbit for all values of ZL0(I), I = 1, 3. ZL0(4) is a scale factor, actually relating the adjoints for the initial high thrust and the low thrust phases of a trajectory. This value will almost always be around 1.

For high thrust with NEP, ZL0(5) is not used but should be set to some arbitrary value such as 1. Two options exist for high thrust and SEP. If the initial Ω is fixed (NODE=1) then ZL0(5) is proportional to the adjoint to Ω . Theoretically this should be zero if non-thrusting perturbations are axially symmetric, and in any case will typically have a magnitude less than 1.0. The other option is to let Ω be free (and therefore the initial adjoint to Ω is zero). ZL0(5) is then Ω . This option may be used if the non-thrust perturbations are not axial symmetric (about the geographic poles). This includes the effect of a tilted radiation field or shadowing. Ω is measured in radians and so ZL0(5) will typically be between $-\pi$ and $+\pi$. Convergence will be facilitated if the correct quadrant is known.

Generally, T will have a magnitude less than 20° so that ZL0(1) will have a magnitude less than .3 or so. ZL0(2) is typically positive around 1. ZL0(3) typically has a magnitude around 1.

Typically values for the NEP program

- .1
1.0
.2
1.5
1.0 (not used)

for the SEP program

- .2
.4
- .1
.9
0.0

The above discussion simply gives an order of magnitude feeling of values for initial guesses. More information can be gained by looking at the definitions of the iteration variables and then looking at particular cases of interest. Also helpful is looking at special cases, e. g., zero eccentricity, constant thrust, for which analytic results are known. The values for similar cases or simplified cases are always useful.

Now a few words will be given on values for which built in nominals exist. There is nothing special about some of these nominals except that the constructors of the program used those values a lot. Comments on a few of these follow.

IPR is a print flag that would normally only be used if there was trouble. It causes all trajectories to be printed (the number of steps in the low thrust portion printed is equal to about $IPR + 1$). Normally, the final (converged) trajectory will be printed anyway.

NIMAX, the number of iterations in the N - R or modified N - R procedure is set to 20 and that is usually enough to get convergence. It can be set to zero, which will cause a printing of the trajectory determined by the values of the input data, bypassing the iterator entirely

TFMAX simply prevents integrations past that time of flight and can be left alone unless you expect the time of flight to be near or greater than 190 days.

DT2, the time step in days with a nominal of 1. Frequently, a different value will be needed depending on the problem.

UEB, the upper error bound for the integrator, nominally set to a high value so that it is usually never reached, allowing the user to determine accuracy by picking the time step. This value is supplied to the SSP integrator and tuning it seemed difficult.

EW, error weights supplied to the SSP integrator, nominally set to 1 for the five orbital elements and zero for the other variables. This is somewhat arbitrary, as the writer has had little experience adjusting these values.

UTKM, the earth's equatorial radius. If you don't believe 6378.16 km, you can change it.

GM, the earth's gravitational constant set to $398601.2 \text{ km}^3/\text{sec}^2$, likewise.

NOP, a flag, if set to 01 then the final conditions include all five orbital elements, specified by the input, if set to 02, then a, e, i are specified only. If any high thrust is included this is automatically set to 02. In this case, this option should not be set by the user.

Oblateness effect is off nominally. If included, J_2 is set to .0010827. As coded, a different value for J_2 would require a recompilation of INPUT.

KSTEP is a flag which indicates whether STEP, which is used to calculate the numerically determined sensitivity matrix in the iterator, is a fraction of the nominal value of the iteration parameters (KSTEP=0) or a fixed step size. Nominally, STEP as a fraction is specified with all elements of the array set to 10^{-6} . This was found

to work pretty well for low thrust only cases. (If the magnitude of the nominal iteration parameter is near zero, the corresponding STEP is considered a step size rather than as a fraction.) When initial high thrust impulses are included a constant step size was more frequently used, especially when the iteration parameters were small in magnitude. Typical values were:

$$10^{-8}, 10^{-4}, 10^{-6}, 10^{-5}, 10^{-5}, 10^{-3}, 10^{-3}, 10^{-6}$$

(The first five being relevant to the NEP program.) STEP (8) remains the fractional variation in the final time used to calculate $\frac{\partial H}{\partial t_f}$.

Changing these values by a couple orders of magnitude had little effect.

Nominally, shadow effect is not included.

Nominally, the sun distance affect on power is not included, i. e., a 1 A. U. distance is assumed. Housekeeping power is nominally zero.

Nominally, low thrust only is assumed. If initial or final high thrust impulses are required, the ΔV 's must be given. For the SEP program, if initial high thrust is included, NODE must be set to 0 if Ω is free, or to 1 if Ω is fixed. Note that if the non-thrust perturbations are symmetric about the equator changing Ω should have no effect on the trajectory, so that if Ω is free, the sensitivity matrix will be singular. This would occur if a radiation model was used which was symmetric about the geographic axis and there was no shadow effect. IPOW is nominally set to 0 for the NEP program. This specifies constant power. The exponential degradation is specified by IPOW=1. For the SEP program, IPOW is nominally set to 01, so the degradation due to radiation is included. A constant power option is available although probably would not generally be used. If shadowing is not included, this is equivalent to a corresponding NEP case, but more expensive to run.

FLIM determines the accuracy of the actual final conditions which determine convergence. The nominal value of 10^{-6} would yield an error in each component of the final conditions of about 10^{-4} . (The "cost" which is compared to FLIM is just the sum of the squares of the errors in the final conditions in internal units, scaled so that typical values have an order of magnitude of zero.) This seems consistent with the general accuracy of the model used.

SGN is used only if there are initial high thrust impulses. It sets the sign of the adjoint to inclination. Nominally this is automatically set negative if the final desired inclination is less than the initial inclination or positive if the opposite is the case.

Because of the functional form of the power versus fluence relation, to avoid having to use a very small time step (at least at the beginning of an integration) or else having numerical difficulties, the initial fluence ($W(7)$) for the SEP input should be nonzero. If the user inputs 0., then a nonzero value will be set by the code. This value is equal to $\dot{N} \cdot T/2$ where \dot{N} is the initial fluence derivative and T is the initial orbital period (after any initial impulses). In order to avoid numerical difficulties, especially when initial impulses are included, it is useful for the user to input some nonzero value for fluence which overrides the internal calculated value. A typical value might be .91 (in units of 10^{14} particles).

At the time of this writing the experience using the final version of the code is limited. Few runs have been performed with the final impulse included. The version of the SEP code used for most runs contained a geographically axial symmetric radiation field which corresponds to modeling the geomagnetic axis as lining up with the geographic axis. This allows a larger time step to be used and effectively averages the spatial dependence of the field over one day. Convergence was sometimes adversely affected by including shadowing. Also when initial impulses were included, convergence was more difficult if the initial Ω was free. Most runs were performed with a fixed initial Ω .

As coded, when the predictor-corrector integrator is used the time step is forced to be an integer division of the final time and thus varies somewhat from one trajectory to another as the time of flight varies. The varying time step can cause varying numerical errors for neighboring trajectories, thus most runs, especially with the more complex SEP program have been performed with the Runge-Kutta integrator for which the time step was fixed. Poor numerical accuracy will adversely affect convergence.

REFERENCES

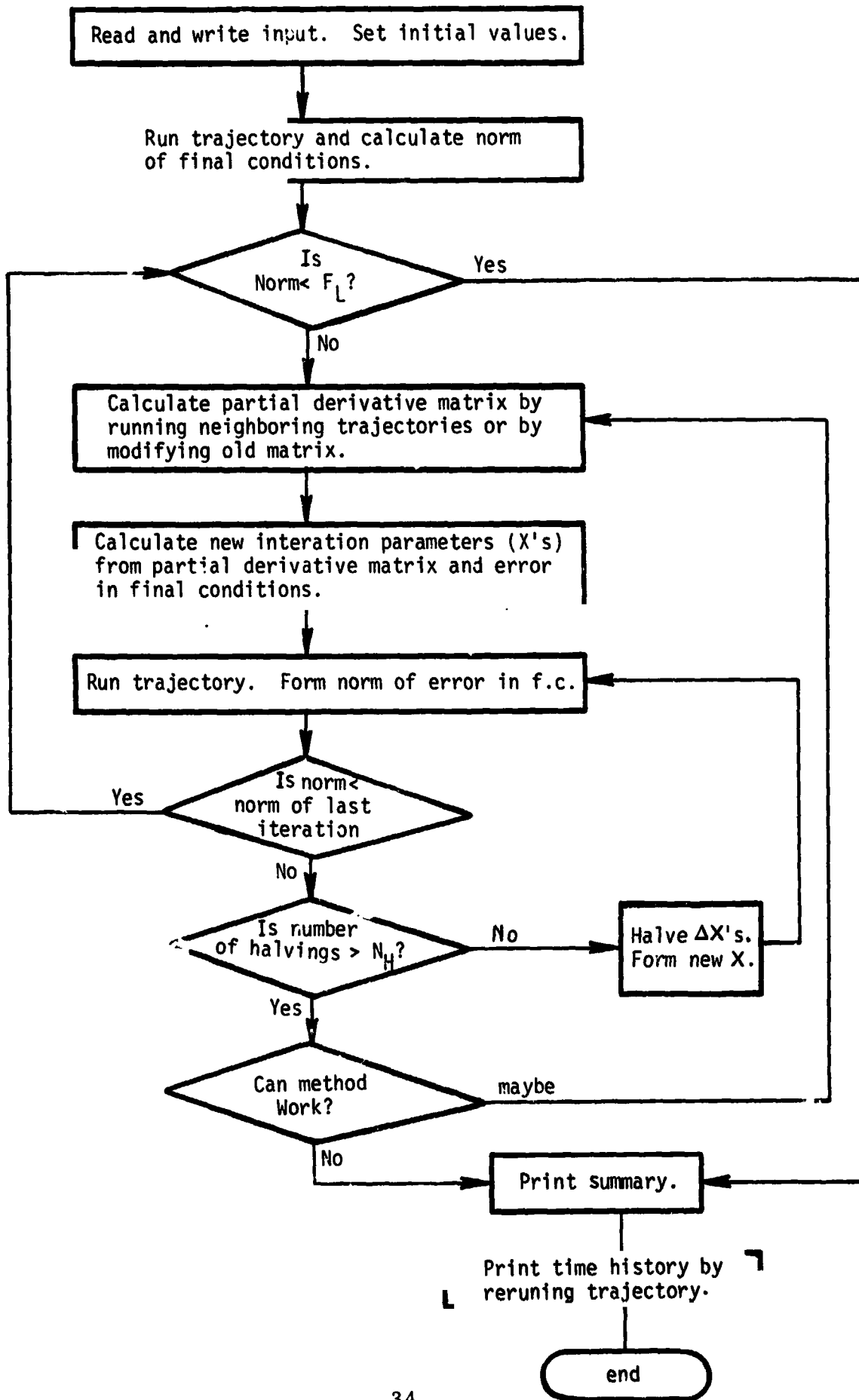
1. Final Technical Report on NASA Contract NAS 5-21791, to be published, C. S. Draper Laboratory, 1974.
2. T. N. Edelbaum, L. L. Sackett, H. L. Malchow, "Optimal Low Thrust Geocentric Transfer," AIAA Paper No. 73-1074, Lake Tahoe, 1973.
3. N. Kryloff and N. Bogoliuboff, Introduction to Non-Linear Mechanics, Princeton University Press, 1947.
4. P. J. Cefola, "Equinoctial Orbit Elements—Application to Artificial Satellite Orbits," AIAA Paper No. 72-937, Palo Alto, 1972.
5. H. W. Small, Minimum Fuel Time-Free Transfer Between Elliptic Orbits, Ph.D. Dissertation, Stanford University, 1972.
6. H. W. Small, "Minimum N-Impulse Time-Free Transfers Between Elliptic Orbits," AIAA J, Vol. 9 No. 4, April, 1971.
7. T. N. Edelbaum, "Minimum Impulse Transfers in the Near Vicinity of a Circular Orbit," J. Astronautical Sciences, Vol. 14, No. 2, pp 66-73, March-April, 1967.
8. System/360 Scientific Subroutine Package Version IV, Programmers Manual, GH20-0205-4, IBM Corp., 5th Ed., 1970.
9. W. G. Melbourne et. al., Constants and Related Information for Astrodynamic Calculations, 1968, NASA Technical Report 32-1306, JPL, Pasadena, 1968.

10. R. H. Battin, Astronautical Guidance, McGraw-Hill, New York, 1964.
11. V. M. Blanco and S. W. McCuskey, Basic Physics of the Solar System, Addison-Wesley, Reading, Massachusetts, 1961.

APPENDIX A

FLOW CHARTS

Three flow charts are included on the following pages. The first is an overall flow chart showing the general flow of the code. The chart is indicative of either the Newton-Raphson or modified Newton-Raphson iterator. The second chart shows the flow of the trajectory calculations, and the third shows the operations taken in calculating the averaged derivative which is called by the integrator routine which extrapolates the low thrust portion of the trajectory.



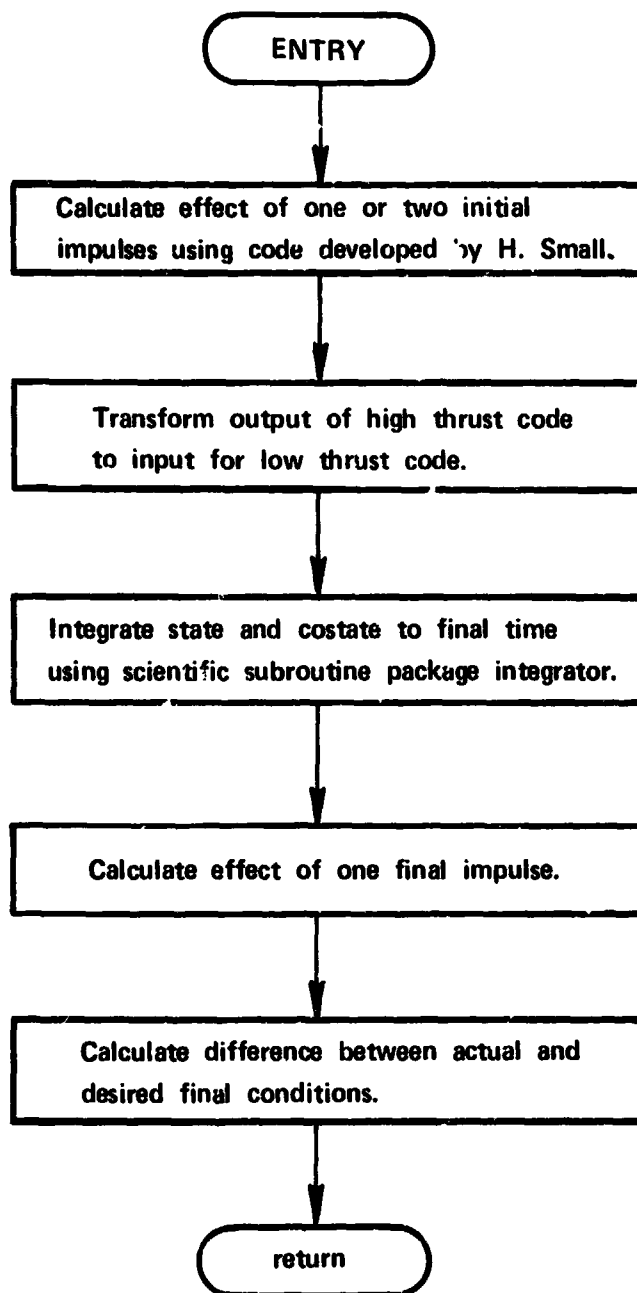


Fig. A-2. Trajectory Flowchart

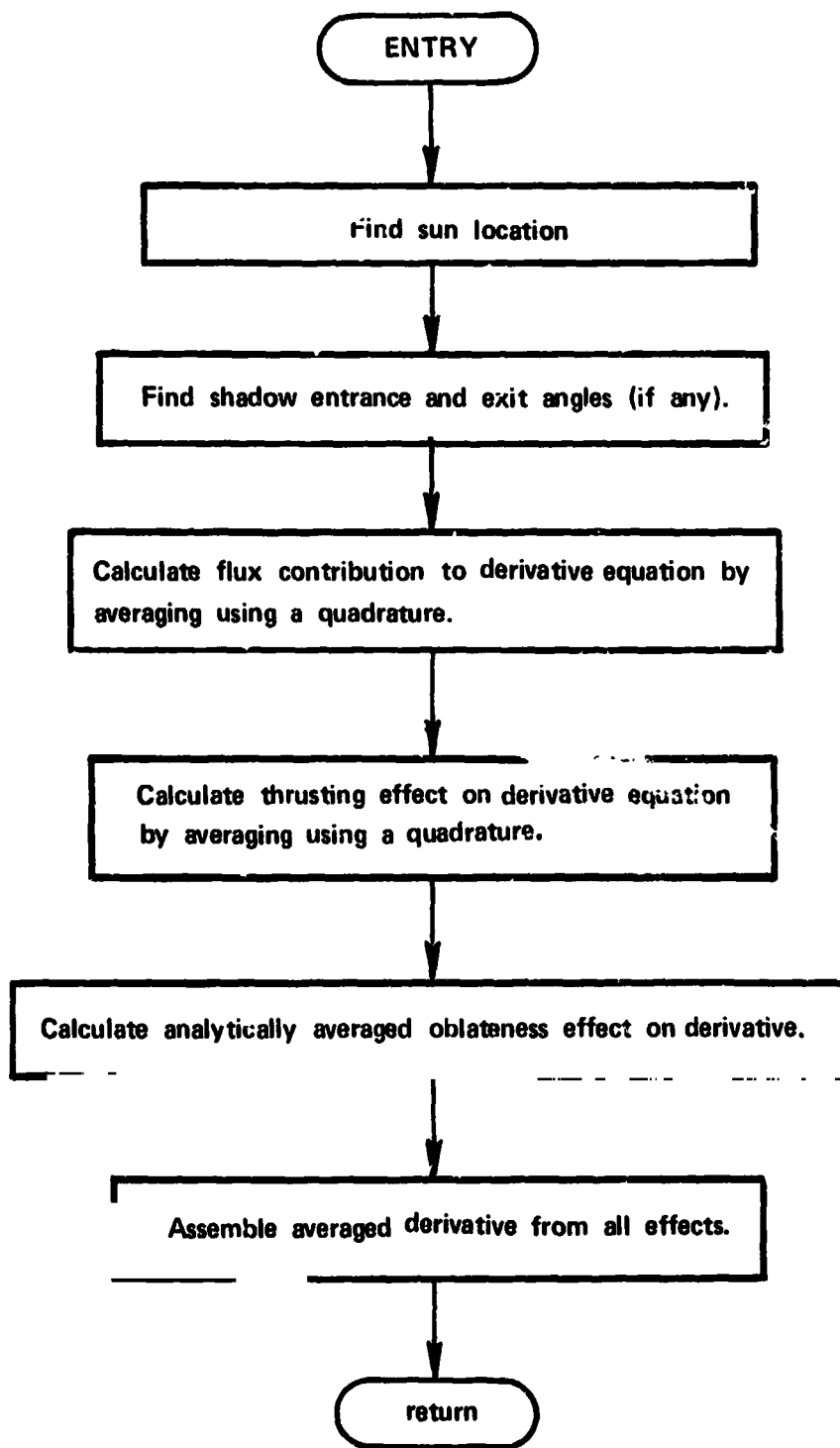


Fig. A-3. Averaged Derivative Calculation Flowchart

APPENDIX B

THRUST DIRECTION ON AN ORBIT

There is no provision in the main deck for printing the thrust direction at points on an individual orbit, prior to the averaging process. This print can be obtained for particular orbits, however, by using a separate main, calling, program, ORBIT, which uses a special version of FCT and EVALMP which contain print statements. The input includes:

Z (I), I = 1, 10, The 5 equinoctial orbital elements, a, h, k, p, q, and their adjoints $\lambda_a, \lambda_h, \lambda_k, \lambda_p, \lambda_q$ which can be taken from the time history output of the optimization program.

A Thrust acceleration, may be set to 1.

N One half the number of points on the orbit for which print is desired.

NCONT, Flag, if 0, stop, if greater than 0 read data for another orbit.

Z and A are in format (F25, 15) and N, NCONT in format (I2).

The output is as follows. Beginning at F (the eccentric longitude) = 0° , at points $180^\circ/N$ apart, the following is printed:

the eccentric longitude, F, the equinoctial coordinates, X1, Y1 (the Z component is zero) and the unit vector indicating the thrust direction in the equinoctial coordinate system.

APPENDIX C

EXAMPLE OF A SEP RUN

This appendix shows an example of actual output for a case including SEP and initial and final impulses. Only the beginning and the end of the iteration print and the time history print is included. The run utilizes modified Newton-Raphson iterator and a 16 point quadrature.

OPTIMUM TRAJECTORY PROGRAM FOR SATELLITE LAUNCH USING SEP AND HIGH THRUST

PERFORM TIME

INITIAL TRAJECTORY DATA

1. ALL INITIAL POSITION DATA ARE SPECIFIED
 2. ALL INITIAL VELOCITY DATA ARE SPECIFIED
 3. ALL INITIAL ACCELERATION DATA ARE SPECIFIED

INITIAL DATA SUMMARY

INITIAL POSITION (X,Y,Z) = 0.000000000000 0.000000000000 0.000000000000 FINAL POSITION (X,Y,Z) = 5000.0000000000

INITIAL VELOCITY DATA SUMMARY

INITIAL VELOCITY (Vx, Vy, Vz) = 0.000000000000 0.000000000000 0.000000000000
 INITIAL ACCELERATION (Ax, Ay, Az) = 0.000000000000 0.000000000000 0.000000000000

INITIAL MASS DATA

INITIAL MASS (M) = 1000.0000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000
 INITIAL MASS FLOW RATE (M-dot) = 0.000000000000
 INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000
 INITIAL MASS FLOW RATE (M-dot) = 0.000000000000
 INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL MASS FLOW RATE (M-dot) = 0.000000000000

INITIAL TIME SUMMARY
 INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS
 INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS
 INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

INITIAL TIME (T) = 0.000000000000 SECONDS = 0.000000000000 UNITS

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

1 1

1. 2. 3. 4. 5. 6. 7. 8. 9. 10. 11. 12. 13. 14. 15. 16. 17. 18. 19. 20. 21. 22. 23. 24. 25. 26. 27. 28. 29. 30. 31. 32. 33. 34. 35. 36. 37. 38. 39. 40. 41. 42. 43. 44. 45. 46. 47. 48. 49. 50. 51. 52. 53. 54. 55. 56. 57. 58. 59. 60. 61. 62. 63. 64. 65. 66. 67. 68. 69. 70. 71. 72. 73. 74. 75. 76. 77. 78. 79. 80. 81. 82. 83. 84. 85. 86. 87. 88. 89. 90. 91. 92. 93. 94. 95. 96. 97. 98. 99. 100. 101. 102. 103. 104. 105. 106. 107. 108. 109. 110. 111. 112. 113. 114. 115. 116. 117. 118. 119. 120. 121. 122. 123. 124. 125. 126. 127. 128. 129. 130. 131. 132. 133. 134. 135. 136. 137. 138. 139. 140. 141. 142. 143. 144. 145. 146. 147. 148. 149. 150. 151. 152. 153. 154. 155. 156. 157. 158. 159. 160. 161. 162. 163. 164. 165. 166. 167. 168. 169. 170. 171. 172. 173. 174. 175. 176. 177. 178. 179. 180. 181. 182. 183. 184. 185. 186. 187. 188. 189. 190. 191. 192. 193. 194. 195. 196. 197. 198. 199. 200. 201. 202. 203. 204. 205. 206. 207. 208. 209. 210. 211. 212. 213. 214. 215. 216. 217. 218. 219. 220. 221. 222. 223. 224. 225. 226. 227. 228. 229. 230. 231. 232. 233. 234. 235. 236. 237. 238. 239. 240. 241. 242. 243. 244. 245. 246. 247. 248. 249. 250. 251. 252. 253. 254. 255. 256. 257. 258. 259. 260. 261. 262. 263. 264. 265. 266. 267. 268. 269. 270. 271. 272. 273. 274. 275. 276. 277. 278. 279. 280. 281. 282. 283. 284. 285. 286. 287. 288. 289. 290. 291. 292. 293. 294. 295. 296. 297. 298. 299. 300. 301. 302. 303. 304. 305. 306. 307. 308. 309. 310. 311. 312. 313. 314. 315. 316. 317. 318. 319. 320. 321. 322. 323. 324. 325. 326. 327. 328. 329. 330. 331. 332. 333. 334. 335. 336. 337. 338. 339. 340. 341. 342. 343. 344. 345. 346. 347. 348. 349. 350. 351. 352. 353. 354. 355. 356. 357. 358. 359. 360. 361. 362. 363. 364. 365. 366. 367. 368. 369. 370. 371. 372. 373. 374. 375. 376. 377. 378. 379. 380. 381. 382. 383. 384. 385. 386. 387. 388. 389. 390. 391. 392. 393. 394. 395. 396. 397. 398. 399. 400. 401. 402. 403. 404. 405. 406. 407. 408. 409. 410. 411. 412. 413. 414. 415. 416. 417. 418. 419. 420. 421. 422. 423. 424. 425. 426. 427. 428. 429. 430. 431. 432. 433. 434. 435. 436. 437. 438. 439. 440. 441. 442. 443. 444. 445. 446. 447. 448. 449. 450. 451. 452. 453. 454. 455. 456. 457. 458. 459. 460. 461. 462. 463. 464. 465. 466. 467. 468. 469. 470. 471. 472. 473. 474. 475. 476. 477. 478. 479. 480. 481. 482. 483. 484. 485. 486. 487. 488. 489. 490. 491. 492. 493. 494. 495. 496. 497. 498. 499. 500. 501. 502. 503. 504. 505. 506. 507. 508. 509. 510. 511. 512. 513. 514. 515. 516. 517. 518. 519. 520. 521. 522. 523. 524. 525. 526. 527. 528. 529. 530. 531. 532. 533. 534. 535. 536. 537. 538. 539. 540. 541. 542. 543. 544. 545. 546. 547. 548. 549. 550. 551. 552. 553. 554. 555. 556. 557. 558. 559. 560. 561. 562. 563. 564. 565. 566. 567. 568. 569. 570. 571. 572. 573. 574. 575. 576. 577. 578. 579. 580. 581. 582. 583. 584. 585. 586. 587. 588. 589. 590. 591. 592. 593. 594. 595. 596. 597. 598. 599. 600. 601. 602. 603. 604. 605. 606. 607. 608. 609. 610. 611. 612. 613. 614. 615. 616. 617. 618. 619. 620. 621. 622. 623. 624. 625. 626. 627. 628. 629. 630. 631. 632. 633. 634. 635. 636. 637. 638. 639. 640. 641. 642. 643. 644. 645. 646. 647. 648. 649. 650. 651. 652. 653. 654. 655. 656. 657. 658. 659. 660. 661. 662. 663. 664. 665. 666. 667. 668. 669. 670. 671. 672. 673. 674. 675. 676. 677. 678. 679. 680. 681. 682. 683. 684. 685. 686. 687. 688. 689. 690. 691. 692. 693. 694. 695. 696. 697. 698. 699. 700. 701. 702. 703. 704. 705. 706. 707. 708. 709. 710. 711. 712. 713. 714. 715. 716. 717. 718. 719. 720. 721. 722. 723. 724. 725. 726. 727. 728. 729. 730. 731. 732. 733. 734. 735. 736. 737. 738. 739. 740. 741. 742. 743. 744. 745. 746. 747. 748. 749. 750. 751. 752. 753. 754. 755. 756. 757. 758. 759. 760. 761. 762. 763. 764. 765. 766. 767. 768. 769. 770. 771. 772. 773. 774. 775. 776. 777. 778. 779. 780. 781. 782. 783. 784. 785. 786. 787. 788. 789. 790. 791. 792. 793. 794. 795. 796. 797. 798. 799. 800. 801. 802. 803. 804. 805. 806. 807. 808. 809. 810. 811. 812. 813. 814. 815. 816. 817. 818. 819. 820. 821. 822. 823. 824. 825. 826. 827. 828. 829. 830. 831. 832. 833. 834. 835. 836. 837. 838. 839. 840. 841. 842. 843. 844. 845. 846. 847. 848. 849. 850. 851. 852. 853. 854. 855. 856. 857. 858. 859. 860. 861. 862. 863. 864. 865. 866. 867. 868. 869. 870. 871. 872. 873. 874. 875. 876. 877. 878. 879. 880. 881. 882. 883. 884. 885. 886. 887. 888. 889. 890. 891. 892. 893. 894. 895. 896. 897. 898. 899. 900. 901. 902. 903. 904. 905. 906. 907. 908. 909. 910. 911. 912. 913. 914. 915. 916. 917. 918. 919. 920. 921. 922. 923. 924. 925. 926. 927. 928. 929. 930. 931. 932. 933. 934. 935. 936. 937. 938. 939. 940. 941. 942. 943. 944. 945. 946. 947. 948. 949. 950. 951. 952. 953. 954. 955. 956. 957. 958. 959. 960. 961. 962. 963. 964. 965. 966. 967. 968. 969. 970. 971. 972. 973. 974. 975. 976. 977. 978. 979. 980. 981. 982. 983. 984. 985. 986. 987. 988. 989. 990. 991. 992. 993. 994. 995. 996. 997. 998. 999. 1000.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

INFO NO. TRAJECTORY CALC

4 12

X 4.4437782657087-07 3.7436447222790-01 -5.77548970097160-01 1.238222706567660 00 1.484763403516450-01
-1.674905242749607-05 -2.91644464095670 02

Y 4.0687919411589850-03 -3.942472560696130-04 -4.2408000251658320-04 -2.9362340892589590-04 8.9850648436607830-04
-1.3790723270403790-04 1.278164478615660-05 1.7246350056384100-03

TF = 1.7727151148758700 C3

FO = 2.048973743118590-05

INITIAL (X,Y) VELOCITY

-2.7772274269917500 01 -3.8724547980.6440 00 4.0489858309914850 00 9.8030807731482400-02 -4.5850097942719710-02
-1.888377701755670-04 -4.9425372898661150-04 1.3809103881386260-03
-8.407578199767970-01 -4.144447768539790-01 -7.8419562202031100-02 9.8195391204238680-03 3.324114782358130-04
1.5471444828823870-08 4.87044797998830-05 -1.2071685480302030-04
-4.842444351811750-01 4.933882199030300-02 6.8416466385731200-02 -1.7129004448181800-04 1.0380684497946560-02
4.1144278876438810-04 -4.844447768539790-01 -4.8345770334783150-05
3.4004887873047610 01 -1.8749777612688330-01 4.4444004442138240-01 5.0074853474788960-05 -3.8796544977397670-01
-2.7772274269917500 01 4.1779447777700 00 4.4637850126098830-05
-4.0899124110448820 00 2.180472671991670-01 -1.8140626010052360-01 -5.6983303579214460-04 1.0466244347488280 00
-4.433303030303000-04 -1.443771049399110-05 -2.3903974626955940-04
-2.844772274269917500 00 4.740277763769940-01 8.2310619679981600-02 8.8374872816779080-01 -5.47224922020285460-02
1.0300072470292770-01 4.144447768539790-01 1.0683658995544700-03
-1.5471444828823870 00 4.87044797998830-05 1.005271813007070-01 1.9646271794961420-01 -7.812452874209490-03
-1.290437167524760-05 4.9118047268750-04 6.2148255265650090-05
-2.279878527638310 00 -4.441162406077010-02 -9.8231162758814500-01 8.9261170715655240-01 -2.3190911149406470-01
-1.404447768539790-05 4.144447768539790-01 3.79557699441310-04

PERCENTAGE = -0.22867577248470-10

INITIAL X

-2.8004894661841220-04
1.244817087944380-03
8.87777221338480-04
-4.3444410010350990-04
-1.444447768539790-01
5.7777777777777770 00
1.444447768539790-01

DE TF = -6.7894047277777770 C3

FO = 1.177447133148007-05

INFO NO. TRAJECTORY CALC

4 13

X 4.4437782657087-07 3.7436447222790-01 -5.77548970097160-01 1.238222706567660 00 1.484763403516450-01
-1.674905242749607-05 -2.91644464095670 02

Y 4.148047106115410-04 3.21442771055540-07 2.105299136097760-06 9.6335403439046030-07 1.4556914024203160-05
-1.02044715776427-05 -2.07340775037510-06 -0.707705513114830-05

TF = 1.177447133148007-05

FO = 1.177447133148007-05

FO = 1.177447133148007-05

CONVERGENCE VALUES FOR INTEGRATED TRAJECTORY

INITIAL FINAL VELOCITY COMPONENTS ARE

F (X,Y) X (DEG) I (DEG) I (ON ASC NODE (DEG)) ARG PERIG (DEG)
9.57402444898600 04 4.000077145.8840-01 1.8210353644688860 01 -1.1659153134752080 01 -8.5815114548318580 01
X (X,Y) P (X) P (Y) P (Z)
4.4437782657087-07 3.7436447222790-01 -5.77548970097160-01 1.238222706567660 00 1.484763403516450-01

THE ERROR IN THE FINAL P, Q, IS (ACTUAL - DESIRED)

F (X,Y) X (DEG) I (DEG) I (ON ASC NODE (DEG)) ARG PERIG (DEG)
2.444447768539790-01 2.2777777777777770 00 3.520448885139930-04
F (X,Y) P (X) P (Y) P (Z) SUB-T(PHASE) (DEG)
4.148047106115410-04 3.21442771055540-07 2.105299136097760-06 9.6335403439046030-07 1.4556914024203160-05

THE CONVERGENCE (INITIAL) DIFFERENCE PARAMETERS ARE
4.4437782657087-07 3.7436447222790-01 -5.77548970097160-01 1.238222706567660 00 1.484763403516450-01
-1.02044715776427-05 -2.07340775037510-06 -0.707705513114830-05

THE INTEGRATED FINAL TIME IS
1.08427091207610 C3 DAYS = 9.386695513.00570 C5 SECONDS = 1.163426406760490 03 UNITS

EQUIVALENT PERTURBATION (M, N) = 0.0252362891183220-01

FINAL MASS = 977.99448707071070 KG, FRACTION OF INITIAL MASS = 0.972994782306

FINAL POWER = 12.72 37417403100 KM, FRACTION OF INITIAL POWER = 0.933982049050279

FINAL THRUST (X,Y) = 0.0473444444444440-01 KM/SEC

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

TIME HISTORY OF ORBITAL TRAJECTORY

HIGH THRUST

ORBIT # 1

POSITIONAL P.E. AND STATE V.F. #1000

| | | | | | | |
|---------------------|-----------------------|----------------------|-------|------|--------------------|----------------------|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.00013400255000-01 | -1.042741121633610-U1 | -5.29654083935010-U2 | J.0 | J.0 | 2.5567646674944-U1 | -0.05095426818590-U1 |

VELOCITY P.E.

| | | | | | | |
|---------------------|------|---------|-------|------|----------------------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.85000000000000-U1 | J.0 | J.0 | J.0 | J.0 | -1.06039435488869-U2 | |

THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0

ORBIT # 2

POSITIONAL P.E. AND STATE V.F. #1000

| | | | | | | |
|---------------------|----------------------|----------------------|----------------------|---------------------|----------------------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 3.42004742155000-01 | -7.02774474929710-U1 | -1.49786281471030-U1 | -1.04345371792170-U2 | 2.54929536122010-U1 | -0.05399109959540-U1 | |

VELOCITY P.E.

| | | | | | | |
|---------------------|---------------------|---------------------|----------------------|----------------------|----------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.43648741175000-04 | 7.30671121436670-U1 | 2.41971720256621-U1 | -2.38118557311185-U1 | -1.03668311656790-U2 | | |

THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0

ORBIT # 3

POSITIONAL P.E. AND STATE V.F. #1000

| | | | | | | |
|---------------------|----------------------|----------------------|----------------------|---------------------|----------------------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 4.14041421976717-00 | -6.70739467366320-U1 | -1.25178394659026-U1 | -5.34201736132040-U2 | 2.07511373296113-U1 | -0.04216363453610-U1 | |

VELOCITY P.E.

| | | | | | | |
|---------------------|---------------------|---------------------|----------------------|----------------------|----------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | | |

THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0

LOW THRUST

| TIME | TIME UNITS | SECONDS | HOURS | DAYS | DV (K/S) | A |
|---|---------------------|----------------------|----------------------|----------------------|----------------------|----------------------|
| THE POSITIONAL ORBITAL ELEMENTS, MASS, AND FLUX ARE | 4.14041421976717-00 | -6.70739467366320-U1 | -1.25178394659026-U1 | -5.34201736132040-U2 | 2.07511373296113-U1 | -0.04216363453610-U1 |
| VELOCITY P.E. | 2.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0 | | | | | | |
| THE ACCELERATION P.E., MASS (M), FLUX (K), THRUST (N), THRUST ACC (1/SEC^2) | 7.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| VELOCITY P.E. | 2.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0 | | | | | | |
| THE ACCELERATION IS | 7.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| VELOCITY P.E. | 2.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0 | | | | | | |
| THE ACCELERATION OF THE STATE IS | 7.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| VELOCITY P.E. | 2.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0 | | | | | | |
| THE ACCELERATION OF THE STATE IS | 7.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| VELOCITY P.E. | 2.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | |
| THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0 | | | | | | |

TIME HISTORY OF ORBITAL TRAJECTORY

HIGH THRUST

ORBIT # 1

POSITIONAL P.E. AND STATE V.F. #1000

| | | | | | | |
|---------------------|-----------------------|----------------------|-------|------|--------------------|----------------------|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.00013400255000-01 | -1.042741121633610-U1 | -5.29654083935010-U2 | J.0 | J.0 | 2.5567646674944-U1 | -0.05095426818590-U1 |

VELOCITY P.E.

| | | | | | | |
|---------------------|------|---------|-------|------|----------------------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.85000000000000-U1 | J.0 | J.0 | J.0 | J.0 | -1.06039435488869-U2 | |

THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0

ORBIT # 2

POSITIONAL P.E. AND STATE V.F. #1000

| | | | | | | |
|---------------------|----------------------|----------------------|----------------------|---------------------|----------------------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 3.42004742155000-01 | -7.02774474929710-U1 | -1.49786281471030-U1 | -1.04345371792170-U2 | 2.54929536122010-U1 | -0.05399109959540-U1 | |

VELOCITY P.E.

| | | | | | | |
|---------------------|---------------------|---------------------|----------------------|----------------------|----------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.43648741175000-04 | 7.30671121436670-U1 | 2.41971720256621-U1 | -2.38118557311185-U1 | -1.03668311656790-U2 | | |

THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0

ORBIT # 3

POSITIONAL P.E. AND STATE V.F. #1000

| | | | | | | |
|---------------------|----------------------|----------------------|----------------------|---------------------|----------------------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 4.14041421976717-00 | -6.70739467366320-U1 | -1.25178394659026-U1 | -5.34201736132040-U2 | 2.07511373296113-U1 | -0.04216363453610-U1 | |

VELOCITY P.E.

| | | | | | | |
|---------------------|---------------------|---------------------|----------------------|----------------------|----------|---|
| TIME UNITS | C.C. | SECONDS | HOURS | DAYS | DV (K/S) | A |
| 2.44097346124800-04 | 4.42151409262400-U1 | 2.41406942574170-U1 | -1.49250007126650-U1 | -0.00022379516660-U1 | | |

THIS ORBIT #1 (U1) UPSILON (U0) DELTA V (K/S) J.0

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

TIME KEPT IN SHADOW = 1.4712257 HOURS, FRACTION OF PERIOD = 0.0892182

ENTRY ANGLE = 117.041325447 EXIT ANGLE = 139.651796644

| TIME | TIME UNITS | SECONDS | HOURS | DAYS | UV (KFS) | N |
|-----------|------------|---------------|---------------|---------------|---------------|---|
| 1.100643P | CS | 8.1493478D 05 | 2.2637C77D 02 | 9.4321155D 00 | 6.9359422D-01 | 6 |

THE POSITIONAL CORRECTION ELEMENTS, MASS, AND FLUX ARE

| | | | | |
|-------------------|--------------------|---------------------|---------------------|--------------------|
| 1.42487927811D 00 | -5.8417A9A7229D-01 | -6.850547323514D-02 | -4.262850023273D-02 | 1.488399671152C-01 |
| 0.73437128194D-01 | 4.45540072761D-01 | | | |

THE ACCEPTED P.E., MASS (KFS), CORN (KFS), THRUST (N), THRUST ACC (1/SEC**2)

| | | | | |
|--------------------|-------------------|--------------------|---------------------|---------------------|
| 1.307969184255D 04 | 5.8917C7C2346D-01 | 1.975865946671D 01 | -1.416983186C73D 01 | -6.251860574616D 01 |
| 0.73437128194D 02 | 1.77740075236D 01 | 9.36114706506D-01 | 9.524658302611D-04 | |

THE COEFFICIENT

| | | | | |
|--------------------|--------------------|-------------------|--------------------|---------------------|
| 2.251079249122D 02 | 1.45517127468D 05 | 2.27497827418D 06 | 2.196987666678D 03 | -8.051053294632C 03 |
| -1.47221074102D 02 | -0.82477407146D 00 | | | |

THE DERIVATIVE OF THE STATE IS

| | | | | |
|--------------------|-------------------|--------------------|--------------------|---------------------|
| 1.55874057744D-03 | 1.14777777777D-04 | 4.307186683326D-05 | 1.612479781471D-05 | -4.515374363240D-05 |
| -2.17711406435D-05 | 1.45540075236D-01 | | | |

THE DERIVATIVE OF THE STATE IS

| | | | | |
|--------------------|--------------------|--------------------|-------------------|---------------------|
| -1.11713482777D-01 | -6.88487777777D-02 | -8.77389467360D-02 | -5.0898557493D-01 | -2.064326996205D 00 |
| 0.73437128194D-01 | 4.45540075236D-01 | | | |

| INITIAL TIME | PERIOD (HRS) | PERIGEE (KFS) | APOGEE (KFS) | DIV. TIME STEP |
|----------------|--------------------|------------------|-----------------|----------------|
| 1.000000000000 | 17.91577777777D 02 | 1.3991879429D 04 | 5.399504256D 04 | 0 |

TIME KEPT IN SHADOW = 1.74181839 HOURS, FRACTION OF PERIOD = 0.07287314

ENTRY ANGLE = 120.202000000 EXIT ANGLE = 138.872007625

| TIME | TIME UNITS | SECONDS | HOURS | DAYS | UV (KFS) | N |
|-----------|------------|---------------|---------------|---------------|---------------|---|
| 1.7342647 | CS | 9.3866555D 05 | 2.6074154D 02 | 1.0864231D 01 | 8.0473865D-01 | 7 |

THE POSITIONAL CORRECTION ELEMENTS, MASS, AND FLUX ARE

| | | | | |
|--------------------|--------------------|--------------------|--------------------|--------------------|
| 1.473747731865D 00 | -5.8417A9A7229D-01 | -6.22747095620D-02 | -4.01310789013D-02 | 1.416474356700D-01 |
| 0.73437128194D-01 | 4.45540075236D-01 | | | |

THE ACCEPTED P.E., MASS (KFS), CORN (KFS), THRUST (N), THRUST ACC (1/SEC**2)

| | | | | |
|-------------------|-------------------|-------------------|--------------------|--------------------|
| 1.67078848442D 04 | 5.8917C7C2346D-01 | 1.87121356308D 01 | -1.39425238665D 01 | -6.23620552550D 01 |
| 0.73437128194D 02 | 1.77740075236D 01 | 9.33459041941D-01 | 9.59882035443D-04 | |

THE COEFFICIENT

| | | | | |
|--------------------|--------------------|-------------------|-------------------|--------------------|
| 2.12724900327D 02 | 1.45517127468D 05 | 2.18620240506D 02 | 2.09211733364D 03 | -8.42743874056C 03 |
| -1.15207447137D-02 | -2.05740075236D-05 | | | |

THE DERIVATIVE OF THE STATE IS

| | | | | |
|--------------------|-------------------|-------------------|-------------------|---------------------|
| 1.55874057744D-03 | 1.14777777777D-04 | 3.80465458304D-05 | 1.63546128743D-05 | -4.877609198704D-05 |
| -2.17711406435D-05 | 1.45540075236D-01 | | | |

THE DERIVATIVE OF THE STATE IS

| | | | | |
|--------------------|--------------------|---------------------|---------------------|--------------------|
| -0.74218427877D-02 | -7.08477777777D-02 | -1.741945153816D-02 | -9.077697052473D-01 | -2.96063561223D 01 |
| 1.07384848484D 00 | 4.41490000000D-02 | | | |

| INITIAL TIME | PERIOD (HRS) | PERIGEE (KFS) | APOGEE (KFS) | DIV. TIME STEP |
|----------------|------------------|------------------|-----------------|----------------|
| 0.000000000000 | 18.940810560D 00 | 1.5221524132D 04 | 5.534424578D 04 | 0 |

TIME KEPT IN SHADOW = 0.6740756 HOURS, FRACTION OF PERIOD = 0.0526571

ENTRY ANGLE = 124.771743043 EXIT ANGLE = 137.594338443

HIGH THRUST FINISH POINT

POSITIONAL CORRECTION ELEMENTS, MASS, AND FLUX ARE

| | | | | |
|---------------------|--------------------|--------------------|----------------------|---------------------|
| 1.6078796138205D 00 | -5.8417A9A7229D-01 | -7.28453699600D-02 | -3.2388164402200D-02 | 1.5699950580154D-01 |
| 0.73437128194D-01 | 4.45540075236D-01 | | | |

ACCEPTED P.E., MASS (KFS), CORN (KFS), THRUST (N), THRUST ACC (1/SEC**2)

| | | | | |
|---------------------|-------------------|---------------------|-----------------------|-----------------------|
| 1.8740244444444D 04 | 5.8917C7C2346D-01 | 1.8210353646887D 01 | -1.16591591347521D 01 | -8.58151145483186D 01 |
| 0.73437128194D 02 | 1.77740075236D 01 | 9.33459041941D-01 | 9.59882035443D-04 | |

COEFFICIENT

| | | | | |
|--------------------|--------------------|-------------------|-------------------|--------------------|
| 2.12724900327D 02 | 1.45517127468D 05 | 2.18620240506D 02 | 2.09211733364D 03 | -8.42743874056C 03 |
| -1.15207447137D-02 | -2.05740075236D-05 | | | |

DERIVATIVE OF THE STATE IS

| | | | | |
|--------------------|-------------------|-------------------|-------------------|---------------------|
| 1.55874057744D-03 | 1.14777777777D-04 | 3.80465458304D-05 | 1.63546128743D-05 | -4.877609198704D-05 |
| -2.17711406435D-05 | 1.45540075236D-01 | | | |

DERIVATIVE OF THE STATE IS

| | | | | |
|--------------------|--------------------|---------------------|---------------------|--------------------|
| -0.74218427877D-02 | -7.08477777777D-02 | -1.741945153816D-02 | -9.077697052473D-01 | -2.96063561223D 01 |
| 1.07384848484D 00 | 4.41490000000D-02 | | | |

| INITIAL TIME | PERIOD (HRS) | PERIGEE (KFS) | APOGEE (KFS) | DIV. TIME STEP |
|----------------|------------------|------------------|-----------------|----------------|
| 0.000000000000 | 18.940810560D 00 | 1.5221524132D 04 | 5.534424578D 04 | 0 |

POSITIONAL CORRECTION ELEMENTS, MASS, AND FLUX ARE

APPENDIX D

EXAMPLE OF A NEP RUN

This appendix shows an example of actual output for a case including NEP and initial high thrust. Only the beginning and the end of the iteration print and the time history print is included. This run utilized the modified Newton-Raphson iterator, the predictor-corrector integrator, and the 16 point quadrature.

OPTIMAL TRAJECTORY PROGRAM FOR NEP AND HIGH THRUST SATELLITE RAISING

MINIMUM TIME NEP

FINAL CONDITION OPTIONS

1. ALL 5 FINAL ORBITAL ELEMENTS SPECIFIED
 2. A.F.I. SPECIFIED, LON ASC NONE AND ARG PERI FREE
 FOR THIS RUN, OPTION = 2

TOTAL DELT FOR INITIAL IMPULSE (M/S) = 1405.803900000000 FINAL IMPULSE (M/S) = 0.0

THE INITIAL ORBITAL ELEMENTS ARE

| A (KM) | E | I (DEG) | LON ASC NONE (DEG) | ARG PERIG (DEG) |
|-----------------------|-----|-----------------------|--------------------|-----------------|
| 6.5414700000000000 03 | 0.0 | 2.8500000000000000 01 | 0.0 | |

INITIAL A (EARTH RADII) = 1.02905333707530 00

THE DESIRED FINAL ORBITAL ELEMENTS ARE

| A (KM) | E | I (DEG) | LON ASC NONE (DEG) | ARG PERIG (DEG) |
|---|-----------------------|-----------------------|--------------------|-----------------|
| 4.2158140000000000 04 | 1.0000000000000000-04 | 1.7900000000000000-01 | | |
| P (EARTH PERI) COST(IMP)=2444421 SQRT(IMP)=404923 | | | | |
| 6.409764314441280 10 | 1.0000000000000000-04 | 8.726644475712710-04 | | |

INITIAL DIRECTION FOR HIGH THRUST, C.F. ONE
 -1.8170000000000000-01 4.7130000000000000-11 -2.1840000000000000-01 1.1830000000000000 00 1.0000000000000000 00

SIGN OF INITIAL LAMDA I IS -1.0

FINAL TIME (ESTIMATE)

4.3333333333333333 01 DAYS = 4.7151999999999999 06 SECONDS = 4.604774438247110 03 UNITS

CRUISE IMP = 1.0000000000000000 02 EFF. CRUISE VEL = 2.479470035214990 01 KM/S

INITIAL MASS (KG) = 1000.000000000000

INITIAL POWER (KW) = 1.4447000000000000 01

ACCELERATION LEVEL = 9.9997417420346140-04 M/SEC/SEC = 1.0705678189167990-04 G'S

J2 = 1.08770000-03

INITIAL TIME IS 0.0 DAYS = 0.0 SECONDS = 0.0 UNITS

TERMIN IS 1.0 DAYS = 0.0 SECONDS = 0.0 UNITS

TERMIN IS 1.0000000000000000 02 DAYS = 1.6416000000000000 07 SECONDS = 2.0546678835504540 04 UNITS

STEP SIZE FOR NUMERICAL DIFFERENTIATION, KSTEP = 1
 1.0000000000000000-05 1.0000000000000000-05 1.0000000000000000-04 1.0000000000000000-04

TIME STEP FOR INTEGRATION
 1.3333333333333333 03 DAYS = 2.5920000000000000 05 SECONDS = 3.2126335003428220 02 UNITS

NUMBER OF POINTS FOR INTEGRATION = 1.0000000000 10

WEIGHTS FOR INTEGRATION ARE
 1.000000 1.000000 1.000000 0.0 0.0 0.0 0.0 0.0 0.0

DIMENSION = 10

MAXIMUM NUMBER OF ITERATION = 20

NOISE LIMIT IN ITER = 1.000000-08

J2 = 1.08270000-03

1. PERTURBATION = 4.1751400000000000 KM

NO IMP = 398601.2000000000 KM^3/SEC^2

UTM = 0.7014000000000000 UTW = 306.8167706095579 UTH = 0.93381333433894000-02 UTM52 = 9.798233450716636 .UTW5

ITERATION 1000

ITER NO. TRAJ. CALLS

X -1.6130000000000000-01 4.7130000000000000-01 -2.1840000000000000-01 1.1837000000000000 00 1.0000000000000000 00

Y -2.0067100000000000-01 5.2107758922102690-02 1.2079912417463440-02 6.5640468896787000-04 0.0
-6.4162700000000000-03

TF= 6.40677661142471100 03

FO= 1.1651100012701900-02

PARTIAL DERIV MATRIX

-1.6129000000000000-01 5.2106678196991330-02 1.2080330178273750-02 6.5547195200675970-04 0.0
-2.8059000000000000-01
-6.4162700000000000-03

4.7131000000000000-01 5.2101091449577570-02 1.2091479750081610-02 6.56273272827490-04 0.0
-2.805817444703100-01
-6.4131397290974800-03

-2.1819000000000000-01 5.210707702669120-02 1.207995974447880-02 6.5670248572356800-04 0.0
-2.8061593106760670-01
-6.4162700000000000-03

1.1831000000000000 00 5.2107758922102690-02 1.2079912417463440-02 6.5640468896787000-04 0.0
-2.806713944148150-01
-6.4162700000000000-03

1.0000000000000000 00 5.2107758922102690-02 1.2079912417463440-02 6.5640468896787000-04 0.0
-2.806713944148150-01
-6.4162700000000000-03

PARTIAL DERIV MATRIX

3.1799515103350 01 4.05476446462900 00 5.461922747691180 00 0.0
1.449021022574450-03
9.2027644761055100-01 -5.867477720429470-01 2.1815026664198930-01 0.0

-2.6057540246642570-04 1.9673376179681790-01 4.6827015221136190-02 0.0
4.1771081019690690-01
-6.4052577822560930-04

-1.186516900223150 00 -1.118656856703560-02 7.6241682676192950-01 6.5640468896787000-04 0.0
-3.0657786265548200-04
0.0

0.0 0.0 0.0 0.0 1.0000000000000000 00
-6.79625751187700-01 6.091073191013350-01 -6.378454324468200-02 9.9358377066247060-01 0.0
3.121810449876160-05

DETERMINANT = 1.1764975364475040-03

DELTA X

-3.02794777396650-04
1.71707026617750-03
-5.62478974786190-04
-2.067761012731570-03
0.0

DEL TF = 1.004703049077670 33

FI= 2.3616223816213250-05

ITER NO. TRAJ. CALLS

2 2

X -1.6130000000000000-01 4.7130000000000000-01 -2.1840000000000000-01 1.1837000000000000 00 1.0000000000000000 00

Y -4.5147149711102110-02 1.7547586742929410-03 -2.7661785980574880-04 -8.7846774036586610-05 0.0
3.493488476583700-04

TF= 6.7067030333448700 03

FO= 2.3616223816213250-05

PARTIAL DERIV MATRIX

3.1799515103350 01 5.2105768443761420 33 5.461922747691180 00 2.1011548910342000-10 0.0
1.449021022574450-03
9.2027644761055100-01 -5.867477720429470-01 2.1815026664198930-01 -0.0111737280912930-10 0.0

-2.617119902918350-04 1.9673376179681790-01 4.6827015221136190-02 1.2482577807707050-11 0.0
4.1771081019690690-01
-6.776006072733550-04

-1.186516900223150 00 -1.1718656764974500-03 7.6241682676192950-01 6.5640468896787000-04 0.0
-3.0644416709732600-04
0.0

0.0 0.0 0.0 0.0 1.0000000000000000 00
-6.79625751187700-01 6.091073191013350-01 -6.378454324468200-02 9.9358377066247060-01 0.0
3.1060449432001580-05

DETERMINANT = 1.1500661679084040-03

DELTA X

-1.757971519005200-04
2.0248762443725630-01
-6.608021779101200-05
-1.7400137444950010-03
0.0

DEL TF = 1.5510760249129000 33

FI= 1.2745988736358630-05

ITER NO. TRAJ. CALLS

3 3

X -1.6130000000000000-01 4.7130000000000000-01 -2.1840000000000000-01 1.1837000000000000 00 1.0000000000000000 00

Y 1.2590100576071960-04 2.5489319221587600-02 -5.8041051918577830-06 1.3312251669122920-05 0.0
2.541515140881410-04

TF= 6.7067569807354100 03

FO= 1.2745988736358630-05

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

PARTIAL PERIODE MATRIX

| | | | | |
|------------------------|-----------------------|-----------------------|------------------------|-----------------------|
| 3.119685154725037-01 | 4.014576737899327-00 | 5.463362920979012-00 | 6.4718799003507987-08 | 0.0 |
| 1.1947845116265627-07 | -5.467447688473537-01 | 2.181503684454247-01 | -2.1785941071104257-04 | 0.0 |
| 9.127740857120770-01 | 1.2133652573119900-01 | 4.682701477978970-02 | 4.184135238801580-09 | 0.0 |
| 4.1771081026762947-01 | 1.947332470740200-01 | 4.682701477978970-02 | 4.184135238801580-09 | 0.0 |
| -8.8396730619128997-05 | -1.166516903968627-00 | -3.171864979202115-07 | 7.874168272902696-01 | 0.0 |
| -1.166516903968627-00 | 0.0 | 0.0 | 0.0 | 1.0000000000000000 00 |
| -1.864826110653527-06 | 0.0 | 0.0 | 0.0 | 0.0 |
| 0.0 | -6.090109936915170-01 | -6.378659871876570-02 | 9.0258263341655077-01 | 0.0 |
| -6.2962576251894437-01 | 1.0106677877827220-04 | | | |

DELTA PERIODE = -3.4076794966475740-03

DELTA PERIODE

| | | | | |
|------------------------|--|--|--|--|
| -1.3982876735046937-05 | | | | |
| 7.975940764789150-05 | | | | |
| -1.800484848474848-05 | | | | |
| -6.3066660116672657-05 | | | | |
| 0.0 | | | | |

DELTA PERIODE = 1.257121294367390-01

PERIODE = 1.3557281736290917-10

PERIODE

| | |
|---|----|
| 7 | 12 |
|---|----|

X

| | | | | |
|-----------------------|----------------------|------------------------|-----------------------|-----------------------|
| -1.814150506786997-01 | 4.742770647766967-01 | -2.1845405726623627-01 | 1.1814367552981266-00 | 1.0000000000000000 00 |
|-----------------------|----------------------|------------------------|-----------------------|-----------------------|

Y

| | | | | |
|-----------------------|------------------------|------------------------|-----------------------|-----|
| -7.264094303168077-12 | -1.1542366208531507-05 | -6.1359874202952467-09 | 2.4292375643389547-08 | 0.0 |
| 8.125991827948367-07 | | | | |

PERIODE = 4.703030303030303-03

PERIODE = 1.3557281736290909-10

PERIODE = 1.3557281736290909-10

COMPARISON VALUES FOR INITIAL AND FINAL PERIODE

INITIAL PERIODE INITIAL PERIODE

| X (MM) | Y (MM) | Z (MM) | PERIODE (SEC) | PERIODE (MM) | PERIODE (MM) |
|-----------------------|-----------------------|------------------------|------------------------|-----------------------|--------------|
| 4.314155564731145-04 | 1.240733701466447-05 | 0.0950067685227305-02 | -2.8496595300447010-01 | 1.998308566967937-07 | 0 |
| 6.6097676076362387-00 | 1.3429410088986007-05 | -6.7513476700952195-05 | -6.1902462502767630-04 | 7.6547246835238280-04 | 0 |

THE DIFFERENCE IN THE PERIODE IS (ACTUAL - DESIRED)

| X (MM) | Y (MM) | Z (MM) | PERIODE (SEC) | PERIODE (MM) | PERIODE (MM) |
|------------------------|----------------------|------------------------|---------------|--------------|--------------|
| -6.4356789534467347-03 | -1.15933672093155-05 | -6.8731477271653590-07 | | | |

PERIODE (MM)

| | | | | | |
|-----------------------|-----------------------|------------------------|--|--|--|
| -7.264094303168077-12 | 1.1542366208531507-05 | -6.1359874202952467-09 | | | |
|-----------------------|-----------------------|------------------------|--|--|--|

THE DIFFERENCE IN INITIAL PERIODE AND FINAL PERIODE

| | | | | |
|------------------------|----------------------|------------------------|-----------------------|-----------------------|
| -1.2141530306246944-11 | 4.742770647766967-01 | -2.1845405726623627-01 | 1.1814367552981266-00 | 1.0000000000000000 00 |
|------------------------|----------------------|------------------------|-----------------------|-----------------------|

THE MEAN PERIODE FINAL TIME IS

4.4737187630973977-01 DAYS = 4.8452005377161570 ON SECONDS = 4.7908038867728510 ON UNITS

FINAL MASS = 848.50484848484848 PER CENT OF INITIAL MASS = 0.363504948607217

FINAL POWER = 14.4973000000000000 KW FRACTION OF INITIAL POWER = 1.0000000000000000

LOW THRUST PERIODE = 4.144055420709543-00 KW/SEC

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

TIME HISTORY OF ORBITAL TRAJECTORY

HIGH THRUST

ORBIT # 1

EQUINOCTIAL POS. AND CONSTANTS, P = 1000, U = 0, 0.0, 2.539676444749440-01, -5.315417852239657-01

CLASSICAL POS. 0.0, 0.0, 2.850000000000000 01, 0.0, -2.32469192246190 02

TIME SCALING, DELTA V (KM/S), 0.0, 0.0, 0.0, 0.0

ORBIT # 2

EQUINOCTIAL POS. AND CONSTANTS, P = 1000, U = 0, 0.0, 2.539676444749440-01, -5.315417852239657-01

CLASSICAL POS. 0.0, 0.0, 2.850000000000000 01, 0.0, -2.32469192246190 02

TIME SCALING, DELTA V (KM/S), 0.0, 0.0, 0.0, 0.0

LOW THRUST

TIME UNITS, SECONDS, HOURS, DAYS, DV (K/S), N

THE DERIVATIVE ELEMENTS ARE LISTED, FOLLOWED BY CL. IN KM E DEG, 1.00526702000000 00, -1.424305444441-01, 1.5032420792690-01, 1.92701565337780-03, 2.1544449814710-01

1.007140474000 04, 1.000000000000-01, 2.6511636469700 01, 1.9273407057280 00, -2.5443007774180 01

MASS (KG), POWER (KW), THRUST (N), THRUST ACC (M/S^2), 1.001000000000 00, 1.400000000000 01, 0.00007617420150-01, 7.00007617420150-04

THE ACCELERATION, 1.703750453730 03, -0.870713471463 00, 2.0290641267780 01, -2.111246566610 02, -4.2740101184300 03

THE POSITIVES OF THE DERIVATIVE ELEMENTS, 3.740066741000-04, 5.431145057410-05, 1.6712721475040-05, -2.2119733965140-05, -2.914519787470-05

THE POSITIVES OF THE DERIVATIVE ELEMENTS, -2.184610540270-04, -2.476577160000-04, -2.097537507010-01, 2.005742017400 00, -1.500654314840-01

PERIOD (HRS), PERIOD (DAYS), 3.8623070000 00, 1.6264674200 01, 4.56740001140 01, 1.4279268130 04, 0

TIME UNITS, SECONDS, HOURS, DAYS, DV (K/S), N

1.1343300 00, 2.5000000 01, 2.2000000 01, 1.0000000 00, 2.6074999-01, 3

THE DERIVATIVE ELEMENTS ARE LISTED, FOLLOWED BY CL. IN KM E DEG, 1.707600100000 00, -1.247013713067-01, 3.541746151280-01, -1.1651747145640-02, 2.2444447754000-01

1.14333314000 04, 1.755107135000-01, 2.435514207770 01, -2.9491004450350 00, -1.6447780483870 01

MASS (KG), POWER (KW), THRUST (N), THRUST ACC (M/S^2), 0.114720404444 00, 1.447700000000 01, 0.0007617420150-01, 1.0007617420150-04

THE ACCELERATION, 1.481120151530 03, -1.863764669030 01, -5.140107061170 01, 3.8483770013040 02, -4.2976622380220 03

THE POSITIVES OF THE DERIVATIVE ELEMENTS, 4.140074101650-04, 2.601370411740-05, 5.960851836700-04, -5.158434941200-05, -3.7158536810400-05

THE POSITIVES OF THE DERIVATIVE ELEMENTS, -4.270754344470-01, -2.164799445937-02, -2.3846634445740-01, 1.6104444453810 00, -7.4856429902000-03

PERIOD (HRS), PERIOD (DAYS), 0.875961175101, 1.374924378344, 2.13948722920 01, 1.47281191100 04, 0

TIME UNITS, SECONDS, HOURS, DAYS, DV (K/S), N

4.4742670 02, 5.1849000 01, 1.4400000 02, 4.0000000 00, 5.23013100-01, 3

THE DERIVATIVE ELEMENTS ARE LISTED, FOLLOWED BY CL. IN KM E DEG, 1.015640474000 00, -1.012746589440-01, 3.5452003184000-01, -2.5093596466340-02, 2.1200250374150-01

1.73437031000 04, 1.407712137350-01, 2.410149126740 01, -6.750395579400 00, -4.499914779600 00

MASS (KG), POWER (KW), THRUST (N), THRUST ACC (M/S^2), 0.124655449911 00, 1.447700000000 01, 0.0007617420150-01, 1.010277403550-03

THE ACCELERATION, 1.144747412130 03, -1.192157407500 01, -1.732476107190 02, 4.444993251067 02, -4.288181137690 03

THE DERIVATIVES OF THE ORBITAL ELEMENTS ARE
 1.60557701200-01 3.3427740045120-05 -1.9756077073140-04 1.98 967032230-05 -6.9046717114121-05

THE DERIVATIVE OF THE COSTATE IS
 -1.166712066470-01 2.6077172105050-02 -1.2300154501700-01 3.5610736721340-02 -2.4331851308970-02

HAMILTONIAN PERIOD (HRS) PERIGEE (KMS) APOGEE (KMS) DIV. TIME STEP
 3.900960618210 19.011747474199 3.07644952600 04 4.16131277440 04 0

TIME TIME UNIT SECONDS HOURS DAYS DV (K/S) N
 4.44744601 03 3.62640000 06 1.00800000 03 4.70000000 01 1.07301970 00 15

THE ORBITAL ELEMENTS ARE (EO), FOLLOWED BY CL IN KM & DEG
 6.176070455300 00 -1.1660271047770-02 7.6719717029610-02 -6.8345372705120-03 1.9392108629810-02

MASS (KG) POWER (KW) THRUST (N) THRUST ACC (M/S**2)
 6.7655711667380 02 1.4657000000000 01 9.4997617420350-01 1.1408062960980-03

THE COSTATE IS
 1.748316957770 02 2.2147767152610 02 -1.37930 333530 03 2.1592524679220 03 -6.2866772649100 03

THE DERIVATIVES OF THE ORBITAL ELEMENTS ARE
 1.6517897406150-01 3.8707731194400-05 -2.4467895377340-04 1.1490246389900-05 -6.2056379007520-05

THE DERIVATIVE OF THE COSTATE IS
 -2.140910215600-02 -2.798709942320-02 5.4304744724120-03 2.3476763427680-02 -1.0874960410440-02

HAMILTONIAN PERIOD (HRS) PERIGEE (KMS) APOGEE (KMS) DIV. TIME STEP
 0.900927478050 21.613175739704 3.63345113730 04 4.74497897630 04 0

TIME TIME UNIT SECONDS HOURS DAYS DV (K/S) N
 4.44474450 03 3.74706590 06 1.04084600 03 4.37689820 01 4.00822540 00 16

THE ORBITAL ELEMENTS ARE (EO), FOLLOWED BY CL IN KM & DEG
 6.104477246630 00 -5.4897017670500-03 7.9511177670500-02 -3.6834478974220-03 1.0200379129380-02

MASS (KG) POWER (KW) THRUST (N) THRUST ACC (M/S**2)
 6.7252043261550 02 1.4657000000000 01 9.4997617420350-01 1.1408062960980-03

THE COSTATE IS
 1.2139139658540 02 2.1873277762400 02 -1.3742646273380 03 2.1623393959760 03 -6.2877608012880 03

THE DERIVATIVES OF THE ORBITAL ELEMENTS ARE
 1.4789949667800-03 3.0974022971070-05 -2.6764930557820-04 2.1967820786120-05 -6.3144985033400-05

THE DERIVATIVE OF THE COSTATE IS
 -1.0487469996700-02 -3.7071221444440-02 6.2631002107110-02 1.8701762896190-02 -3.8071710399490-03

HAMILTONIAN PERIOD (HRS) PERIGEE (KMS) APOGEE (KMS) DIV. TIME STEP
 0.900927478050 21.613175739704 3.63345113730 04 4.74497897630 04 0

TIME TIME UNIT SECONDS HOURS DAYS DV (K/S) N
 4.70000000 03 3.94240000 06 1.07240000 03 4.47772600 01 4.14409580 00 17

THE ORBITAL ELEMENTS ARE (EO), FOLLOWED BY CL IN KM & DEG
 6.1057675076360 00 1.367509008360-04 -8.745134767000-05 -4.1902462502770-04 7.6547246839240-04

MASS (KG) POWER (KW) THRUST (N) THRUST ACC (M/S**2)
 6.645864666730 02 1.4657000000000 01 9.4997617420350-01 1.1413778977680-03

THE COSTATE IS
 1.063972968420 02 2.1190486337460 02 -1.3608896894720 03 2.1647444552040 03 -6.2877548291420 03

THE DERIVATIVES OF THE ORBITAL ELEMENTS ARE
 1.4449790165200-03 4.0507471012740-05 -3.7719411157270-04 1.1117267008380-05 -6.5475114743270-05

THE DERIVATIVE OF THE COSTATE IS
 -1.4031915056680-02 -5.3281444736270-02 1.1983902816440-02 1.4740758779340-02 4.0709340114770-02

HAMILTONIAN PERIOD (HRS) PERIGEE (KMS) APOGEE (KMS) DIV. TIME STEP
 1.073900812994 23.924376702038 4.21546782670 04 4.72618224570 04 0

PROGRAM HAS RUN SUCCESSFULLY

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR.

APPENDIX E

SOURCE CODE LISTINGS

The source code listings for all the subprograms are given in this appendix. There are five principle groupings. The first is used only for SEP, the second with NEP, the third is a common group used both with SEP and NEP, the fourth includes optional versions of ITER, TRAJ, and QUAD and finally the fifth group includes the driver program ORBIT and special versions of FCT and EVAMP containing print statements which are used to get thrust direction at points on particular orbits.

The SEP group includes:

- INPUT
- OUTPC
- ITER (modified Newton Raphson)
- PRTN
- DCROUT
- TRAJ (Runga-Kutta version)
- OUTP
- FUNCT
- EARTH (non-inclined magnetic axis)
- SUN
- SHADOW
- DQRTIC
- DCUBIC
- FLUX

The NEP group includes:

- INPUT
- OUTPC
- ITER (modified Newton-Raphson)

The NEP group includes:

PRTN
DCROUT
TRAJ (Runga-Kutta version)
OUTP
FUNCT

The common group includes:

CONTL
OBLATE
QUAD (16 point)
FCT
EVALMP
MAINE
START
TIME
SWITCH
DTDU
OUTH
IMPLS
YF

The option group includes:

ITER (Newton-Raphson for SEP)
ITER (Newton-Raphson for NEP)
TRAJ (predictor-corrector for SEP)
TRAJ (predictor-corrector for NEP)
QUAD (4 point)
QUAD (8 point)
QUAD (32 point)
EARTH (inclined magnetic axis)

The thrust direction group includes:

ORBIT

FCT (print version)

EVALMP (print version)

| | | |
|---|----------------------------------|----------|
| C INPUT/INPUTS | | 00000010 |
| C | | 00000020 |
| C SEP AND HIGH THRUST. | | 00000030 |
| C THIS SUBPROGRAM IS CALLED BY CONTE AND READS AND PRINTS | | 00000040 |
| C ALL INITIAL DATA AS WELL AS SETS INITIAL CONSTANTS. | | 00000050 |
| C THE UNITS ARE BASED ON INTERNAL MU=1.0, INTERNAL DISTANCE | | 00000060 |
| C UNIT=1 EARTH RADII, AND EXTERNAL MU= 398601.2 KM*KM*KM/ | | 00000070 |
| C SEC*SEC AND EARTH RADII= 6378.16 KM. A CIRCULAR | | 00000080 |
| C ORBIT AT 1 EARTH RADII WOULD HAVE A PERIOD OF 2 PI INTERNAL | | 00000090 |
| C TIME UNITS. | | 00000100 |
| C 14 DIMENSIONAL WITH MASS AND FLUX | | 00000110 |
| C INPUT | | 00000120 |
| C LOW/HIGH | HIGH/LOW/HIGH | 00000130 |
| C | INITIAL ORBIT | 00000140 |
| C A (KM) | | 00000150 |
| C E | SET TO ZERO | 00000160 |
| C I (DEG) | | 00000170 |
| C LONG. OF NODE (DEG) | NOT USED | 00000180 |
| C ARG. OF PERIGEE (DEG) | NOT USED | 00000190 |
| C MASS (KG) | | 00000200 |
| C FLUX | | 00000210 |
| C | INITIAL GUESSES | 00000220 |
| C LAMBDA A | LIKE UPSILON | 00000230 |
| C LAMBDA H | LIKE SMALL'S K | 00000240 |
| C LAMBDA K | LIKE SMALL'S J | 00000250 |
| C LAMBDA P | SCALE FACTOR | 00000260 |
| C LAMBDA Q | NODAL ANGLE (RAD) OR ITS ADJOINT | 00000270 |
| C LAMBDA M | | 00000280 |
| C LAMBDA N | | 00000290 |
| C | FINAL ORBIT | 00000300 |
| C A | | 00000310 |
| C E | | 00000320 |
| C I | | 00000330 |
| C NODE (NOT USED IF NOP=2) | NOT USED | 00000340 |
| C PERIGEE (NOT USED IF NOP=2) | NOT USED | 00000350 |
| C | | 00000360 |
| C TF2 (DAYS), GUESS FOR FINAL TIME | | 00000370 |
| C PKW (KW), POWER | | 00000380 |
| C SPIM (SEC), SPECIFIC IMPULSE OF SEP | | 00000390 |
| C TL JULIAN DATE AT INITIAL TIME | | 00000400 |
| C IRDFLG | NOMINAL | 00000410 |
| C 1 END OF INPUT | | 00000420 |
| C 2 IPR, PRINT FLAG | 0 | 00000430 |
| C 3 NIMAX, MAX. NO. OF ITERATIONS | 20 | 00000440 |
| C 4 TFMAX2 (DAYS), MAX. TF | 190. | 00000450 |
| C 5 DT2 (DAYS), TIME STEP FOR D.E. | 1. | 00000460 |
| C 6 UEB, UPPER ERROR BOUND FOR D.E. | 1.010 | 00000470 |
| C 7 EW, ERROR WEIGHTS FOR D.E. | 1.,1.,1.,1.,0.,... | 00000480 |
| C 8 R ETKM, EQUATORIAL EARTH RADIUS (KM) | 6378.16 | 00000490 |
| C 9 GM (KM**3/SEC**2) EARTH GRAV.CONST. | 398601.2 | 00000500 |
| C 10 NOP = 1, 5 O.E. SPECIFIED AT TF | 1 | 00000510 |
| C = 2, 3 O.E. SPECIFIED AT TF | | 00000520 |
| C 11 SETS OBLATENESS. AJ2. = 1.08270-3 | 0. | 00000530 |
| C 12 STEP(8), STEP SIZE FOR NUMERICAL DIFF. | 1.0-6 | 00000540 |
| C KSTEP = 0, STEP AS FRACTION IN ITER | 0 | 00000550 |
| C = 1, STEP AS CONSTANT IN ITER | | 00000560 |
| C 13 ISON = 0, SHADOW EFFECT OFF | 0 | 00000570 |

```

C          = 1. SHADOW EFFECT ON                00000580
C 14 ISUN = 0 SUN DIST EFFECT ON POWER OFF      0      00000590
C          = 1 EFFECT ON                        00000600
C PH (KW) HOUSEKEEPING POWER                   0.      00000610
C 15 IHI = 1, LOW THRUST ONLY                    1      00000620
C          = 2, HIGH/LOW                        00000630
C          = 3, HIGH/LOW/HIGH                   00000640
C          = 4, LOW/HIGH                         00000650
C DVI1 (M/S) TOTAL INITIAL HIGH THRUST DEL V   0.      00000660
C DVI2 (M/S) DEL V FOR FINAL IMPULSE           0.      00000670
C NODE = 0, INITIAL LINE OF NODES FREE         00000680
C          = 1, INITIAL LINE OF NODES FIXED    00000690
C          NODE MEANINGFUL ONLY IF IHI= 2 OR 3 00000700
C 16 IPOW = 0, CONSTANT POWER                    1      00000710
C          = 1, DEGRADATION EFFECT             00000720
C 17 FLIM, NORM LIMIT IN ITERATION ROUTINE     1.0-6    00000730
C 18 SGN = -1., INITIAL LAM I IS NEGATIVE      00000740
C          = +1., INITIAL LAM I IS POSITIVE    00000750
C                                                00000760
C                                                00000770
C SUBROUTINE INPUT                               00000780
C                                                00000790
C IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)        00000800
C                                                00000810
C NAMELIST /UN/UTKM,UTS,UTD,UTKG,UTKW,UTMS2     00000820
C                                                00000830
C COMMON /XMM/ZLO(7), STFP(R), ZERF(R)          00000840
C COMMON /ELEM/ZPO(7), ZPF(5)                   00000850
C COMMON /INT/ITF,IPR,IDIM,IOIM2,NIMAX          00000860
C COMMON /TRA/TFMA>.DT,UER,FW(14)              00000870
C COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,UTKG,UTKW,UTMS2 00000880
C COMMON /T/TF,S,TC,TFMIN                       00000890
C COMMON /A/A,AMU,PI                             00000900
C COMMON /WF/WF(5)                               00000910
C COMMON /J2/ AJ2                                00000920
C COMMON /TC/NOP                                 00000930
C COMMON /JD/TL                                  00000940
C COMMON /POWER/PO,C,POW,PH,ISUN,ISON,IPOW     00000950
C COMMON /HIGH/DVI1,DVI2,IHI                    00000960
C COMMON /ACOM/AF(10)                            00000970
C COMMON /F/FLIM,KSTEP                           00000980
C COMMON /NOD/NODE                               00000990
C COMMON /SG/SGN                                 00010000
C                                                00010100
C DIMENSION W(7)                                00010200
C                                                00010300
C INTEGER CONSTANTS                             00010400
C                                                00010500
C IDIM=14                                        00010600
C IOIM2=7                                       00010700
C IOIM3=8                                       00010800
C IPR=0                                          00010900
C ITF=3                                          00011000
C NIMAX=20                                      00011100
C NOP= 1                                        00011200
C ISON= 0                                       00011300
C ISUN= 0                                       00011400

```

```

      IMI= 1
      IPOW= 1
      KSTEP= 0
C
C REAL CONSTANTS
C
      AJ2= 0.00
      AMU=1.000
      UEB= 1.00+10
      DO 10 I=1,10IM2
        EW(I+10IM2)= 0.00
10      EW(I)= 1.000
        EW(6)= 0.00
        EW(7)= 0.00
      DO 12 I= 1,10IM3
12      STEP(I)= 1.0-6
        DT2= 1.000
        GM=398601.200
        UTKM= 6378.1600
        DTR= .01745329251994329600
        PI= 3.141592653589793200
        TFMAX2= 190.000
        TFMIN2= 0.000
        T02=0.000
        TL= 0.00
        PH= 0.00
        FLIM= 1.0-6
        DV11= 0.00
        DV12= 0.00
        AF(1)= 7.202
        AF(2)= -1.402
        AF(3)= -1.8301
        AF(4)= 2.0601
        AF(5)= 9.0501
        AF(6)= -9.1601
        AF(7)= -2.27336D3
        AF(8)= 5.97087D3
        AF(9)= -4.25392D3
        AF(10)= 1.19494D-21
C
C ALL READ STATEMENTS FOLLOW
C
      READ (5,1000) (W(I),I= 1,10IM2)
      READ (5,1000) (ZLO(I),I=1,10IM2)
      READ (5,1000) (WF(I),I= 1,5)
      SGN= WF(3)-W(3)
      IF (SGN.NE.0.00) SGN= SGN/DABS(SGN)
      READ (5,1001) TF2
      READ (5,1001) PKW
      READ (5,1001) SPIM
      READ (5,1001) TL
C
      TL MUST RE BETWEEN ABOUT A.D. 1950 AND 2000
      IF ((TL.LT.2.433D6).OR.(TL.GT.2.452D6)) GO TO 230
20  READ (5,1002) IRDFLG
      IF ((IRDFLG.GT.18).OR.(IRDFLG.LT.1)) GO TO 200
25  GO TO (150,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48),
1  IRDFLG

```

```

00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710

```

| | | |
|----|--|----------|
| C | | 00001720 |
| C | THESE VALUES ARE READ ONLY IF INDICATED BY IPR=FLG | 00001730 |
| C | | 00001740 |
| 32 | READ (5,1002) IPR | 00001750 |
| | IF (IPR.LT.0) GO TO 210 | 00001760 |
| | GO TO 20 | 00001770 |
| 33 | READ (5,1002) NIMAX | 00001780 |
| | IF ((NIMAX.LT.0).OR.(NIMAX.GT.50)) GO TO 220 | 00001790 |
| | GO TO 20 | 00001800 |
| 34 | READ (5,1001) TFMX2 | 00001810 |
| | IF ((TFMX2.LT.0.00).OR.(TFMX2.GT.1.03)) GO TO 220 | 00001820 |
| | GO TO 20 | 00001830 |
| 35 | READ (5,1001) DT2 | 00001840 |
| | IF ((DT2.LT.0.00).OR.(DT2.GT.1.03)) GO TO 220 | 00001850 |
| | GO TO 20 | 00001860 |
| 36 | READ (5,1001) UER | 00001870 |
| | GO TO 20 | 00001880 |
| 37 | READ (5,1003) EW | 00001890 |
| | GO TO 20 | 00001900 |
| 38 | READ (5,1001) UTKM | 00001910 |
| | GO TO 20 | 00001920 |
| 39 | READ (5,1001) GM | 00001930 |
| | GO TO 20 | 00001940 |
| 40 | READ (5,1002) NOP | 00001950 |
| | IF ((NOP.LT.1).OR.(NOP.GT.2)) GO TO 220 | 00001960 |
| | GO TO 20 | 00001970 |
| 41 | AJ2= 1.0827D-3 | 00001980 |
| | GO TO 20 | 00001990 |
| 42 | READ (5,1001) (STEP(I),I=1,10IM3) | 00002000 |
| | READ (5,1002) KSTEP | 00002010 |
| | IF ((KSTEP.LT.0).OR.(KSTEP.GT.1)) GO TO 220 | 00002020 |
| | GO TO 20 | 00002030 |
| 43 | READ (5,1002) ISUN | 00002040 |
| | IF ((ISUN.LT.0).OR.(ISUN.GT.1)) GO TO 220 | 00002050 |
| | GO TO 20 | 00002060 |
| 44 | READ (5,1002) ISUN | 00002070 |
| | IF ((ISUN.LT.0).OR.(ISUN.GT.1)) GO TO 220 | 00002080 |
| | READ (5,1001) PH | 00002090 |
| | GO TO 20 | 00002100 |
| 45 | READ (5,1002) IH1 | 00002110 |
| | IF ((IH1.LT.1).OR.(IH1.GT.4)) GO TO 220 | 00002120 |
| | READ (5,1001) DV11,DV12 | 00002130 |
| | IF (IH1.LT.3) DV12= 0.00 | 00002140 |
| | IF ((IH1.EQ.1).OR.(IH1.EQ.4)) DV11= 0.00 | 00002150 |
| | NOP= 2 | 00002160 |
| | READ (5,1002) NODE | 00002170 |
| | IF ((NODE.LT.0).OR.(NODE.GT.1)) GO TO 220 | 00002180 |
| | GO TO 20 | 00002190 |
| 46 | READ (5,1002) IPOW | 00002200 |
| | IF ((IPOW.LT.0).OR.(IPOW.GT.1)) GO TO 220 | 00002210 |
| | GO TO 20 | 00002220 |
| 47 | READ (5,1003) FLIM | 00002230 |
| | IF (FLIM.LT.0.00) GO TO 220 | 00002240 |
| | GO TO 20 | 00002250 |
| 48 | READ (5,1001) SGN | 00002260 |
| | IF (DABS(SGN).NE.1.00) GO TO 220 | 00002270 |
| | GO TO 20 | 00002280 |

```

C
C
C
C
C
C TIME VALUES ARE CHANGED FROM DAYS TO OTHER UNITS
C
150 UTS= DSQRT(UTKM**3/GM)
    UTM=UTS/60.00
    UTH=UTS/3600.00
    UTD=UTH/24.00
    TO= TO2/UTD
    TF= TF2/UTD
    TFMIN= TFMIN2/UTD
    TFMAX= TFMAX2/UTD
    DT= DT2/UTD
    T01= T0*UTS
    TF1= TF*UTS
    TFMIN1= TFMIN*UTS
    TFMAX1= TFMAX*UTS
    DT1= DT*UTS
C
    AF(10)= AF(10)*UTS
C
C MORE CONVERSIONS
C
    UTMS2= UTKM*1.03/(UTS**2)
    UTKG= 1.03
    UTKW= UTKG*UTMS2*UTKM/UTS
C
C
C
C CALL EARTH
C
C THE PRINTING OF ALL INITIAL VALUES FOLLOWS
C
    WRITE (6,2000)
    WRITE (6,2001)
    WRITE (6,2030)
    WRITE (6,2033)
    WRITE (6,2034)
    WRITE (6,2035) NOP
    IF (ISON.EQ.1) WRITE (6,2042)
    WRITE (6,2045) DV11,DV12
    WRITE (6,2002)
    WRITE (6,2003)
    IF ((IH1.EQ.1).OR.(IH1.EQ.4)) GO TO 158
C
    W(1) IS SEMIMAJOR AXIS
C
    W(2) IS ECC. SET TO 0
C
    W(3) IS INCLINATION
C
    W(4) IS OMEGA IF NODE= 1
C
    ZLO SHOULD BE Y1 (UP),Y2 (XK),YS (XJ),C,(LAM) OMEGA,LAM M, LAM N
C
    W(2)= 0.00
    III=3+NODE
    WRITE (6,2004) (W(I),I= 1,III)
    ZPO(1)= W(1)/UTKM

```

```

00002240
00002300
00002310
00002320
00002330
00002340
00002350
00002360
00002370
00002380
00002390
00002400
00002410
00002420
00002430
00002440
00002450
00002460
00002470
00002480
00002490
00002500
00002510
00002520
00002530
00002540
00002550
00002560
00002570
00002580
00002590
00002600
00002610
00002620
00002630
00002640
00002650
00002660
00002670
00002680
00002690
00002700
00002710
00002720
00002730
00002740
00002750
00002760
00002770
00002780
00002790
00002800
00002810
00002820
00002830
00002840
00002850

```

| | | |
|-----|---|----------|
| | ZPO(2)= W(3)*DTR | 00002860 |
| | WRITE (6,2046) ZPO(1) | 00002870 |
| | ZPO(6)= W(6)/UTKG | 00002880 |
| | IF (NODE.EQ.1) ZPO(3)= W(4)*DTR | 00002890 |
| | GO TO 162 | 00002900 |
| C | | 00002910 |
| 158 | WRITE (6,2004) (W(I),I=1,5) | 00002920 |
| C | | 00002930 |
| C | CHANGE FROM CLASSICAL O.F. TO EQUINOCTIAL O.E. | 00002940 |
| | DO 160 I=3,5 | 00002950 |
| 160 | W(I)= W(I)*DTR | 00002960 |
| | ZPO(1)= W(1)/UTKM | 00002970 |
| | ZPO(2)= W(2)*DSIN(W(5)+W(4)) | 00002980 |
| | ZPO(3)= W(2)*DCOS(W(5)+W(4)) | 00002990 |
| | ZPO(4)= DTAN(W(3)/2.000)*DSIN(W(4)) | 00003000 |
| | ZPO(5)= DTAN(W(3)/2.000)*DCOS(W(4)) | 00003010 |
| C | | 00003020 |
| | ZPO(6)= W(6)/UTKG | 00003030 |
| C | | 00003040 |
| C | | 00003050 |
| | WRITE (6,2005) | 00003060 |
| | WRITE (6,2004) (ZPO(I),I=1,5) | 00003070 |
| 162 | DV11= (DVI1/(UTMS2*UTS))*DSQRT(ZPO(1)/AMU) | 00003080 |
| | DV12= DV12/(UTMS2*UTS) | 00003090 |
| | ZPO(7)= W(7) | 00003100 |
| C | | 00003110 |
| | WRITE (6,2038) W(6) | 00003120 |
| | IF (IPOW.EQ.0) WRITE (6,2049) | 00003130 |
| | WRITE (6,2039) PKW | 00003140 |
| | WRITE (6,2043) PH | 00003150 |
| | WRITE (6,2040) W(7) | 00003160 |
| C | | 00003170 |
| | PO= PKW/UTKW | 00003180 |
| | PH= PH/UTKW | 00003190 |
| | IF (ISUN.NE.0) WRITE (6,2044) | 00003200 |
| C | | 00003210 |
| C | WRITE FINAL CONDITIONS, CHANGE TO EQUINOCTIAL FINAL COND. | 00003220 |
| C | | 00003230 |
| | WRITE (6,2006) | 00003240 |
| | ZPF(1)= WF(1)/UTKM | 00003250 |
| | GO TO (165,170), NOP | 00003260 |
| C | | 00003270 |
| 165 | WRITE (6,2003) | 00003280 |
| | WRITE (6,2004) (WF(I),I= 1,5) | 00003290 |
| | DO 166 I=3,5 | 00003300 |
| 166 | WF(I)= WF(I)*DTR | 00003310 |
| | ZPF(2)= WF(2)*DSIN(WF(5)+WF(4)) | 00003320 |
| | ZPF(3)= WF(2)*DCOS(WF(5)+WF(4)) | 00003330 |
| | ZPF(4)= DTAN(WF(3)/2.000)*DSIN(WF(4)) | 00003340 |
| | ZPF(5)= DTAN(WF(3)/2.000)*DCOS(WF(4)) | 00003350 |
| | DO 167 I=3,5 | 00003360 |
| 167 | WF(I)=WF(I)/DTR | 00003370 |
| | WRITE (6,2005) | 00003380 |
| | WRITE (6,2004) (ZPF(I),I=1,5) | 00003390 |
| | GO TO 190 | 00003400 |
| C | | 00003410 |
| 170 | ZPF(2)= WF(2) | 00003420 |

```

7PF(3)= DARS(DTAN(WF(3)*DTR/2.00)) 00003430
7PF(4)= 0.00 00003440
7PF(5)=0.00 00003450
WRITE (6,2031) 00003460
WRITE (6,2004) (WF(I),I=1,3) 00003470
WRITE (6,2032) 00003480
WRITE (6,2004) (7PF(I),I=1,3) 00003490
C 00003500
C 00003540
190 WRITE (6,2007) 00003550
196 WRITE (6,2011) ZLO 00003560
IF ((IH1.GT.1).AND.(IH1.LT.4)) WRITE (6,2050)SGN 00003570
WRITE (6,2008) 00003580
WRITE (6,2009) TF2,TF1,TF 00003590
C= SPIM/UTS 00003600
CC= C*UTMS2*UTS 00003610
WRITE (6,2012) SPIM,CC,C 00003620
A= UTMS2*2.00*(PO-PH)/(C*ZPO(6)) 00003630
WRITE (6,2041) A 00003640
WRITE (6,2037) TL 00003650
WRITE (6,2036) AJ2 00003660
WRITE (6,2013) 00003670
WRITE (6,2009) T02,T01,T0 00003680
WRITE (6,2014) 00003690
WRITE (6,2009) TFMIN2,TFMIN1,TFMIN 00003700
WRITE (6,2015) 00003710
WRITE (6,2009) TFMAX2,TFMAX1,TFMAX 00003720
WRITE (6,2010) KSTEP 00003730
WRITE (6,2011) STEP 00003740
WRITE (6,2016) 00003750
WRITE (6,2009) DT2,DT1,DT 00003760
WRITE (6,2017) UEB 00003770
WRITE (6,2018) 00003780
WRITE (6,2019) EW 00003790
WRITE (6,2020) IDIM 00003800
WRITE (6,2022) NIMAX 00003810
WRITE (6,2048) FLIM 00003820
WRITE (6,2026) UTKM 00003830
WRITE (6,2027) GM 00003840
WRITE (6,UN) 00003850
RETURN 00003860
200 WRITE (6,2023) IRDFLG 00003870
STOP 00003880
210 WRITE (6,2024) IPR 00003890
STOP 00003900
220 WRITE (6,2025) IRDFLG 00003910
STOP 00003920
230 WRITE (6,2037) TL 00003930
STOP 00003940
C 00003950
1000 FORMAT (F25.15) 00003960
1001 FORMAT (F25.15) 00003970
1002 FORMAT (I2) 00003980
1003 FORMAT (7D6.1) 00003990
1004 FORMAT (F18.11) 00004000
2000 FORMAT (1H1,22X,75H OPTIMAL TRAJECTORY PROGRAM FOR SATELLITE RAISI 00004010
ING USING SEP AND HIGH THRUST) 00004020

```


2001 FORMAT (1H0,53X,13H MINIMUM TIME) 00004030
2002 FORMAT (34H0 THE INITIAL ORBITAL ELEMENTS ARE) 00004040
2003 FORMAT (1H0,10X,6HA (KM),19X,1HE,20X,7HI (DEG),10X,18H LON ASC NODE 00004050
1 (DEG),6X,15H ARG PERIG (DEG)) 00004060
2004 FORMAT (1P5D23.14) 00004070
2005 FORMAT (1H0,6X,13HA (EARTH RAD),16X,1HH,22X,1HK,22X,1HP,22X,1HQ) 00004080
2006 FORMAT (40H0 THE DESIRED FINAL ORBITAL ELEMENTS ARE) 00004090
2007 FORMAT (32H0 INITIAL GUESSED PARAMETERS ARE) 00004100
2008 FORMAT (21H0 FINAL TIME ESTIMATE) 00004110
2009 FORMAT (1H ,10X,1PD22.15,7H DAYS =,1PD22.15,10H SECONDS =,1PD22.1500004120
1,6H UNITS) 00004130
2010 FORMAT (50H0 STEP SIZE FOR NUMERICAL DIFFERENTIATION, KSTEP =,12) 00004140
2011 FORMAT (1P5D23.14) 00004150
2012 FORMAT (12H0 SPEC IMP =, 1PD23.14,15H SEC, EXH VEL =,1PD23.14, 00004160
16H M/S =,1PD23.14, 10H F.R./T.U.) 00004170
2013 FORMAT (17H0 INITIAL TIME IS) 00004180
2014 FORMAT (10H0 TFMIN IS) 00004190
2015 FORMAT (10H0 TFMAX IS) 00004200
2016 FORMAT (27H0 TIME STEP FOR INTEGRATION) 00004210
2017 FORMAT (36H0 UPPER ERROR BOUND ON INTEGRATION =,1PD20.10) 00004220
2018 FORMAT (35H0 ERROR WEIGHTS FOR INTEGRATION ARE) 00004230
2019 FORMAT (1P14D8.1) 00004240
2020 FORMAT (13H0 DIMENSION =,15) 00004250
2022 FORMAT (31H0 MAXIMUM NUMBER OF ITERATION =,15) 00004260
2023 FORMAT (44H0 IRDFLG SHOULD BE BETWEEN 1 AND 16, IT IS =,15) 00004270
2024 FORMAT (28H0 IPR SHOULD BE < 0, IT IS =,15) 00004280
2025 FORMAT (27H0 BAD INPUT DATA, IRDFLG = ,13) 00004290
2026 FORMAT (17H0 1 EARTH RADIUS=,F25.12,3H KM) 00004300
2027 FORMAT (11H0 MU (GM) =,F25.10,13H KM**3/SEC**2) 00004310
2030 FORMAT (25H0 FINAL CONDITION OPTIONS) 00004320
2033 FORMAT (43H 1. ALL 5 FINAL ORBITAL ELEMENTS SPECIFIED) 00004330
2034 FORMAT (51H 2. A.E.1 SPECIFIED, LON ASC NODE AND ARG PER FREE) 00004340
2035 FORMAT (24H FOR THIS RUN, OPTION =,14) 00004350
2031 FORMAT (1H0,10X,6HA (KM),19X,1HE,20X,7HI (DEG)) 00004360
2032 FORMAT (1H0,6X,13HA (EARTH RAD),9X,15HSORT(H**2+K**2),8X, 00004370
1 15HSORT(P**2+Q**2)) 00004380
2037 FORMAT (32H0 JULIAN DATE AT INITIAL TIME IS, F20.8) 00004390
2036 FORMAT (6H0 J2 =,1PD15.7) 00004400
2038 FORMAT (21H0 INITIAL MASS (<G) =, F18.11) 00004410
2039 FORMAT (22H0 INITIAL POWER (KW) =, 1PD23.14) 00004420
2040 FORMAT (17H0 INITIAL FLUX = ,1PD23.14) 00004430
2041 FORMAT (24H0 INITIAL ACC (1 A.U.) =,1PD23.15,7H M/S**2) 00004440
2042 FORMAT (24H0 SHADOW EFFECT INCLUDED) 00004450
2043 FORMAT (27H0 HOUSEKEEPING POWER (KW) =,1PD23.14) 00004460
2044 FORMAT (39H0 SUN DISTANCE EFFECT ON POWER INCLUDED) 00004470
2045 FORMAT (41H0 TOTAL DELV FOR INITIAL IMPULSES (M/S) =, F20.12,5X, 00004480
1 21H FINAL IMPULSES (M/S) =,F20.12) 00004490
2046 FORMAT (27H0 INITIAL A (EARTH RAD)) =, 1PD25.14) 00004500
C2047 FORMAT (79H0 INITIAL GUESS FOR IMPULSE PARAMETERS, S.F., LONG. NOD 00004510
C 1E, MASS AND FLUX COSTATE) 00004520
2048 FORMAT (23H0 NORM LIMIT IN ITER = ,1PD12.5) 00004530
2049 FORMAT (22H0 NO POWER DEGRADATION) 00004540
2050 FORMAT (30H0 SIGN OF INITIAL LAMBDA I IS ,F6.1) 00004550
END 00004560

```

C OUTPC/OUTPCS                                00000010
C THIS SUBPROGRAM WRITES THE VALUES FOR THE FINAL (CONVERGED) 00000020
C TRAJECTORY. IT IS CALLED BY THE MAIN PROGRAM CONTL.          00000030
C TO BE USED WITH A DIM. ZERF.                                  00000040
C 14 DIMENSIONAL WITH MASS AND FLUX                             00000050
C                                                                00000060
C                                                                00000070
C                                                                00000080
C                                                                00000090
C SUBROUTINE OUTPC                                             00000100
C                                                                00000110
C IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)                      00000120
C                                                                00000130
C COMMON /XMM/ZLO(7),STEP(R),ZERF(R)                           00000140
C COMMON /Z/ZF(14),OZ(14)                                       00000150
C COMMON /T/TF,S,TC,TF*IN                                       00000160
C COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,UTKG,UTKW,UTMSZ       00000170
C COMMON /ELEM/ZPO(7),ZPF(5)                                       00000180
C COMMON /WF/WF(5)                                               00000190
C COMMON /A/A,AMU,P1                                             00000200
C COMMON /TC/NOP                                                 00000210
C COMMON /POWER/PO,C,POW,PH,ISUN,ISON,IPOW                      00000220
C                                                                00000230
C DIMENSION DELZF(5),DELWF(5),WFC(5)                            00000240
C                                                                00000250
C WFC(1)= ZF(1)*UTKM                                             00000260
C WFC(2)= 0.000                                                  00000270
C DUMMY= ZF(2)**2 + ZF(3)**2                                       00000280
C IF (DUMMY.GT.1.0D-40) WFC(2)=DSORT(DUMMY)                       00000290
C WFC(3)= 0.000                                                  00000300
C DUMMY= ZF(4)**2 + ZF(5)**2                                       00000310
C IF (DUMMY.GT.1.0D-40) WFC(3)= 2.000*DATAN(DSORT(DUMMY))/DTR   00000320
C WFC(4)=0.000                                                  00000330
C IF ((DABS(ZF(4)).GT.1.0D-8).AND.(DABS(ZF(5)).GT.1.0D-8))      00000340
1 WFC(4)= DATAN2(ZF(4),ZF(5))/DTR                                  00000350
C WFC(5)= 0.000                                                  00000360
C IF ((DABS(ZF(2)).GT.1.0D-8).AND.(DABS(ZF(3)).GT.1.0D-8))    00000370
1 WFC(5)=DATAN2(ZF(2),ZF(3))/DTR                                  00000380
C WFC(5)=WFC(5)-WFC(4)                                           00000390
C DO 10 I=1,5                                                    00000400
C   DELWF(I)= WFC(I) - WF(I)                                       00000410
10 DELZF(I)= ZF(I) - ZPF(I)                                       00000420
C   TF2= TF*UTD                                                    00000430
C   RMASS= ZF(6)/ZPO(6)                                             00000440
C   AMASS= ZF(6)*UTKG                                              00000450
C   RPOW= POW/PO                                                    00000460
C   AKWPOW= POW*UTKW                                               00000470
C   TF1= TF*UTS                                                    00000480
C   DELV= C*DLOG(ZPO(6)/ZF(6))*UTKM/UTS                          00000490
C                                                                00000500
C WRITE (6,3000)                                                  00000510
C WRITE (6,3001)                                                  00000520
C WRITE (6,3002) WFC                                              00000530
C WRITE (6,3003)                                                  00000540
C WRITE (6,3002) (ZF(I),I=1,5)                                    00000550
C WRITE (6,3004)                                                  00000560
C WRITE (6,3001)                                                  00000570

```

```

IF (NOP.EQ.1) WRITE (6,3002) DELWF                                00000580
IF (NOP.EQ.2) WRITE (6,3002) (DELWF(1),I=1,3)                    00000590
GO TO (20, 30), NOP                                              00000600
C                                                                    00000610
20 WRITE (6,3003)                                                 00000620
WRITE (6,3002) DELZF                                             00000630
GO TO 100                                                         00000640
C                                                                    00000650
30 DELZF(2)= DSQRT(7F(2)**2+7F(3)**2) -ZPF(2)                    00000660
DELZF(3)= DSQRT(7F(4)**2+7F(5)**2) -ZPF(3)                      00000670
WRITE (6,3011)                                                   00000680
WRITE (6,3002) (DELZF(1),I=1,3)                                  00000690
C                                                                    00000700
100 WRITE (6,3006)                                                00000710
WRITE (6,3002) ZLO                                               00000720
WRITE (6,3008)                                                    00000730
WRITE (6,3009) TF2,TF1,TF                                        00000740
WRITE (6,3012) ZF(7)                                             00000750
WRITE (6,3013) AMASS,RMASS                                       00000760
WRITE (6,3014) AKWPOW,RPOW                                       00000770
WRITE (6,3010) DELV                                              00000780
RETURN                                                            00000790
3000 FORMAT (35H0 ACTUAL FINAL ORBITAL ELEMENTS ARE)             00000800
3001 FORMAT (1H0,10X,6A (KM),18X,1HE,20X,7HI (DEG),10X,18HLON ASC NODE00000810
1 (DEG),6X,15HARG PERIG (DEG))                                  00000820
3002 FORMAT (1P5D23.15)                                          00000830
3003 FORMAT (1H0,5X,17HA (EARTH RAD),16X,1HH,22X,1HK,22X,1HP,22X,1HQ) 00000840
3004 FORMAT (51H0 THE ERROR IN THE FINAL O.E. IS (ACTUAL - DESIRED)) 00000850
3005 FORMAT (60H0 CLASSICAL O.E. MAY HAVE DISCREPANCY OF MULTIPLES OF 900000860
10 DEG)                                                         00000870
C                                                                    00000880
3006 FORMAT (46H0 THE CONVERGED INITIAL GUESSED PARAMETERS ARE) 00000880
3008 FORMAT (29H0 THE MINIMIZED FINAL TIME IS)                  00000890
3009 FORMAT (1H ,10X,1PD22.15,7H DAYS =,D22.15,10H SECONDS =,1PD22.15,600000900
1H UNITS)                                                       00000910
3010 FORMAT (18H0 LOW THRUST DELV=,1PD25.14,7H KM/SEC)          00000920
3011 FORMAT (1H0,5X,13HA (EARTH RAD),9X,15HSQRT(H**2+K**2),8X, 00000930
1 15HSQRT(P**2+Q**2))                                          00000940
3012 FORMAT (34H0 EQUIVALENT PARTICLES (*1.0-14) =,1PD22.15) 00000950
3013 FORMAT (14H0 FINAL MASS =,F22.15,31H KG, FRACTION OF INITIAL MASS 00000960
1=,F22.15)                                                     00000970
3014 FORMAT (15H0 FINAL POWER =,F22.15,32H KW, FRACTION OF INITIAL POWE00000980
1R =,F22.15)                                                   00000990
END                                                                00001000

```

| | | | | |
|----|---|--|--|------------|
| C | ITER | MODNRS | | 00000010 |
| C | MODNR/MODNRS | MODIFIED N-R ITERATOR | | 00000020 |
| C | | | | 00000030 |
| C | | | | 00000040 |
| C | HXR | VERSION | | 00000050 |
| C | | | | 00000060 |
| | SUBROUTINE ITER(KOUNT,NI,FUNCT,PRTN) | | | 00000070 |
| C | | | | 00000080 |
| | IMPLICIT REAL*(A-H,O-S) | | | 00000090 |
| C | | | | 00000100 |
| C | | | | 00000110 |
| C | X | VALUES OF THE INDEPENDENT VARIABLES(INITIAL,CURRENT,FINAL) | | DP00000120 |
| C | XS | STEP SIZE TO PERTURB X'S TO COMPUTE PARTIAL DERIVATIVES | | SP00000130 |
| C | Y | VALUES OF THE DEPENDENT VARIABLES(CURRENT,FINAL) | | DP00000140 |
| | | COMMON/XMMM/X(7),XS(M),Y(R) | | 00000150 |
| | | COMMON/INT/ITF,IPR,IDIM,IDIM2,MAXNOI | | 00000160 |
| | | COMMON/T/TF,S,TO,TFMIN | | 00000170 |
| | | COMMON/DY/DYDT(R) | | 00000180 |
| | | COMMON/F/FLIM,KSTEP | | 00000190 |
| C | | | | 00000200 |
| | DIMENSION YNOM(R),XN(7),P(R,B),COEF(R),DYDTN(B) | | | 00000210 |
| | | N=7 | | 00000220 |
| | | M=8 | | 00000230 |
| | | INORM=ITF | | 00000240 |
| | | IR=1 | | 00000250 |
| | | ICONS=1 | | 00000260 |
| | | ISW=0 | | 00000270 |
| | | NI=1 | | 00000280 |
| | | KOUNT=0 | | 00000290 |
| | | CALL FUNCT | | 00000300 |
| | | ITF=3 | | 00000310 |
| | | KOUNT=KOUNT+1 | | 00000320 |
| | | F0=0.00 | | 00000330 |
| | | DO 15 I=1,M | | 00000340 |
| 15 | | F0=F0+Y(I)**2 | | 00000350 |
| 9 | | DO 16 I=1,N | | 00000360 |
| | | DYDTN(I)=DYDT(I) | | 00000370 |
| | | XN(I)=X(I) | | 00000380 |
| | 16 | YNOM(I)=Y(I) | | 00000390 |
| | | YNOM(M)=Y(M) | | 00000400 |
| | | TFN=TF | | 00000410 |
| | | DYDTN(M)=DYDT(M) | | 00000420 |
| 10 | | CALL PRN(KOUNT,NI) | | 00000430 |
| | | WRITE(6,1011)F0 | | 00000440 |
| | | IF(F0.LE.FLIM)GO TO 90 | | 00000450 |
| | | IF(NI.GT.MAXNOI)GO TO 80 | | 00000460 |
| | | IF(ISW.NE.0)GO TO 27 | | 00000470 |
| C | COMPUTE NUMERICAL PARTIAL DERIVATIVES | | | 00000480 |
| | | DO 17 I=1,M | | 00000490 |
| 17 | | P(I,M)=DYDT(I) | | 00000500 |
| | | WRITE(6,1013) | | 00000510 |
| | | DO 25 J=1,N | | 00000520 |
| | | TEMP=X(J) | | 00000530 |
| | | STEP=XS(J)*DABS(X(J)) | | 00000540 |
| | | IF((DABS(X(J)).LT.1.D-10).OR.(KSTEP.EQ.1))STEP=XS(J) | | 00000550 |
| C | | IF(DABS(X(J)).LT.1.D-10)WRITE(6,1014) | | 00000560 |

| | |
|---|----------|
| X(J)=X(J)+STEP | 00000570 |
| CALL FUNCT | 00000580 |
| WRITE(6,1000)X(J) | 00000590 |
| WRITE(6,1001)(Y(I),I=1,M) | 00000600 |
| DO 20 I=1,M | 00000610 |
| 20 P(I,J)=(Y(I)-YNOM(I))/STEP | 00000620 |
| 25 X(J)=TEMP | 00000630 |
| KOUNT=KOUNT | 00000640 |
| 27 WRITE(6,1002) | 00000650 |
| DO 30 I=1,M | 00000660 |
| WRITE(6,1001)(P(I,J),J=1,M) | 00000670 |
| 30 CONTINUE | 00000680 |
| DO 35 I=1,M | 00000690 |
| 35 COEF(I)=-YNOM(I) | 00000700 |
| CALL DCROUT(P,COEF,OUT,0.00,M,1,IND) | 00000710 |
| IF(IND.NE.0)GO TO 85 | 00000720 |
| WRITE(6,1015)DET | 00000730 |
| DO 40 I=1,M | 00000740 |
| 40 IF (DARS(COEF(I)).LT.1.D-10) COEF(I)= 0.0C | 00000750 |
| C | 00000760 |
| IF (DABS(XN(1)).GT.1.D2) GO TO 47 | 00000770 |
| RATS= 1.00 | 00000780 |
| DO 45 I= 1,5 | 00000790 |
| RAT= DARS(COEF(I))/(1.D0+DABS(XN(I))+.1D0) | 00000800 |
| IF (RAT.GT.RATS) RATS= RAT | 00000810 |
| 45 CONTINUE | 00000820 |
| DO 46 I= 1,R | 00000830 |
| 46 COEF(I)= COEF(I)/RATS | 00000840 |
| C | 00000850 |
| WRITE(6,1016) RATS | 00000860 |
| C | 00000870 |
| 47 WRITE(6,1003)(COEF(I),I=1,N) | 00000880 |
| SN= COEF(M) | 00000890 |
| WRITE(6,1012) SN | 00000900 |
| DO 50 J=1,N | 00000910 |
| 50 X(J)=XN(J)+COEF(J) | 00000920 |
| TF=TFN+SN | 00000930 |
| IHALV=0 | 00000940 |
| 51 IF (INORM.EQ.1) ITF=1 | 00000950 |
| CALL FUNCT | 00000960 |
| ITF=3 | 00000970 |
| KOUNT=KOUNT+1 | 00000980 |
| F1=0.00 | 00000990 |
| DO 52 I=1,M | 00001000 |
| 52 F1=F1+Y(I)**2 | 00001010 |
| WRITE(6,1010)F1 | 00001020 |
| IF(F1.LT.F0)GO TO 55 | 00001030 |
| WRITE(6,1008) | 00001040 |
| IF(IHALV.EQ.6)GO TO 95 | 00001050 |
| IHALV=IHALV+1 | 00001060 |
| DO 53 J=1,N | 00001070 |
| COEF(J)=COEF(J)/2.D0 | 00001080 |
| WRITE(6,1000)COEF(J) | 00001090 |
| 53 X(J)=XN(J)+COEF(J) | 00001100 |
| SN=SN/2.000 | 00001110 |
| WRITE(6,1012) SN | 00001120 |
| TF=TFN+SN | 00001130 |
| GO TO 51 | |

| | | |
|------|---|----------|
| 55 | IF(NI-MAXNOI)70,70,80 | 00001140 |
| 70 | NI=NI+1 | 00001150 |
| | ICONS=NI | 00001160 |
| | F0=F1 | 00001170 |
| | SUMDX=0.00 | 00001180 |
| | DO 76 J=1,M | 00001190 |
| 76 | SUMDX=COEF(J)**2+SUMDX | 00001200 |
| | DO 77 I=1,M | 00001210 |
| | DO 77 J=1,M | 00001220 |
| | P(I,J)=P(I,J)+(Y(I)*COEF(J))/SUMDX | 00001230 |
| 77 | CONTINUE | 00001240 |
| | ISW=1 | 00001250 |
| | GO TO 9 | 00001260 |
| 80 | NI=9999 | 00001270 |
| | WRITE(6,1006) | 00001280 |
| | RETURN | 00001290 |
| 95 | NI=9999 | 00001300 |
| | WRITE(6,1007) | 00001310 |
| | RETURN | 00001320 |
| 90 | WRITE(6,1005)F0 | 00001330 |
| | RETURN | 00001340 |
| 95 | IF(NI.EQ.1.OR.IR.EQ.10.(R.ICONS.NE.NI)GO TO 100 | 00001350 |
| | ICONS=ICONS+1 | 00001360 |
| | IR=IR+1 | 00001370 |
| | DO 96 J= 1,N | 00001380 |
| | DYDT(J)= DYDTN(J) | 00001390 |
| | X(J)= XN(J) | 00001400 |
| 96 | Y(J)= YNOM(J) | 00001410 |
| | Y(M)= YNOM(M) | 00001420 |
| | DYDT(M)=DYDTN(M) | 00001430 |
| | TF= TFN | 00001440 |
| | ISW=0 | 00001450 |
| | WRITE(6,1004) | 00001460 |
| | GO TO 10 | 00001470 |
| 100 | NI=9999 | 00001480 |
| | WRITE(6,1009) | 00001490 |
| | RETURN | 00001500 |
| 1000 | FORMAT(/1X,1PD23.15) | 00001510 |
| 1001 | FORMAT(1X,1P5D23.15) | 00001520 |
| 1002 | FORMAT(21HOPARTIAL DERIV MATRIX) | 00001530 |
| 1003 | FORMAT(11HODELX: S ARE/(1X,1PD23.15)) | 00001540 |
| 1004 | FORMAT(35HOFORM NEW PARTIAL DERIVATIVE MATRIX) | 00001550 |
| 1005 | FORMAT(4HOF0=,1PD22.15,23H-CASE CONVERGED...FERTIG) | 00001560 |
| 1006 | FORMAT(38HOEXCEEDED MAXIMUM NUMBER OF ITERATIONS) | 00001570 |
| 1007 | FORMAT(16HOMATRIX SINGULAR) | 00001580 |
| 1008 | FORMAT(11HODELX: S ARE) | 00001590 |
| 1009 | FORMAT(19HOMETHOD CANNOT WORK) | 00001600 |
| 1010 | FORMAT(4HOF1=,1PD23.15) | 00001610 |
| 1011 | FORMAT(4HOF0=,1PD23.15) | 00001620 |
| 1012 | FORMAT (10HO DEL TF =,1PD23.15) | 00001630 |
| 1013 | FORMAT (40HO X(I) = X(I) FOLLOWED BY CORRESPONDING Y) | 00001640 |
| 1014 | FORMAT (24HO Y(I) = Y(I) SO DX(I)=XS(I)) | 00001650 |
| 1015 | FORMAT (15HO RESIDUAL =,1PD23.15) | 00001660 |
| 1016 | FORMAT (8HO RATS =,1PD23.15) | 00001670 |
| | END | 00001680 |

```

C PRN/PRNS
C
C THIS PROGRAM IS CALLED BY THE ITERATOR AND PRINTS
C 14 DIMENSIONAL WITH MASS AND FLUX
C
C
C SUBROUTINE PRN(KOUNT,N)
C
C IMPLICIT REAL*8(A-H,O-S)
COMMON /XMM/X(7),XS(R),Y(8)
COMMON /T/TF,S,TO,TFMIN
C
C N=7
M=R
WRITE (6,1000)
WRITE (6,1001) N01,KOUNT
WRITE (6,1002)
WRITE (6,1003) (X(J),J=1,N)
WRITE (6,1004)
WRITE (6,1003) (Y(J),J=1,M)
WRITE (6,1005) TF
1000 FORMAT (29H0 ITER N01. TRAJECTORY CALLS)
1001 FORMAT (1H0,16,5X,16)
1002 FORMAT (2H0X)
1003 FORMAT (1X,1P5D23.15)
1004 FORMAT (2H0Y)
1005 FORMAT (5H0 TF=,1PD26.16)
RETURN
END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
00000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290

```

| | | |
|-----|---------------------------------------|----------|
| C | DCRUIT/DCROUT14 R DIM. | 0000000 |
| | SUBROUTINE DCRUIT(AA,B, EPS,NI,M,IND) | 0000000 |
| | DOUBLE PRECISION A,-EPS,T,S,P,DT,AA | 0000000 |
| | DIMENSION A(M,R),-B(1),AA(R,R) | |
| 5 | IND=0 | 36110050 |
| | N=NI | 36110060 |
| | DO 6 I=1,N | 0000000 |
| | DO 6 J=1,N | 0000000 |
| 6 | A(I,J)=AA(I,J) | 0000000 |
| | IF(M)10,25,25 | 36110080 |
| 10 | M=N | 36110090 |
| | DO 20 I=1,N | 36110100 |
| | DO 15 J=1,N | 36110110 |
| 15 | R(I,J)=0.DO | 0000000 |
| 20 | R(I,I)=1.DO | 0000000 |
| 25 | IC=0 | 36110140 |
| | II=0 | 36110150 |
| | T=DABS(A(1,1)) | 36110160 |
| | DO 35 I=2,N | 36110170 |
| | IF(T-DABS(A(1,I)))30,35,35 | 36110180 |
| 30 | II=I | 36110190 |
| | T=DABS(A(1,II)) | 36110020 |
| 35 | CONTINUE | 36110210 |
| | IF(II)40,65,40 | 36110220 |
| 40 | IC=IC+1 | 36110230 |
| | IF(M)45,55,45 | 36110240 |
| 45 | DO 50 J=1,M | 36110250 |
| | S=R(1,J) | 36110260 |
| | R(1,J)=R(II,J) | 36110270 |
| 50 | R(II,J)=S | 36110280 |
| 55 | DO 60 J=1,N | 36110290 |
| | S=A(1,J) | 36110300 |
| | A(1,J)=A(II,J) | 36110310 |
| 60 | A(II,J)=S | 36110320 |
| 65 | P=A(1,1) | 36110330 |
| | IF(DABS(P)-EPS)70,70,75 | 36110340 |
| 70 | IND=1 | 36110350 |
| | D=0.DO | 0000000 |
| | GO TO 200 | 36110370 |
| 75 | DO 80 J=2,N | 36110380 |
| 80 | A(1,J)=A(1,J)/P | 36110390 |
| | IF(M)85,95,85 | 36110400 |
| 85 | DO 90 J=1,M | 36110410 |
| 90 | R(1,J)=R(1,J)/P | 36110420 |
| 95 | DO 170 K=2,N | 36110430 |
| | KM=K-1 | 36110440 |
| | T=-1.DO | 0000000 |
| | DO 105 I=K,N | 36110460 |
| | DO 98 J=1,KM | 36110470 |
| 98 | A(I,K)=A(I,K)-A(I,J)*A(J,K) | 36110480 |
| | IF(T-DABS(A(I,K)))100,105,105 | 36110490 |
| 100 | T=DABS(A(I,K)) | 36110500 |
| | II=I | 36110510 |
| 105 | CONTINUE | 36110520 |
| | IF(II-K)110,135,110 | 36110530 |
| 110 | IC=IC+1 | 36110540 |
| | IF(M)115,125,115 | 36110550 |

| | | |
|-----|--------------------------------|----------|
| 115 | DO 120 J=1,M | 36110560 |
| | S=R(K,J) | 36110570 |
| | R(K,J)=R(II,J) | 36110580 |
| 120 | R(II,J)=S | 36110590 |
| 125 | DO 130 J=1,N | 36110600 |
| | S=A(K,J) | 36110610 |
| | A(K,J)=A(II,J) | 36110620 |
| 130 | A(II,J)=S | 36110630 |
| 135 | DT=A(K,K) | 36110640 |
| | IF(DARS(DT)-EPS)70,70,140 | 36110650 |
| 140 | P=P*DT | 36110660 |
| | IF(K-N)145,155,145 | 36110670 |
| 145 | KP=K+1 | 36110680 |
| | DO 150 J=KP,N | 36110690 |
| | DO 148 I=1,KM | 36110700 |
| 148 | A(K,J)=A(K,J)-A(K,I)*A(I,J) | 36110710 |
| 150 | A(K,J)=A(K,J)/DT | 36110720 |
| 155 | IF(M)160,170,160 | 36110730 |
| 160 | DO 165 J=1,M | 36110740 |
| | DO 162 I=1,KM | 36110750 |
| 162 | R(K,J)=R(K,J)-A(K,I)*R(I,J) | 36110760 |
| 165 | R(K,J)=R(K,J)/DT | 36110770 |
| 170 | CONTINUE | 36110780 |
| | IF(MOD(IC,2))175,180,175 | 36110790 |
| 175 | P=-P | 36110800 |
| 180 | D=P | 36110810 |
| | IF(M)185,200,185 | 36110820 |
| 185 | II=N | 36110830 |
| | DO 190 K=2,N | 36110840 |
| | KP=II | 36110850 |
| | II=II-1 | 36110860 |
| | DO 190 J=1,M | 36110870 |
| | DO 190 I=KP,N | 36110880 |
| 190 | R(II,J)=R(II,J)-A(II,I)*R(I,J) | 36110890 |
| 200 | RETURN | 36110900 |
| | END | 36110910 |

```

C TRAJ/TRAJRKS                                00000010
C                                              00000020
C SEP AND HIGH THRUST                        00000030
C 7/14 DIM. WITH POWER EXTRAPOLATION.       00000040
C THIS ROUTINE SETS UP THE INPUT TO THE INTEGRATOR WHICH 00000050
C   EXTRAPOLATES THE TRAJECTORY FROM INITIAL TIME TO    00000060
C   FINAL TIME. IT ALSO EVALUATES THE CHANGE IN TF AND  00000070
C   THE ERROR IN THE FINAL CONDITIONS.                00000080
C THIS PROGRAM IS CALLED BY ITER OR BY CONTROL 00000090
C IT CALL THE SUBPROGRAM TRKGS (RUNGA-KUTTA) 00000100
C MIN J, MAX H.                                  00000110
C R DIM. ZERF. T.C. OPTIONS.                   00000120
C NOP=1--ALL 5 FINAL O.F. FIXED. =2--A,E,I ONLY FIXED. 00000130
C                                              00000140
C                                              00000150
C   SUBROUTINE TRAJ                               00000160
C                                              00000170
C   IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)       00000180
C                                              00000190
C   COMMON /XMM/ ZLO(7), STEP(8), ZERF(8)        00000200
C   COMMON /TRA/TFMAX, DT0, UER, EW(14)          00000210
C   COMMON /Z/ Z(14), DERZ(14)                  00000220
C   COMMON /INT/ITF, IPR, IDIM, IDIM2, NIMAX    00000230
C   COMMON /T/TF, SD, TD, TFIN                  00000240
C   COMMON /ELEM/ZPO(7), ZPF(5)                 00000250
C   COMMON /DY/DYDT(R)                          00000260
C   COMMON /TC/NOP                               00000270
C   COMMON /HIGH/DV1],D [12,1H1                00000280
C   COMMON /A/A,AMU,P]                          00000290
C   COMMON /NOD/NODE                             00000300
C   EXTERNAL FUNCT, OUTP                        00000310
C   DIMENSION PRMT(5), AUX(R,14), DERZ1(14), DP(5) 00000320
C   IF ((IHI.EQ.1).OR.(IHI.EQ.4)) GO TO 9       00000330
C   IF ((IHI.EQ.1).OR.(IHI.EQ.4)) GO TO 9       00000340
C   IF ((IHI.EQ.1).OR.(IHI.EQ.4)) GO TO 9       00000350
C   IF ((IHI.EQ.1).OR.(IHI.EQ.4)) GO TO 9       00000360
C HIGH THRUST                                  00000370
C                                              00000380
C   UP= (PI/2.D0)*ZLO(1)/DSQRT(1.D0+ZLO(1)**2) 00000390
C   CUP= DCOS(UP)                               00000400
C   XK= CUP*(.75D0+.25D0*ZLO(2)/DSQRT(1.D0+ZLO(2)**2)) 00000410
C   DUM= (1.D0+CUP*XK)*DSQRT((CUP-XK)/(CUP+XK)) 00000420
C   XJ= DUM*ZLO(3)/DSQRT(1.D0+ZLO(3)**2)        00000430
C   JM= 2                                        00000440
C   VAR= ZPO(3)                                 00000450
C   IF (NODE.EQ.0) VAR= ZLO(5)                  00000460
C   CALL MAINE(0.D0,0.D0,XK,UP,XJ,1.D0,1,JM,DP,DV1) 00000470
C   CALL OUTHI(JM,PI,ZPO(1),ZPO(2),VAR,IPR,Z, IDIM2) 00000480
C   IF (NODE.EQ.0) GO TO 4                      00000490
C   Z(IDIM2+4)= Z(IDIM2+4)+ZLO(5)*Z(5)         00000500
C   Z(IDIM2+5)= Z(IDIM2+5)-ZLO(5)*Z(4)         00000510
C   DO 5 I= 1,5                                 00000520
C     Z(I+IDIM2)= ZLO(4)+Z(I+IDIM2)*1.D4      00000530
C   DO 6 I= 6,7                                 00000540
C     Z(I)= ZPO(I)                              00000550
C   DO 6 I= 6,7                                 00000560
C     Z(I+IDIM2)= ZLO(I)                       00000570
C

```

| | |
|--|----------|
| C LOW THRUST | 00000580 |
| C | 00000590 |
| 9 PRMT(1)= TO | 00000600 |
| PRMT(2)= TF | 00000610 |
| PRMT(3)= DTO | 00000620 |
| PRMT(4)= UER | 00000630 |
| C | 00000640 |
| IF ((IHI.EQ.2).OR.(IHI.FQ.3)) GO TO 15 | 00000650 |
| C | 00000660 |
| C Z IS A VECTOR OF STATE AND COSTATE | 00000670 |
| DO 10 I=1, IDIM2 | 00000680 |
| Z(I)=ZP0(I) | 00000690 |
| 10 Z(I+IDIM2)= ZLO(I) | 00000700 |
| C | 00000710 |
| C EW ARE ERROR WEIGHTS--INPUT TO THE INTEGRATOR | 00000720 |
| C | 00000730 |
| 15 DO 20 I=1, IDIM | 00000740 |
| 20 DERZ(I)=EW(I) | 00000750 |
| C | 00000760 |
| C CALL THE R-K INTEGRATOR | 00000770 |
| C | 00000780 |
| CALL DRKGS(PRMT,7,DERZ, IDIM, IHLF, FUNCT, OUTP, AUX) | 00000790 |
| IF (IHLF.GT.10) GO TO 100 | 00000800 |
| C | 00000810 |
| C Z IS NOW THE FINAL O.E.. | 00000820 |
| C ZERO THE ERROR IN THE FINAL CONDITIONS | 00000830 |
| C | 00000840 |
| H=0.00 | 00000850 |
| DO 30 I=1, IDIM2 | 00000860 |
| 30 H= H + Z(I+7)*DERZ(I) | 00000870 |
| ZERF(6)= Z(13)*1.D-3 | 00000880 |
| ZERF(7)= Z(14)*1.D-3 | 00000890 |
| DYDT(6)= DERZ(13)*1.D-3 | 00000900 |
| DYDT(7)= DERZ(14)*1.D-3 | 00000910 |
| TF1=TF*(STEP(R)+1.00) | 00000920 |
| CALL FUNCT(TF1,Z,DERZ1) | 00000930 |
| H1=0.00 | 00000940 |
| DO 35 I=1, IDIM2 | 00000950 |
| 35 H1=H1+Z(I+7)*DERZ1(I) | 00000960 |
| DYDT(8)= (H1-H)/(TF1-TF) | 00000970 |
| ZERF(8)= H -1.00 | 00000980 |
| C | 00000990 |
| C FINAL CONDITION OPTION BRANCH | 00001000 |
| C | 00001010 |
| GO TO (40,50), NOP | 00001020 |
| C | 00001030 |
| 40 DO 45 I=1,5 | 00001040 |
| ZERF(I)= Z(I) -7PF(I) | 00001050 |
| 45 DYDT(I)= DERZ(I) | 00001060 |
| RETURN | 00001070 |
| C | 00001080 |
| 50 ZERF(4)= (Z(3)*Z(9)-7(2)*Z(10))*1.D-3 | 00001090 |
| ZERF(5)= (Z(5)*Z(11)-7(4)*Z(12))*1.D-3 | 00001100 |
| DYDT(4)= DERZ(3)*7(4)+Z(3)*DERZ(9)-DERZ(2)*Z(10)-Z(2)*DERZ(10) | 00001110 |
| DYDT(4)= DYDT(4)*1.D-3 | 00001120 |
| DYDT(5)= DERZ(5)*7(11)+7(5)*DERZ(11)-DERZ(4)*Z(12)-Z(4)*DERZ(12) | 00001130 |
| DYDT(5)= DYDT(5)*1.D-3 | 00001140 |

| | | |
|------|---|----------|
| | IF (IHI.LT.3) GO TO 60 | 00001150 |
| C | | 00001160 |
| C | FINAL HIGH THRUST IMPULSE | 00001170 |
| C | | 00001180 |
| | CALL IMPLS(DVIZ,IPK,0.7,DERZ,IDIM2) | 00001190 |
| C | | 00001200 |
| 60 | ZERF(1)= Z(1) - 7PF(1) | 00001210 |
| | DUM1= DSORT(Z(2)**2 + Z(3)**2) | 00001220 |
| | ZERF(2)= DUM1 - 7PF(2) | 00001230 |
| | DUM2= DSORT(Z(4)**2 + Z(5)**2) | 00001240 |
| | ZERF(3)= DUM2 - 7PF(3) | 00001250 |
| | DYDT(1)= DERZ(1) | 00001260 |
| | DYDT(2)= 0.00 | 00001270 |
| | DYDT(3)= 0.00 | 00001280 |
| | IF (DUM1.GT.1.0-12) DYDT(2)= (Z(2)*DERZ(2) + Z(3)*DERZ(3))/DUM1 | 00001290 |
| | IF (DUM2.GT.1.0-12) DYDT(3)= (Z(4)*DERZ(4) + Z(5)*DERZ(5))/DUM2 | 00001300 |
| C | | 00001310 |
| C | SPECIAL CASE, E=0 AND/OR I=0 | 00001320 |
| C | | 00001330 |
| | IF (ZPF(2).NE.0.00) GO TO 70 | 00001340 |
| | ZERF(2)= Z(2) | 00001350 |
| | ZERF(4)= Z(3) | 00001360 |
| | DYDT(2)= DERZ(2) | 00001370 |
| | DYDT(4)= DERZ(3) | 00001380 |
| 70 | IF (ZPF(3).NE.0.00) RETURN | 00001390 |
| | ZERF(3)= Z(4) | 00001400 |
| | ZERF(5)= Z(5) | 00001410 |
| | DYDT(3)= DERZ(4) | 00001420 |
| | DYDT(5)= DERZ(5) | 00001430 |
| | RETURN | 00001440 |
| C | | 00001450 |
| 100 | IF 'IHLF.EQ.11) WRITE (6,1000) | 00001460 |
| | IF 'IHLF.EQ.12) WRITE (6,1001) | 00001470 |
| | IF 'IHLF.EQ.13) WRITE (6,1002) | 00001480 |
| | STOP | 00001490 |
| C | | 00001500 |
| 1000 | FORMAT (68H0 THE NUMBER OF BISECTIONS OF THE ORIGINAL INCREMENT HAS | 00001510 |
| | EXCEEDED 10) | 00001520 |
| 1001 | FORMAT (27H0 INITIAL INCREMENT IS ZERO) | 00001530 |
| 1002 | FORMAT (54H0 INITIAL INCREMENT HAS WRONG SIGN OR BOUNDS ARE WRONG) | 00001540 |
| | END | 00001550 |

```

C OUTP/OUTPS 00000010
C 00000020
C SFP 00000030
C 14 DIMENSIONAL WITH MASS, FLUX 00000040
C THIS IS THE OUTP PROGRAM FOR THE P-C OR R-K 00000050
C INTEGRATOR--FIXED TIME ONLY (ITF=3) 00000060
C EQUINDUCTIAL O.E. AND CUSTATE ARE USED. 00000070
C INCLUDES SHADOW TIME. 00000080
C 00000090
C 00000100
C SUBROUTINE OUTP(1.7,DERZ,IHLF,IDIM,PRMT) 00000110
C 00000120
C IMPLICIT REAL*(A-H,O-S), INTEGER (I-N) 00000130
C COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,UTKG,UTKW,UTMS2 00000140
C COMMON /INT/ITF,IPR,IO,IOIM2,NIMAX 00000150
C COMMON /A/A,AMU,PI 00000160
C COMMON /SHAD/ FEN,FFX,DFEN(5),DFEX(5),ISHAD 00000170
C COMMON/POWER/PO,C.POW,PH,ISUN,ISON,IPOW 00000180
C COMMON /ELEM/ Z(12) 00000190
C COMMON /SOL/RS(4) 00000200
C 00000210
C DIMENSION PRMT(5), Z(14), DERZ(14), W(9), Z10(10) 00000220
C 00000230
C IF (ITF.NE.3) GO TO 10 00000240
C IF (IPR.EQ.0) RETURN 00000250
C IF (T.EQ.PRMT(1)) N=0 00000260
C IF (T.EQ.PRMT(1)) M=0 00000270
C N=N+1 00000280
C IF ((T.LT.(.9999999999D0=DFLOAT(M)*(PRMT(2)-PRMT(1))/DFLOAT(IPR))) 00000290
1 .AND.(IHLF.LT.1)).AND.(T.LT.(.9999999999D0*PRMT(2)))) RETURN 00000300
C M=M+1 00000310
C CALL SUN(T,Z) 00000320
C IF (ISON.EQ.0) GO TO 2 00000330
C DO 1 I= 1,5 00000340
C Z10(I)= Z(I) 00000350
1 Z10(I+5)= Z(I+IOIM2) 00000360
C CALL SHADOW(Z10) 00000370
2 IF (IPOW) 4,3,4 00000380
3 POW= PO 00000390
C IF (ISUN.EQ.1) POW=POW/RS(4)**2 00000400
C GO TO 5 00000410
4 DUM1= DLOG10(Z(7))+14.D0 00000420
C POW= PO*DEXP(-.4364D-12*DUM1**10) 00000430
C IF (ISUN.EQ.1) POW= POW/RS(4)**2 00000440
5 POW= POW-PH 00000450
C A= 2.D0*POW/(C*Z(6)) 00000460
C 00000470
6 TS=UTS*T 00000480
C TM=UTM*T 00000490
C TH=UTH*T 00000500
C TD=UTD*T 00000510
C DV= C*DLOG(.20(6)/Z(6))*UTKM/UTS 00000520
C H= 0.D0 00000530
C DO 7 I=1,IOIM2 00000540
7 H= H + Z(I+IOIM2)*DFRZ(I) 00000550
C W(1)= Z(1)*UTKM 00000560
C W(2)=0.D0 00000570

```

```

DUMMY=7(2)**2+7(3)**2                                00000580
IF (DUMMY.GT.1.D-40) W(2)=DSORT(DUMMY)                00000590
W(3)=0.D0                                              00000600
DUMMY=7(4)**2 + 7(5)**2                               00000610
IF (DUMMY.GT.1.D-40) W(3)= 2.D0*DATAN(DSORT(DUMMY))/DTR 00000620
W(4)=0.D0                                              00000630
IF ((Z(4).NE.0.D0).(W(5).NE.0.D0)) W(4)=DATAN2(Z(4),Z(5))/DTR 00000640
W(5)= 0.D0                                             00000650
IF ((Z(2).NE.0.D0).(W(3).NE.0.D0)) W(5)=DATAN2(Z(2),Z(3))/DTR 00000660
W(5)=W(5)-W(4)                                        00000670
IDIM3=IDIM2+1                                         00000680
C
C
W(6)= Z(6)*UTKG                                       00000700
W(7)= POW*UTKW                                        00000710
W(9)= A*UTMS2                                         00000720
W(8)= W(9)*W(6)                                       00000730
C
C
WRITE (6,1001)                                         00000750
WRITE (6,1002)                                         00000760
WRITE (6,1003) T, TS, TH, TD, DV, N                  00000770
WRITE (6,1004)                                         00000780
WRITE (6,1005) (Z(1),I=1,IDIM2)                      00000790
WRITE (6,1014)                                         00000800
WRITE (6,1005) W                                       00000810
WRITE (6,1006)                                         00000820
WRITE (6,1005) (Z(1),I=IDIM3,IDIM)                   00000830
WRITE (6,1007)                                         00000840
WRITE (6,1005) (DFR7(I),I=1,IDIM2)                   00000850
WRITE (6,1008)                                         00000860
WRITE (6,1005) (DFR7(I),I=IDIM3,IDIM)                 00000870
WRITE (6,1009)                                         00000880
PER=2.D0*PI*DSORT(Z(1)**3/AMU)*UTM                   00000890
AP= W(1)*(1.D0+W(2))                                  00000900
PE= W(1)*(1.D0-W(2))                                  00000910
WRITE (6,1010) H,PER,PE,AP,IMLF                       00000920
C
IF (ISHAD.EQ.0) GO TO 30                               00000930
IF (FEX.LT.FEN) FEX=FEX+2.D0*PI                       00000940
TSHAD= (FEX-FEN+7(2))*(DCOS(FEX)-DCOS(FEN))-Z(3)*(DSIN(FEX) 00000950
-DSIN(FEN)))/PER/(2.D0*PI)                           00000960
FPER= TSHAD/PER                                       00000970
WRITE (6,1013) TSHAD, FPER                             00000980
FEXD= FEX/DTR                                         00000990
FEND= FEN/DTR                                         0001000
WRITE (6,1015) FEND,FEXD                              0001001
C
30 RETURN                                             0001002
10 WRITE (6,1000)                                       0001003
STOP                                                  0001004
C
1000 FORMAT (56H0 ITF MUST EQUAL 3--I.E. NEED FIXED ESTIMATED FINAL TIM0001005
IE)                                                  0001006
1001 FORMAT (70H0 *****0001007
[*****])                                           0001008
1002 FORMAT (5H TIME,10X,10M TIME UNITS,15X,7HSECONDS,9X 0001009
0001100
0001110
0001120
0001130
0001140

```

| | | |
|------|--|----------|
| 1 | ,5HHOURS,11X,4HDAYS,9X,RHOV (K/S),10X,1HN) | 00001150 |
| 1003 | FORMAT (1P2025.7,1P415.7,19//) | 00001160 |
| 1004 | FORMAT (54HO THE ELLIPTICAL ORBITAL ELEMENTS, MASS, AND FLUX ARE) | 00001170 |
| 1005 | FORMAT (1P5022.12//) | 00001180 |
| 1006 | FORMAT (16HO THE COSTATE IS) | 00001190 |
| 1007 | FORMAT (32HO THE DERIVATIVE OF THE STATE IS) | 00001200 |
| 1008 | FORMAT (34HO THE DERIVATIVE OF THE COSTATE IS) | 00001210 |
| 1009 | FORMAT (1HO,7X,11HMANILIONIAN,8X,12HPERIOD (HRS),7X, | 00001220 |
| 1 | 12HPERIGEE (KM),9X,11HAPOGEE (KM),5X,14HDIV. TIME STEP) | 00001230 |
| 1010 | FORMAT (2F20.12,1P2020.10,19//) | 00001240 |
| 1012 | FORMAT (1P4022.12) | 00001250 |
| 1013 | FORMAT (24HO TIME SPENT IN SHADOW =, F15.8,29H HOURS, FRACTION OF | 00001260 |
| 1 | PERIOD =, F15.8//) | 00001270 |
| 1014 | FORMAT (78HO THE CLASSICAL D.E., MASS (KG), POWER (KW), THRUST (N) | 00001280 |
| 1, | THRUST ACC (M/SFC**2)) | 00001290 |
| 1015 | FORMAT (16HO ENTRY ANGLE = ,F20.10,14H EXIT ANGLE = ,F20.10) | 00001300 |
| | END | 00001310 |

```

C FUNCT/FUNCTS                                00000010
C                                                00000020
C SEP                                          00000030
C COMPLEX MODEL --FLUX AS Z(7)              00000040
C 14 DIM. VERSION WITH FLUX AND MASS INCLUDED. 00000050
C THIS SUBROUTINE IS AN INTERFACE BETWEEN THE INTEGRATOR ROUTINE
C   AND THE QUADREATURE ROUTINE.            00000060
C INCLUDES SHADOW EFFECT.                   00000070
C THIS ROUTINE ADD THE EFFECT OF ORLATENESS (J2) TO THE DERIV. 00000080
C ORLATE CALCULATES THE EFFECT OF J2. RETURNED AS DZJ2. 00000090
C Z IS A VECTOR OF THE AVERAGED STATE AND COSTATE 00000100
C DERZ IS THE AVERAGED DERIVITIVE OF Z      00000120
C                                             00000130
C                                             00000140
C                                             00000150
C SUBROUTINE FUNCT(X,Z,DERZ)                00000160
C                                             00000170
C                                             00000180
C IMPLICIT REAL*8(A-H,O-S)                 00000190
C COMMON /A/A,AM(),PI                      00000200
C COMMON /J2/AJ2                            00000210
C COMMON/JD/ TL                             00000220
C COMMON /SHAD/ FEN,FEF,DFEN(5),DFEF(5),ISHAD 00000230
C COMMON/RCOM/R(9)                          00000240
C COMMON/POWER/PO,C,POW,PH,ISUN,ISON,IPW    00000250
C COMMON/ACOM/ AF(10)                       00000260
C COMMON /SOL/RSUN(3),RS                    00000270
C                                             00000280
C NAMELIST/PRINT/TT,PO,C,710,BETA,DUM1,POW,DP,DERZ10,HZ,MM 00000290
C                                             00000300
C DIMENSION Z(14), DERZ(14), G(10), H(10), DZJ2(10),GEX(10),GEN(10) 00000310
C DIMENSION Z10(10),DFRZ10(10),G6(6),H6(6),DFL1(6),DFL(6) 00000320
C                                             00000330
C EXTERNAL FCT,FLUX                         00000340
C                                             00000350
C SET UP COEFFS OF COSF AND SINF IN X1 AND Y1 AND PARTIALS 00000360
C                                             00000370
C                                             00000380
C BETA= 1.00/(1.00+DSQRT(1.00-Z(2)**2-Z(3)**2)) 00000390
C B(1)= 1.00-Z(2)**2*B-TA                   00000400
C R(2)= Z(2)*Z(3)*BETA                     00000410
C B(3)= 1.00-Z(3)**2*B-TA                 00000420
C BETA3= BETA**3/(1.00-BETA)               00000430
C A1= Z(2)**2*BETA3                        00000440
C A2= Z(3)**2*BETA3                        00000450
C A3= BETA+A1                              00000460
C A4= BETA+A2                              00000470
C R(4)= -Z(2)*(BETA+A3)                   00000480
C R(5)= Z(3)*A3                            00000490
C B(6)= -Z(2)*A2                           00000500
C R(7)= -Z(3)*A1                          00000510
C R(8)= Z(2)*A4                            00000520
C B(9)= -Z(3)*(BETA+A4)                   00000530
C                                             00000540
C                                             00000550
C                                             00000560
C                                             00000570

```


| | | |
|----|--|----------|
| C | | 00000580 |
| C | | 00000590 |
| | DO 10 I=1,5 | 00000600 |
| | Z10(I)=Z(I) | 00000610 |
| 10 | Z10(I+5)= Z(I+7) | 00000620 |
| C | | 00000640 |
| | DUM= 1.00 | 00000650 |
| | QEX= -PI | 00000660 |
| | QEN= PI | 00000670 |
| C | | 00000680 |
| | CALL SUN(X,Z) | 00000700 |
| | ISHAD=0 | 00000710 |
| | IF (ISUN.EQ.0) GO TO 12 | 00000720 |
| | CALL SHADOW(Z10) | 00000730 |
| C | | 00000740 |
| C | CALCULATE POWER, DPDN AND ACCELERATION | 00000750 |
| C | | 00000760 |
| 12 | IF (IPDW) 20,15,20 | 00000770 |
| C | | 00000780 |
| C | CONSTANT POWER | 00000790 |
| C | | 00000800 |
| 15 | DO 16 I= 1,6 | 00000810 |
| 16 | DFL(I)= 0.00 | 00000820 |
| | POW= P0 | 00000830 |
| | IF (ISUN.EQ.1) POW= POW/RS**2 | 00000840 |
| | DP= 0.00 | 00000850 |
| | DERZ(7)= 0.00 | 00000855 |
| | GO TO 23 | 00000860 |
| C | | 00000870 |
| 20 | IF= 4 | 00000880 |
| | AIF= 2.00*PI/FLOAT(IF) | 00000890 |
| | IFV= 0 | 00000900 |
| | F1= -PI | 00000910 |
| | DO 34 I= 1,6 | 00000920 |
| 34 | DFL(I)= 0.00 | 00000930 |
| 35 | F2= F1+AIF | 00000940 |
| | CALL QUAD(F1,F2,FLUX,DFL1,Z10,G6,H6,6) | 00000950 |
| | IFV= IFV+1 | 00000960 |
| | F1= F2 | 00000970 |
| | DO 36 I= 1,6 | 00000980 |
| 36 | DFL(I)= DFL(I)+DFL1(I) | 00000990 |
| | IF (IFV.LT. IF) GO TO 35 | 00001000 |
| C | | 00001010 |
| | DERZ(7)= AF(10)*DFL(1) | 00001020 |
| | IF (Z(7).EQ.0.00) Z(7)= .500*DERZ(7)*Z(1)**1.500 | 00001030 |
| | DERZ(7)= DERZ(7)/(2. 0*PI) | 00001040 |
| C | | 00001050 |
| C | | 00001060 |
| | DUM1= DLOG10(Z(7))+14.00 | 00001070 |
| | POW= P0*DEXP(-.4364)-12*DUM1**10) | 00001080 |
| | IF (ISUN.EQ.0) GO TO 22 | 00001090 |
| | POW= POW/RS**2 | 00001100 |
| C | | 00001110 |
| 22 | DP= -POW*.43640-11*DUM1**9/(Z(7)*DLOG(10.00)) | 00001120 |
| C | | 00001130 |
| 23 | POW= POW-PH | 00001140 |
| | IF (POW.LE.0.00) GO TO 500 | 00001150 |

```

C
C
C      A= 2.00*PI/(C*Z(6))
C
C      IF (ISHAD.EQ.0) GO TO 30
C
C SHADOW INFLUENCE
C
C      IF (FEN.LE.FEX) FEN= FEN+ 2.00*PI
C      QEX= FEX
C      QFN= FEN
C      DDC= DCOS(FEN)-DCOS(FEX)
C      DDS= DSIN(FEN)-DSIN(FEX)
C      DUM= (FEN-FEX+Z(2)*DDC-Z(3)*DDS)/(2.00*PI)
C
C      SN= (1.00-Z(3)*DCOS(FEN)-Z(2)*DSIN(FEN))/(2.00*PI)
C      SX= (1.00-Z(3)*DCOS(FEX)-Z(2)*DSIN(FEX))/(2.00*PI)
C
C
C 30 CALL QUAD(QEX,QFN,FC1,DERZ10,Z10,G,H,10)
C
C      HM= -2.00*PI/C**2
C      DFRZ(6)= HM*DUM
C      HM= Z(3)*HM
C
C      IF (ISHAD.EQ.0) GO TO 60
C
C SHADOW INFLUENCE
C
C      CALL FCT(GEN,QEX,Z10,GEN,GEX)
C      HWX=0.00
C      HWN= 0.00
C      DO 40 I=1,5
C          HWX= HWX+Z10(I+5)*GFN(I)
C          HWN= HWN+Z10(I+5)*GFN(I)
C          HWX= HWX+HM*SX
C          HWN= HWN+HM*SN
C
C      DO 50 I=1,5
C 50  DERZ10(I+5)= DERZ10(I+5)-HWN*DFEN(I)+HWX*DFEX(I)
C      DFRZ10(7)= DFRZ10(7)-HM*DDC/(2.00*PI)
C      DFRZ10(8)= DFRZ10(8)+HM*DDS/(2.00*PI)
C      HM= HM*DUM
C
C
C
C 60 IF (AJ2.LE.0.00) GO TO 90
C
C OBLATENESS EFFECT
C
C 70 CALL OBLATE(AJ2,Z10,DZJ2,1)
C      DO 80 I=1,10
C 80  DERZ10(I)= DERZ10(I)+DZJ2(I)
C
C
C
C 90  HZ= 0.00
C      DO 100 I=1,5

```

```

00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710
00001720

```

| | | |
|------|---|----------|
| 100 | M7 = MZ + Z10(1+5) * DERZ10(1) | 00001730 |
| C | | 00001740 |
| | DERZ(13) = M7 / Z(6) | 00001750 |
| | DERZ(14) = -DP * (M7 + HM) / POW | 00001760 |
| | DO 110 I = 1,5 | 00001770 |
| | DERZ(I) = DERZ10(I) | 00001780 |
| 110 | DERZ(1+7) = DERZ10(1+5) - AF(10) * Z(14) * DEL(1+1) * (2.00 * PI) | 00001790 |
| C | | 00001800 |
| C | | 00001810 |
| C | TT = X | 00001820 |
| C | WRITE (6, PRINT) | 00001830 |
| C | | 00001840 |
| | RETURN | 00001850 |
| C | | 00001860 |
| 500 | WRITE (6, 1000) | 00001870 |
| 1000 | FORMAT('EITHER POW OR TL <= 0') | 00001880 |
| | STOP | 00001890 |
| | END | 00001900 |

```

C EARTH'S                                0000010
C                                          0000020
C                                          0000030
C THIS SUBPROGRAM SETS THE VALUES FOR EARTH'S ORBITAL ELEMENTS 0000040
C AND CALCULATES THE MEAN ANOMALY AT THE INITIAL TIME           0000050
C I.E. TAKEN FROM BATTIN, 1964, EPOCH 1960 JAN. 1.5, JD=2436935. 0000060
C INPUT                                                                0000070
C   TL--INITIAL TIME, BEGINNING OF LOW THRUST TRAJECTORY         0000080
C OUTPUT                                                                0000090
C   C(1)--EARTH'S SEMIMAJOR AXIS                                  0000100
C   C(2)--EARTH'S ECCENTRICITY                                    0000110
C   C(3)--ARGUMENT OF PERIHELIUM                                0000120
C   C(4)--MEAN ORBITAL MOTION                                     0000130
C   C(5)--MEAN ANOMALY AT TL                                     0000140
C   C(6)--COSINE OF ANGLE OF ORBILQUITY                         0000150
C   C(7)--SINE OF ANGLE OF ORBILQUITY                          0000160
C   C(17)--LONG. OF N. MAG. POLE AT TIME TL                    0000170
C   C(18)--LONG. OF S. MAG. POLE *** SET TO 90 DEGREES         0000180
C   C(19)--EARTH ROT. FRAC. *** SET TO ZERO                   0000190
C   C(20)--2*PI                                                 0000200
C                                                                    0000210
C                                                                    0000220
C                                                                    0000230
C SUBROUTINE EARTH                                                  0000240
C                                                                    0000250
C   IMPLICIT REAL*(A-H,O-S)                                       0000260
C                                                                    0000270
C   COMMON /JD/ TL                                                0000280
C   COMMON /TERRA/ C(20)                                          0000290
C   COMMON /UNITS/UTS,UTH,UTM,UTD,UTKM,DTR,UTKG,UTKW,UTMS2      0000300
C                                                                    0000310
C                                                                    0000320
C   C(1) = 1.00                                                  0000330
C   C(2) = .01672600                                             0000340
C   C(3) = 102.2525300                                           0000350
C   C(4) = .98560900                                             0000360
C                                                                    0000370
C MEAN ANOMALY AT EPOCH                                           0000380
C   AN= 100.1581500-C(3)                                         0000390
C   BO= AN                                                         0000400
C MEAN ANOMALY AT TIME TL                                         0000410
C   AN= AN+C(4)*(TL-2436935.00)                                  0000420
C   AN=AN/360.00                                                 0000430
C   AN=AN-IDINT(AN)                                              0000440
C                                                                    0000450
C   C(5)= AN*360.00*DTR                                          0000460
C   C(3)= C(3)*DTR                                               0000470
C   C(4)= C(4)*DTR*UTD                                           0000480
C                                                                    0000490
C   DUM= 23.45*DTR                                               0000500
C   C(6)= DCOS(DUM)                                              0000510
C   C(7)= DSIN(DUM)                                              0000520
C                                                                    0000530
C CALCULATE THE ROTATION MATRIX FROM EQUATORIAL TO GMT COORD. 0000540
C                                                                    0000550
C   I0= BO+(2.00*C(2)-.2500*C(2)**3)*DSIN(PI)+1.2500*C(2)**2 0000560
C   I1= *DSIN(2.00*BO)+1.06333333333333300*C(2)**3*DSIN(3.00*BO) 0000570

```

| | | |
|---|--|----------|
| | RD= RD+C(3) | 00000580 |
| | RX= -DCOS(RO) | 00000590 |
| | RY= -DSIN(RO)*C(6) | 00000600 |
| | RHD= DSQRT(RX**2+RY**2) | 00000610 |
| | RX= RX/RHD | 00000620 |
| | RY= RY/RHD | 00000630 |
| | C(8)= RX | 00000640 |
| | C(9)= -RY | 00000650 |
| | C(10)= 0.00 | 00000660 |
| | C(11)= RY | 00000670 |
| | C(12)= RX | 00000680 |
| | C(13)= 0.00 | 00000690 |
| | C(14)= 0.00 | 00000700 |
| | C(15)= 0.00 | 00000710 |
| | C(16)= 1.00 | 00000720 |
| C | | 00000730 |
| | C(17)= 289.900*DTR | 00000740 |
| | C(18)= 90.00*DTR | 00000750 |
| | C(19)= 0.00 | 00000760 |
| | C(20)= 360.00*DTR | 00000770 |
| C | | 00000780 |
| | C(17)= C(17)+C(19)*(TL-2436935.00)/UTD | 00000790 |
| | RETURN | 00000800 |
| | END | 00000810 |

```

C SUN 00000010
C 00000020
C 00000030
C THIS PROGRAM CALCULATES THE EARTH TO SUN DIRECTION AND 00000040
C DISTANCE FOR A GIVEN TIME. OUTPUT IN THE EQUINOCTIAL 00000050
C COORDINATE FRAME. 00000060
C ALSO CALCULATES FLUX FACTORS.
C INPUT 00000070
C Z--10 VECTOR OF EQ. O.E. AND COSTATE(NOT USED) 00000080
C AE--EARTH ORBIT SEMIMAJOR AXIS 00000090
C EC--EARTH ORBIT ECCENTRICITY 00000100
C W--LONG. OF PERIH. 00000110
C ENE--MEAN ORBITAL MOTION 00000120
C AN--MEAN ANOMALY AT BEGINNING OF TRAJECTORY (T0) 00000130
C COB--COS OF ANGLE OF ORBILITY 00000140
C SOB--SIN OF ANGLE OF ORBILITY 00000150
C CMA--ROTATION MATRIX FROM EQUAT. TO GMT COORD. 00000160
C A1--LONG. OF N MAG POLE AT TL 00000170
C A2--LAT OF N MAG POLE AT TL 00000180
C WE--EARTH ROTATIONAL FREQ. 00000190
C TUP1--2*PI 00000200
C T--PRESENT TIME 00000210
C OUTPUT 00000220
C RS--UNIT VECTOR FROM EARTH TO SUN, EQUINOCTIAL COORD. 00000230
C R--DISTANCE FROM EARTH TO SUN AT TIME T 00000240
C C--FLUX FACTOR
C 00000250
C 00000260
C 00000270
C SUBROUTINE SUN(T,Z) 00000280
C 00000290
C IMPLICIT REAL*8(A-H,O-S) 00000300
C 00000310
C COMMON /SOL/ RS(3), R 00000320
C COMMON/TERRA/ AE,EC,W,ENE,AN,COB,SOB,CMA(3,3),A1,A2,WE,TUP1 00000330
C COMMON/CCOM/C(6) 00000340
C 00000350
C DIMENSION RS1(3),CM(3,3),Z(14) 00000360
C DIMENSION VN(3),V2(3),CMP(3,2),CMO(3,2) 00000370
C 00000380
C 00000390
C MEAN ANOMALY AT TIME T 00000400
C AA= AN+ENE*T 00000410
C 00000420
C TRUE ANOMALY--CORRECT THRU ECCENTRICITY CURVED 00000430
C F=AA+(2.00*EC-.2500*EC**3)*DSIN(AA)+1.2500*EC**2*DSIN(2.00*AA) 00000440
C 1 +1.0R333333333300*EC**3*DSIN(3.00*AA) 00000450
C B=F+W 00000460
C 00000470
C DISTANCE BETWEEN EARTH AND SUN 00000480
C R=AE*(1.00-EC**2)/(1.00+EC*DCOS(F)) 00000490
C 00000500
C UNIT VECTOR TO SUN, EQUATORIAL COORD. 00000510
C RS1(1)=-DCOS(B) 00000520
C RS1(2)=-COB*DSIN(B) 00000530
C RS1(3)=-SOB*DSIN(B) 00000540
C 00000550

```

```

C TRANSFORM TO EQUINOCTIAL COORD.
AR= 1.00+Z(4)**2+Z(5)**2
CM(1,1)= (1.00-Z(4)**2-Z(5)**2)/AR
CM(2,1)= 2.00*Z(4)*Z(5)/AR
CM(3,1)= -2.00*Z(4)/AR
CM(1,2)= CM(2,1)
CM(2,2)= (1.00+Z(4)**2-Z(5)**2)/AR
CM(3,2)= 2.00*Z(5)/AR
CM(1,3)= -CM(3,1)
CM(2,3)= -CM(3,2)
CM(3,3)= (1.00-Z(4)**2-Z(5)**2)/AR
DO 10 I=1,3
RS(I)= 0.00
DO 10 J=1,3
10 RS(I)= RS(I)+CM(J,1)*RS1(J)
C
C
C CALCULATION OF C(6), FLUX FACTORS
C
THETA= WE*T+A1
THETA= THETA/TUPI
THETA= THETA-IDJNT(THETA)
THETA= THETA*TUPI
C
C VN IS VECTOR THRU N MAG POLE AT TIME T IN GMT COORD
A3= DCOS(A2)
VN(1)= DCOS(THETA)*A3
VN(2)= DSIN(THETA)*A3
VN(3)= DSIN(A2)
C
C V2 IS N IN EQUATORIAL COORDS.
DO 20 I=1,3
V2(I)= 0.00
DO 20 J=1,3
20 V2(I)= V2(I)+VN(J)*CMA(J,I)
C
C CALC PARTIALS OF F AND G UNIT VECTORS
AR= 2.00/AB
DO 30 I=1,3
CMP(I,1)= -AB*(Z(5)*CM(1,2)+CM(1,3))
CMP(I,2)= AB*Z(5)*CM(1,1)
CMQ(I,1)= AB*Z(4)*CM(1,2)
30 CMQ(I,2)= AB*(-Z(4)*CM(1,1)+CM(1,3))
C
C C(1),C(2) ARE COEFF OF X1,Y1 IN EQN FOR SIN(ALPHA)
C C(3),C(4) ARE COEFF OF X1,Y1 IN DSIN DP
C C(5),C(6) " " " DO
DO 40 K=1,2
C(K)= 0.00
C(K+2)= 0.00
C(K+4)= 0.00
DO 40 J=1,3
C(K)= C(K)+V2(J)*CM(J,K)
C(K+2)= C(K+2)+V2(J)*CMP(J,K)
40 C(K+4)= C(K+4)+V2(J)*CMQ(J,K)
C
RETURN
END

```

```

00000560
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810
00000820
00000830
00000840
00000850
00000860
00000870
00000880
00000890
00000900
00000910
00000920
00000930
00000940
00000950
00000960
00000970
00000980
00000990
00001000
00001010
00001020
00001030
00001040
00001050
00001060
00001070
00001080
00001090
00001100
00001110
00001120
00001130

```

```

C SHADOW/SHADOW1                                00000010
C                                                    00000020
C THIS PROGRAM DETERMINES IF A GIVEN ORBIT PASSES THROUGH 00000030
C THE EARTH'S SHADOW AND IF SO WHAT THE ENTRY AND EXIT 00000040
C ANGLES ARE. IT ALSO CALCULATES THE PARTIAL DERIVATIVE 00000050
C VECTOR OF F WRT THE O.F. AT THE ENTRY AND EXIT POINTS. 00000060
C INPUT AND OUTPUT IN EQUINOCTIAL COORD. 00000070
C INPUT 00000080
C Z-- 10 VECTOR OF O.F. AND COSTATE (COSTATE NOT USED) 00000090
C XSUN,YSUN,ZSUN--SUN'S DIRECTION IN EQ. COORD. (UNIT VECTOR) 00000100
C OUTPUT 00000110
C FEN--ENTRY ANGLE 00000120
C FEX--EXIT ANGLE 00000130
C DFEN--DIRIVITIVE OF F AT ENTRY 00000140
C DFEX--DIRIVITIVE OF F AT EXIT 00000150
C ISHAD--FLAG=0 IF ORBIT NOT INTERSECT SHADOW 00000160
C =2 IF ORBIT ENTER AND EXIT FROM SHADOW 00000170
C 00000180
C 00000190
C 00000200
C SUBROUTINE SHADOW(Z) 00000210
C 00000220
C IMPLICIT REAL*8(A-H,O-S) 00000222
C 00000224
C COMMON /SDL/ XSUN,YSUN,ZSUN,RSUN 00000240
C COMMON /SHAD/ FEN,FEX,DFEN(5),DFEX(5),ISHAD 00000250
C 00000260
C DIMENSION DSDX(5),AP(4),RT(4),Z(10) 00000270
C 00000280
C NAMELIST /DUMFEN/ FEN,DUM 00000290
C NAMELIST /DUMFEX/ FEX,DUM 00000300
C NAMELIST /EQ/ AP,RT,ARE 00000305
C NAMELIST /EQ2/BETA,R1,B2,B3,D1,D2,D3,H1,H2,H3,G1,G2,A0 00000306
C NAMELIST /PR4/ DUM,DSDX 00000307
C NAMELIST /PR5/ I,II,CF,SF,X1,Y1 00000308
C NAMELIST /PR6/ I,II,FON 00000309
C 00000310
C 00000320
C CALCULATE POLYNOMIAL COEFF. 00000330
C 00000340
C BETA= DSQRT(1.00-Z(2)**2-Z(3)**2) 00000350
C BETA= 1.00/(1.00+BETA) 00000360
C B1= 1.00-Z(2)**2*BETA 00000370
C B2= Z(2)*Z(3)*BETA 00000380
C B3= 1.00-Z(3)**2*BETA 00000390
C D1= 1.00-XSUN**2 00000400
C D2= 1.00-YSUN**2 00000410
C D3= 2.00*XSUN*YSUN 00000420
C C1=B2**2 00000430
C C2=B3**2 00000440
C C3=B2*B3 00000450
C C4=B1*B2 00000460
C H1=D1*(B1**2-C1)+D2*(C1-C2)-D3*(C4-C3) 00000470
C H2=-2.00*(D1*B1*Z(3)+D2*B2*Z(2))+D3*(B2*Z(3)+B1*Z(2)) 00000480
C H3=D1*(C1+Z(3)**2)+D2*(C2+Z(2)**2)-D3*(C3+Z(2)*Z(3)) 00000490
C 1 -1.00/Z(1)**2 00000495
C G1= 2.00*(D1*C4+D2*C3)-D3*(C1+B1*B3) 00000500

```



```

G2= -2.00*(D1*B2*7(4)+D2*B3*7(2))+D3*(B3*Z(3)+H2*Z(2))      00000510
C1= G1**2                                                       00000520
C2= G2**2                                                       00000530
C3= G1*G2                                                       00000540
A0= H1**2+C1                                                    00000550
C WRITE (6,EQ2)                                                 00000555
AP(1)=2.00*(H1*H2+C3)/A0                                        00000560
AP(2)= (H2**2+2.00*H2*H1-C1+C2)/A0                            00000570
AP(3)= 2.00*(H3*H2-C3)/A0                                      00000580
AP(4)= (H3**2-C2)/A0                                          00000590
C                                                                 00000600
C CALL SUBROUTINE TO SOLVE A QUARTIC EQN.                       00000610
C                                                                 00000620
C CALL DORIC(AP,RT,NRF)                                         00000630
C                                                                 00000632
C WRITE (6,EQ)                                                  00000634
C                                                                 00000640
C NRE= NUMBER OF REAL ROOTS. MUST BE EQUAL TO 0,2,OR 4        00000650
C ROOTS ARE RT(1), OR RT(1),RT(2),RT(3)+-RT(4)*I, OR         00000660
C RT(1)+-RT(2)*I,RT(3)+-RT(4)*I                               00000670
C IF ((NRE.EQ.1).OR.(NRE.EQ.3)) GO TO 130                       00000680
C                                                                 00000690
C FEN= 100.00                                                  00000700
C FEX= 100.00                                                  00000710
C ISHAD= 0                                                      00000720
C I=0                                                            00000730
10 I=I+1                                                        00000740
C IF ((ISHAD.EQ.2).OR.(I.EQ.(NRF+1))) GO TO 120               00000750
C                                                                 00000760
C CF= RT(1)                                                     00001000
C SF= DSORT(1.00-CF**2)                                         00001010
C                                                                 00001020
C HEMISPHERE CHECK                                             00001030
C II=1                                                           00001040
20 X1= B1*CF+B2*SF-7(3)                                         00001050
C Y1= B3*SF+B2*CF-7(2)                                         00001060
C WRITE (6,PR5)                                                 00001065
C IF ((X1*XSUN+Y1*YSUN).LT.0.00) GO TO 40                      00001070
30 IF ((II.EQ.2) GO TO 10                                       00001080
C II=2                                                           00001090
C SF=-SF                                                         00001100
C GO TO 20                                                       00001110
C                                                                 00001120
C IS SHADOW EQUATION ZERO?                                     00001130
40 EQN= D1*X1**2+D2*Y1**2-D3*X1*Y1-Z(1)**(-2)                 00001140
C WRITE (5,PR6)                                                 00001145
C IF (DABS(EQN).GT.1.0E-6) GO TO 30                             00001150
C                                                                 00001160
C ROOT HAS PASSED TESTS--NOW CHECK TO SEE IF EXIT OR ENTRY ANGLE
C DXDF= -B1*SF+B2*CF                                           00001170
C DYDF= -B2*SF+B3*CF                                           00001180
C USDF= (2.00*D1*X1-D3*Y1)*DXDF +(2.00*D2*Y1-D3*X1)*DYDF    00001190
C DUM= DATAN2(SF,CF)                                           00001200
C IF (DSDF) 70,50,60                                           00001210
C                                                                 00001220
C ORBIT IS TANGENT TO SHADOW                                   00001230
50 WRITE (6,1010)                                              00001240
C                                                                 00001250

```

| | |
|---|----------|
| GO TO 30 | 00001260 |
| C | 00001270 |
| C IS FEX ALREADY FOUND? | 00001280 |
| 60 IF (FEX.EQ.1.D2) GO TO 40 | 00001290 |
| C YES | 00001300 |
| WRITE (6,DUMFEX) | 00001310 |
| GO TO 30 | 00001320 |
| C | 00001330 |
| C IS FEN ALREADY FOUND? | 00001340 |
| 70 IF (FEN.EQ.1.D2) GO TO 40 | 00001350 |
| C YES | 00001360 |
| WRITE (6,DUMFEN) | 00001370 |
| GO TO 30 | 00001380 |
| C | 00001390 |
| C CALCULATE DSDX | 00001400 |
| 80 ZETA= Z(3)*SF-Z(2)*CF | 00001410 |
| BETA3= BETA**3/(1.D0-BETA) | 00001420 |
| PZ5= Z(2)*BETA3 | 00001430 |
| PZ6= Z(3)*BETA3 | 00001440 |
| DXDH= -2.D0*Z(2)*BETA*CF+7(3)*BETA*SF+PZ5*ZETA*Z(2) | 00001450 |
| DXDK= Z(2)*BETA*SF-1.D0+PZ6*Z(2)*ZETA | 00001460 |
| DYDH= Z(3)*BETA*CF-1.D0-PZ5*Z(3)*ZETA | 00001470 |
| DYDK= -2.D0*Z(3)*BETA*SF+Z(2)*BETA*CF-PZ6*Z(3)*ZETA | 00001480 |
| DSDX(1)= 2.D0*Z(1)**(-3) | 00001490 |
| DUM1= 2.D0*D1*X1-D3*Y1 | 00001492 |
| DUM2= 2.D0*D2*Y1-D3*X1 | 00001494 |
| DSDX(2)= DUM1*DXDH+DUM2*DYDH | 00001500 |
| DSDX(3)= DUM1*DXDK+DUM2*DYDK | 00001510 |
| D=2.D0/(1.D0+Z(4)**2+7(5)**2) | 00001520 |
| DXSP= (-YSUN*Z(5)-ZSUN)*D | 00001530 |
| DXSQ= YSUN*Z(4)*D | 00001540 |
| DYSP= XSUN*Z(5)*D | 00001550 |
| DYSQ= (-XSUN*Z(4)+ZSUN)*D | 00001560 |
| DUM1= -2.D0*X1*(X1*XSUN+Y1*YSUN) | 00001570 |
| DUM2= -2.D0*Y1*(Y1*YSUN+X1*XSUN) | 00001580 |
| DSDX(4)= DUM1*DXSP+DUM2*DYSQ | 00001590 |
| DSDX(5)= DUM1*DXSQ+DUM2*DYSQ | 00001600 |
| C | 00001605 |
| WRITE (6,PR4) | 00001610 |
| ISHAD=ISHAD+1 | 00001620 |
| IF (DSD.F.LT.0.D0) GO TO 100 | 00001630 |
| C | 00001640 |
| C EXIT ANGLE AND DERIVATIVE | 00001650 |
| FEX=DUM | 00001660 |
| DO 90 J=1,5 | 00001670 |
| 90 D EX(J)= DSDX(J)/DSDF | 00001680 |
| GO TO 10 | 00001690 |
| C | 00001700 |
| C ENTRY ANGLE AND DERIVATIVE | 00001710 |
| 100 FEN=DUM | 00001720 |
| DO 110 J=1,5 | 00001730 |
| 110 DFEN(J)= DSDX(J)/DSDF | 00001740 |
| GO TO 10 | 00001750 |
| C | 00001760 |
| C | 00001770 |
| 120 IF ((ISHAD.EQ.0).OR.(ISHAD.EQ.2)) RETURN | 00001780 |
| WRITE (6,1020) ISHAD | 00001785 |
| ISHAD= 0 | |

| | | |
|------|--|----------|
| | RETURN | 00001790 |
| C | | 00001800 |
| 130 | WRITE (6,1030) NPF | 00001802 |
| | STOP | 00001804 |
| C | | 00001806 |
| C | | 00001810 |
| 1010 | FORMAT (33H0 DSDP=0. DRAIT TANGENT TO SHADOW) | 00001820 |
| 1020 | FORMAT (15H0 ERROR--(SHAD)=,14) | 00001830 |
| 1030 | FORMAT (49H0 DORTIC HAS RETURNED WITH NUMBER OF REAL ROOTS =,14) | 00001832 |
| | END | 00001840 |

| | | |
|----|--|----------|
| | SUBROUTINE DORTIC(C,R,NRE) | 3C120020 |
| C | | 3C120030 |
| C | SOLVES POLYNOMIAL EQUATION OF THE TYPE | 3C120050 |
| C | $X^{**4}+C(1)*X^{**3}+C(2)*X^{**2}+C(3)*X+C(4)=0$ | 3C120060 |
| C | | 3C120070 |
| C | THE COEFFICIENT OF X^{**4} IS ASSUMED TO BE 1 | 3C120080 |
| C | | 3C120090 |
| C | R CONTAINS THE ROOTS | 3C120100 |
| C | | 3C120110 |
| C | NRE CONTAINS THE NUMBER OF REAL ROOTS | 3C120120 |
| C | | 3C120130 |
| C | IF THERE ARE TWO REAL ROOTS THEY WILL BE IN | 3C120140 |
| C | R(1) AND R(2), WITH THE COMPLEX ROOTS R(3)+-R(4)*I | 3C120150 |
| C | | 3C120160 |
| C | | 3C120170 |
| C | IF THERE ARE NO REAL ROOTS, THE COMPLEX | 3C120180 |
| C | ROOTS ARE R(1)+-R(2)*I AND R(3)+-R(4)*I | 3C120190 |
| C | | 3C120200 |
| | DIMENSION C(4),R(4),CP(2),Y(3) | 3C120210 |
| | DOUBLE PRECISION C,R,CP,Y,C1SQ,A,B,D,E,F,REAL,DSCR,RAD | 3C120220 |
| | C1SQ=C(1)**2 | 3C120240 |
| | CP(1)=-C(2) | 3C120250 |
| | CP(2)=C(1)*C(3)-4.00*C(4) | 3C120260 |
| | CP(3)=(4.00*C(2)-C1SQ)*C(4)-C(3)**2 | 3C120270 |
| | 5 CALL DCUBIC(CP,Y,0.0) | 3C120280 |
| | R A=C1SQ/4.00-C(2)+Y(1) | 3C120290 |
| | B=.500*C(1)*Y(1)-C(3) | 3C120300 |
| | D=.2500*Y(1)*Y(1)-C(4) | 3C120310 |
| | IF(A) 10,10,15 | 3C120320 |
| 10 | E=0. | 3C120330 |
| | GO TO 20 | 3C120340 |
| 15 | E=DSQRT(A) | 3C120350 |
| 20 | F(D) 25,25,30 | 3C120360 |
| 25 | F=0. | 3C120370 |
| | GO TO 50 | 3C120380 |
| 30 | F=DSIGN(DSQRT(D),R) | 3C120390 |
| 50 | NRE=0 | 3C120400 |
| | REAL=-.2500*C(1)+.500*E | 3C120410 |
| | DSCR=REAL*REAL-.500*Y(1)+F | 3C120420 |
| 53 | RAD=DSQRT(DABS(DSCR)) | 3C120430 |
| | IF(DSCR)60,55,55 | 3C120440 |
| 55 | NRE=2 | 3C120450 |
| | R(1)=REAL+RAD | 3C120460 |
| | R(2)=REAL-RAD | 3C120470 |
| | GO TO 65 | 3C120480 |
| 60 | R(3)=REAL | 3C120490 |
| | R(4)=RAD | 3C120500 |
| 65 | REAL=REAL-E | 3C120510 |
| | DSCR=REAL*REAL-.500*Y(1)-F | 3C120520 |
| 68 | RAD=DSQRT(DABS(DSCR)) | 3C120530 |
| | IF(DSCR)80,70,70 | 3C120540 |
| 70 | NRE=NRE+2 | 3C120550 |
| | R(NRE)=REAL-RAD | 3C120560 |
| | R(NRE-1)=REAL+RAD | 3C120570 |
| | GO TO 90 | 3C120580 |
| 80 | R(NRE+1)=REAL | 3C120590 |
| | R(NRE+2)=RAD | 3C120600 |
| 90 | RETURN | 3C120610 |
| | END | 3C120620 |

| | | |
|----|--|----------|
| C | SUBROUTINE DCURIC(C,R,NRE) | 3C110020 |
| C | | 3C110030 |
| C | SOLVES POLYNOMIAL EQUATION OF THE TYPE | 3C110050 |
| C | $X^3+C(1)*X^2+C(2)*X+C(3)=0$ | 3C110060 |
| C | | 3C110070 |
| C | THE COEFFICIENT OF X^3 IS ASSUMED TO BE 1 | 3C110080 |
| C | | 3C110090 |
| C | R CONTAINS THE ROOTS | 3C110100 |
| C | | 3C110110 |
| C | NRE CONTAINS THE NUMBER OF REAL ROOTS | 3C110120 |
| C | | 3C110130 |
| C | IF THERE IS ONE REAL ROOT IT WILL BE | 3C110140 |
| C | IN R(1), WITH THE COMPLEX ROOTS R(2)+R(3)*I | 3C110150 |
| C | | 3C110160 |
| C | | 3C110170 |
| C | | 3C110180 |
| C | | 3C110190 |
| | DIMENSION C(3),R(3) | 3C110210 |
| | DOUBLE PRECISION C,R,C1SQ,P,Q,DEL,T,A,CRTA,CRTB,PHI3,CON,Y | 3C110220 |
| | DOUBLE PRECISION R,S,HQ | 3C110230 |
| | C1SQ=C(1)**2 | 3C110240 |
| | P=C(2)-C1SQ/3.D0 | 3C110250 |
| | Q=C(3)-(C(2)/3.D0-2.D0*C1SQ/27.D0)*C(1) | 3C110260 |
| | DEL=4.D0*P**3+27.D0*Q**2 | 3C110270 |
| | T=C(1)/3.D0 | 3C110280 |
| 5 | IF(DEL)20,10,10 | 3C110290 |
| 10 | SQ=DSQRT(DEL/108.D0) | 3C110300 |
| | HQ=.5D0*Q | 3C110310 |
| | A=-HQ+SQ | 3C110320 |
| | B=-HQ-SQ | 3C110330 |
| | CRTA=DSIGN(DABS(A)**.3333333333333333D0,A) | 3C110340 |
| | CRTB=DSIGN(DABS(B)**.3333333333333333D0,B) | 3C110350 |
| 15 | Y=CRTA+CRTB | 3C110360 |
| | R(1)=Y-T | 3C110370 |
| | R(2)=-.5D0*Y-T | 3C110380 |
| | R(3)=.86602540378443865D0*(CRTA-CRTB) | 3C110390 |
| | NRE=1 | 3C110400 |
| | GO TO 40 | 3C110410 |
| 20 | PHI3=DATAN2(DSQRT(-DEL/27.D0),-Q)/3.D0 | 3C110420 |
| 22 | CON=2.D0*DSQRT(-P/3.D0) | 3C110430 |
| 30 | R(1)=CON*DCOS(PHI3)-T | 3C110440 |
| | R(2)=-CON*DCOS(1.0471975511965977D0-PHI3)-T | 3C110450 |
| | R(3)=-CON*DCOS(1.0471975511965977D0+PHI3)-T | 3C110460 |
| | NRE=3 | |
| 40 | RETURN | |
| | END | |

```

C FLUX                                00000010
C                                     00000020
C                                     00000030
C                                     00000040
C THIS PROGRAM CALCULATES THE FLUX DERIVATIVE AND          00000050
C THE INTERAND OF THE FLUX CONTRIBUTION TO THE O.F.        00000060
C COSTATE EQUATION                                         00000070
C ONLY G(1), PFPX, PFPY, AND PFPZA ARE DEPENDENT          00000080
C ON THE FORM OF N DOT                                     00000090
C                                                         00000100
C                                                         00000110
C SUBROUTINE FLUX(F1,F2,Z,H,G)                               00000120
C                                                         00000130
C IMPLICIT REAL*8(A-H,O-S)                                  00000140
C COMMON /RCOM/R(9)                                         00000150
C COMMON /CCOM/C(6)                                         00000160
C COMMON /ACOM/AF(10)                                       00000170
C DIMENSION FI(9),FII(5),FIV(9),FIW(9)                    00000180
C DIMENSION Z(1),H(1),G(1),PF(5)                          00000190
C                                                         00000200
C                                                         00000210
C M=0                                                         00000220
C F= F1                                                       00000230
C                                                         00000240
C CALCULATE GEOMAGNETIC COORDINATES                        00000250
C                                                         00000260
5  CF= DCOS(F)                                               00000270
   SF= DSIN(F)                                               00000280
   X1= Z(1)*(R(1)*CF+R(2)*SF-Z(3))                          00000290
   Y1= Z(1)*(R(2)*CF+R(3)*SF-Z(2))                          00000300
   R= DSQRT(X1**2+Y1**2)                                     00000310
   U= 1.00/R                                                 00000320
   SA= (C(1)*X1+C(2)*Y1)/R                                   00000330
   SGN= 0.00                                                 00000340
   DUM1= DABS(SA)                                            00000350
   IF (DUM1.GT.1.0-10) SGN= SA/DUM1                         00000360
   SA= DUM1                                                  00000370
   CA= 0.00                                                  00000380
   IF (SA.LT.1.00) CA= DSQRT(1.00-SA**2)                    00000390
   IF (CA.LE.1.0-10) CA= 1.0-10                            00000400
   S= 1.00-Z(3)*CF-Z(2)*SF                                  00000410
C                                                         00000420
C FLUX RATE EQUATION                                       00000430
C                                                         00000440
C                                                         00000450
C FI(1)= U*CA                                               00000460
C FI(2)= U**2*CA                                           00000470
C FI(3)= FI(2)*SA                                          00000480
C FI(4)= U*SA                                               00000490
C FI(5)= U**3*SA**2                                        00000500
C FI(6)= FI(5)*SA                                          00000510
C FI(7)= DSQRT(U)*CA                                       00000520
C FI(8)= U**2.2500*CA                                       00000530
C FI(9)= U**2.0000*CA                                       00000540
C                                                         00000550
C SUM= 0.00                                                 00000560
C DO 6 I=1,9                                                00000570
6  SUM= SUM+AF(I)*FI(I)

```

| | | |
|---|--------------------------------------|----------|
| C | G(1)= DEXP(SUM) | 00000580 |
| C | | 00000590 |
| C | | 00000600 |
| C | PARTIAL OF X1 AND Y1 WRT H AND K | 00000610 |
| C | | 00000620 |
| | PXPH= Z(1)*(R(4)*(F+K(5)*SF) | 00000630 |
| | PYPH= Z(1)*(R(5)*(F+K(6)*SF-1.00) | 00000640 |
| | PXPK= Z(1)*(B(7)*(C+R(8)*SF-1.00) | 00000650 |
| | PYPK= Z(1)*(B(8)*(C+R(9)*SF) | 00000660 |
| C | | 00000670 |
| C | | 00000680 |
| | FIU(1)= CA | 00000690 |
| | FIU(2)= 2.00*U*CA | 00000700 |
| | FIU(3)= FIU(2)*SA | 00000710 |
| | FIU(4)= SA | 00000720 |
| | FIU(5)= 3.00*F1(5)/U | 00000730 |
| | FIU(6)= FIU(5)*SA | 00000740 |
| | FIU(7)= .500*F1(7)/U | 00000750 |
| | FIU(8)= .2500*F1(8)/U | 00000760 |
| | FIU(9)= .200*F1(9)/U | 00000770 |
| C | | 00000780 |
| C | | 00000790 |
| | FIV(1)= 0.00 | 00000800 |
| | FIV(2)= 0.00 | 00000810 |
| | FIV(3)= U**2*CA | 00000820 |
| | FIV(4)= U | 00000830 |
| | FIV(5)= 2.00*U**2*SA | 00000840 |
| | FIV(6)= FIV(5)*SA*1.500 | 00000850 |
| | FIV(7)= 0.00 | 00000860 |
| | FIV(8)= 0.00 | 00000870 |
| | FIV(9)= 0.00 | 00000880 |
| C | | 00000890 |
| | FIW(1)= U | 00000900 |
| | FIW(2)= U**2 | 00000910 |
| | FIW(3)= FIW(2)*SA | 00000920 |
| | FIW(4)= 0.00 | 00000930 |
| | FIW(5)= 0.00 | 00000940 |
| | FIW(6)= 0.00 | 00000950 |
| | FIW(7)= U**2.500 | 00000960 |
| | FIW(8)= U**2.2500 | 00000970 |
| | FIW(9)= U**2.00 | 00000980 |
| C | | 00000990 |
| | SUM= 0.00 | 00001000 |
| | DO 7 I= 1,9 | 00001010 |
| 7 | SUM= SUM+AF(I)*FIU(I) | 00001020 |
| | SUM= -G(1)*SUM/R**3 | 00001030 |
| C | | 00001040 |
| | PFPX= SUM*X1 | 00001050 |
| | PFPY= SUM*Y1 | 00001060 |
| C | | 00001070 |
| | SUM= 0.00 | 00001080 |
| | DO 8 I= 1,9 | 00001090 |
| R | SUM= SUM+AF(I)*(FIV(I)-FIW(I)*SA/CA) | 00001100 |
| C | | 00001110 |
| | PFPSA= G(1)*SUM*SGN | 00001120 |
| C | | 00001130 |
| C | | 00001140 |

| | | |
|----|--|----------|
| C | PF(1) = (PFPX*X + (PY*Y))/Z(1) | 00001150 |
| | DUM1 = PFPX + PFPSA*(C(1) - Y)*SA/R)/R | 00001160 |
| | DUM2 = PFPY + PFPSA*(C(2) - Y)*SA/R)/R | 00001170 |
| | PF(2) = DUM1*PXPH + DUM2*PYPH | 00001180 |
| | PF(3) = DUM1*PXPX + DUM2*PYPK | 00001190 |
| | PF(4) = PFPSA*(C(3)*X + C(4)*Y)/R | 00001200 |
| | PF(5) = PFPSA*(C(5)*X + C(6)*Y)/R | 00001210 |
| C | | 00001220 |
| | DO 10 I = 1,5 | 00001230 |
| 10 | G(I+1) = PF(I)*S | 00001240 |
| | G(3) = G(3) - G(1)*SF | 00001250 |
| | G(4) = G(4) - G(1)*CF | 00001260 |
| | G(1) = G(1)*S | 00001270 |
| C | | 00001280 |
| | IF (M.EQ.1) RETURN | 00001290 |
| | F = F2 | 00001300 |
| | M = 1 | 00001310 |
| | DO 20 J = 1,6 | 00001320 |
| 20 | H(1) = G(1) | 00001330 |
| | GO TO 5 | 00001340 |
| | END | 00001350 |
| | | 00001360 |

| | | |
|--|---------------------------|----------|
| C INPUT/INPUTN | | 00000010 |
| C | | 00000020 |
| C NEP AND HIGH THRUST. | | 00000070 |
| C THIS SUBPROGRAM IS CALLED BY CONTL AND READS AND PRINTS | | 00000040 |
| C ALL INITIAL DATA AS WELL AS SETS INITIAL CONSTANTS. | | 00000050 |
| C THE UNITS ARE BASED ON INTERNAL MU=1.0, INTERNAL DISTANCE | | 00000060 |
| C UNIT=1 EARTH RADII, AND EXTERNAL MU= 398601.2 KM*KM*KM/ | | 00000070 |
| C SEC*SEC AND EARTH RADII= 6378.16 KM. A CIRCULAR | | 00000080 |
| C ORB., AT 1 EARTH RADII WOULD HAVE A PERIOD OF 2 PI INTERNAL | | 00000090 |
| C TIME UNITS. | | 00000100 |
| C TO BE USED WITH 6 DIM. ZERF. | | 00000110 |
| C INPUT | | 00000120 |
| C LOW/HIGH HIGH/LOW/HIGH | | 00000130 |
| C INITIAL ORBIT | | 00000140 |
| C A (KM) | | 00000150 |
| C E | | 00000160 |
| C I (DEG) | | 00000170 |
| C LONG. ASC. NODE (DEG) NOT USED | | 00000180 |
| C ARG. OF PERIGEE (DEG) NOT USED | | 00000190 |
| C INITIAL GUESSES | | 00000200 |
| C LAMBDA A LIKE UPSILON | | 00000210 |
| C LAMBDA H LIKE SMALL'S K | | 00000220 |
| C LAMBDA K LIKE SMALL'S J | | 00000230 |
| C LAMBDA P SCALE FACTOR | | 00000240 |
| C LAMBDA G DUMMY--NOT USED | | 00000250 |
| C DESIRED FINAL ORBIT | | 00000260 |
| C A | | 00000270 |
| C E | | 00000280 |
| C I | | 00000290 |
| C NODE (NOT USED IF NOP=2) NOT USED | | 00000300 |
| C PERIGEE (NOT USED IF NOP=2) NOT USED | | 00000310 |
| C | | 00000320 |
| C TF2 (DAYS), GUESS FOR FINAL TIME | | 00000330 |
| C PKW (KW), POWER | | 00000340 |
| C SPIM (SEC), SPECIFIC IMPULSE OF NEP | | 00000350 |
| C AMO (KG), INITIAL MASS (NEP) | | 00000360 |
| C IROFLG NOMINAL | | 00000370 |
| C 1 END OF INPUT | | 00000380 |
| C 2 IPR, PRINT FLAG | 0 | 00000390 |
| C 3 NIMAX, MAX. NO. OF ITERATIONS | 20 | 00000400 |
| C 4 TFMAX2 (DAYS), MAX. TF | 190. | 00000410 |
| C 5 DT2 (DAYS), TIME STEP FOR D.E. | 1. | 00000420 |
| C 6 UEB, UPPER ERROR BOUND FOR D.E. | 1.010 | 00000430 |
| C 7 EW, ERROR WEIGHTS | 1.,1.,1.,1.,1.,1.,0.,.... | 00000440 |
| C 8 RUTKM, EQUATORIAL EARTH RADIUS | 6378.16 | 00000450 |
| C 9 GM (KM**3/SEC**2), EARTH GRAV. CONST. | 398601.2 | 00000460 |
| C 10 NOP = 1, 5 D.E. SPECIFIED AT TF | 1 | 00000470 |
| C = 2, 3 D.E. SPECIFIED AT TF | | 00000480 |
| C 11 SETS ORBITALNESS COEFF. AJ2.=1.0827D-5 | 0. | 00000490 |
| C 12 STEP, STEP SIZE FOR NUM. DIFF. (5 DIM) | 1.0-6 | 00000500 |
| C KSTEP = 0, STEP AS FRACTION IN ITER | 0 | 00000510 |
| C = 1, STEP AS CONSTANT IN ITER | | 00000520 |
| C 13 IPOWER = 0, CONSTANT POWER | 0 | 00000530 |
| C = 1, EXPONENTIAL DEGRADATION | | 00000540 |
| C BB (SEC), TIME CONSTANT | 1.07 | 00000550 |
| C PH (KW), HOUSEKEEPING POWER | 0. | 00000560 |
| C 14 EMPTY | | 00000570 |

```

C 15 IHI = 1. LOW THRUST ONLY          1          00000580
C      = 2. HIGH/LOW                   1          00000590
C      = 3. HIGH/LOW/HIGH               1          00000600
C      = 4. LOW/HIGH                    1          00000610
C      DVI1 (M/S) TOTAL INITIAL DEL. V  0.          00000620
C      DVI2 (M/S), DEL. FOR FINAL IMPULSE 0.          00000630
C 16 EMPTY                               1          00000640
C 17 FLIM NORM LIMIT IN ITER ROUTINE    1.0-06      00000650
C 18 SGN = -1., INITIAL LAM I NEGATIVE  1          00000660
C      = +1., INITIAL LAM I POSITIVE    1          00000670
C                                          1          00000680
C                                          1          00000690
C                                          1          00000700
C      SUBROUTINE INPUT                  1          00000710
C                                          1          00000720
C      IMPLICIT REAL*8(A-H,O-S), INTEGER (I-N) 1          00000730
C                                          1          00000740
C      NAMELIST /UN/UTKM,UTS,UTD,UTMS2,UTKW 1          00000750
C                                          1          00000760
C                                          1          00000770
C      COMMON /XMM/2LO(5), STEP(5), ZERF(6) 1          00000780
C      COMMON /ELEM/2PO(5), ZPF(5)          1          00000790
C      COMMON /INT/ITF,IPR,IDIM,IDIM2,NIMAX 1          00000800
C      COMMON /TRA/TFMAX,DT,UEB,EW(10)     1          00000810
C      COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,DUMMY,UTKW,UTMS2 1          00000820
C      COMMON /T/TF,S,TO,TFMIN             1          00000830
C      COMMON /A/A,AMU,PI                  1          00000840
C      COMMON /WF/WF(5)                    1          00000850
C      COMMON /J2/ AJ2                      1          00000860
C      COMMON /TC/NOP                       1          00000870
C      COMMON /POWER/PKW,CC,AMO,A0,A1,A2,A3,IPOW 1          00000880
C      COMMON/HIGH/DVI1,DVI2,IHI,ITR       1          00000890
C      COMMON /F/FLIM,KSTEP                 1          00000900
C      COMMON /SG/SGN                       1          00000910
C                                          1          00000920
C      DIMENSION W(5)                       1          00000930
C                                          1          00000940
C      INTEGER CONSTANTS                    1          00000950
C                                          1          00000960
C      IHI= 1                               1          00000970
C      IDIM=10                              1          00000980
C      IDIM2=5                              1          00000990
C      IDIM3=6                              1          00001000
C      IPR=0                                1          00001010
C      ITF=3                                1          00001020
C      NIMAX=20                             1          00001030
C      NOP= 1                               1          00001040
C      IPOW= 0                              1          00001050
C      KSTEP= 0                             1          00001060
C                                          1          00001070
C      REAL CONSTANTS                       1          00001080
C                                          1          00001090
C      AJ2= 0.00                            1          00001100
C      AMU=1.000                            1          00001110
C      UEB= 1.00+10                         1          00001120
C      DO 10 I=1,IDIM2                      1          00001130
C      EW(I+IDIM2)= 0.00                   1          00001140

```

5-2

| | | |
|----|---|----------|
| 10 | FW(1)= 1.000 | 00001150 |
| | DO 12 1=1.101M2 | 00001160 |
| 12 | STEP(1)= 1.0-6 | 00001170 |
| | DT2= 1.000 | 00001180 |
| | GM=398601.200 | 00001190 |
| | UTKM= 6378.1600 | 00001200 |
| | DTR= .01745329251994329600 | 00001210 |
| | PI= 3.141592653589793200 | 00001220 |
| | TFMAX2= 190.000 | 00001230 |
| | TFMIN2= 0.000 | 00001240 |
| | T02=0.000 | 00001250 |
| | BR= 1.07 | 00001260 |
| | DV11= 0.00 | 00001270 |
| | DV12= 0.00 | 00001280 |
| | PH= 0.00 | 00001290 |
| | FLIM= 1.0-6 | 00001300 |
| C | | 00001310 |
| C | ALL READ STATEMENTS FOLLOW | 00001320 |
| C | | 00001330 |
| | READ (5,1001) W | 00001340 |
| | READ (5,1001) ZLO | 00001350 |
| | READ (5,1001) WF | 00001360 |
| | SGN= WF(3)-W(3) | 00001370 |
| | IF (SGN.NE.0.00) SGN= SGN/DABS(SGN) | 00001380 |
| | READ (5,1001) TF2 | 00001390 |
| | READ (5,1001) PKW | 00001400 |
| | READ (5,1001) SPIM | 00001410 |
| | READ (5,1001) AMO | 00001420 |
| 20 | READ (5,1002) IRDFLG | 00001430 |
| | IF ((IRDFLG.GT.1P).OR.(IRDFLG.LT.1)) GO TO 200 | 00001440 |
| 25 | GO TO (150,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48), | 00001450 |
| | 1 IRDFLG | 00001460 |
| C | | 00001470 |
| C | THESE VALUES ARE READ ONLY IF INDICATED BY IRDFLG | 00001480 |
| C | | 00001490 |
| 32 | READ (5,1002) IPR | 00001500 |
| | IF (IPR.LT.0) GO TO 210 | 00001510 |
| | GO TO 20 | 00001520 |
| 33 | READ (5,1002) NIMAX | 00001530 |
| | IF ((NIMAX.LT.0).OR.(NIMAX.GT.50)) GO TO 220 | 00001540 |
| | GO TO 20 | 00001550 |
| 34 | READ (5,1001) TFMAX2 | 00001560 |
| | IF ((TFMAX2.LT.0.00).OR.(TFMAX2.GT.1.)) GO TO 220 | 00001570 |
| | GO TO 20 | 00001580 |
| 35 | READ (5,1001) DT2 | 00001590 |
| | IF ((DT2.LT.0.00).OR.(DT2.GT.1.03)) GO TO 220 | 00001600 |
| | GO TO 20 | 00001610 |
| 36 | READ (5,1001) UEB | 00001620 |
| | IF (UEB.LT.0.00) GO TO 220 | 00001630 |
| | GO TO 20 | 00001640 |
| 37 | READ (5,1003) EW | 00001650 |
| | GO TO 20 | 00001660 |
| 38 | READ (5,1001) UTKM | 00001670 |
| | GO TO 20 | 00001680 |
| 39 | READ (5,1001) GM | 00001690 |
| | GO TO 20 | 00001700 |
| 40 | READ (5,1002) NOP | 00001710 |

| | | |
|-----|--|----------|
| | IF ((NOP.LT.0).OR.(NOP.GT.1)) GO TO 220 | 00001720 |
| | GO TO 20 | 00001730 |
| 41 | AJ2= 1.0R27D-3 | 00001740 |
| | GO TO 20 | 00001750 |
| 42 | READ (5,1001) (STEP1),I=1,5) | 00001760 |
| | READ (5,1002) KSTEP | 00001770 |
| | IF ((KSTEP.LT.0).OR.(KSTEP.GT.1)) GO TO 220 | 00001780 |
| | GO TO 20 | 00001790 |
| 43 | READ (5,1002) IPDW | 00001800 |
| | READ (5,1001) BR,PH | 00001810 |
| C | BB SHOULD BE IN SECONDS | 00001820 |
| | IF ((IPDW.LT.0).OR.(IPDW.GT.1)) GO TO 230 | 00001830 |
| | GO TO 20 | 0000184 |
| 44 | GO TO 20 | 0000185C |
| 45 | READ (5,1002) IMI | 00001860 |
| | IF ((IMI.LT.0).OR.(IMI.GT.4)) GO TO 220 | 00001870 |
| | NOP= 2 | 00001880 |
| | READ (5,1001) DV11, DV12 | 00001890 |
| | IF ((IMI.LT.3) DV12= 0.00 | 00001900 |
| | IF ((IMI.EQ.1).OR.(IMI.EQ.4)) DV11= 0.00 | 00001910 |
| | GO TO 20 | 00001920 |
| 46 | GO TO 20 | 00001930 |
| 47 | READ (5,1003) FLIM | 00001940 |
| | IF (FLIM.LT.0.00) GO TO 220 | 00001950 |
| | GO TO 20 | 00001960 |
| 48 | READ (5,1001) SGN | 00001970 |
| | IF ((SGN.EQ.1.00).AND.(SGN.EQ.-1.00)) GO TO 220 | 00001980 |
| | GO TO 20 | 00001990 |
| C | | 00002000 |
| C | | 00002010 |
| C | TIME VALUES ARE CHANGED FROM DAYS TO OTHER UNITS | 00002020 |
| C | | 00002030 |
| 150 | UTS= DSORT(UTKM**3/GM) | 00002040 |
| | UTM=UTS/60.00 | 00002050 |
| | UTH=UTS/3600.00 | 00002060 |
| | UTD=UTH/24.00 | 00002070 |
| | T0= T02/UTD | 00002080 |
| | TF= TF2/UTD | 00002090 |
| | TFMIN= TFMIN2/UTD | 00002100 |
| | TFMAX= TFMAX2/UTD | 00002110 |
| | DT= DT2/UTD | 00002120 |
| | T01= T0*UTS | 00002130 |
| | TF1= TF*UTS | 00002140 |
| | TFMIN1= TFMIN*UTS | 00002150 |
| | TFMAX1= TFMAX*UTS | 00002160 |
| | DT1= DT*UTS | 00002170 |
| C | | 00002180 |
| C | MORE CONVERSIONS | 00002190 |
| | UTMS2= (UTKM*1.03/(UTS**2) | 00002200 |
| | UTKW= UTMS2*UTKM/UTS | 00002210 |
| C | | 00002220 |
| | C= SPIM/UTS | 00002230 |
| | CC= C*UTKM/UTS | 00002240 |
| | AA= 2.00*PKW/(AM0*CC) | 00002250 |
| | AAA= 2.00*(PKW-PH)/(AM0*CC) | 00002260 |
| | BB1= BB/UTS | 00002270 |
| | AO= AA/UTMS2 | 00002280 |

| | |
|---|----------|
| A1= A0/C | 00002290 |
| A0A= AAA/UTMS2 | 00002300 |
| IF (1/POW.EQ.1) A1= A1*RA1 | 00002310 |
| A2= 1.00/RA1 | 00002320 |
| RA2= RA1*UTD | 00002330 |
| A3= PH/PKW | 00002340 |
| C | 00002350 |
| C | 00002360 |
| C THE PRINTING OF ALL INITIAL VALUES FOLLOWS | 00002370 |
| C | 00002380 |
| WRITE (6,2000) | 00002390 |
| WRITE (6,2001) | 00002400 |
| WRITE (6,2030) | 00002410 |
| WRITE (6,2033) | 00002420 |
| WRITE (6,2034) | 00002430 |
| WRITE (6,2035) NOP | 00002440 |
| WRITE (6,2043) DV11,UV12 | 00002450 |
| C | 00002460 |
| WRITE (6,2002) | 00002470 |
| WRITE (6,2003) | 00002480 |
| IF ((1/MI.EQ.1).OR.(1/MI.FQ.4)) GO TO 158 | 00002490 |
| C W(1) IS SEMIMAJOR AXIS | 00002500 |
| C W(2) IS ECC. SET TO 0 | 00002510 |
| C W(3) IS INCLINATION | 00002520 |
| C ZLO SHOULD BE Y1 (UP),Y2 (XK),Y3 (XJ),C,1. | 00002530 |
| C | 00002540 |
| W(2)= 0.00 | 00002550 |
| W(4)= 0.00 | 00002560 |
| WRITE (6,2004) (W(I),I=1,4) | 00002570 |
| ZPO(1)= W(1)/UTKM | 00002580 |
| ZPO(2)= W(3)*DTR | 00002590 |
| WRITE (6,2044) ZPO(1) | 00002600 |
| ZLO(5)= 1.00 | 00002610 |
| GO TO 162 | 00002620 |
| 158 WRITE (6,2004) W | 00002630 |
| C | 00002640 |
| C CHANGE FROM CLASSICAL O.E. TO EQUINOCTIAL O.E. | 00002650 |
| 159 DO 160 I=3,5 | 00002660 |
| 160 W(I)= W(I)*DTR | 00002670 |
| ZPO(1)= W(1)/UTKM | 00002680 |
| ZPO(2)= W(2)*DSIN(W(5)+W(4)) | 00002690 |
| ZPO(3)= W(2)*DCOS(W(5)+W(4)) | 00002700 |
| ZPO(4)= DTAN(W(3)/2.000)*DSIN(W(4)) | 00002710 |
| ZPO(5)= DTAN(W(3)/2.000)*DCOS(W(4)) | 00002720 |
| C | 00002730 |
| WRITE (6,2005) | 00002740 |
| WRITE (6,2004) ZPO | 00002750 |
| C | 00002760 |
| C WRITE FINAL CONDITIONS. CHANGE TO EQUINOCTIAL FINAL COND. | 00002770 |
| C | 00002780 |
| 162 DV11= (DV11/(UTMS2*UTS))*DSQRT(ZPO(1)/AMU) | 00002790 |
| DV12= DV12/(UTMS2*UTS) | 00002800 |
| C | 00002810 |
| WRITE (6,2006) | 00002820 |
| ZPF(1)= WF(1)/UTKM | 00002830 |
| GO TO (165,170), NOP | 00002840 |
| C | 00002850 |

| | | |
|-----|---|----------|
| 165 | WRITE (6,2003) | 00002860 |
| | WRITE (6,2004) WF | 00002870 |
| | DO 166 I=3,5 | 00002880 |
| 166 | WF(I)=WF(I)*DTR | 00002890 |
| | ZPF(2)=WF(2)*DSIN(WF(5)+WF(4)) | 00002900 |
| | ZPF(3)=WF(2)*DCOS(WF(5)+WF(4)) | 00002910 |
| | ZPF(4)=DTAN(WF(3)/2.000)*DSIN(WF(4)) | 00002920 |
| | ZPF(5)=DTAN(WF(3)/2.000)*DCOS(WF(4)) | 00002930 |
| | DO 167 I=3,5 | 00002940 |
| 167 | WF(I)=WF(I)/DTR | 00002950 |
| | WRITE (6,2005) | 00002960 |
| | WRITE (6,2004) ZPF | 00002970 |
| | GO TO 190 | 00002980 |
| C | | 00002990 |
| 170 | ZPF(2)=WF(2) | 00003000 |
| | ZPF(3)=DABS(DTAN(WF(3)*DTR/2.00)) | 00003010 |
| | ZPF(4)=0.00 | 00003020 |
| | ZPF(5)=0.00 | 00003030 |
| | WRITE (6,2031) | 00003040 |
| | WRITE (6,2004) (WF(I),I=1,3) | 00003050 |
| | WRITE (6,2032) | 00003060 |
| | WRITE (6,2004) (ZPF(I),I=1,3) | 00003070 |
| C | | 00003080 |
| 190 | IF (IMI.EQ.1) GO TO 194 | 00003090 |
| | WRITE (6,2045) | 00003100 |
| | GO TO 196 | 00003110 |
| 194 | WRITE (6,2007) | 00003120 |
| 196 | WRITE (6,2011) ZLO | 00003130 |
| | IF ((IMI.GT.1).AND.(IMI.LT.4)) WRITE (6,2047) SGN | 00003140 |
| | WRITE (6,2008) | 00003150 |
| | WRITE (6,2009) TF2,TF1,TF | 00003160 |
| | WRITE (6,2040) SPIM,CC | 00003170 |
| | WRITE (6,2038) AMO | 00003180 |
| | WRITE (6,2039) PKW | 00003190 |
| | IF (IPDW.EQ.1) WRITE (6,2041) BB,BB2 | 00003200 |
| | WRITE (6,2012) AAA,ADA | 00003210 |
| | WRITE (6,2036) AJ2 | 00003220 |
| | WRITE (6,2013) | 00003230 |
| | WRITE (6,2009) T02,T01,TO | 00003240 |
| | WRITE (6,2014) | 00003250 |
| | WRITE (6,2009) TFMIN2,TFMIN1,TFMIN | 00003260 |
| | WRITE (6,2015) | 00003270 |
| | WRITE (6,2009) TFMAX2,TFMAX1,TFMAX | 00003280 |
| | WRITE (6,2010) KSTEP | 00003290 |
| | WRITE (6,2011) STEP | 00003300 |
| | WRITE (6,2016) | 00003310 |
| | WRITE (6,2009) DT2,DT1,DT | 00003320 |
| | WRITE (6,2017) UEB | 00003330 |
| | WRITE (6,2018) | 00003340 |
| | WRITE (6,2019) EW | 00003350 |
| | WRITE (6,2020) IDIM | 00003360 |
| | WRITE (6,2022) NIMAX | 00003370 |
| | WRITE (6,2046) FLIM | 00003380 |
| | WRITE (6,2036) AJ2 | 00003390 |
| | WRITE (6,2026) UTKM | 00003400 |
| | WRITE (6,2027) GM | 00003410 |
| | WRITE (6,UN) | 00003420 |

| | |
|--|----------|
| RETURN | 00003430 |
| 200 WRITE (6,2023) IRDFLG | 00003440 |
| STOP | 00003450 |
| 210 WRITE (6,2024) IPR | 00003460 |
| STOP | 00003470 |
| 220 WRITE (6,2025) IRDFLG | 00003480 |
| STOP | 00003490 |
| 230 WRITE (6,2042) IPOW | 00003500 |
| STOP | 00003510 |
| C | 00003520 |
| 1001 FORMAT (F25.15) | 00003530 |
| 1002 FORMAT (I2) | 00003540 |
| 1003 FORMAT (5D6.1) | 00003550 |
| 2000 FORMAT (1H1,22X,69H OPTIMAL TRAJECTORY PROGRAM FOR NEP AND HIGH THROUST SATELLITE RAISING) | 00003560 |
| 2001 FORMAT (1H0,40X,39H MINIMUM TIME NEP) | 00003580 |
| 2002 FORMAT (34H0 THE INITIAL ORBITAL ELEMENTS ARE) | 00003590 |
| 2003 FORMAT (1H0,10X,6HA (KM),19X,1HE,20X,7HI (DEG),10X,18HLON ASC NODE 1 (DEG),6X,15HARG PERIG (DEG)) | 00003600 |
| 2004 FORMAT (1P5D23.14) | 00003610 |
| 2005 FORMAT (1H0,6X,13HA (EARTH RAD),16X,1HM,22X,1MK,22X,1MP,22X,1HD) | 00003630 |
| 2006 FORMAT (40H0 THE DESIRED FINAL ORBITAL ELEMENTS ARE) | 00003640 |
| 2007 FORMAT (32H0 INITIAL GUESSED PARAMETERS ARE) | 00003650 |
| 2008 FORMAT (21H0 FINAL TIME ESTIMATE) | 00003660 |
| 2009 FORMAT (1H ,10X,1PD22.15,7H DAYS =,1PD22.15,10H SECONDS =,1PD22.15,16H UNITS) | 00003670 |
| 2010 FORMAT (50H0 STEP SIZE FOR NUMERICAL DIFFERENTIATION, KSTEP =,12) | 00003690 |
| 2011 FORMAT (1P5D23.14) | 00003700 |
| 2012 FORMAT (22H0 ACCELERATION LEVEL =,1PD25.15,12H M/SEC/SEC =,1PD25.15,4H G'S) | 00003710 |
| 2013 FORMAT (17H0 INITIAL TIME IS) | 00003730 |
| 2014 FORMAT (10H0 TFMIN IS) | 00003740 |
| 2015 FORMAT (10H0 TFMAX IS) | 00003750 |
| 2016 FORMAT (27H0 TIME STEP FOR INTEGRATION) | 00003760 |
| 2017 FORMAT (36H0 UPPER ERROR BOUND IN INTEGRATION =,1PD20.10) | 00003770 |
| 2018 FORMAT (35H0 ERROR WEIGHTS FOR INTEGRATION ARE) | 00003780 |
| 2019 FORMAT (1P10D12.4) | 00003790 |
| 2020 FORMAT (13H0 DIMENSION =,15) | 00003800 |
| 2022 FORMAT (31H0 MAXIMUM NUMBER OF ITERATION =,15) | 00003810 |
| 2023 FORMAT (44H0 IRDFLG SHOULD BE BETWEEN 1 AND 15, IT IS =,15) | 00003820 |
| 2024 FORMAT (28H0 IPR SHOULD BE < 0, IT IS =,15) | 00003830 |
| 2025 FORMAT (27H0 BAD INPUT DATA, IRDFLG = ,13) | 00003840 |
| 2026 FORMAT (17H0 1 EARTH RADIUS =,F25.12,3H KM) | 00003850 |
| 2027 FORMAT (11H0 MU (GM) =,F25.10,13H KM**3/SEC**2) | 00003860 |
| 2030 FORMAT (25H0 FINAL CONDITION OPTIONS) | 00003870 |
| 2033 FORMAT (43H 1. ALL 5 FINAL ORBITAL ELEMENTS SPECIFIED) | 00003880 |
| 2034 FORMAT (51H 2. A.E.1 SPECIFIED, LON ASC NODE AND ARG PER FREE) | 00003890 |
| 2035 FORMAT (24H FOR THIS RUN, OPTION =,14) | 00003900 |
| 2031 FORMAT (1H0,10X,6HA (KM),19X,1HE,20X,7HI (DEG)) | 00003910 |
| 2032 FORMAT (1H0,6X,13HA (EARTH RAD),9X,15HSORT(H**2+K**2),8X, 1 15HSORT(P**2+Q**2)) | 00003920 |
| 2036 FORMAT (6H0 J2 =,1PD15.7) | 00003940 |
| 2038 FORMAT (21H0 INITIAL MASS (KG) =, F1R.11) | 00003950 |
| 2039 FORMAT (22H0 INITIAL POWER (KW) =, 1PD23.14) | 00003960 |
| 2040 FORMAT (12H0 SPEC I-P =,1PD23.14,15H SEC, EXH VEL =,1PD23.14, 1 5H KM/S) | 00003970 |
| 2041 FORMAT (33H0 POWER INTEGRATION TIME CONSTANT =,1PD23.15,6H SEC =, | 00003980 |
| | 00003990 |

| | | |
|------|--|----------|
| 1 | F20.10,4HDAYS) | 00004000 |
| 2042 | FORMAT (30HD IPW SHOULD BE 0 OR 1, IPW=, 13) | 00004010 |
| 2043 | FORMAT (41HD TOTAL DELV FOR INITIAL IMPULSES (M/S) =, F20.12,5X, | 00004020 |
| 1 | 21HFINAL IMPULS. (M/S) =, F20.12) | 00004030 |
| 2044 | FORMAT (27HD INITIAL A (EARTH EARTH) =, IPD25.14) | 00004040 |
| 2045 | FORMAT (42HD INITIAL GUESS FOR HIGH THRUST, S.F., INE) | 00004050 |
| 2046 | FORMAT (23HD NORM LIMIT IN ITER =, IPD12.5) | 00004060 |
| 2047 | FORMAT (29HD SIGN OF INITIAL LAMBDA 1 IS, Fh.1) | 00004070 |
| | END | 00004080 |


```

C OUTPC/OUTPCN                                00000010
C                                                00000020
C NEP AND HIGH THRUST                          00000025
C THIS SUBPROGRAM WRITES THE VALUES FOR THE FINAL CONVERGED
C TRAJECTORY. IT IS CALLED BY THE MAIN PROGRAM CONTL. 00000030
C TO BE USED WITH 6 DIM. ZERF.                00000040
C                                                00000050
C                                                00000070
C                                                00000080
C                                                00000090
C SUBROUTINE OUTPC                             00000100
C                                                00000110
C IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)      00000120
C                                                00000130
C COMMON /XMM/ZLO(5),STEP(5),ZERF(6)         00000140
C COMMON /Z/ZF(10),DZ(10)                    00000150
C COMMON /T/TF,S,TD,TF*IN                     00000160
C COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,DUMMY,UTKW,UTMS2 00000170
C COMMON /ELEM/ZPO(5),ZPF(5)                  00000180
C COMMON /WF/WF(5)                             00000190
C COMMON /A/A,AMU,PI                           00000200
C COMMON /TC/NDP                                00000210
C COMMON /POWER/PKW,CC,AM0,A0,A1,A2,A3,IP0W   00000220
C                                                00000230
C DIMENSION DELZF(5),DELWF(5),WFC(5)         00000240
C                                                00000250
C WFC(1)= ZF(1)*UTKM                           00000260
C WFC(2)= 0.000                                00000270
C DUMMY= ZF(2)**2 + ZF(3)**2                   00000280
C IF (DUMMY.GT.1.0D-40) WFC(2)=DSQRT(DUMMY)    00000290
C WFC(3)= 0.000                                00000300
C DUMMY= ZF(4)**2 + ZF(5)**2                   00000310
C IF (DUMMY.GT.1.0D-40) WFC(3)= 2.0D0*DATAN(DSQRT(DUMMY))/DTR 00000320
C WFC(4)=0.000                                00000330
C IF ((DABS(ZF(4)).GT.1.0D-8).AND.(DABS(ZF(5)).GT.1.0D-8)) 00000340
1 WFC(4)= DATAN2(ZF(4),ZF(5))/DTR              00000350
C WFC(5)= 0.000                                00000360
C IF ((DABS(ZF(2)).GT.1.0D-8).AND.(DABS(ZF(3)).GT.1.0D-8)) 00000370
1 WFC(5)=DATAN2(ZF(2),ZF(3))/DTR              00000380
C WFC(5)=WFC(5)-WFC(4)                         00000390
C DO 10 J=1,5                                  00000400
C   DELWF(J)= WFC(J) - WF(J)                   00000410
10  DELZF(J)= ZF(J) - ZPF(J)                   00000420
C   TF2= TF*UTD                                00000430
C   TF1= TF*UTS                                00000440
C   IF (IP0W.EQ.1) B1= DEXP(-A2*TF)            00000450
C   IF (IP0W.EQ.1) RMASS= 1.0D0+A1*(B1-1.0D0+A3*A2*TF) 00000460
C   IF (IP0W.EQ.0) RMASS= 1.0D0-A1*TF        00000470
C   AMASS= AM0*RMASS                           00000480
C   DELV= -CC*DLOG(RMASS)                     00000490
C   POW= PKW                                    00000500
C   IF (IP0W.EQ.1) POW= POW*(B1-A3)           00000510
C   RPOW= POW/PKW                              00000520
C                                                00000530
C WRITE (6,3000)                               00000540
C WRITE (6,3001)                               00000550
C WRITE (6,3002) WFC                          00000560
C WRITE (6,3003)                               00000570

```

```

WRITE (6,3002) (ZF(I),I=1,5) 00000590
WRITE (6,3004) 00000590
WRITE (6,3001) 00000600
IF (NOP.EQ.1) WRITE (6,3002) DELWF 00000610
IF (NOP.EQ.2) WRITE (6,3002) (DELWF(I),I=1,3) 00000620
GO TO (20, 30), NOP 00000630
C 00000640
20 WRITE (6,3003) 00000650
WRITE (6,3002) DELZF 00000660
GO TO 100 00000670
C 00000680
30 DELZF(2)= DSQRT(7F(2)**2+7F(3)**2) -ZPF(2) 00000690
DELZF(3)= DSQRT(7F(4)**2+ZF(5)**2) -ZPF(3) 00000700
WRITE (6,3011) 00000710
WRITE (6,3002) (DELZF(I),I=1,3) 00000720
C 00000730
100 WRITE (6,3006) 00000740
WRITE (6,3002) ZLO 00000750
WRITE (6,3008) 00000760
WRITE (6,3009) TF2,TF1,TF 00000770
WRITE (6,3013) AMASS,RMASS 00000780
WRITE (6,3014) POW,PPOW 00000790
WRITE (6,3010) DELV 00000800
RETURN 00000810
3000 FORMAT (35H0 ACTUAL FINAL ORBITAL ELEMENTS ARE) 00000820
3001 FORMAT (1H0,10X,6HA (KM),18X,1HE,20X,7H) (DEG),10X,18HLON ASC NODE00000830
1 (DEG),6X,15HARG PERIG (DEG)) 00000840
3002 FORMAT (1P5D23.15) 00000850
3003 FORMAT (1H0,5X,13HA (EARTH RAD),16X,1HH,22X,1HK,22X,1HP,22X,1HO) 00000860
3004 FORMAT (51H0 THE ERROR IN THE FINAL O.E. IS (ACTUAL - DESIRED)) 00000870
C3005 FORMAT (60H0 CLASSICAL O.E. MAY HAVE DISCREPANCY OF MULTIPLES OF 900000880
10 DEG) 00000890
C 00000900
3006 FORMAT (46H0 THE CONVERGED INITIAL GUESSED PARAMETERS ARE) 00000900
3008 FORMAT (29H0 THE MINIMIZED FINAL TIME IS) 00000910
3009 FORMAT (1H ,10X,1PD22.15,7H DAYS =,D22.15,10H SECONDS =,1PD22.15,600000920
1H UNITS) 00000930
3010 FORMAT (18H0 LOW THRUST DELV=,1PD25.14,7H KM/SEC) 00000940
3011 FORMAT (1H0,5X,13HA (EARTH AD),9X,15HSQRT(H**2+K**2),8X,
1 15HSQRT(P**2+Q**2)) 00000950
3013 FORMAT (14H0 FINAL MASS =,F22.15,31H KG, FRACTION OF INITIAL MASS 00000970
1=,F22.15) 00000980
3014 FORMAT (15H0 FINAL POWER =, F22.15,32H KW, FRACTION OF INITIAL POW00000990
1ER =,F22.15) 00001000
END 00001010

```

| | | |
|----|--|------------|
| C | ITER MODNRN | 00000070 |
| C | | 00000030 |
| C | | 00000040 |
| C | 6X6 VERSION | 00000050 |
| C | | 00000060 |
| C | SUBROUTINE ITER(KOUNT,NI,FUNCT,PRIN) | 00000070 |
| C | | 00000080 |
| C | IMPLICIT REAL*(A-H,O-S) | 00000090 |
| C | | 00000100 |
| C | | 00000110 |
| C | VALUES OF THE INDEPENDENT VARIABLES(INITIAL,CURRENT,FINAL) | DP00000120 |
| C | STEP SIZE TO PERTURB X:S TO COMPUTE PARTIAL DERIVATIVES | SP00000130 |
| C | VALUES OF THE DEPENDENT VARIABLES(CURRENT,FINAL) | DP00000140 |
| C | COMMON/XMM/X(5),XS(5),Y(6) | 00000150 |
| C | COMMON /INT/ITF,IPR,IDIM,IDIM2,MAXNDI | 00000160 |
| C | COMMON /T/TF,S,TC,TFMIN | 00000170 |
| C | COMMON /DY/ DYDT(6) | 00000180 |
| C | COMMON /HIGH/DV11,DV12,IM1,ITR | 00000190 |
| C | COMMON /F/FLIM,KSIEP | 00000200 |
| C | | 00000210 |
| C | DIMENSION YNOM(6),XN(5),P(6,6),COEF(6),DYDTN(6) | 00000220 |
| C | N=5 | 00000230 |
| C | M=6 | 00000240 |
| C | INDRM=ITF | 00000250 |
| C | IR=1 | 00000260 |
| C | ICONS=1 | 00000270 |
| C | ISW=0 | 00000280 |
| C | N 1 | 00000290 |
| C | KPJNT=0 | 00000300 |
| C | ITR=0 | 00000310 |
| C | CALL FUNCT | 00000320 |
| C | ITF=3 | 00000330 |
| C | KOUNT=KOUNT+1 | 00000340 |
| C | F0=0.DO | 00000350 |
| C | DO 15 I=1,M | 00000360 |
| 15 | F0=F0+Y(I)**2 | 00000370 |
| 9 | DO 16 I=1,N | 00000380 |
| C | DYDTN(I)= DYDT(I) | |
| C | XN(I)=X(I) | 00000400 |
| 16 | YNOM(I)=Y(I) | 00000410 |
| C | YNOM(M)= Y(M) | 00000420 |
| C | DYDTN(M)= DYDT(M) | 00000430 |
| C | TFN=TF | 00000440 |
| 10 | CALL PRIN(KOUNT,NI) | 00000450 |
| C | WRITE(6,1011)F0 | 00000460 |
| C | IF(F0.LE.FLIM)GO TO 90 | 00000470 |
| C | IF (NI.GT.MAXNDI) GO TO 80 | 00000480 |
| C | IF(ISW.NE.0)GO TO 27 | 00000490 |
| C | | 00000500 |
| C | COMPUTE NUMERICAL PARTIAL DERIVATIVES | |
| C | DO 17 I=1,M | 00000510 |
| 17 | P(I,M)= DYDT(I) | 00000520 |
| C | WRITE (6,1013) | 00000530 |
| C | DO 25 J=1,N | 00000540 |
| C | ITR=J | 00000550 |
| C | TEMP=X(J) | 00000560 |
| C | STEP=X(S(J))*DABS(X(J)) | 00000570 |

| | | |
|----|--|----------|
| | IF ((DABS(X(J)).LT.1.D-10).OR.(XSTEP.FO.))STEP=XS(J) | 00000580 |
| C | IF (DABS(X(J)).LT.1.D-10) WRITE (6,1014) | 00000590 |
| | X(J)=X(J)+STEP | 00000600 |
| | CALL FUNCT | 00000610 |
| | WRITE(6,1000)X(J) | 00000620 |
| | WRITE(6,1001)(Y(I),I=1,M) | 00000630 |
| | DO 20 I=1,M | 00000640 |
| 20 | P(I,J)=(Y(I)-YNOM(I))/STEP | 00000650 |
| 25 | X(J)=TEMP | 00000660 |
| | KOUNT=KOUNT+N | 00000670 |
| 27 | WRITE(6,1002) | 00000680 |
| | DO 30 I=1,M | 00000690 |
| | WRITE(6,1001)(P(I,J),J=1,M) | 00000700 |
| 30 | CONTINUE | 00000710 |
| | DO 35 I=1,M | 00000720 |
| 35 | COEF(I)=-YNOM(I) | 00000730 |
| | CALL DCR0UT(P,COEF,1,1,0,DO,M,1,IND) | 00000740 |
| | IF(IND.NE.0)GO TO 45 | 00000750 |
| | WRITE (6,1015) DE1 | 00000760 |
| | DO 40 I=1,M | 00000770 |
| 40 | IF (DABS(COEF(I)).LT.1.D-10) COEF(I)= 0.D0 | 00000780 |
| | WRITE(6,1003)(COEF(I),I=1,N) | 00000790 |
| | SN= COEF(M) | 00000800 |
| | WRITE (6,1012) SN | 00000810 |
| | DO 50 J=1,N | 00000820 |
| 50 | X(J)=XN(J)+COEF(J) | 00000830 |
| | TF=TF+SN | 00000840 |
| | IHALV=0 | 00000850 |
| 51 | IF (INORM.EQ.1) IIF=1 | 00000860 |
| | ITR=0 | 00000870 |
| | CALL FUNCT | 00000880 |
| | ITF=3 | 00000890 |
| | KOUNT=KOUNT+1 | 00000900 |
| | F1=0.D0 | 00000910 |
| | DO 52 I=1,M | 00000920 |
| 52 | F1=F1+Y(I)**2 | 00000930 |
| | WRITE(6,1010)F1 | 00000940 |
| | IF(F1.LT.F0)GO TO 55 | 00000950 |
| | WRITE(6,1008) | 00000960 |
| | IF(IHALV.EQ.8)GO TO 95 | 00000970 |
| | IHALV=IHALV+1 | 00000980 |
| | DO 53 J=1,N | 00000990 |
| | COEF(J)=COEF(J)/2.D0 | 00001000 |
| | WRITE(6,1000)COEF(J) | 00001010 |
| 53 | X(J)=XN(J)+COEF(J) | 00001020 |
| | SN=SN/2.D0 | 00001030 |
| | WRITE (6,1012) SN | 00001040 |
| | TF=TF+SN | 00001050 |
| | GO TO 51 | 00001060 |
| 55 | IF(NI-MAX(J))70,70,70 | 00001070 |
| 70 | NI=NI+1 | 00001080 |
| | ICONS=NI | 00001090 |
| | F0=F1 | 00001100 |
| | SUMDX=0.D0 | 00001110 |
| | DO 76 J=1,M | 00001120 |
| 76 | SUMDX=COEF(J)**2+SUMDX | 00001130 |
| | DO 77 I=1,M | 00001140 |

| | | |
|----|---|----------|
| | DO 77 J=1,M | 00001150 |
| | P(I,J)=P(I,J)+(Y(I)*COFF(J))/SUMDX | 00001160 |
| | 77 CONTINUE | 00001170 |
| | ISW=1 | 00001180 |
| | GO TO 9 | 00001190 |
| | 80 NI=9999 | 00001200 |
| | WRITE(6,1006) | 00001210 |
| C | PRINT 1006 | 000012 0 |
| | RETURN | 00001230 |
| | 85 NI=9999 | 00001240 |
| | WRITE(6,1007) | 00001250 |
| C | PRINT 1007 | 00001260 |
| | RETURN | 00001270 |
| | 90 WRITE(6,1005)FO | 00001280 |
| C | 90 PRINT 1005,FO | 00001290 |
| | RETURN | 00001300 |
| | 95 IF(NI.EQ.1.OR.IR.EQ.10.OR.ICONS.NE.NI)GO TO 100 | 00001310 |
| | ICONS=ICONS+1 | 00001320 |
| | IR=IR+1 | 00001330 |
| | DO 96 J= 1,N | 00001340 |
| | DYDT(J)= DYDTN(J) | 00001350 |
| | X(J)= XN(J) | 00001360 |
| 96 | Y(J)= YNOM(J) | 00001370 |
| | Y(M)= YNOM(M) | 00001380 |
| | TF= TFN | 00001390 |
| | DYDT(M)= DYDTN(M) | 00001400 |
| | ISW=0 | 00001410 |
| C | PRINT 1004 | 00001420 |
| | WRITE(6,1004) | 00001430 |
| | GO TO 10 | 00001440 |
| | 100 NI=9999 | 00001450 |
| | WRITE(6,1009) | 00001460 |
| C | PRINT 1009 | 00001470 |
| | RETURN | 00001480 |
| | 1000 FORMAT(/1X,1PD23.15) | 00001490 |
| | 1001 FORMAT(1X,1P5D23.15) | 00001500 |
| | 1002 FORMAT(21'OPARTIAL DERIV MATRIX) | 00001510 |
| | 1003 FORMAT(11HMODEL X S ARE/(1X,1PD23.15)) | 00001520 |
| | 1004 FORMAT(35HOFORM NEW PARTIAL DERIVATIVE MATRIX) | 00001530 |
| | 1005 FORMAT(4HOF0=,1PD22.15,23HCASE CONVERGED...FERTIG) | 00001540 |
| | 1006 FORMAT(38H0EXCEEDED MAXIMUM NUMBER OF ITERATIONS) | 00001550 |
| | 1007 FORMAT(16H0M=TRIX SINGULAR) | 00001560 |
| | 1008 FORMAT(11H0ELX S ARE) | 00001570 |
| | 1009 FORMAT(19H0METHOD CANNOT WORK) | 00001580 |
| | 1010 FORMAT(4HOF1=,1PD23.15) | 00001590 |
| | 1011 FORMAT(4HOF0=,1PD23.15) | 00001600 |
| | 1012 FORMAT (10H0 DEL TF =,1PD23.15) | 00001610 |
| | 1013 FORMAT (40H0 X(I)+DX(I) FOLLOWED BY CORRESPONDING Y) | 00001620 |
| | 1014 FORMAT (24H0 X(I)=0. SO DX(I)=XS(I)) | 00001630 |
| | 1015 FORMAT (15H0 DETERMINANT =,1PD23.15) | 00001640 |
| | END | 00001650 |

| | | |
|------|---|----------|
| C | PRTN/PRTNN | 00000010 |
| C | | 00000020 |
| C | THIS PROGRAM IS CALLED BY THE ITERATOR AND PRINTS | 00000030 |
| C | USED WITH 6 DIM. Y. | 00000035 |
| C | | 00000040 |
| C | | 00000050 |
| C | SUBROUTINE PRTN(KOUNT,NOI) | 00000060 |
| C | IMPLICIT REAL*8(A-M,O-S) | 00000070 |
| | COMMON /XMMM/X(5),XS(5),Y(6) | 00000080 |
| C | COMMON /T/TF,S,TO,TFMIN | 00000085 |
| | N=5 | 00000100 |
| | M=6 | 00000104 |
| | WRITE (6,1000) | 00000108 |
| | WRITE (6,1001) NOI,KOINT | 00000110 |
| | WRITE (6,1002) | 00000120 |
| | WRITE (6,1003) (X(J),J=1,N) | 00000130 |
| | WRITE (6,1004) | 00000140 |
| | WRITE (6,1003) (Y(J),J=1,M) | 00000150 |
| | WRITE (6,1005) TF | 00000160 |
| 1000 | FORMAT (24H0 ITER NO. TRAJ. CALLS) | 00000165 |
| 1001 | FORMAT (1H0,16,5X,16) | 00000170 |
| 1002 | FORMAT (2HOX) | 00000180 |
| 1003 | FORMAT (1X,1P5023.15) | 00000190 |
| 1004 | FORMAT (2HOY) | 00000200 |
| 1005 | FORMAT (5HO TF=,1PD26.16) | 00000210 |
| | RETURN | 00000215 |
| | END | 00000220 |
| | | 00000230 |

| | |
|--|----------|
| C DCR0UT/DCR0UT3 6 DIM. | |
| SUBROUTINE DCR0UT(AA,R,I,EPS,NI,M,IND) | 0000000 |
| DOUBLE PRECISION A,K,I,EPS,T,S,P,DT,AA | 0000000 |
| DIMENSION A(6,4),R(6,1),AA(6,6) | 0000000 |
| 5 IND=0 | 36110050 |
| N=N | 36110060 |
| DO 6 I=1,N | 0000000 |
| DO 6 J=1,N | 0000000 |
| 6 A(I,J)=AA(I,J) | 0000000 |
| IF(M)10,25,25 | 36110080 |
| 10 M=N | 36110090 |
| DO 20 I=1,N | 36110100 |
| DO 15 J=1,N | 36110110 |
| 15 R(I,J)=0.DO | 0000000 |
| 20 R(I,I)=1.DO | 0000000 |
| 25 IC=0 | 36110140 |
| II=0 | 36110150 |
| T=DABS(A(I,I)) | 36110160 |
| DO 35 I=2,N | 36110170 |
| IF(T-DABS(A(I,I)))30,35,35 | 36110180 |
| 30 II=I | 36110190 |
| T=DABS(A(I,II)) | 36110020 |
| 35 CONTINUE | 36110210 |
| IF(II)40,65,40 | 36110220 |
| 40 IC=IC+1 | 36110230 |
| IF(M)45,55,45 | 36110240 |
| 45 DO 50 J=1,M | 36110250 |
| S=R(I,J) | 36110260 |
| R(I,J)=R(II,J) | 36110270 |
| 50 R(II,J)=S | 36110280 |
| 55 DO 60 J=1,N | 36110290 |
| S=A(I,J) | 36110300 |
| A(I,J)=A(II,J) | 36110310 |
| 60 A(II,J)=S | 36110320 |
| 65 P=A(I,I) | 36110330 |
| IF(DABS(P)-EPS)70,70,75 | 36110340 |
| 70 IND=1 | 36110350 |
| D=0.DO | 0000000 |
| GO TO 200 | 36110370 |
| 75 DO 80 J=2,N | 36110380 |
| 80 A(I,J)=A(I,J)/P | 36110390 |
| IF(M)85,95,85 | 36110400 |
| 85 DO 90 J=1,M | 36110410 |
| 90 R(I,J)=R(I,J)/P | 36110420 |
| 95 DO 170 K=2,N | 36110430 |
| KM=K-1 | 36110440 |
| T=-1.DO | 0000000 |
| DO 105 I=K,N | 36110460 |
| DO 98 J=1,KM | 36110470 |
| 98 A(I,K)=A(I,K)-A(I,J)*A(J,K) | 36110480 |
| IF(T-DABS(A(I,K)))100,105,105 | 36110490 |
| 100 T=DABS(A(I,K)) | 36110500 |
| II=I | 36110510 |
| 105 CONTINUE | 36110520 |
| IF(II-K)110,135,110 | 36110530 |
| 110 IC=IC+1 | 36110540 |
| IF(M)115,125,115 | 36110550 |

| | | |
|-----|--------------------------------|----------|
| 115 | DO 120 J=1,M | 36110560 |
| | S=R(K,J) | 36110570 |
| | R(K,J)=R(II,J) | 36110580 |
| 120 | R(II,J)=S | 36110590 |
| 125 | DO 130 J=1,N | 36110600 |
| | S=A(K,J) | 36110610 |
| | A(K,J)=A(II,J) | 36110620 |
| 130 | A(II,J)=S | 36110630 |
| 135 | DT=A(K,K) | 36110640 |
| | IF(DABS(DT)-EPS)70,70,140 | 36110650 |
| 140 | P=P*DT | 36110660 |
| | IF(K-N)145,155,145 | 36110670 |
| 145 | KP=K+1 | 36110680 |
| | DO 150 J=KP,N | 36110690 |
| | DO 148 I=1,KM | 36110700 |
| 148 | A(K,J)=A(K,J)-A(K,I)*A(I,J) | 36110710 |
| 150 | A(K,J)=A(K,J)/DT | 36110720 |
| 155 | IF(M)160,170,160 | 36110730 |
| 160 | DO 165 J=1,M | 36110740 |
| | DO 162 I=1,KM | 36110750 |
| 162 | R(K,J)=R(K,J)-A(K,I)*R(I,J) | 36110760 |
| 165 | R(K,J)=R(K,J)/DT | 36110770 |
| 170 | CONTINUE | 36110780 |
| | IF(MOD(IC,2))175,180,175 | 36110790 |
| 175 | P=-P | 36110800 |
| 180 | D=P | 36110810 |
| | IF(M)185,200,185 | 36110820 |
| 185 | II=N | 36110830 |
| | DO 190 K=2,N | 36110840 |
| | KP=II | 36110850 |
| | II=II-1 | 36110860 |
| | DO 190 J=1,M | 36110870 |
| | DO 190 I=KP,N | 36110880 |
| 190 | R(II,J)=R(II,J)-A(II,I)*R(I,J) | 36110890 |
| 200 | RETURN | 36110900 |
| | END | 36110910 |


```

C TRAJ/TRAJRNK                                00000010
C                                                00000020
C NFP AND HIGH THRUST.                        00000030
C THIS ROUTINE SETS UP THE INPUT TO THE INTEGRATOR WHICH 00000040
C   EXTRAPOLATES THE TRAJECTORY FROM INITIAL TIME TO    00000050
C   FINAL TIME. IT ALSO EVALUATES THE CHANGE IN TF AND  00000060
C   THE ERROR IN THE FINAL CONDITIONS.                  00000070
C THIS PROGRAM IS CALLED BY ITER OR BY CONTROL          00000080
C MIN J, MAX H.                                       00000090
C 6 DIM. ZERF, T.C. OPTIONS.                       00000100
C NOP=1--ALL 5 FINAL O.E. FIXED, =2--A,E,I ONLY FIXED. 00000110
C DYDT USED ONLY IF ITR= 0                          00000120
C IHI= 1 NO HIGH THRUST                              00000130
C   = 2 2 INITIAL IMPULSES                            00000140
C   = 3 2 INITIAL AND ONE FINAL IMPULSE               00000150
C   = 4 FINAL IMPULSE                                  00000160
C WHEN IHI=2 OR 3, ZLO(5) AND ZERF(5) ARE DUMMIES.    00000170
C                                                      00000180
C                                                      00000190
C SUBROUTINE TRAJ                                     00000200
C IMPLICIT REAL*8(A-H,O-S), INTEGER (I-N)            00000210
C                                                      00000220
C COMMON /XMMH/ZLO(5), DUMMY(5), ZERF(6)             00000230
C COMMON /TRA/TFMAX, DTO, UEB, EW(10)                 00000240
C COMMON /Z/Z(10), DERZ(10)                           00000250
C COMMON /INT/ITF, IPR, IDIM, IDIM2, NIMAX            00000260
C COMMON /T/TF, SD, TO, TFMIN                        00000270
C COMMON /ELEM/ZPO(5), ZPF(5)                         00000280
C COMMON /DY/DYDT(6)                                  00000290
C COMMON /TC/NOP                                       00000300
C COMMON /POWER/PKW,CC,AMO,A0,A1,A2,A3,IPOW          00000310
C COMMON /A/A,AMU,PI                                    00000320
C COMMON /HIGH/DV11,DV12,IHI,ITR                     00000330
C COMMON /F/FLIM,KSTEP                                00000340
C                                                      00000350
C EXTERNAL FUNCT, OUTP                                00000360
C DIMENSION PRMT(5), AUX(R,10), DP(5), ZS(6)         00000370
C                                                      00000380
C IF ((IHI.EQ.1).OR.(IHI.FQ.4)) GO TO 9              00000390
C IF (ITR.NE.1) GO TO 4                               00000400
C DO 3 I=1,6                                          00000410
C   ZS(I)= ZERF(I)                                    00000420
C   IF (ITR.EQ.4) GO TO 70                            00000430
C   IF (ITR.EQ.5) GO TO *0                            00000440
C                                                      00000450
C HIGH THRUST                                         00000460
C                                                      00000470
C UP= (PI/2.DO)*ZLO(1)/DSQRT(1.DO+ZLO(1)**2)         00000480
C CUP= DCOS(UP)                                       00000490
C XK= CUP*(.75DO+.25DO*ZLO(2)/DSQRT(1.DO+ZLO(2)**2)) 00000500
C DUM= (1.DO+CUP*XK)*ISQRT((CUP-XK)/(CUP+XK))         00000510
C XJ= DUM*ZLO(3)/DSQRT(1.DO+ZLO(3)**2)               00000520
C JM= 2                                               00000530
C CALL MAINE(0.DO,0.DO,XK,UP,XJ,1.DO,1,JM,DP,DV11)    00000540
C CALL OUTHI(JM,PI,ZPO(1),ZPO(2),0.DO,IPR,Z,IDIM2)    00000550
C DO 5 I= 1,IDIM2                                    00000560
C                                                      00000570

```

```

5      Z(I+IDIM2)= ZL0(4)*Z(I+IDIM2)*1.04      00000580
C
C NEP      00000590
C          00000600
C          00000610
9      PRMT(1)= T0      00000620
      PRMT(2)= TF      00000630
      PRMT(3)= DTO      00000640
      PRMT(4)= UEB      00000650
C          00000660
C Z IS A VECTOR OF STATE AND COSTATE      00000670
C          00000680
      IF ((IHL.EQ.2).OR.(IHL.EQ.3)) GO TO 15      00000690
      DO 10 I=1, IDIM2      00000700
      Z(I)=ZP0(I)      00000710
10     Z(I+IDIM2)= ZL0(I)      00000720
C          00000730
C EW ARE ERROR WEIGHTS--INPUT TO THE INTEGRATOR      00000740
C          00000750
15     DO 20 I=1, IDIM      00000760
20     DERZ(I)=EW(I)      00000770
C          00000780
C FOR ITF LT 3 USE NORM CUTOFF CONDITION      00000790
C          00000800
      IF (ITF.LT.3) TF=TFMAX      00000810
C          00000820
C CALL THE R-K INTEGRATOR      00000830
C          00000840
      CALL DRKGS(PRMT,7.0E+7, IDIM, IHLF, FUNCT, OUTP, AUX)      00000850
      IF (IHLF.GT.10) GO TO 100      00000860
C          00000870
C Z IS NOW THE FINAL O.E..      00000880
C ZERF THE ERROR IN THE FINAL CONDITIONS      00000890
C          00000900
      H=0.00      00000910
      DO 30 I=1, IDIM2      00000920
30     H= H + Z(I+5)*DERZ(I)      00000930
      ZERF(6)= H -1.00      00000940
      IF (I=1.LT.3) GO TO 35      00000950
C          00000960
C FINAL IMPULSE      00000970
C          00000980
      PSI= (Z(3)*Z(7)-Z(2)*Z(8))*1.0-3      00000990
      PSIDOT= DERZ(3)*Z(7)+Z(3)*DERZ(7)-Z(2)*DERZ(8)-DERZ(2)*Z(8)      00001000
      PSIDOT= PSIDOT*1.0-3      00001010
      CALL IMPLS(DV12, IPR, ITR, Z, DERZ, IDIM2)      00001020
C          00001030
C FINAL CONDITION OPTION BRANCH      00001040
C          00001050
35     GO TO (40,50), NOP      00001060
C          00001070
40     DO 45 I=1, IDIM2      00001080
      ZERF(I)= Z(I) -ZPF(I)      00001090
45     DYDT(I)= DERZ(I)      00001100
      GO TO 55      00001110
C          00001120
50     ZERF(1)= Z(1) - ZPF(1)      00001130
      DUM1= DSQRT(Z(2)**2 + Z(3)**2)      00001140

```

| | | |
|-----|---|----------|
| | ZERF(2)= DUM1 - ZPF(2) | 00001150 |
| | DUM2= DSQRT(Z(4)**2 + Z(5)**2) | 00001160 |
| | ZERF(3)= DUM2 - ZPF(3) | 00001170 |
| | ZERF(4)= (Z(3)*Z(7) - Z(2)*Z(8))*1.D-3 | 00001180 |
| | ZERF(5)= (Z(5)*Z(9) - Z(4)*Z(10))*1.D-3 | 00001190 |
| | DYDT(1)= DERZ(1) | 00001200 |
| | DYDT(2)= 0.D0 | 00001210 |
| | DYDT(3)= 0.D0 | 00001220 |
| | IF (DUM1.GT.1.D-40) DYDT(2)= (Z(2)*DERZ(2) + Z(3)*DERZ(3))/DUM1 | 00001230 |
| | IF (DUM2.GT.1.D-40) DYDT(3)= (Z(4)*DERZ(4) + Z(5)*DERZ(5))/DUM2 | 00001240 |
| | DYDT(4)= DERZ(3)*Z(7)+Z(3)*DERZ(7)-DERZ(2)*Z(8)-Z(2)*DERZ(8) | 00001250 |
| | DYDT(4)=DYDT(4)*1.D-3 | 00001260 |
| | DYDT(5)= DERZ(5)*Z(9)+Z(5)*DERZ(9)-DERZ(4)*Z(10)-Z(4)*DERZ(10) | 00001270 |
| | DYDT(5)=DYDT(5)*1.D-3 | 00001280 |
| | IF (IMI.EQ.1) GO TO 55 | 00001290 |
| | IF (IMI.EQ.4) GO TO 52 | 00001300 |
| C | | 00001310 |
| C | DUM1Y | 00001320 |
| C | | 00001330 |
| | ZERF(5)= ZLO(5)-1.D0 | 00001340 |
| | DYDT(5)= 0.D0 | 00001350 |
| | IF (IMI.LT.3) GO TO 53 | 00001360 |
| C | | 00001370 |
| 52 | ZERF(4)= PSI | 00001380 |
| | DYDT(4)= PSIDOT | 00001390 |
| 53 | IF (ITR.GT.0) RETURN | 00001400 |
| C | | 00001410 |
| | DO 54 I= 1,6 | 00001420 |
| 54 | ZS(I)= ZERF(I) | 00001430 |
| C | | 00001440 |
| 55 | IF (IPDW.EQ.) GO TO 60 | 00001450 |
| | A4= 1.D0+TF | 00001460 |
| | DYDT(6)= A1*H/A4 | 00001470 |
| | RETURN | 00001480 |
| C | | 00001490 |
| 60 | B1= DEXP(-A2*TF) | 00001500 |
| | A4= 1.D0+A1*(B1-1.D0+A2*A3*TF) | 00001510 |
| | DYDT(6)= A2*H*(-B1/(B1-A3)+A1*(B1-A3)/A4) | 00001520 |
| | RETURN | 00001530 |
| C | | 00001540 |
| C | | 00001550 |
| C | | 00001560 |
| 70 | DO 75 I= 1,3 | 00001570 |
| 75 | ZERF(I)= ZS(I) | 00001580 |
| | DUM= DUMMY(4) | 00001590 |
| | IF (K.EP.EQ.0) DUM= DUM*ZLO(4) | 00001600 |
| | ZERF(4)= (1.D0+DUM)*ZS(4) | 00001610 |
| | ZERF(5)= ZS(5) | 00001620 |
| | ZERF(6)= (1.D0+DUM)*(ZS(6)+1.D0)-1.D0 | 00001630 |
| | RETURN | 00001640 |
| C | | 00001650 |
| | DO 85 I= 1,6 | 00001660 |
| 85 | ZERF(I)= ZS(I) | 00001670 |
| | ZERF(5)= ZS(5)+DUMMY(5) | 00001680 |
| | RETURN | 00001690 |
| C | | 00001700 |
| 100 | IF (IMLF.EQ.11) WRITE (6,1000) | 00001710 |

| | |
|--|----------|
| IF (IHLF.EQ.12) WRITE (6,1001) | 00001720 |
| IF (IHLF.EQ.13) WRITE (6,1002) | 00001730 |
| STOP | 00001740 |
| C | 00001750 |
| 1000 FORMAT (68H0 THE NUMBER OF BISECTIONS OF THE ORIGINAL INCREMENT HAS | 00001760 |
| 1S EXCEEDED 10) | 00001770 |
| 1001 FORMAT (27H0 INITIAL INCREMENT IS ZERO) | 00001780 |
| 1002 FORMAT (54H0 INITIAL INCREMENT HAS WRONG SIGN OR BOUNDS ARE WRONG) | 00001790 |
| END | 00001800 |

```

C OUTP/OUTPN                                00000010
C                                             00000020
C NEP                                        00000030
C THIS IS THE OUTP PROGRAM FOR THE          00000040
C INTEGRATOR--FIXED GUESS TIME ONLY (ITF=3) 00000050
C                                             00000060
C                                             00000070
C                                             00000080
C                                             00000090
C SUBROUTINE OUTP(I,Z,DERZ,IHLF,IDIM,PRMT)  00000100
C                                             00000110
C IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)   00000120
C COMMON /UNITS/UTS,UTM,UTH,UTI,UTKM,DTR,DUMMY,UTKW,UTMS2 00000130
C COMMON /INT/ITF,IPR,ID,IDIM2,NIMAX        00000140
C COMMON /POWER/PKW,CC,AMO,A0,A1,A2,A3,IPOW 00000150
C COMMON /A/A,AMU,PI                        00000160
C                                             00000170
C DIMENSION PRMT(5), Z(10), DERZ(10), W(5) 00000180
C                                             00000190
C IF (ITF.NE.3) GO TO 10                     00000200
C IF (IPR.EQ.0) RETURN                       00000210
C IF (T.EQ.PRMT(1)) N=0                     00000220
C IF (T.EQ.PRMT(1)) M=0                     00000230
C N=N+1                                       00000240
C IF ((T.LT.(.9999999999D0*DFLOAT(M))*(PRMT(2)-PRMT(1))/DFLOAT(IPR)))
1 .AND. (IHLF.LI.11).AND.(T.LT.(.9999999999D0*PRMT(2)))) RETURN 00000250
C M=M+1                                       00000260
C TS=UTS*T                                   00000270
C TM=UTM*T                                   00000280
C TH=UTH*T                                   00000290
C TD=UTD*T                                   00000300
C IF (IPOW.EQ.1) B1= DEXP(-A2*T)             00000310
C IF (IPOW.EQ.1) RMASS= 1.D0+A1*(B1-1.D0+A3*A2*T) 00000320
C IF (IPOW.EQ.0) RMASS= 1.D0-A1*T           00000330
C DV= -CC*DLOG(RMASS)                       00000340
C AMASS= AMO*RMASS                           00000350
C POW= PKW                                    00000360
C IF (IPOW.EQ.1) POW= (B1-A3)*POW           00000370
C THR= 2.D0*POW/CC                           00000380
C AA= THR/AMASS                              00000390
C H= 0.D0                                    00000400
C DO 5 I=1,5                                 00000410
5 H= H + Z(I+IDIM2)*DERZ(I)                 00000420
C W(1)= Z(1)*UTKM                           00000430
C W(2)=0.D0                                  00000440
C DUMMY=Z(2)**2+Z(3)**2                     00000450
C IF (DUMMY.GT.1.D-40) W(2)=DSORT(DUMMY)    00000460
C W(3)=0.D0                                  00000470
C DUMMY=Z(4)**2 + Z(5)**2                   00000480
C IF (DUMMY.GT.1.D-40) W(3)= 2.D0*DATAN(DSORT(DUMMY))/DTR 00000490
C W(4)=0.D0                                  00000500
C IF ((Z(4).NE.0.D0).OR.(Z(5).NE.0.D0)) W(4)= DATAN2(Z(4),Z(5))/DTR 00000510
C W(5)= 0.D0                                 00000520
C IF ((Z(2).NE.0.D0).OR.(Z(3).NE.0.D0)) W(5)= DATAN2(Z(2),Z(3))/DTR 00000530
C W(5)=W(5)-W(4)                            00000540
C IDIM3=IDIM2+1                             00000550
C                                             00000560
C                                             00000570

```

| | |
|--|----------|
| WRITE (6,1001) | 00000580 |
| WRITE (6,1002) | 00000590 |
| WRITE (6,1003) I. IS. TH. TD. DV. N | 00000600 |
| WRITE (6,1004) | 00000610 |
| WRITE (6,1005) (Z(I),I=1,IDIM2) | 00000620 |
| WRITE (6,1005) W | 00000630 |
| WRITE (6, 1013) | 00000640 |
| WRITE (6,1012) AMASS.POW,THR,AA | 00000650 |
| WRITE (6,1006) | 00000660 |
| WRITE (6,1005) (Z(I),I=1,IDIM3,IDIM) | 00000670 |
| WRITE (6,1007) | 00000680 |
| WRITE (6,1005) (DERZ(I),I=1,IDIM2) | 00000690 |
| WRITE (6,1008) | 00000700 |
| WRITE (6,1005) (DERZ(I),I=1,IDIM3,IDIM) | 00000710 |
| WRITE (6,1009) | 00000720 |
| PER=2.00*PI*DSORT(Z(I)**3/AMU)*UTH | 00000730 |
| AP= W(1)*(1.00+W(2)) | 00000740 |
| PE= W(1)*(1.00-W(2)) | 00000750 |
| WRITE (6,1010) H. PER,PE,AP,IMLF | 00000760 |
| C | 00000770 |
| RETURN | 00000780 |
| 10 WRITE (6,1000) | 00000790 |
| STOP | 00000800 |
| C | 00000810 |
| 1000 FORMAT (56H0 ITF MUST EQUAL 3--I.E. NEED FIXED ESTIMATED FINAL TIME | 00000820 |
| 1E) | 00000830 |
| 1001 FORMAT (70H0 *****00000840 | |
| 1*****) | 00000850 |
| 1002 FORMAT (5H TIME,10X,10HTIME UNITS,15X,7HSECONDS,9X | 00000860 |
| 1 ,5HHOURS,11X,4HDAYS,9X,8HIV (K/S),10X,1HN) | 00000870 |
| 1003 FORMAT (1P2025.7,1P3015.7,19//) | 00000880 |
| 1004 FORMAT (60H0 THE ORBITAL ELEMENTS ARE (EQ. FOLLOWED BY CL. IN KM | 00000890 |
| 1 DEG)) | 00000900 |
| 1005 FORMAT (1P5022.12//) | 00000910 |
| 1006 FORMAT (16H0 THE COSTATE IS) | 00000920 |
| 1007 FORMAT (45H0 THE DERIVATIVE OF THE ORBITAL ELEMENTS ARE) | 00000930 |
| 1008 FORMAT (34H0 THE DERIVATIVE OF THE COSTATE IS) | 00000940 |
| 1009 FORMAT (1H0,7X,11HHA ILTON: AN,8X,12HPERIOD (HRS),7X, | 00000950 |
| 1 12HPERIGEE (KM),9X,11HAPUGEE (KM),5X,14HDIV. TIME STEP) | 00000960 |
| 1010 FORMAT (2F20.12,1P2020.10,19//) | 00000970 |
| 1012 FORMAT (1P4022.12) | 00000980 |
| 1013 FORMAT (9X,9HMASS (KG),12X,10HPOWER (KW),12X,10HTHRUST (N), | 00000990 |
| 1 8X,19HTHRUST ACC (M/S**2)) | 00001000 |
| END | 00001010 |

| | |
|--|----------|
| C FUNCT/FUNCTN | 00000010 |
| C | 00000020 |
| C NEP | 00000030 |
| C THIS SUBROUTINE IS AN INTERFACE BETWEEN THE INTEGRATOR ROUTINE | 00000040 |
| C AND THE QUADRATURE ROUTINE. | 00000050 |
| C THIS ROUTINE ADDS THE EFFECT OF ORLATENESS (J2) TO THE DERIV. | 00000060 |
| C ORLATE CALCULATES THE EFFECT OF J2, RETURNED AS DZJ2. | 00000070 |
| C Z IS A VECTOR OF THE AVERAGED STATE AND COSTATE | 00000080 |
| C DERZ IS THE AVERAGED DERIVATIVE OF Z | 00000090 |
| C -PI AND PI AND THE LIMITS OF INTEGRATION IN THE QUADRATURE | 00000100 |
| C IO IS THE DIMENSION OF Z | 00000110 |
| C | 00000120 |
| C | 00000130 |
| C | 00000140 |
| SUBROUTINE FUNCT(X,Z,DERZ) | 00000150 |
| C | 00000160 |
| IMPLICIT REAL*(A-H,O-S) | 00000170 |
| COMMON /A/A,AMU,PI | 00000180 |
| COMMON /J2/AJ2 | 00000190 |
| COMMON /POWER/PKW,CC,AMO,AO,A1,A2,A3,IPOW | 00000200 |
| DIMENSION Z(10), DERZ(10), G(10), H(10), DZJ2(10) | 00000210 |
| EXTERNAL FCT | 00000220 |
| R1= DEXP(-A2*X) | 00000230 |
| C | 00000240 |
| IF (IPOW.EQ.0) A= AO/(1.00-A1*X) | 00000250 |
| IF ((IPOW.EQ.1).AND.(A3.GE.B1)) GO TO 20 | 00000260 |
| IF (IPOW.EQ.1) A= AO*(R1-A3)/(1.00+A1*(R1-1.00+A3*A2*X)) | 00000270 |
| DO 2 I=1,10 | 00000280 |
| 2 DERZ(I)= 0.00 | 00000290 |
| C | 00000300 |
| CALL QUAD(-PI,PI,FCT,DERZ,Z,G,H,10) | 00000310 |
| IF (AJ2.LE.0.00) RETURN | 00000320 |
| 5 CALL ORLATE(AJ2,Z,DZJ2,1) | 00000330 |
| DO 10 I=1,10 | 00000340 |
| 10 DERZ(I)= DERZ(I) + DZJ2(I) | 00000350 |
| RETURN | 00000360 |
| 20 WRITE (6,1000) | 00000370 |
| STOP | 00000380 |
| 1000 FORMAT (' POWER IS NEGATIVE') | 00000390 |
| END | 00000400 |

| | | |
|------|--|----------|
| C | CONTL | 00000010 |
| C | | 00000020 |
| C | SEP OR NEP, HIGH THRUST MAY BE INCLUDED. | 00000030 |
| C | THIS IS THE MAIN CONTROLLING PROGRAM FOR FINDING THE | 00000040 |
| C | OPTIMAL TRAJECTORY FOR SATELLITE RAISING | 00000050 |
| C | USING ELECTRIC PROPULSION AND HIGH THRUST. | 00000060 |
| C | | 00000070 |
| C | | 00000080 |
| C | | 00000090 |
| C | | 00000100 |
| C | IMPLICIT REAL*(A-H,O-S), INTEGER (I-N) | 00000110 |
| C | COMMON /INT/ITF, IPR, IDIM, IDIM2, NIMAX | 00000120 |
| C | EXTERNAL TRAJ, PRIN | 00000130 |
| C | | 00000140 |
| C | | 00000150 |
| C | | 00000160 |
| C | ALL SETTING OF CONSTANTS AND INITIAL READ AND WRITES | 00000170 |
| C | ARE DONE BY THE SUBROUTINE INPUT | 00000180 |
| C | | 00000190 |
| C | CALL INPUT | 00000200 |
| C | IF (NIMAX.EQ.0) GO TO 10 | 00000210 |
| C | WRITE (6,1001) | 00000220 |
| C | | 00000230 |
| C | THE ITERATOR ROUTINE SOLVES THE 2PBVP FOR THE OPTIMAL TRAJECTORY | 00000240 |
| C | | 00000250 |
| C | CALL ITER(KOUNT, NI, TRAJ, PRIN) | 00000260 |
| C | IF (NI.EQ.9999) WRITE (6,1002) | 00000270 |
| C | WRITE (6,1003) | 00000280 |
| C | | 00000290 |
| C | OUTPC PRINTS A SUMMARY OF THE OPTIMAL TRAJECTORY VALUES | 00000300 |
| C | | 00000310 |
| C | CALL OUTPC(KOUNT) | 00000320 |
| C | | 00000330 |
| C | TIME HISTORY OF THE OPTIMAL TRAJECTORY IS CALCULATED AND PRINTED | 00000340 |
| C | | 00000350 |
| 10 | WRITE (6,1004) | 00000360 |
| | IPR=1000000 | 00000370 |
| | ITF=3 | 00000380 |
| | CALL TRAJ | 00000390 |
| C | | 00000400 |
| | IF (NI.EQ.9999) WRITE (6,1002) | 00000410 |
| | IF (NI.NE.9999) WRITE (6,1005) | 00000420 |
| | STOP | 00000430 |
| 1001 | FORMAT (18H) ITERATION BEGINS) | 00000440 |
| 1002 | FORMAT (29HC OPTIMIZATION NOT SUCCESSFUL) | 00000450 |
| 1003 | FORMAT (43H) CONVERGED VALUES FOR OPTIMIZED TRAJECTORY) | 00000460 |
| 1004 | FORMAT (36H) TIME HISTORY OF OPTIMAL TRAJECTORY) | 00000470 |
| 1005 | FORMAT (30HO PROGRAM HAS RUN SUCCESSFULLY) | 00000480 |
| | END | 00000490 |

| | |
|--|----------|
| C ORLATE | 00000010 |
| C | 00000020 |
| C | 00000030 |
| C THIS SUBPROGRAM EVALUATES THE AVERAGED DERIVATIVE OF THE | 00000040 |
| C STATE AND CSTATE DUE TO THE ORLATENESS. J2 TERM | 00000050 |
| C ASSUMES EARTH EQUATORIAL RADIUS=1, MU=1. IF NOT C1 MUST BE | 00000052 |
| C MODIFIED. | 00000054 |
| C IF IJ=2, EVALUATE DZ(I), I=1,5 ONLY. | 00000056 |
| C | 00000060 |
| C | 00000070 |
| C SUBROUTINE ORLATE(AJ2,Z,DZJ2,IJ) | 00000080 |
| C | 00000090 |
| C IMPLICIT REAL*(A-H,O-S) | 00000100 |
| C DIMENSION Z(10),DZJ2(10),PJ(4,5) | 00000120 |
| C | 00000130 |
| C | 00000140 |
| C | 00000150 |
| C C1= 1.500*AJ2/Z(1)**3.5 | 00000160 |
| C B1=Z(4)**2+Z(5)**2 | 00000170 |
| C B7= 1.00-6.00*B1 + 3.00*B1**2 | 00000180 |
| C D2= 1.00 - Z(2)**2 -Z(3)**2 | 00000190 |
| C B2= 1.00/D2**2 | 00000200 |
| C B6= 1.00/(1.00+B1) | 00000210 |
| C C2= B2*B6 | 00000220 |
| C B4= 1.00 - B1 | 00000230 |
| C C3= C1*C2 | 00000240 |
| C D16= B6*B7 | 00000250 |
| C D3= C3*D16 | 00000260 |
| C | 00000270 |
| C DZJ2(1)= 0.00 | 00000280 |
| C DZJ2(2)= Z(3)*D3 | 00000290 |
| C DZJ2(3)= -Z(2)*D3 | 00000300 |
| C | 00000310 |
| C D4=B4*C3 | 00000320 |
| C | 00000330 |
| C DZJ2(4)= -Z(5)*D4 | 00000340 |
| C DZJ2(5)= Z(4)*D4 | 00000350 |
| C | 00000360 |
| C IF (IJ.EQ.2) RETURN | 00000370 |
| C | 00000380 |
| C D5= -3.500*C1/Z(1) | 00000390 |
| C B8= C2*D16*D5 | 00000400 |
| C | 00000410 |
| C PJ(1,1)=Z(3)*B8 | 00000420 |
| C PJ(2,1)= -Z(2)*B8 | 00000430 |
| C | 00000440 |
| C B12= C2*B4*D5 | 00000450 |
| C | 00000460 |
| C PJ(3,1)= -Z(5)*B12 | 00000470 |
| C PJ(4,1)= Z(4)*B12 | 00000480 |
| C | 00000490 |
| C D2= .2500*D2**3 | 00000500 |
| C D7= B6*Z(2)/D2 | 00000510 |
| C B9= C1*D16 | 00000520 |
| C | 00000530 |
| C PJ(1,2)= Z(3)*B9*D7 | 00000540 |
| C PJ(2,2)= -B9*(Z(2)*D7 + C2) | 00000550 |

| | | |
|----|--|----------|
| C | | 00000560 |
| | R13= C1*D7*B4 | 00000570 |
| C | | 00000580 |
| | PJ(3,2)= -Z(5)*R13 | 00000590 |
| | PJ(4,2)= Z(4)*R13 | 00000600 |
| C | | 00000610 |
| | D9= B6*Z(3)/D2 | 00000620 |
| C | | 00000630 |
| | PJ(1,3)= B9*(Z(3)*D9 + C2) | 00000640 |
| | PJ(2,3)= -Z(2)*B9*D9 | 00000650 |
| C | | 00000660 |
| | B14= C1*B4*D9 | 00000670 |
| C | | 00000680 |
| | PJ(3,3)= -Z(5)*B14 | 00000690 |
| | PJ(4,3)= Z(4)*B14 | 00000700 |
| C | | 00000710 |
| | D10= -2.00*B2*B6**2 | 00000720 |
| | D11=Z(4)*D10 | 00000730 |
| | D12= -12.00*B4 | 00000740 |
| | D13= C1*B6 | 00000750 |
| | D15=C2*D12 | 00000760 |
| | B10= D13*(2.00*B7*D11 + Z(4)*D15) | 00000770 |
| C | | 00000780 |
| | PJ(1,4)= Z(3)*B10 | 00000790 |
| | PJ(2,4)= -Z(2)*B10 | 00000800 |
| C | | 00000810 |
| | B15= B4*D11 - 2.00*Z(4)*C2 | 00000820 |
| C | | 00000830 |
| | PJ(3,4)= -Z(5)*C1*B15 | 00000840 |
| C | | 00000850 |
| | B16= C2*B4 | 00000860 |
| C | | 00000870 |
| | PJ(4,4)= C1*(B16 + Z(4)*B15) | 00000880 |
| C | | 00000890 |
| | D14= Z(5)*D10 | 00000900 |
| | D15= C2*D12 | 00000910 |
| | B17= D13*(2.00*B7*D14 + Z(5)*D15) | 00000920 |
| C | | 00000930 |
| | PJ(1,5)= Z(3)*B11 | 00000940 |
| | PJ(2,5)= -Z(2)*B11 | 00000950 |
| C | | 00000960 |
| | B17= D14*B4 - 2.00*Z(5)*C2 | 00000970 |
| C | | 00000980 |
| | PJ(3,5)= -C1*(B16 + Z(5)*B17) | 00000990 |
| | PJ(4,5)= Z(4)*C1*B17 | 00001000 |
| C | | 00001010 |
| | | 00001020 |
| C | | 00001030 |
| | DO 10 J=1,5 | 00001035 |
| | DZJ2(J+5)= 0.00 | 00001040 |
| | DC 10 I=1,4 | 00001050 |
| 10 | DZJ2(J+5)= DZ(2(J+5) - Z(I+6))*PJ(I,J) | 00001060 |
| | RETURN | 00001070 |
| | END | |

| | | |
|-----|--|----------|
| C | QUAD16 | 00000010 |
| C | | 00000020 |
| C | | 00000030 |
| C | THIS IS A MODIFIED QUADRATURE INTEGRATION PROGRAM FOR | 00000040 |
| C | VECTOR VALUED FUNCTIONS OF ONE VARIABLE. IT INTEGRATES | 00000050 |
| C | G (OR H) OVER X FROM XL TO XU. THE RESULT IS Y. | 00000060 |
| C | EVALUATION IS BY A 16 POINT GAUSS QUADRATURE FORMULA. | 00000070 |
| C | | 00000080 |
| C | SUBROUTINE QUAD(XL,XU,FCT,Y,Z,G,H,N) | 00000090 |
| C | | 00000100 |
| C | IMPLICIT REAL*(A-Z,D-S) | 00000110 |
| C | DIMENSION Y(1),Z(1),G(1),H(1) | 00000120 |
| C | | 00000130 |
| | A= .5D0*(XU+XL) | 00000140 |
| | B= XU-XL | 00000150 |
| | C= .49470046749582497D0*B | 00000160 |
| | K=1 | 00000170 |
| | GO TO 500 | 00000180 |
| 10 | DO 20 I=1,N | 00000190 |
| 20 | Y(I)= .13576229705877047D-1*G(I) | 00000200 |
| | C= .47228751155661629D0*B | 00000210 |
| | K=2 | 00000220 |
| | GO TO 500 | 00000230 |
| 30 | DO 40 I=1,N | 00000240 |
| 40 | Y(I)= Y(I) + .31126761969323946D-1*G(I) | 00000250 |
| | C= .43281560119391587D0*B | 00000260 |
| | K=3 | 00000270 |
| | GO TO 500 | 00000280 |
| 50 | DO 60 I=1,N | 00000290 |
| 60 | Y(I)= Y(I)+ .47579255841246392D-1*G(I) | 00000300 |
| | C= .3770220417750152D0*B | 00000310 |
| | K=4 | 00000320 |
| | GO TO 500 | 00000330 |
| 70 | DO 80 I=1,N | 00000340 |
| 80 | Y(I)= Y(I)+ .62314445627766936D-1*G(I) | 00000350 |
| | C= .30893812220132187D0*B | 00000360 |
| | K=5 | 00000370 |
| | GO TO 500 | 00000380 |
| 90 | DO 100 I=1,N | 00000390 |
| 100 | Y(I)= Y(I) + .747979944082837D-1*G(I) | 00000400 |
| | C= .22900838882867369D0*B | 00000410 |
| | K=6 | 00000420 |
| | GO TO 500 | 00000430 |
| 110 | DO 120 I=1,N | 00000440 |
| 120 | Y(I)= Y(I) + .8457825969750127D-1*G(I) | 00000450 |
| | C= .14080177538962946D0*B | 00000460 |
| | K=7 | 00000470 |
| | GO TO 500 | 00000480 |
| 130 | DO 140 I=1,N | 00000490 |
| 140 | Y(I)= Y(I) + .9130170752246179D-1*G(I) | 00000500 |
| | C= .4750625491871872D-1*B | 00000510 |
| | K=8 | 00000520 |
| | GO TO 500 | 00000530 |
| 150 | DO 160 I=1,N | 00000540 |
| 160 | Y(I)= B*(Y(I) + .9472530522751425D-1*G(I)) | 00000550 |
| | RETURN | 00000560 |
| 500 | CALL FCT(A-C,A+C,Z,H,G) | 00000570 |
| | DO 510 I=1,N | 00000580 |
| 510 | G(I)= G(I) + H(I) | 00000590 |
| | GO TO (10,30,50,70,90,110,130,150),K | 00000600 |
| | STOP | 00000605 |
| | END | 00000610 |

```

C FCT 00000010
C 00000020
C THIS SUBPROGRAM IS CALLED BY THE QUADRATURE PROGRAM AND 00000030
C EVALUATES THE INTEGRAND 00000040
C 00000050
C 00000060
C 00000070
C SUBROUTINE FCT(F1,F2,Z,H,G) 00000080
C IMPLICIT REAL*(A-H,O-S), INTEGER (I-N) 00000082
C 00000084
C COMMON /A/A,AMU,PI 00000090
C 00000100
C DIMENSION Z(10),G(10),H(10),AM(5,3),PAM(5,3,5),VEC(3),X(5),PRA(5) 00000110
C 00000120
C M=0 00000140
C F=F1 00000150
C 00000160
C 00000170
C EVALUATE M AND PARTIAL OF M WRT STATE 00000180
C 00000190
C DD 5 I=1,5 00000192
C 5 X(I)=Z(I) 00000194
C 10 CALL EVALMP(X,F,AMU,AM,PAM,I) 00000200
C 00000220
C EVALUATE THE COMMON SCALAR FACTOR 00000230
C 00000240
C CT=DCOS(F) 00000242
C ST=DSIN(F) 00000244
C RA= 1.000-Z(3)*CT -Z(2)*ST 00000250
C FAC= A*RA/(2.000*PI) 00000260
C 00000270
C EVALUATE THE UNIT OF M TRANSPOSE LAMBDA 00000280
C 00000290
C ABVEC= 0.00 00000300
C DD 30 I=1,3 00000310
C VEC(I)=0.000 00000320
C DD 20 J=1,5 00000330
C 20 VEC(I)= VEC(I) + AM(J,I)*Z(J+5) 00000340
C 30 ABVEC= ABVEC + VEC(I)**2 00000350
C ABVEC= DSQRT(ABVEC) 00000360
C DD 40 I=1,3 00000370
C 40 VEC(I)=VEC(I)/ABVEC 00000380
C 00000390
C EVALUATE THE PARTIAL OF RA WRT X 00000391
C 00000392
C PRA(1)=0.00 00000393
C PRA(2)=-ST 00000394
C PRA(3)=-CT 00000395
C PRA(4)=0.00 00000396
C PRA(5)=0.00 00000397
C ABVEC=ABVEC/RA 00000398
C 00000399
C EVALUATE THE FUNCTION 00000400
C 00000410
C DD 60 I=1,5 00000420
C G(I)= 0.000 00000430
C G(I+5)=0.000 00000440

```

| | | |
|----|---|----------|
| | DO 50 J=1,3 | 00000450 |
| | G(I)= G(I) + AM(I,J)*VFC(J) | 00000460 |
| | DO 50 L=1,5 | 00000470 |
| 50 | G(I+5)= G(I+5) - 2(L+5)*PAM(L,J,I)*VEC(J) | 00000480 |
| | G(I)= FAC*G(I) | 00000490 |
| 60 | G(I+5)=FAC*(G(I+5)-ARVFC*PRA(I)) | 00000500 |
| | IF (M.EQ.1) RETURN | 00000510 |
| | DO 70 I=1,10 | 00000520 |
| 70 | H(I)= G(I) | 00000530 |
| | F=F2 | 00000540 |
| | M=1 | 00000545 |
| | GO TO 10 | 00000550 |
| | END | 00000560 |

```

C      SUBROUTINE EVALMP                                00000010
C      THIS SUBROUTINE EVALUATES THE 5X3 MATRIX M AND THE 00000020
C      5X3X5 PARTIAL OF M WRT X                        00000030
C      C                                               00000040
C      IF IMFLAG=1, BOTH M (AM) AND ITS PARTIAL (PAM) ARE EVALUATED 00000050
C      IF IMFLAG=2, ONLY M (AM) IS EVALUATED          00000060
C      IF IMFLAG=3, ONLY THE PARTIAL OF M (PAM) IS EVALUATED 00000070
C      C                                               00000080
C      C                                               00000090
C      SUBROUTINE EVALMP(X, THETA, AMU, AM, PAM, IMFLAG) 00000100
C      IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)          00000110
C      DIMENSION X(5), AM(5,3), PAM(5,3,5)           00000120
C      COMMON /EVMP/ X1,Y1,X1DOT,Y1DOT                00000130
C      C                                               00000140
C      C                                               00000150
C      EN=DSORT(AMU/X(1)**3)                            00000160
C      RHO= DSORT(1.00- X(2)**2- X(3)**2)              00000170
C      BETA= 1.00/(1.00 +RHO)                          00000180
C      CT= DCOS(THETA)                                  00000190
C      ST= DSIN(THETA)                                  00000200
C      RA= 1.00-X(3)*CT -X(2)*ST                       00000210
C      ZETA= X(3)*ST-X(2)*CT                            00000220
C      BETA3= BETA**3/(1.00 -BETA)                     00000230
C      X1= X(1)*((1.00 -X(2)**2*BETA)*CT +X(2)*X(3)*BETA*ST -X(3)) 00000240
C      Y1= X(1)*((1.00 -X(3)**2*BETA)*ST +X(2)*X(3)*BETA*CT -X(2)) 00000250
C      X1DOT= -((1.00 -X(2)**2*BETA)*ST -X(2)*X(3)*BETA*CT)*EN*X(1)/RA 00000260
C      Y1DOT= ((1.00 -X(3)**2*BETA)*CT -X(2)*X(3)*BETA*ST)*EN*X(1)/RA 00000270
C      PZ1= X(1)*(ZETA*(BETA+X(2)**2*BETA3) -X(2)*BETA -ST)*CT/RA    00000280
C      PZ2= -X(1)*(-ZETA*X(2)*X(3)*BETA3 +1.00 +(ST -X(2)*BETA)*ST/RA) 00000290
C      PZ3= X(1)*(-ZETA*X(2)*X(3)*BETA3-1.00 +(X(3)*BETA -CT)*CT/RA) 00000300
C      PZ4= X(1)*(-ZETA*(BETA +X(3)**2*BETA3) +(CT -X(3)*BETA)*ST/RA) 00000310
C      IF (IMFLAG .EQ. 3) GO TO 10                      00000320
C      C IF DO NOT WANT TO EVALUATE PARTIAL OF M, BRANCH TO 10 00000330
C      AM(1,1)= 2.00*X1DOT/(EN**2*X(1))                00000340
C      AM(1,2)= 2.00*Y1DOT/(EN**2*X(1))                00000350
C      AM(1,3)=0.00                                     00000360
C      DUM= RHO/(EN*X(1)**2)                            00000370
C      AM(2,1)= DUM*(PZ2- X(2)*BETA*X1DOT/EN)          00000380
C      AM(2,2)= DUM*(PZ4 -X(2)*BETA*Y1DOT/EN)          00000390
C      AM(2,3)= DUM*(X(3)*(X(5)*Y1 -X(4)*X1)/RHO**2)  00000400
C      AM(3,1)= -DUM*(PZ1 +X(3)*BETA*X1DOT/EN)         00000410
C      AM(3,2)= -DUM*(PZ3 +X(3)*BETA*Y1DOT/EN)         00000420
C      AM(3,3)= -DUM*(X(2)*(X(5)*Y1 -X(4)*X1)/RHO**2) 00000430
C      AM(4,1)=0.00                                     00000440
C      AM(4,2)=0.00                                     00000450
C      DUM= (1.00 +X(4)**2 +X(5)**2)/(2.00*EN*X(1)**2*RHO) 00000460
C      AM(4,3)= DUM*Y1                                  00000470
C      AM(5,1)=0.00                                     00000480
C      AM(5,2)=0.00                                     00000490
C      AM(5,3)= DUM*X1                                  00000500
C      IF (IMFLAG .EQ. 2) RETURN                        00000510
C      C IF WE ONLY WISH TO EVALUATE M THEN PROGRAM RETURNS HERE 00000520
10 CA= DSORT(AMU/X(1))/RA                               00000530
C      PZ5= X(2)*BETA3                                  00000540
C      PZ6= X(3)*BETA3                                  00000550
C      PZ9= CA*ST/RA                                    00000560
C      PZ10= CA*CT/RA                                   00000570

```

```

PZ20= X(1)*(-2.00*X(2)*BETA*CT +X(3)*BETA*ST +PZ5*ZETA*X(2)) 00000580
PZ26= X(1)*(X(2)*HFTA*SI -1.00 +PZ6*X(2)*ZETA) 00000590
PZ29= X(1)*(X(3)*BETA*CT -1.00 -PZ5*X(3)*ZETA) 00000600
PZ35= X(1)*(-2.00*X(3)*BETA*ST +X(2)*BETA*CT -PZ6*X(3)*ZETA) 00000610
PZ11= -X1DOT/(2.00*X(1)) 00000620
PZ12= -Y1DOT/(2.00*X(1)) 00000630
DUM1= 1.00 -RA 00000640
PZ13= -CA*(-2.00*X(2)*BETA*ST -X(3)*BETA*CT -PZ5*X(2)*DUM1)+PZ9 00000650
1 *X1DOT/CA 00000660
PZ14= -CA*(-X(2)*BETA*CT -PZ6*X(2)*DUM1) +PZ10*X1DOT/CA 00000670
PZ15= -CA*(X(3)*BETA*ST +PZ5*X(3)*DUM1) +PZ9*Y1DOT/CA 00000680
PZ16= -CA*(2.00*X(3)*BETA*CT +X(2)*BETA*ST +PZ6*DUM1*X(3)) 00000690
1 +PZ10*Y1DOT/CA 00000700
DUM= BETA +X(2) *PZ5 00000710
PZ17= 1.00+ PZ5*X(2)*(3.00/BETA +1.00/(1.00-BETA)) 00000720
PZ18= (2.00 +PZ17)*PZ5 00000730
PZ19= PZ17*PZ6 00000740
DUM2= X(2)*BETA -ST 00000750
PZ21= -X(1)*(CT*DUM -ZETA*PZ18 +CT*DUM/RA +CT*ST*DUM2/RA**2) 00000760
PZ22= X(1)*(ST*DUM +ZETA*PZ19 -CT*X(2)*PZ6/RA-CT**2*DUM2/RA**2) 00000770
PZ23= BETA3*(3.00/BETA +1.00/(1.00 -BETA)) 00000780
PZ24= PZ23*PZ5 00000790
PZ25= PZ23*PZ6 00000800
PZ27= X(1)*(-CT*X(2)*X(3)*BETA3 +ZETA*X(3)*(BETA3 +X(2)*PZ24) 00000810
1 +(ST*(BETA +X(2)*PZ5))/RA +ST**2*DUM2/RA**2) 00000820
PZ28= X(1)*(ST*X(2)*X(3)*BETA3 +ZETA*X(2)*(BETA3 +X(3)*PZ25) 00000830
1 +X(2)*ST*PZ6/RA +ST*CT*DUM2/RA**2) 00000840
DUM2= X(3)*BETA-CT 00000850
PZ30= X(1)*(CT*X(2)*X(3)*BETA3 -ZETA*X(3)*(BETA3 +X(2)*PZ24) 00000860
1 +CT*X(3)*PZ5/RA +CT*ST*DUM2/RA**2) 00000870
PZ31= X(1)*(-ST*X(2)*X(3)*BETA3 -ZETA*X(2)*(BETA3 +X(3)*PZ25) 00000880
1 +CT*(BETA +X(3)*PZ6)/RA +CT**2*DUM2/RA**2) 00000890
DUM= BETA +X(3)*PZ6 00000900
PZ32= 1.00 +PZ6*X(3)*(3.00/BETA +1.00/(1.00 -BETA)) 00000910
PZ33= PZ32*PZ5 00000920
PZ34= PZ32*PZ6 +2.00*X(3)*6ETA3 00000930
PZ36= X(1)*(CT*DUM -ZETA*PZ33 -ST*X(3)*PZ5/RA -ST**2*DUM2/RA**2) 00000940
PZ37= X(1)*(-ST*DUM -ZETA*PZ34 -ST*(BETA +X(3)*PZ6)/RA -ST*CT 00000950
1 *DUM2/RA**2) 00000960
DO 20 J=1,2 00000970
20 PAM(1,J,1)= 3.00*AM(1,J)/(2.00*X(1)) 00000980
DUM =2.00*X(1)**2/AMU 00000990
PAM(1,1,2)= PZ13*DUM 00001000
PAM(1,1,3)= PZ14*DUM 00001010
PAM(1,2,2)= PZ15*DUM 00001020
PAM(1,2,3)=PZ16*DUM 00001030
DUM= DSORT(AMU*X(1)) 00001040
CB=RHO/DUM 00001050
PZ38= -X(2)*CB/RHO**2 00001060
PZ39= -X(3)*CB/RHO**2 00001070
PAM(2,1,1)= AM(2,1)/(2.00*X(1)) 00001080
PAM(2,1,2)= -CB*BETA*Y1DOT/EN +PZ38*AM(2,1)/CB +CB*(PZ27 00001090
1 -X(2)*BETA*PZ13/EN -X(2)*X1DOT*PZ5/EN) 00001100
PAM(2,1,3)= PZ39*AM(2,1)/CB +CB*(PZ28 -PZ6*X(2)*X1DOT/EN 00001110
1 -X(2)*BETA*PZ14/EN) 00001120
PAM(2,2,1)= AM(2,2)/(2.00*X(1)) 00001130
PAM(2,2,2)= PZ38*AM(2,2)/CB +CB*(PZ36 -BETA*Y1DOT/EN -X(2) 00001140

```

```

1      *Y1DOT*P75/EN -X(2)*META*P215/EN)                00001150
PAM(2,2,3)= P739*AM(2,2)/CR +CR*(P237 -X(2)*Y1DOT*P26/EN 00001160
1      -X(2)*META*P716/EN)                                00001170
PAM(2,3,1)= AM(2,3)/(2.00*X(1))                            00001180
DUM1= X(5)*Y1 -X(4)*X1                                    00001190
PAM(2,3,2)= X(3)*(X(5)*P229 -X(4)*P220)/(RHO*DUM) +X(2)*X(3) 00001200
1      *DUM1/(RHO**3*DUM)                                  00001210
PAM(2,3,3)= DUM1/(RHO*DUM) +X(3)*(X(5)*P235 -X(4)*P226)/(RHO 00001220
1      *DUM) +X(3)**2*DUM1/(RHO**3*DUM)                    00001230
PAM(2,3,4)= -X(3)*X1/(RHO*DUM)                            00001240
PAM(2,3,5)= X(3)*Y1/(RHO*DUM)                            00001250
PAM(3,1,1)= AM(3,1)/(2.00*X(1))                            00001260
PAM(3,1,2)= P238*AM(3,1)/CR -CR*(P221 +X(3)*X1DOT*P25/EN 00001270
1      +X(3)*META*P213/EN)                                00001280
PAM(3,1,3)= P239*AM(3,1)/CR -CR*(P222 +(META*X1DOT +X(3) 00001290
1      *X1DOT*P76 +X(3)*META*P214)/EN)                      00001300
PAM(3,2,1)= AM(3,2)/(2.00*X(1))                            00001310
PAM(3,2,2)= P238*AM(3,2)/CR -CR*(P230 +X(3)*(Y1DOT*P25 00001320
1      +META*P215)/EN)                                      00001330
PAM(3,2,3)= P239*AM(3,2)/CR -CR*(P231 +(META*Y1DOT +X(3) 00001340
1      *Y1DOT*P76 +X(3)*META*P216)/EN)                      00001350
PAM(3,3,1)= AM(3,3)/(2.00*X(1))                            00001360
PAM(3,3,2)= -DUM1/(RHO*DUM) -X(2)*(X(5)*P229 -X(4)*P220)/ 00001370
1      (RHO*DUM) -X(2)**2*DUM1/(RHO**3*DUM)                00001380
PAM(3,3,3)= -X(2)*(X(5)*P235 -X(4)*P226)/(RHO*DUM) -X(2)*X(3) 00001390
1      *DUM1/(RHO**3*DUM)                                    00001400
PAM(3,3,4)= X(2)*X1/(RHO*DUM)                              00001410
PAM(3,3,5)= -X(2)*Y1/(RHO*DUM)                            00001420
Z5= (1.00 +X(5)**2 +X(4)**2)/(2.00*DUM*RHO)                00001430
PZ40= -Z5/(2.00*X(1))                                       00001440
PZ41= X(2)*Z5/RHO**2                                         00001450
PZ42= X(3)*Z5/RHO**2                                         00001460
PZ43= X(4)/(DUM*RHO)                                         00001470
PZ44= X(5)/(DUM*RHO)                                         00001480
PAM(4,3,1)= AM(4,3)/(2.00*X(1))                              00001490
PAM(4,3,2)= PZ41*Y1+Z5*P229                                  00001500
PAM(4,3,3)= PZ42*Y1 +Z5*P235                                 00001510
PAM(4,3,4)= PZ43*Y1                                          00001520
PAM(4,3,5)= PZ44*Y1                                          00001530
PAM(5,3,1)= AM(5,3)/(2.00*X(1))                              00001540
PAM(5,3,2)= PZ41*X1 +Z5*P220                                  00001550
PAM(5,3,3)= PZ42*X1 +Z5*P226                                  00001560
PAM(5,3,4)= PZ43*X1                                          00001570
PAM(5,3,5)= PZ44*X1                                          00001580
DO 30 K=1,5                                                  00001590
PAM(I,3,K)=0.00                                             00001600
DO 30 I=4,5                                                 00001610
DO 30 J=1,2                                                 00001620
30 PAM(I,J,K)=0.00                                           00001630
DO 40 I=1,3                                                 00001640
DO 40 J=1,2                                                 00001650
DO 40 K=4,5                                                 00001660
40 PAM(I,J,K)=0.00                                           00001670
RETURN                                                       00001680
END                                                           00001690

```


| | | |
|---|---|----------|
| C | MAINE | 00000010 |
| C | | 00000020 |
| C | | 00000030 |
| C | | 00000040 |
| C | | 00000050 |
| C | SUBROUTINE MAINE (E,F,XK,UP,XJ,HR,IMAX,DP,TTOT) | 00000060 |
| C | IMPLICIT REAL*8 (A-H,O-Z) | 00000070 |
| C | COMMON/S1R/S | 00000080 |
| C | COMMON/BPC/BPO,BPT,E1,PI,DTF | 00000090 |
| C | 103 FORMAT(6F15.10,' FAILED GENERAL TESTS') | 00000100 |
| C | CALL START (E,F,XK,UP,XJ,HR,1,1,NF) | 00000110 |
| C | IF (NF) 399,399,100 | 00000120 |
| C | 100 J=0 | 00000130 |
| C | I= 1 | 00000140 |
| C | BPO=(DSORT(2.00*(1.00+E))-1.00-E)/HR | 00000150 |
| C | 202 J=J+1 | 00000160 |
| C | 208 DT1= 0.00 | 00000170 |
| C | 210 CALL TIME(I,J,DT1,SPAN,S(I,J+1,2)) | 00000180 |
| C | 220 CALL SWITCH(I,J,S(I,J+1,2),X,NX,0) | 00000265 |
| C | IF (S(I,J+1,1).GT.TTOT) GO TO 230 | 00000270 |
| C | IF (J+1-JM) 202,307,307 | 00000280 |
| C | 230 JM= J | 00000290 |
| C | 307 DTF= TTOT-S(I,JM,1) | 00000300 |
| C | S(I,JM+1,14)= 1.00 | 00000347 |
| C | S(I,JM+1,15)= 0.00 | 00000360 |
| C | S(I,JM+1,3)= 0.00 | 00000370 |
| C | CALL DTDU(I,JM,DTF,JM+1,1) | 00000371 |
| C | 308 RETURN | 00000380 |
| C | 399 WRITE (6,103) E,F,XK,UP,XJ,HR | 00000381 |
| C | RETURN | 00000382 |
| C | END | 00000384 |
| | | 00000385 |
| | | 00000386 |
| | | 00000387 |
| | | 00000390 |
| | | 00000400 |
| | | 00000410 |
| | | 00000420 |

| | | |
|----|--|----------|
| C | START | 00000010 |
| C | | 00000020 |
| C | | 00000030 |
| C | | 00000040 |
| C | | 00000050 |
| C | SUBROUTINE START (F,F,XK,UP,XJ,HR,I,J,NF) | 00000060 |
| C | | 00000070 |
| C | IMPLICIT REAL*8 (A-H,O-Z) | 00000080 |
| C | | 00000090 |
| C | DIMENSION S(3,7,20) | 00000100 |
| C | | 00000110 |
| C | COMMON/STR/S | 00000120 |
| C | | 00000130 |
| | S(I,J,12)= DCOS(F) | 00000140 |
| | S(I,J,13)= DSIN(F) | 00000150 |
| | S(I,J,10)= 1.00+E*S(I,J,12) | 00000160 |
| | RAD= DSQRT(S(I,J,10)) | 00000170 |
| | S(I,J,5)= E*S(I,J,13)/RAD | 00000180 |
| | S(I,J,9)= RAD*DSIN(UP) | 00000190 |
| | S(I,J,11)= RAD*DCOS(UP)-XK | 00000200 |
| | S(I,J,7)= (XJ*S(I,J,9)+XK*S(I,J,5))/(1.00+XK*(XK+S(I,J,11))) | 00000210 |
| 5 | S(I,J,4)= HR/DSQRT((1.00+S(I,J,7)**2)*S(I,J,10)) | 00000220 |
| | S(I,J,6)= XK | 00000230 |
| | S(I,J,8)= XJ*(XK+S(I,J,11))+(XK*S(I,J,7)-S(I,J,5))*S(I,J,9) | 00000240 |
| | IF (I-1) 31,6,7 | 00000250 |
| 6 | CALL SWITCH(I,1,0.00,0X,NF,1) | 00000260 |
| | IF (NF) 31,31,7 | 00000270 |
| 7 | S(I,J,1)= 0.00 | 00000280 |
| | S(I,J,2)= 0.00 | 00000290 |
| | S(I,J,3)= 0.00 | 00000300 |
| | S(I,J,14)= 1.00 | 00000310 |
| | S(I,J,15)= 0.00 | 00000320 |
| | S(I,J,16)= 1.00 | 00000330 |
| | S(I,J,17)= 0.00 | 00000340 |
| | S(I,J,18)= 0.00 | 00000350 |
| | S(I,J,19)= 0.00 | 00000360 |
| | S(I,J,20)= 0.00 | 00000370 |
| 31 | RETURN | 00000380 |
| | END | 00000390 |

| | | |
|-------|--|----------|
| C | TIME | 00000010 |
| C | | 00000020 |
| C | | 00000030 |
| C | | 00000040 |
| C | | 00000050 |
| C | SUBROUTINE TIME (I,J,DT1,SPAN,DT5) | 00000060 |
| C | IMPLICIT REAL*8 (A-H,O-7) | 00000070 |
| C | | 00000080 |
| C | DIMENSION DT(50),O(50) | 00000090 |
| C | | 00000100 |
| C | | 00000110 |
| 101 | FORMAT(4D16.8) | 00000120 |
| 102 | FORMAT(4I3,6D16.8) | 00000130 |
| 107 | FORMAT(' FAILED KSP') | 00000140 |
| C | | 00000150 |
| | K= 2 | 00000160 |
| | K75= 0 | 00000170 |
| | IKS= 20 | 00000180 |
| | M= 4 | 00000190 |
| | DTNEG= 5.D-10 | 00000200 |
| | IF (DT1) 305,500,600 | 00000210 |
| 500 | DT(1)= 1.D-6 | 00000220 |
| | CALL SWITCH(I,J,DT(1),O(1),NF,1) | 00000230 |
| | IF (NF) 306,306,501 | 00000240 |
| 501 | DT(2)= 1.D-3 | 00000250 |
| 502 | CALL SWITCH(I,J,DT(K),O(K),NF,2) | 00000260 |
| | IF (NF) 510,510,503 | 00000270 |
| 503 | IF (K-7) 504,306,306 | 00000280 |
| 504 | DT(K+1)= DT(K)*5.D0 | 00000290 |
| | K= K+1 | 00000300 |
| | GO TO 502 | 00000310 |
| 510 | SPAN= (DT(K)-DT(K-1))/5.D0 | 00000320 |
| | GO TO 79 | 00000330 |
| 600 | CALL SWITCH(I,J,DT1,O(1),NF,1) | 00000340 |
| | IF (NF) 2,2,1 | 00000350 |
| 1 | DT(1)= DT1 | 00000360 |
| | DT(2)= DT(1)+SPAN | 00000370 |
| | CALL SWITCH(I,J,DT(2),O(2),NF,2) | 00000380 |
| | IF (NF) 50,50,305 | 00000390 |
| 2 | DT(2)= DT1 | 00000400 |
| | O(2)= O(1) | 00000410 |
| | DT(1)= DMAX1(1.D-6,DT1-SPAN) | 00000420 |
| | CALL SWITCH(I,J,DT(1),O(1),NF,2) | 00000430 |
| | IF (NF) 305,305,50 | 00000440 |
| 50 | KS= K-1 | 00000450 |
| C NOW | GOOD AT DT(KS),BAD AT DT(KS)+SPAN | 00000460 |
| | KSP= KS+IKS | 00000470 |
| | DO= O(K)-O(K-1) | 00000480 |
| 51 | DDT= (DT(K-1)-DT(K))/DO*O(K) | 00000490 |
| | IF (DDT*(DTNEG-DDT)) 54,53,53 | 00000500 |
| 53 | IF (NF) 75,75,100 | 00000510 |
| 54 | IF (KSP-K) 304,304,55 | 00000520 |
| 55 | DT(K+1)= DT(K)+DDT-DTNEG/2.D0 | 00000530 |
| | IF ((DT(K+1)-DT(KS)+DTNEG)*(DT(KS)+SPAN-DT(K+1))) 75,75,56 | 00000540 |
| 56 | K= K+1 | 00000550 |
| | CALL SWITCH (I,J,DT(K),O(K),NF,2) | 00000560 |
| | DO= O(K)-O(K-1) | 00000570 |

| | | |
|-----|--|----------|
| | IF (DO) 51,100,51 | 00000580 |
| 75 | IF (K75-M) 76,76,305 | 00000590 |
| 76 | K75= K75+1 | 00000600 |
| | K= KS | 00000610 |
| | SPAN= SPAN/5.DO | 00000620 |
| 78 | K= K+1 | 00000630 |
| 79 | DT(K)= DT(K-1)+SPAN | 00000640 |
| | CALL SWITCH(I,J,DT(K),O(K),NF,2) | 00000650 |
| 80 | IF (NF) 50,50,78 | 00000660 |
| 100 | DT= DT(K)+DTNEG | 00000670 |
| | RETURN | 00000680 |
| 304 | WRITE (6,107) | 00000690 |
| 305 | WRITE (6,102) I,J,K,K75,DT1,SPAN,DT(1),O(1),DT(2),O(2) | 00000700 |
| | WRITE (6,101) (DT(IK),O(IK), IK=1,K) | 00000710 |
| | KK= 5) | 00000720 |
| | O(KK)= 1.DO | 00000730 |
| 306 | RETURN | 00000740 |
| | END | 00000750 |

| | |
|--|----------|
| C SWITCH | 00000010 |
| C | 00000020 |
| C | 00000030 |
| C | 00000040 |
| C | 00000050 |
| C SUBROUTINE SWITCH(I,J,T,01,NF,KDU) | 00000060 |
| C | 00000070 |
| C IMPLICIT REAL*8(A-H,O-Z) | 00000080 |
| C | 00000090 |
| C DIMENSION S(3,7,20) | 00000100 |
| C | 00000110 |
| C COMMON/STR/S | 00000120 |
| C | 00000130 |
| 177 FORMAT(8D16.8) | 00000140 |
| 178 FORMAT(5D25.15) | 00000150 |
| C | 00000160 |
| IF (KDU-1) 40,20,30 | 00000170 |
| 20 ASK= 1.00-S(I,J,9)*S(I,J,9)-S(I,J,6)*S(I,J,6) | 00000180 |
| TF= 1.00-2.00*S(I,J,7)*S(I,J,7) | 00000190 |
| QS= S(I,J,5)+(S(I,J,6)-S(I,J,11))*S(I,J,7) | 00000200 |
| QSR= 8.00*S(I,J,7)*QS | 00000210 |
| ASKO= ASK*(2.00*TF-1.00) | 00000220 |
| THO= -3.00*(2.00*TF-1.00) | 00000230 |
| D22= -(2.00+TF)*S(I,J,7) | 00000240 |
| D21= -D22*S(I,J,6)-QS*TF | 00000250 |
| XE= TF-S(I,J,6)**2-((S(I,J,8)+S(I,J,5)*S(I,J,9))**2+ | 00000260 |
| 1 (S(I,J,5)*S(I,J,6)-S(I,J,7))**2)/S(I,J,10) | 00000270 |
| 30 F= S(I,J,11)+T*S(I,J,4) | 00000280 |
| F2= F*F | 00000290 |
| P= (F+S(I,J,6))*(F+S(I,J,6))+S(I,J,9)*S(I,J,9) | 00000300 |
| AA= P*(ASK+F2)-4.00*F2 | 00000310 |
| XB= S(I,J,7)*(2.00*P*(F+S(I,J,6))-4.00*F)+(S(I,J,5)+ | 00000320 |
| 1 S(I,J,7)*T*S(I,J,4))*(2.00*F2-P) | 00000330 |
| XC= ASKO+QSR*F+THO*F2+P*XE | 00000340 |
| XD= D21+D22*F | 00000350 |
| X4= XA*XE-XB*XD | 00000360 |
| Z4= XE*XB*XB+XA*(XD*XD-XE*XC) | 00000370 |
| O1= (XC*XC+16.00*X4)*X4*X4-Z4*(XC*(XC*XC+16.00*X4)+27.00*Z4) | 00000380 |
| IF (XE) 0,9,777 | 00000390 |
| 777 IF (O1) 9,3,3 | 00000400 |
| 3 PP= 2.00*XC-3.00*XD*XD/XE | 00000410 |
| IF (PP) 4,10,10 | 00000420 |
| 4 IF (XC*XC+12.00*X4-PP*PP) 9,9,10 | 00000430 |
| 9 NF= 0 | 00000440 |
| RETURN | 00000450 |
| 10 NF= 1 | 00000460 |
| RETURN | 00000470 |
| 40 DER= XD*XD/XE/XE | 00000480 |
| TR= DER/2.00*(XC/XE-3.00/4.00*DER)+2.00*(4/XE/XE | 00000490 |
| OP= XC/XE-3.00/2.00*DER | 00000495 |
| OQ= (XD*(DER-XC/XE)+2.00*XB)/XE | 00000500 |
| IF (OQ) 42,41,42 | 00000510 |
| 41 S(I,J+1,3)= 1.020 | 00000520 |
| S(I,J+1,14)= -1.00 | 00000530 |
| S(I,J+1,15)= 0.00 | 00000540 |
| GO TO 43 | 00000550 |
| 42 PPR= OP+DSORT(XC*XC+12.00*X4)/XE | 00000560 |

```
YS= -DABS(D0)/D0*DSJKT(TR/PPR)
S3= YS-X0/2.00/XE
S(I,J+1,3)= 1.00/S3
403 DEN= 1.00+P*S(I,J+1,3)*S(I,J+1,3)
S(I,J+1,14)= 2.00/DEN-1.00
S(I,J+1,15)= 2.00*S(I,J+1,3)*DSQRT(P)/DEN
43 CALL DTDU(I,J,T,J+1,1)
300 RETURN
END
```

```
00000570
00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
```

| | | |
|----|--|----------|
| C | DTDU | 00000010 |
| C | | 00000020 |
| C | | 00000030 |
| C | | 00000040 |
| C | | 00000050 |
| C | SUBROUTINE DTDU(I,L,DT,M,ISW) | 00000060 |
| C | IMPLICIT REAL*8 (A-H,O-7) | 00000070 |
| C | DIMENSION S(3,7,20) | 00000080 |
| C | COMMON/STR/S | 00000090 |
| C | | 00000100 |
| | CDU= 1.00 | 00000110 |
| | SDU= 0.00 | 00000120 |
| | DF= S(I,L,4)*DT | 00000130 |
| | VL= 1.00+DF*(S(I,L,6)+S(I,L,11))/S(I,L,10) | 00000140 |
| | VH= DF*S(I,L,9)/S(I,L,10) | 00000150 |
| | RAD2= VL*VL+VH*VH | 00000160 |
| | S(I,M,1)= S(I,L,1)+DT | 00000170 |
| | S(I,M,2)= DT | 00000180 |
| | S(I,M,5)= S(I,L,5)+DF*S(I,L,7) | 00000190 |
| | S(I,M,8)= S(I,L,8)*VL-VH*(S(I,L,10)-1.00)*S(I,L,7)- | 00000200 |
| | 1 S(I,L,11)*S(I,L,5) | 00000210 |
| | S(I,M,10)= S(I,L,10)*RAD2 | 00000220 |
| | S(I,M,11)= S(I,L,11)+DF | 00000230 |
| | RAD= DSQRT(RAD2) | 00000240 |
| | CX= VL/RAD | 00000250 |
| | SX= VH/RAD | 00000260 |
| | S(I,M,16)= CX*S(I,L,16)-SX*S(I,L,17) | 00000270 |
| | S(I,M,17)= CX*S(I,L,17)+SX*S(I,L,16) | 00000280 |
| | S(I,M,18)= S(I,L,18) | 00000290 |
| | TX= (CX+S(I,M,16))/(1.00+S(I,L,16)) | 00000300 |
| | S(I,M,19)= TX*S(I,L,19)+SX*S(I,L,12) | 00000310 |
| | S(I,M,20)= TX*S(I,L,20)+SX*S(I,L,13) | 00000320 |
| | IF (ISW) 12,12,11 | 00000330 |
| 11 | CDU= S(I,L+1,14) | 00000340 |
| | SDU= S(I,L+1,15) | 00000350 |
| | SOR= SDU/DSQRT(S(I,M,10)) | 00000360 |
| | SRT= (1.00-CDU)/S(I,M,10) | 00000370 |
| | TP= CDU+SRT-SOR*S(I,M,5) | 00000380 |
| | TTO= CDU*S(I,L,7)+SOR*S(I,M,11) | 00000390 |
| | CSOC= DSQRT(1.00+S(I,L,7)*S(I,L,7)-TTO*TTO) | 00000400 |
| | FPR= CSOC/DSQRT(TP) | 00000410 |
| | S(I,M,4)= FPR*S(I,L,4) | 00000420 |
| | S(I,M,5)= (CDU*S(I,M,5)+(S(I,M,10)-1.00)*SOR)*FPR/CSOC | 00000430 |
| | S(I,M,6)= (S(I,L,6)-SOR*S(I,L,7)-SRT*S(I,M,11))/FPR/TP | 00000440 |
| | S(I,M,7)= TTO/CSOC | 00000450 |
| | S(I,M,9)= (CDU*S(I,L,9)+SOR*S(I,M,8))/FPR/TP | 00000460 |
| | S(I,M,8)= (CDU*S(I,M,8)-SOR*S(I,L,9)*S(I,M,10))/CSOC | 00000470 |
| | S(I,M,11)= (CDU*S(I,M,11)-SOR*S(I,L,7)*S(I,M,10))/FPR | 00000480 |
| | S(I,M,10)= TP*S(I,M,10) | 00000490 |
| | T1= S(I,M,17) | 00000500 |
| | S(I,M,17)= CDU*S(I,M,17)-SDU*S(I,M,18) | 00000510 |
| | S(I,M,18)= CDU*S(I,M,18)+SDU*T1 | 00000520 |
| 12 | SSI= S(I,M,19)*S(I,M,19)+S(I,M,20)*S(I,M,20) | 00000530 |
| | IF (SSI) 13,13,14 | 00000540 |
| | | 00000550 |
| | | 00000560 |
| | | 00000570 |

| | | |
|-----|---|---------|
| 13 | $S(I,M,12) = CDU * S(I,L,12) - SDU * S(I,L,13)$ | 0000580 |
| | $S(I,M,13) = CDU * S(I,L,13) + SDU * S(I,L,12)$ | 0000590 |
| | GO TO 300 | 0000600 |
| 14 | $S(I,M,12) = (S(I,M,17) - S(I,M,19) - S(I,M,18) * S(I,M,20)) / SS1$ | 0000610 |
| | $S(I,M,13) = (S(I,M,17) * S(I,M,19) - S(I,M,18) * S(I,M,20)) / SS1$ | 0000620 |
| 300 | RETURN | 0000630 |
| | END | 0000640 |

| | | |
|-----|--|----------|
| C | OUTH1 | 00000010 |
| C | | 00000020 |
| C | | 00000030 |
| C | | 00000040 |
| C | SUBROUTINE OUTH1(JM,P),AO,A11,OM1,IPR,Z,IDIM2) | 00000050 |
| C | IMPLICIT REAL*8 (A-H,O-Z) | 00000060 |
| C | | 00000070 |
| C | DI MENSION S(3,7,20),TR(3,3),TR1(3,3),TR3(3,3),ZP(10),Z(1) | 00000080 |
| C | | 00000090 |
| C | COMMON/STP/S | 00000100 |
| C | COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,UTKG,UTKW,UTMS2 | 00000110 |
| C | COMMON /SG/SGN | 00000120 |
| C | | 00000130 |
| C | | 00000140 |
| C | NAMelist/OUT/J,A,H,F,AI,OM,W,F,U,PHI,UP,DT,TT | 00000150 |
| C | NAMelist/OUT2/A13,GM3,W3,ALAM3,ALAM4,ALAM5,Y2,ALA,ALE,ALI | 00000160 |
| C | NAMelist/OUT3/ZP | 00000170 |
| C | | 00000180 |
| C | | 00000190 |
| C | JM1=JM+1 | 00000200 |
| C | IF (IPR.EQ.0) GO TO 12 | 00000210 |
| C | DO 10 K= 1,20 | 00000220 |
| C10 | WRITE (6,999) K,(S(1,J,K), J=1,JM1) | 00000230 |
| C | WRITE (6,1000) | 00000240 |
| C | WRITE (6,1001) | 00000250 |
| C | WRITE (6,1000) | 00000260 |
| C | | 00000270 |
| C | 12 DO 100 J= 1,JM1 | 00000280 |
| C | IF ((IPR.EQ.0).AND.(J.GT.1).AND.(J.LT.JM1)) GO TO 100 | 00000290 |
| C | AI= DATAN2(DSORT(S(1,J,19)**2+S(1,J,20)**2),S(1,J,16)) | 00000300 |
| C | OM= 0.00 | 00000310 |
| C | IF (S(1,J,16).NE.1.00) OM= DATAN2(S(1,J,20),S(1,J,19)) | 00000320 |
| C | DUM= DATA2(S(1,J,13),S(1,J,12)) | 00000330 |
| C | U= DUM-OM | 00000340 |
| C | DUM1= S(1,J,10)-1.00 | 00000350 |
| C | DUM2= DSORT(S(1,J,10))*S(1,J,5) | 00000360 |
| C | E= DSORT(DUM1**2+DUM2**2) | 00000370 |
| C | F=U | 00000380 |
| C | IF (E.NE.0.00) F= DATAN2(DUM2,DUM1) | 00000390 |
| C | W= U-F | 00000400 |
| C | PHI= DATAN(S(1,J,7)) | 00000410 |
| C | H= S(1,J,4)*DSORT(S(1,J,10))/DCOS(PHI) | 00000420 |
| C | M= DABS(H) | 00000430 |
| C | UP= DATAN2(S(1,J,9),S(1,J,11))+S(1,J,6) | 00000440 |
| C | DT= S(1,J,2) | 00000450 |
| C | TT= S(1,J,1) | 00000460 |
| C | IF (J.GT.1) GO TO 40 | 00000470 |
| C | | 00000480 |
| C | CPII= DCOS(PHI) | 00000490 |
| C | G1= S(1,1,8)*CPII | 00000500 |
| C | G2= S(1,1,9)*DSORT(S(1,1,10))*CPII | 00000510 |
| C | | 00000520 |
| C | SGN= +1.00 IMPLIES ALAM4 IS POSITIVE | 00000530 |
| C | SGN= -1.00 IMPLIES ALAM4 IS NEGATIVE | 00000540 |
| C | | 00000550 |
| C | ALAM5= (SGN+1.00)*PI/2.00 | 00000560 |
| C | ALAM4= SGN*DSORT(G1**2+G2**2) | 00000570 |

| | | |
|----|--|----------|
| | IF (ALAM4.EQ.0.D0) GO TO 14 | 00000580 |
| | ALAM5= DATAN2(SGN*G2.SGN*G1) | 00000590 |
| 14 | WI= ALAM5-PI/2.D0 | 00000600 |
| | AL1= DSORT(1.D0/A0)*ALAM4 | 00000610 |
| | WD= W | 00000620 |
| | AID= A11 | 00000630 |
| | OMD= OM1 | 00000640 |
| | GO TO 110 | 00000650 |
| 20 | DO 30 M=1,3 | 00000660 |
| | DO 30 N= 1,3 | 00000670 |
| | TR3(M,N)= TR(M,N) | 00000680 |
| 30 | TR1(M,N)= TR(M,N) | 00000690 |
| | IF (IPR) 52,100,52 | 00000700 |
| C | | 00000710 |
| 40 | WD= W | 00000720 |
| | AID= AI | 00000730 |
| | OMD= OM | 00000740 |
| | GO TO 110 | 00000750 |
| 50 | DO 51 M= 1,3 | 00000760 |
| | DO 51 N= 1,3 | 00000770 |
| | TR3(M,N)= 0.D0 | 00000780 |
| | DO 51 IK=1,3 | 00000790 |
| 51 | TR3(M,N)= TR3(M,N)+TR(M,IK)*TR1(IK,N) | 00000800 |
| 52 | S1= DSORT(TR3(3,1)**2+TR3(3,2)**2) | 00000810 |
| | A13= DATAN2(S1,TR3(3,3)) | 00000820 |
| | OM3= 0.D0 | 00000830 |
| | W3= DATAN2(TR3(1,2),TR3(1,1)) | 00000840 |
| | IF (DABS(S1).LT.1.D-10) GO TO 60 | 00000850 |
| | OM3= DATAN2(TR3(3,1),-TR3(3,2)) | 00000860 |
| | W3= DATAN2(TR3(1,3),TR3(2,3)) | 00000870 |
| 60 | Z(2)= E*DSIN(W3+OM3) | 00000880 |
| | Z(3)= E*DCOS(W3+OM3) | 00000890 |
| | Z(4)= TR3(1,1)/(1.D0+TR3(3,3)) | 00000900 |
| | Z(5)= -TR3(3,2)/(1.D0+TR3(3,3)) | 00000910 |
| | ALAM3= S(1,J,4)*(S(1,J,10)*S(1,J,6)-S(1,J,11))/M | 00000920 |
| | Y2= DSIN(F)*DSIN(PHI)+DCOS(F)*DCOS(PHI)*S(1,J,11)/DSORT(| 00000930 |
| 1 | S(1,J,10)) | 00000940 |
| | DUM= 1.D0-E**2 | 00000950 |
| | A= A0*H*H/DUM | 00000960 |
| | Z(1)= A | 00000970 |
| | ALA= .5D0*DSORT(1.D0/(A**3*(UM)))*(ALAM3-E*Y2) | 00000980 |
| | ALE= DSORT(1.D0/(A*DUM**3))*(Y2-E*ALAM3) | 00000990 |
| | Z(1+IDIM2)= ALA | 00001000 |
| | Z(2+IDIM2)= DSIN(W3+OM3)*ALE | 00001010 |
| | Z(3+IDIM2)= DCOS(W3+OM3)*ALE | 00001020 |
| | DUM= 2.D0*(DCOS(A1/2.D0))**2*AL1 | 00001030 |
| | Z(4+IDIM2)= DSIN(OM3)*DUM | 00001040 |
| | Z(5+IDIM2)= DCOS(OM3)*DUM | 00001050 |
| | IF (IPR.EQ.0) GO TO 100 | 00001060 |
| | W3= W3/DTR | 00001070 |
| | OM3= OM3/DTR | 00001080 |
| | A13= A13/DTR | 00001090 |
| | A1= A1/DTR | 00001100 |
| | OM= OM/DTR | 00001110 |
| | W= W/DTR | 00001120 |
| | F= F/DTR | 00001130 |
| | U= U/DTR | 00001140 |

| | | |
|-----|---|---------|
| | PHI= PHI/DTR | 0001150 |
| | UP= UP/DTP | 0001160 |
| C | | 0001170 |
| | DD 90 I= 1.5 | 0001180 |
| | ZP(1)= Z(1) | 0001190 |
| 90 | ZP(1+5)= Z(1+10IM2) | 0001200 |
| C | WRITE (6,OUT) | 0001210 |
| C | WRITE (6,OUT2) | 0001220 |
| C | WRITE (6,OUT3) | 0001230 |
| | A= A*UTKM | 0001240 |
| | DV= (DT*UTKM/UTS)*DSORT(1,DD/A0) | 0001250 |
| | WRITE (6,1002) J | 0001260 |
| | WRITE (6,1003) | 0001270 |
| | WRITE (6,1010) (Z(1), I= 1.5) | 0001280 |
| | WRITE (6,1010) (Z(1+10IM2), I= 1.5) | 0001290 |
| | WRITE (6,1004) | 0001300 |
| | WRITE (6,1010) A,E,413,OM3,W3 | 0001310 |
| | WRITE (6,1005) | 0001320 |
| | WRITE (6,1010) F,PHI,UP,DV | 0001330 |
| 100 | CONTINUE | 0001340 |
| C | | 0001350 |
| | IF (IPR.EQ.0) RETURN | 0001360 |
| | WRITE (6,1000) | 0001370 |
| | WRITE (6,1000) | 0001380 |
| | WRITE (6,1006) | 0001390 |
| C | | 0001400 |
| | RETURN | 0001410 |
| C | | 0001420 |
| 110 | CW= DCOS(WD) | 0001430 |
| | SW= DSIN(WD) | 0001440 |
| | CI= DCOS(AID) | 0001450 |
| | SI= DSIN(AID) | 0001460 |
| | COM= DCOS(OMD) | 0001470 |
| | SOM= DSIN(OMD) | 0001480 |
| | TR(1,3)= SW*SI | 0001490 |
| | TR(2,3)= CW*SI | 0001500 |
| | TR(3,3)= CI | 0001510 |
| | TR(3,1)= SI*SOM | 0001520 |
| | TR(3,2)= -SI*COM | 0001530 |
| | TR(1,1)= CW*COM-SW*CI*SOM | 0001540 |
| | TR(1,2)= CW*SOM+SW*CI*COM | 0001550 |
| | TR(2,1)= -SW*COM-CW*CI*SOM | 0001560 |
| | TR(2,2)= -SW*SOM+CW*CI*COM | 0001570 |
| | IF (J.EQ.1) GO TO 20 | 0001580 |
| | GO TO 50 | 0001590 |
| C | | 0001600 |
| C | | 0001610 |
| C | | 0001620 |
| | 999 FORMAT (14,1P5D20.R) | 0001630 |
| | 1000 FORMAT (1H0,'*****') | 0001640 |
| | 1001 FORMAT (1H0,'HIGH THRUST') | 0001650 |
| | 1002 FORMAT (1H0,'ORBIT =',I2) | 0001660 |
| | 1003 FORMAT (1H0,'EQUINOCTIAL D.E. AND COSTATE/S.F.*1000.') | 0001670 |
| | 1004 FORMAT (1H0,'CLASSICAL D.E.') | 0001680 |
| | 1005 FORMAT (1H0,6X,'TRUE ANOMALY',13X,'PHI (DEG)',12X,'UPSILON (DEG)', | 0001690 |
| | 1 9X,'DELTA V (KM/S)') | 0001700 |
| | 1006 FORMAT (1H0,'LOW THRUST') | 0001710 |
| | 1010 FORMAT (1P5D23.14) | 0001720 |
| | END | 0001730 |

| | | |
|----|---|----------|
| C | IMPLS | 00000010 |
| C | | 00000070 |
| C | | 00000030 |
| C | | 00000040 |
| C | CALCULATES EFFECT OF ONE FINAL IMPULSE AT MAX. OF PRIMER VECTOR | 00000050 |
| C | | 00000060 |
| C | | 00000070 |
| C | SUBROUTINE IMPLS(DV12,IPR,ITR,Z,DERZ,IDIM2) | 00000080 |
| C | | 00000090 |
| C | IMPLICIT REAL*8(A-H,O-S) | 00000100 |
| C | | 00000110 |
| C | DIMENSION Y(22),UY(3),V2(3),R(3),CM(3,3),M(3),E(3),C1(3) | 00000120 |
| C | DIMENSION ZZ(5),ZS(5),M1(3),V1(3),Z(1),DFRZ(1),DZS(5),Z10(10) | 00000130 |
| C | | 00000140 |
| C | COMMON /EVMP/ RT(2),VT(2) | 00000150 |
| C | COMMON /A/ A,AMU,PI | 00000160 |
| C | COMMON/UNIT5/UTS,UTM,UTH,UT),UTKM,DTR,UTKG,UTKW,UTMS2 | 00000170 |
| C | | 00000180 |
| C | NAMelist/FIM1/F,UY,ZZ,R1,V1,R,V2,H,E,Y,YS,FS,IS,YO,CM | 00000190 |
| C | NAMelist/FIM2/DZS | 00000200 |
| C | | 00000210 |
| C | | 00000220 |
| C | DO 4 I= 1,5 | 00000230 |
| | Z10(I)= Z(I) | 00000240 |
| 4 | Z10(I+5)= Z(I+IDIM2) | 00000250 |
| | KK= 0 | 00000260 |
| | DF= PI/10.00 | 00000270 |
| 5 | F= -DF | 00000280 |
| | YS= -1.00 | 00000290 |
| | DO 10 J= 2,21 | 00000300 |
| | F= F+DF | 00000310 |
| | CAL YF(F,Y(1),UY,Z10) | 00000320 |
| | IF (Y(1).LE.YS) GO TO 10 | 00000330 |
| | YS= Y(1) | 00000340 |
| | IS= J | 00000350 |
| | FS= F | 00000350 |
| 10 | CONTINUE | 00000370 |
| C | | 00000380 |
| | Y(1)= Y(21) | 00000390 |
| | Y(22)= Y(2) | 00000400 |
| | FN= FS-DF | 00000410 |
| | FP= FS+DF | 00000420 |
| | C= (Y(IS)*FP**2-Y(IS+1)*FS**2)-(Y(IS-1)*FP**2-Y(IS+1)*FN**2) | 00000430 |
| 1 | + (Y(IS-1)*FS**2-Y(IS)*FN**2) | 00000440 |
| | B= (FS*Y(IS+1)-FP*Y(IS))-(FN*Y(IS+1)-FP*Y(IS-1)) | 00000450 |
| 1 | + (FN*Y(IS)-FS*Y(IS-1)) | 00000460 |
| | F= C/(2.00*B) | 00000470 |
| C | | 00000480 |
| | CALL YF(F,YO,UY,Z10) | 00000490 |
| | DO 15 i= 1,2 | 00000500 |
| | R1(i)= RT(i) | 00000510 |
| 15 | V1(i)= VT(i) | 00000520 |
| | R1(3)= 0.00 | 00000530 |
| | V1(3)= 0.00 | 00000540 |
| | DO 20 I= 1,3 | 00000550 |
| 20 | V1(i)= V1(i)+DV12*UY(i) | 00000560 |
| | P= Z(4) | 00000570 |

| | | |
|-----|--|----------|
| | C= Z(5) | 00000580 |
| | IFLAG= 1 | 00000590 |
| | GO TO 200 | 00000600 |
| C | | 00000610 |
| 25 | DO 30 I= 1,3 | 00000620 |
| | V2(I)= 0.00 | 00000630 |
| | R(I)= 0.00 | 00000640 |
| | DO 30 J=1,3 | 00000650 |
| | V2(I)= V2(I)+CM(I,J)*V1(J) | 00000660 |
| 30 | R(I)= R(I)+CM(I,J)*R1(J) | 00000670 |
| | DO 40 I= 1,3 | 00000680 |
| | I1= I+1 | 00000690 |
| | I2= I+2 | 00000700 |
| | IF (I1.GT.3) I1= I1-3 | 00000710 |
| | IF (I2.GT.3) I2= I2-3 | 00000720 |
| 40 | H(I)= R(I1)*V2(I2)-R(I2)*V2(I1) | 00000730 |
| | DO 50 I= 1,3 | 00000740 |
| | I1= I+1 | 00000750 |
| | I2= I+2 | 00000760 |
| | IF (I1.GT.3) I1= I1-3 | 00000770 |
| | IF (I2.GT.3) I2= I2-3 | 00000780 |
| 50 | E(I)= (V2(I1)*H(I2)-V2(I2)*H(I1))/AMU | 00000790 |
| | HM= 0.00 | 00000800 |
| | VM= 0.00 | 00000810 |
| | RAD= 0.00 | 00000820 |
| | DO 60 I= 1,3 | 00000830 |
| | VM= VM+V2(I)**2 | 00000840 |
| | HM= HM+H(I)**2 | 00000850 |
| 60 | RAD= RAD+R(I)**2 | 00000860 |
| | RAD= DSORT(RAD) | 00000870 |
| | HM= DSORT(HM) | 00000880 |
| | DO 70 I= 1,3 | 00000890 |
| | E(I)= E(I)-R(I)/RAD | 00000900 |
| 70 | H(I)= H(I)/HM | 00000910 |
| | ZZ(1)= 1.00/(2.00/RAD-VM/AMU) | 00000920 |
| | DUM= 1.00+H(3) | 00000930 |
| | ZZ(4)= H(1)/DUM | 00000940 |
| | ZZ(5)= -H(2)/DUM | 00000950 |
| C | | 00000960 |
| | P= ZZ(4) | 00000970 |
| | Q= ZZ(5) | 00000980 |
| | IFLAG= 2 | 00000990 |
| | GO TO 200 | 00001000 |
| 75 | DO 80 I=1,2 | 00001010 |
| | ZZ(4-I)= 0.00 | 00001020 |
| | DO 80 J= 1,3 | 00001030 |
| 80 | ZZ(4-I)= ZZ(4-I)+E(J)*CM(J,I) | 00001040 |
| | KK= KK+1 | 00001050 |
| C | IF (IPR.GT.0) WRITE (6,FIM1) | 00001060 |
| C | | 00001070 |
| | IF (ITR.GT.0) GO TO 140 | 00001080 |
| | IF (KK-1) 120, 100,120 | 00001090 |
| 100 | DO 110 I= 1,5 | 00001100 |
| | ZS(I)= ZZ(I) | 00001110 |
| | Z10(I)= Z10(I)+DERZ(I)*1.0-2 | 00001120 |
| 110 | Z10(I+5)= Z(I+IDIM2)+DERZ(I+IDIM2)*1.0-2 | 00001130 |
| | GO TO 5 | 00001140 |

```

C
120 DO 130 I= 1,5
    DFRZ(I)= (ZZ(I)-75(I))*1.02
    DZ5(I)=DERZ(I)
130 Z(I)= Z5(I)
C IF (IPR.GT.0) WRITE (6,FIM2)
    GO TO 160
C
140 DO 150 I= 1,5
    Z(I)= ZZ(I)
150 CONTINUE
C
160 IF (IPR.EQ.0) RETURN
    WRITE (6,1000)
    WRITE (6,1001)
    WRITE (6,1000)
    WRITE (6,1002)
    WRITE (6,1010) (Z(I),I= 1,5)
    ZZ(1)= Z(1)*UTKM
    ZZ(2)= 0.00
    DUMMY= Z(2)**2+Z(3)**2
    IF (DUMMY.GT.1.0-40) ZZ(2)= DSORT(DUMMY)
    ZZ(3)= 0.00
    DUMMY= Z(4)**2+Z(5)**2
    IF (DUMMY.GT.1.0-40) ZZ(3)= 2.00*DATAN(DSORT(DUMMY))/DTR
    ZZ(4)= 0.00
    IF ((Z(4).NE.0.00).OR.(Z(5).NE.0.00)) ZZ(4)=DATAN2(Z(4),Z(5))/DTR
    ZZ(5)= 0.00
    IF ((Z(2).NE.0.00).OR.(Z(3).NE.0.00)) ZZ(5)=DATAN2(Z(2),Z(3))/DTR
    ZZ(5)= ZZ(5)-ZZ(4)
    WRITE (6,1003)
    WRITE (6,1010) ZZ
    F= F/DTR
    WRITE (6,1004) F
    WRITE (6,1005) R1
    WRITE (6,1006) UY
    WRITE (6,1000)
    RETURN
C
C
C CALCULATES TRANSFORMATION MATRIX TO EQUINOCTIAL COORD FRAME
C
200 AB= 1.00+P**2+Q**2
    CM(1,1)= (1.00-P**2+Q**2)/AB
    CM(2,1)= 2.00*P*Q/AB
    CM(3,1)= -2.00*P/AB
    CM(1,2)= CM(2,1)
    CM(2,2)= (1.00+P**2-Q**2)/AB
    CM(3,2)= 2.00*Q/AB
    CM(1,3)= -CM(2,1)
    CM(2,3)= -CM(3,2)
    CM(3,3)= (1.00-P**2-Q**2)/AB
    GO TO (25,75),IFLAG
C
1000 FORMAT (1H0,'*****')
1001 FORMAT (1H0,'HIGH THRUST FINAL ORBIT')
1002 FORMAT ('0 EQUINOCTIAL O.E.')

```

```

00001150
00001160
00001170
00001180
00001190
00001200
00001210
00001220
00001230
00001240
00001250
00001260
00001270
00001280
00001290
00001300
00001310
00001320
00001330
00001340
00001350
00001360
00001370
00001380
00001390
00001400
00001410
00001420
00001430
00001440
00001450
00001460
00001470
00001480
00001490
00001500
00001510
00001520
00001530
00001540
00001550
00001560
00001570
00001580
00001590
00001600
00001610
00001620
00001630
00001640
00001650
00001660
00001670
00001680
00001690
00001700
00001710

```

1003 FORMAT ('O CLASSICAL O.F.')

1004 FORMAT ('O ECCENTRIC LONG. (DEG) = ',F20.10)

1005 FORMAT ('O RADIUS VECTOR = ',1P3023.14)

1006 FORMAT ('O IMPULSE DIRECTION = ',1P3023.14)

1010 FORMAT (1P5023.14)

C

END

00001720

00001730

00001740

00001750

00001760

00001770

00001780

| | | |
|----|---|----------|
| C | YF | 00000010 |
| C | | 00000020 |
| C | | 00000030 |
| C | CALL BY IMPLS, CALLS FVALMP | 00000040 |
| C | | 00000050 |
| C | | 00000060 |
| C | SUBROUTINE YF(F,Y,U,Z) | 00000070 |
| C | | 00000080 |
| C | IMPLICIT REAL*8(A-H,O-S) | 00000090 |
| C | | 00000100 |
| C | DIMENSION U(3),R(3),V(3),AM(5,3),DUMMY(75), X(5),Z(1) | 00000110 |
| C | | 00000120 |
| C | | 00000140 |
| C | | 00000150 |
| | DO 5 I= 1,5 | 00000160 |
| 5 | X(I)= Z(I) | 00000170 |
| | CALL EVALMP(X,F,1.00,AM,DUMMY,Z) | 00000180 |
| | Y= 0.00 | 00000190 |
| | DO 20 I= 1,3 | 00000200 |
| | U(I)= 0.00 | 00000210 |
| | DO 10 K= 1,5 | 00000220 |
| 10 | U(I)= U(I)+AM(K,I)*7(5+K) | 00000230 |
| 20 | Y= Y+U(I)**2 | 00000240 |
| | Y= DSORT(Y) | 00000250 |
| | DO 30 I= 1,3 | 00000260 |
| 30 | U(I)= U(I)/Y | 00000270 |
| | RETURN | 00000280 |
| | END | |


```

C ITER NRS
C NEWTON RAPHSON NRS 00000010
C 00000020
C SPECIAL VERSION FOR MAX PARTIAL DER. MATRIX, R DIM. Y. 00000030
C 00000040
C SUBROUTINE ITER(KOUNT,NI,FUNCT,PRTN) 00000050
C 00000060
C IMPLICIT REAL*(A-H,O-S), INTEGER (I-N) 00000070
C 00000080
C 00000090
C X VALUES OF THE INDEPENDENT VARIABLES(INITIAL,CURRENT,FINAL) DP00000100
C XS STEP SIZE TO PERFORM XS TO COMPUTE PARTIAL DERIVATIVES SP00000110
C Y VALUES OF THE DEPENDENT VARIABLES(CURRENT,FINAL) DP00000120
COMMON/XMMM/X(7),XS(R),Y(R) 00000130
COMMON /INT/ITF,IPR,IDIM,IDIM2,MAXNOI 00000140
COMMON /T/TF,S,TO,TFMIN 00000150
COMMON /DY/DYDT(R) 00000160
COMMON /F/FLIM,KSTEP 00000170
C 00000180
C 00000190
C ARRAYS USED INTERNALLY BY THE ITERATOR 00000200
DIMENSION YNOM(R),XN(7),P(R,R),COEF(R) 00000210
EQUIVALENCE(YNOM,COEF) 00000220
C 00000230
N= 7 00000240
M= 8 00000250
INORM= ITF 00000260
NI=1 00000270
KOUNT=0 00000280
CALL FUNCT 00000290
ITF=3 00000300
KOUNT=KOUNT+1 00000310
FO=0.00 00000320
DO 15 I=1,M 00000330
15 FO=FO+Y(I)**2 00000340
DO 10 I=1,N 00000350
10 XN(I)=X(I) 00000360
16 YNOM(I)=Y(I) 00000370
YNOM(M)= Y(M) 00000380
TFN=TF 00000390
CALL PRTN(KOUNT,NI) 00000400
WRITE(6,1011)FO 00000410
IF(FO.LE.FLIM)GO TO 90 00000415
IF (NI.GT.MAXNOI) GO TO 80 00000420
C COMPUTE NUMERICAL PARTIAL DERIVATIVES 00000430
WRITE (6,1013) 00000440
DO 17 I=1,M 00000450
17 P(I,M)= DYDT(I) 00000460
DO 25 J=1,N 00000470
TEMP=X(J) 00000480
STEP=XS(J)*DABS(X(J)) 00000490
IF ((DABS(X(J)).LT.1.D-10).OR.(KSTEP.EQ.1))STEP=XS(J) 00000500
IF (DABS(X(J)).LT.1.D-10) WRITE (6,1014) 00000510
X(J)=X(J)+STEP 00000520
CALL FUNCT 00000530
WRITE(6,1000)X(J) 00000540
WRITE(6,1001)(Y(I),I=1,M)

```

| | |
|---|----------|
| DO 20 I=1,M | 00000550 |
| 20 P(I,J)=(Y(I)-YNOM(I))/STEP | 00000560 |
| 25 X(J)=TEMP | 00000570 |
| KOUNT=KOUNT+N | 00000580 |
| WRITE(6,1002) | 00000590 |
| DO 30 I=1,M | 00000600 |
| WRITE(6,1001)(P(I,J),J=1,M) | 00000610 |
| 30 CONTINUE | 00000620 |
| DO 35 I=1,M | 00000630 |
| 35 COEF(I)=-YNOM(I) | 00000640 |
| CALL DCROUT(P,COEF,DEF,0,DO,M,1,IND) | 00000650 |
| IF(IND.NE.0)GO TO 85 | 00000660 |
| WRITE(6,1015) DEF | 00000670 |
| DO 40 I=1,M | 00000680 |
| 40 IF (DABS(COEF(I)).LT.1.D-12) COEF(I)= 0.D0 | 00000690 |
| WRITE(6,1003)(COEF(I),I=1,N) | 00000700 |
| SN= COEF(M) | 00000710 |
| WRITE(6,1012) SN | 00000720 |
| DO 50 J=1,N | 00000730 |
| 50 X(J)=XN(J)+COEF(J) | 00000740 |
| TF=TFN + SN | 00000750 |
| IHALV=0 | 00000760 |
| 51 IF (INORM.EQ.1) ITF=1 | 00000770 |
| CALL FUNCT | 00000780 |
| ITF=3 | 00000790 |
| KOUNT=KOUNT+1 | 00000800 |
| F1=0.D0 | 00000810 |
| DO 52 I=1,M | 00000820 |
| 52 F1=F1+Y(I)**2 | 00000830 |
| WRITE(6,1010)F1 | 00000840 |
| IF(F1.LT.F0)GO TO 55 | 00000850 |
| WRITE(6,1008) | 00000860 |
| IF (IHALV.EQ.10) GO TO 95 | 00000870 |
| IHALV=IHALV+1 | 00000880 |
| DO 53 J=1,N | 00000890 |
| COEF(J)=COEF(J)/2.D0 | 00000900 |
| WRITE(6,1000)COEF(J) | 00000910 |
| 53 X(J)=XN(J)+COEF(J) | 00000920 |
| SN= SN/2.D0 | 00000930 |
| WRITE(6,1012) SN | 00000940 |
| TF= TFN + SN | 00000950 |
| GO TO 51 | 00000960 |
| 55 IF(NI-MAXNOI)70,70,80 | 00000970 |
| 70 NI=NI+1 | 00000980 |
| F0=F1 | 00000990 |
| GO TO 10 | 00010000 |
| 80 NI=9999 | 00010100 |
| WRITE(6,1006) | 00010200 |
| RETURN | 00010300 |
| 85 NI=9999 | 00010400 |
| WRITE(6,1007) | 00010500 |
| RETURN | 00010600 |
| 90 WRITE(6,1005)F0 | 00010700 |
| RETURN | 00010800 |
| 95 NI=9999 | 00010900 |
| WRITE(6,1009) | 00011000 |
| RETURN | 00011100 |

| | | |
|------|--|----------|
| 1000 | FORMAT(/1X,1PD23.15) | 00001120 |
| 1001 | FORMAT(1X,1P5D23.15) | 00001130 |
| 1002 | FORMAT(21HOPARTIAL DERIV MATRIX) | 00001140 |
| 1003 | FORMAT(11HODELX:S APP / (1X,1PD23.15)) | 00001150 |
| 1005 | FORMAT(4HOF0=,1PD22.15,23HCASE CONVERGED...FERTIG) | 00001160 |
| 1006 | FORMAT(38HOEXCEEDED MAXIMUM NUMBER OF ITERATIONS) | 00001170 |
| 1007 | FORMAT(16HOMATRIX SINGULAR) | 00001180 |
| 1008 | FORMAT(11HODELX:S APP) | 00001190 |
| 1009 | FORMAT(19HOMETHOD CANNOT WORK) | 00001200 |
| 1010 | FORMAT(4HOF1=,1PD23.15) | 00001210 |
| 1011 | FORMAT(4HOF0=,1PD23.15) | 00001220 |
| 1012 | FORMAT (10HO DEL IF =,1PD23.15) | 00001230 |
| 1013 | FORMAT (40HO X(1)+DX(1) FOLLOWED BY CORRESPONDING Y) | 00001240 |
| 1014 | FORMAT (24HO X(1)=0. SO DX(1)=XS(1)) | 00001250 |
| 1015 | FORMAT (15HO DETERMINENT =,1PD23.15) | 00001260 |
| | END | 00001270 |

```

C ITER      NRN
C NEWTON RAPHSON      NRN
C
C SPECIAL VERSION FOR 6x6 PARTIAL DER. MATRIX, 6 DIM. Y.
C
C      SUBROUTINE ITER(KOUNT,NI,FUNCT,PRTN)
C
C      IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)
C
C X  VALUES OF THE INDEPFDENT VARIABLES(INITIAL,CURRENT,FINAL)
C XS STEP SIZE TO PERTURB X:S TO COMPUTE PARTIAL DERIVATIVES
C Y  VALUES OF THE DEPENDENT VARIABLES(CURRENT,FINAL)
COMMON/XMMM/X(5),XS(5),Y(6)
COMMON /INT/ITF,IPR,IDIM, IDIM2,MAXNOI
COMMON /T/TF,S,T0,TFMIN
COMMON /DY/DYDT(6)
COMMON /HIGH/DV11,DV12,IHI,ITR
COMMON /F/FLIM,KSTEP
C
C
C      ARRAYS USED INTERNALLY BY THE ITERATOR
C      DIMENSION YNOM(6),XN(5),P(6,6),COEF(6)
C      EQUIVALENCE(YNOM,COEF)
C
C      N= 5
C      M= 6
C      INORM= ITF
C      NI=1
C      KOUNT=0
C      ITR= 0
C      CALL FUNCT
C      ITF=3
C      KOUNT=KOUNT+1
C      FO=0.00
C      DO 15 I=1,M
15  FO=FO+Y(I)**2
C      DO 16 I=1,N
16  XN(I)=X(I)
C      YNOM(I)=Y(I)
C      YNOM(M)= Y(M)
C      TFN=TF
C      CALL PRTN(KOUNT,NI)
C      WRITE(6,1011)FO
C      IF(FO.LE.FLIM)GO TO 90
C      IF (NI.GT.MAXNOI) GO TO 80
C COMPUTE NUMERICAL PARTIAL DERIVATIVES
C      WRITE (6,1013)
C      DO 17 I=1,M
17  P(I,M)= DYDT(I)
C      DO 25 J=1,N
C      ITR= J
C      TEMP=X(J)
C      STEP=XS(J)*DABS(X(J))
C      IF ((DABS(X(J)).LT.1.0-10).OR.(KSTEP.EQ.1)) STEP=XS(J)
C      IF (DABS(X(J)).LT.1.0-10) WRITE (6,1014)
C      X(J)=X(J)+STEP

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
DP00000100
SP00000110
DP00000120
00000130
00000140
00000150
00000160
00000170
00000175
00000180
00000180
00000190
00000200
00000210
00000220
00000230
00000240
00000250
00000260
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360
00000370
00000380
00000390
00000400
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000550

```

| | |
|---|----------|
| CALL FUNCT | 00000560 |
| WRITE(6,1000)X(J) | 00000570 |
| WRITE(6,1001)(Y(I),I=1,M) | 00000580 |
| DO 20 I=1,M | 00000590 |
| 20 P(I,J)=(Y(I)-YNOM(I))/STEP | 00000600 |
| 25 X(J)=TEMP | 00000610 |
| KOUNT=KOUNT+N | 00000620 |
| WRITE(6,1002) | 00000630 |
| DO 30 I=1,M | 00000640 |
| WRITE(6,1001)(P(I,J),J=1,M) | 00000650 |
| 30 CONTINUE | 00000660 |
| DO 35 I=1,M | 00000670 |
| 35 COEF(I)=-YNOM(I) | 00000680 |
| CALL DCROUT(P,COEF,OPT,0.00,M,1,IND) | 00000690 |
| IF(IND.NE.0)GO TO 85 | 00000700 |
| DO 40 I=1,M | 00000710 |
| 40 IF (DABS(COEF(I)).LT.1.0-10) COEF(I)= 0.00 | 00000720 |
| WRITE(6,1003)(COEF(I),I=1,N) | 00000730 |
| SN= COEF(M) | 00000740 |
| WRITE (6,1012) SN | 00000750 |
| DO 50 J=1,N | 00000760 |
| 50 X(J)=XN(J)+COEF(J) | 00000770 |
| TF=TFN + SN | 00000780 |
| IHALV=0 | 00000790 |
| 51 IF (INORM.EQ.1) ITF=1 | 00000800 |
| ITR= 0 | 00000810 |
| CALL FUNCT | 00000820 |
| ITF=3 | 00000830 |
| KOUNT=KOUNT+1 | 00000840 |
| F1=0.00 | 00000850 |
| DO 52 I=1,M | 00000860 |
| 52 F1=F1+Y(I)**2 | 00000870 |
| WRITE(6,1010)F1 | 00000880 |
| IF(F1.LT.F0)GO TO 55 | 00000890 |
| WRITE(6,1008) | 00000900 |
| IF (IHALV.EQ.10) GO TO 95 | 00000910 |
| IHALV=IHALV+1 | 00000920 |
| DO 53 J=1,N | 00000930 |
| COEF(J)=COEF(J)/2.00 | 00000940 |
| WRITE(6,1000)COEF(J) | 00000950 |
| 53 X(J)=XN(J)+COEF(J) | 00000960 |
| SN= SN/2.00 | 00000970 |
| WRITE (6,1012) SN | 00000980 |
| TF= TFN + SN | 00000990 |
| GO TO 51 | 00001000 |
| 55 IF(NI-MAXNDI)70,70,00 | 00001010 |
| 70 NI=NI+1 | 00001020 |
| F0=F1 | 00001030 |
| GO TO 10 | 00001040 |
| 80 NI=9999 | 00001050 |
| WRITE(6,1006) | 00001060 |
| RETURN | 00001070 |
| 85 NI=9999 | 00001080 |
| WRITE(6,1007) | 00001090 |
| RETURN | 00001100 |
| 90 WRITE(6,1005)F0 | 00001110 |
| RETURN | 00001120 |

| | |
|--|----------|
| 95 NI=9999 | 00001130 |
| WRITE(6,1009) | 00001140 |
| RETURN | 00001150 |
| 1000 FORMAT(1X,1PD23.15) | 00001160 |
| 1001 FORMAT(1X,1P5D23.15) | 00001170 |
| 1002 FORMAT(21HOPARTIAL DERIV MATRIX) | 00001180 |
| 1003 FORMAT(11HMODELX:S ARE/(1X,1FD23.15)) | 00001190 |
| 1005 FORMAT(4HOF0=,1PD22.15,23HCASE CONVERGED...FERTIG) | 00001200 |
| 1006 FORMAT(38HDEXCEEDED MAXIMUM NUMBER OF ITERATIONS) | 00001210 |
| 1007 FORMAT(16HMATRIX SINGULAR) | 00001220 |
| 1008 FORMAT(11HMODELX:S ARE) | 00001230 |
| 1009 FORMAT(19HMETHOD CANNOT WORK) | 00001240 |
| 1010 FORMAT(4HOF1=,1PD23.15) | 00001250 |
| 1011 FORMAT(4HOF0=,1PD23.15) | 00001260 |
| 1012 FORMAT(10HO DEL TF =,1PD23.15) | 00001270 |
| 1013 FORMAT(40HO X(I)+DX(I) FOLLOWED BY CORRESPONDING Y) | 00001280 |
| 1014 FORMAT(24HO X(I)=0. SO DX(I)=XS(I)) | 00001290 |
| END | 00001300 |

| | |
|--|----------|
| C TRAJ/TRAJPCS | 00000010 |
| C | 00000020 |
| C SEP AND HIGH THRUST | 00000030 |
| C 7/14 DIM. WITH POWER DEGRADATION. | 00000040 |
| C THIS ROUTINE SETS UP THE INPUT TO THE INTEGRATOR WHICH | 00000050 |
| C EXTRAPOLATES THE TRAJECTORY FROM INITIAL TIME TO | 00000060 |
| C FINAL TIME. IT ALSO EVALUATES THE CHANGE IN TF AND | 00000070 |
| C THE ERROR IN THE FINAL CONDITIONS. | 00000080 |
| C THIS PROGRAM IS CALLED BY ITER OR BY CONTROL | 00000090 |
| C IT CALL THE SUBPROGRAM UMPCG (PRED-CORR) | 00000100 |
| C MIN J, MAX H. | 00000110 |
| C P DIM, ZERF, T.C. OPTIONS. | 00000120 |
| C NOP=1--ALL 5 FINAL O.E. FIXED, =2--A,E,I ONLY FIXED. | 00000130 |
| C | 00000140 |
| C | 00000150 |
| C SUBROUTINE TRAJ | 00000160 |
| C | 00000170 |
| C IMPLICIT REAL*8(A-H,O-S), INTEGER (I-N) | 00000180 |
| C | 00000190 |
| C COMMON /XMM/ZLO(7), STFP(R), ZERF(R) | 00000200 |
| C COMMON /TRA/TFMAX, DTO, UEB, EW(14) | 00000210 |
| C COMMON /Z/Z(14), DERZ(14) | 00000220 |
| C COMMON /INT/ITF, IPR, IDIM, IDIM2, NIMAX | 00000230 |
| C COMMON /T/TF, SD, TO, TFMIN | 00000240 |
| C COMMON /ELEM/ZPO(7), ZPF(5) | 00000250 |
| C COMMON /DY/DYDT(R) | 00000260 |
| C COMMON /TC/NOP | 00000270 |
| C COMMON /HIGH/DV11,DV12,IHI | 00000280 |
| C COMMON /A/A,AMU,PI | 00000290 |
| C COMMON /NOD/NODE | 00000300 |
| C | 00000310 |
| C EXTERNAL FUNCT, OUTP | 00000320 |
| C DIMENSION PRMT(5), AUX(16,14),DERZ1(14),DP(5) | 00000330 |
| C | 00000340 |
| C IF ((IHI.EQ.1).OR.(IHI.EQ.4)) GO TO 9 | 00000350 |
| C | 00000360 |
| C P1 THRUST | 00000370 |
| C | 00000380 |
| C UP= (PI/2.DO)*ZLO(1)/DSQRT(1.DO+ZLO(1)**2) | 00000390 |
| C CUP= DCOS(UP) | 00000400 |
| C XK= CUP*(.75DO+.25DO*ZLO(2)/DSQRT(1.DO+ZLO(2)**2)) | 00000410 |
| C DUM= (1.DO+CUP*XK)*DSQRT((CUP-XK)/(CUP+XK)) | 00000420 |
| C XJ= DUM*ZLO(3)/DSQRT(1.DO+ZLO(3)**2) | 00000430 |
| C JM= 2 | 00000440 |
| C VAR= ZPO(3) | 00000450 |
| C IF (NODE.EQ.0) VAR= ZLO(5) | 00000460 |
| C CALL MAINE(0.DO,0.DO,XK,UP,XJ,1.DO,1,JM,DP,DV11) | 00000470 |
| C CALL OUTH(JM,PI,ZPO(1),ZPO(2),VAR,IPR,Z,IDIM2) | 00000480 |
| C IF (NODE.EQ.0) GO TO 4 | 00000490 |
| C Z(IDIM2+4)= Z(IDIM2+4)+ZLO(5)*Z(5) | 00000500 |
| C Z(IDIM2+5)= Z(IDIM2+5)-ZLO(5)*Z(4) | 00000510 |
| 4 DO 5 I= 1,5 | 00000520 |
| 5 Z(I+IDIM2)= ZLO(4)*Z(I+IDIM2)*1.D4 | 00000530 |
| DO 6 I= 6,7 | 00000540 |
| Z(I)= ZPO(I) | 00000550 |
| 6 Z(I+IDIM2)= ZLO(I) | 00000560 |
| C | 00000570 |

| | |
|--|----------|
| C LOW THRUST | 00000580 |
| C | 00000590 |
| 9 DT= (TF-T0)/(IDIM*(TF-T0)/DT0)+1.00 | 00000600 |
| PRMT(1)= T0 | 00000610 |
| PRMT(2)= TF | 00000620 |
| PRMT(3)= DT | 00000630 |
| PRMT(4)= UEB | 00000640 |
| C | 00000650 |
| IF ((IMI.EQ.2).OR.(IMI.FQ.3)) GO TO 15 | 00000660 |
| C | 00000670 |
| C Z IS A VECTOR OF STATE AND COSTATE | 00000680 |
| DO 10 I=1, IDIM2 | 00000690 |
| Z(I)=ZPO(I) | 00000700 |
| 10 Z(I+IDIM2)= ZLO(I) | 00000710 |
| C | 00000720 |
| C EW ARE ERROR WEIGHTS--INPUT TO THE INTEGRATOR | 00000730 |
| C | 00000740 |
| 15 DO 20 I=1, IDIM | 00000750 |
| 20 DERZ(I)=EW(I) | 00000760 |
| C | 00000770 |
| C CALL THE P-C INTEGRATOR | 00000780 |
| C | 00000790 |
| CALL DMPCG(PRMT,7,DERZ, IDIM, IMLF, FUNCT-OUTP, AUX) | 00000800 |
| IF (IMLF.GT.10) GO TO 100 | 00000810 |
| C | 00000820 |
| C Z IS NOW THE FINAL O.E.. | 00000830 |
| C ZERF THE ERROR IN THE FINAL CONDITIONS | 00000840 |
| C | 00000850 |
| H=0.00 | 00000860 |
| DO 30 I=1, IDIM2 | 00000870 |
| 30 H= H + Z(I+7)*DERZ(I) | 00000880 |
| ZERF(6)= Z(13)*1.0-3 | 00000890 |
| ZERF(7)= Z(14)*1.0-3 | 00000900 |
| DYDT(6)= DERZ(13)*1.0-3 | 00000910 |
| DYDT(7)= DERZ(14)*1.0-3 | 00000920 |
| TF1=TF+(STEP(8)+1.00) | 00000930 |
| CALL FUNCT(TF1,Z,DERZ) | 00000940 |
| H1=0.00 | 00000950 |
| DO 35 I=1, IDIM2 | 00000960 |
| 35 H1=H1+Z(I+7)*DERZ(I) | 00000970 |
| DYDT(8)= (H1-H)/(TF1-TF) | 00000980 |
| ZERF(8)= H -1.00 | 00000990 |
| C | 00001000 |
| C FINAL CONDITION OPTION BRANCH | 00001010 |
| C | 00001020 |
| GO TO (40,50), NOP | 00001030 |
| C | 00001040 |
| 40 DO 45 J=1,5 | 00001050 |
| ZERF(I)= Z(I) -7PF(I) | 00001060 |
| 45 DYDT(I)= DERZ(I) | 00001070 |
| RETURN | 00001080 |
| C | 00001090 |
| 50 ZERF(4)= (Z(3)*Z(9)-Z(2)*Z(10))*1.0-3 | 00001100 |
| ZERF(5)= (Z(5)*Z(11)-Z(4)*Z(12))*1.0-3 | 00001110 |
| DYDT(4)= DERZ(3)*Z(9)+Z(3)*DERZ(9)-DERZ(2)*Z(10)-Z(2)*DERZ(10) | 00001120 |
| DYDT(4)= DYDT(4)*1.0-3 | 00001130 |
| DYDT(5)= DERZ(5)*Z(11)+Z(5)*DERZ(11)-DERZ(4)*Z(12)-Z(4)*DERZ(12) | 00001140 |

| | | |
|------|---|----------|
| | DYDT(5)= DYDT(5)*1.D-3 | 00001150 |
| | IF (IHLT.3) GO TO 60 | 00001160 |
| C | | 00001170 |
| C | FINAL HIGH THRUST IMPULSE | 00001180 |
| C | | 00001190 |
| | CALL IMPLS(DV12,IPR,0.2,DERZ,IDIM2) | 00001200 |
| C | | 00001210 |
| 60 | ZERF(1)= Z(1) - 7PF(1) | 00001220 |
| | DUM1= DSORT(Z(2)**2 + Z(3)**2) | 00001230 |
| | ZERF(2)= DUM1 - 7PF(2) | 00001240 |
| | DUM2= DSORT(Z(4)**2 + Z(5)**2) | 00001250 |
| | ZERF(3)= DUM2 - 7PF(3) | 00001260 |
| | DYDT(1)= DERZ(1) | 00001270 |
| | DYDT(2)= 0.D0 | 00001280 |
| | DYDT(3)= 0.D0 | 00001290 |
| | IF (DUM1.GT.1.D-12) DYDT(2)= (Z(2)*DERZ(2) + Z(3)*DERZ(3))/DUM1 | 00001300 |
| | IF (DUM2.GT.1.D-12) DYDT(3)= (Z(4)*DERZ(4) + Z(5)*DERZ(5))/DUM2 | 00001310 |
| C | | 00001320 |
| C | SPECIAL CASE, E=0 AND/OR I=0 | 00001330 |
| C | | 00001340 |
| | IF (7PF(2).NE.0.D0) GO TO 70 | 00001350 |
| | ZERF(2)= Z(2) | 00001360 |
| | ZERF(4)= Z(3) | 00001370 |
| | DYDT(2)= DERZ(2) | 00001380 |
| | DYDT(4)= DERZ(3) | 00001390 |
| 70 | IF (7PF(3).NE.0.D0) RETURN | 00001400 |
| | ZERF(3)= Z(4) | 00001410 |
| | ZERF(5)= Z(5) | 00001420 |
| | DYDT(3)= DERZ(4) | 00001430 |
| | DYDT(5)= DERZ(5) | 00001440 |
| | RETURN | 00001450 |
| C | | 00001460 |
| 100 | IF (IHLF.EQ.11) WRITE (6,1000) | 00001470 |
| | IF (IHLF.EQ.12) WRITE (6,1001) | 00001480 |
| | IF (IHLF.EQ.13) WRITE (6,1002) | 00001490 |
| | STOP | 00001500 |
| C | | 00001510 |
| 1000 | FORMAT (68H) THE NUMBER OF BISECTIONS OF THE ORIGINAL INCREMENT HAS | 00001520 |
| | EXCEEDED 10) | 00001530 |
| 1001 | FORMAT (27H) INITIAL INCREMENT IS ZERO: | 00001540 |
| 1002 | FORMAT (54H) INITIAL INCREMENT HAS WRONG SIGN OR BOUNDS ARE WRONG) | 00001550 |
| | END | 00001560 |

```

C TRAJ/TRAJPCN                                00000010
C                                                00000020
C NEP AND HIGH THRUST.                          00000030
C THIS ROUTINE SETS UP THE INPUT TO THE INTEGRATOR WHICH 00000040
C   EXTRAPOLATES THE TRAJECTORY FROM INITIAL TIME TO    00000050
C   FINAL TIME. IT ALSO EVALUATES THE CHANGE IN YF AND   00000060
C   THE ERROR IN THE FINAL CONDITIONS.                  00000070
C THIS PROGRAM IS CALLED BY ITER OR BY CONTROL          00000080
C IT CALLS THE SUBPROGRAM HMPCG (PREDICTOR CORRECTOR)    00000090
C MIN J, MAX H.                                        00000100
C 6 DIM. ZERF, T.C. OPTIONS.                          00000110
C ICP=1--ALL 5 FINAL O.F. FIXED. =2--A,E,I ONLY FIXED. 00000120
C DYDT USED ONLY IF ITR= 0                            00000130
C IHI= 1 NO HIGH THRUST                               00000140
C   = 2 2 INITIAL IMPULSES                            00000150
C   = 3 2 INITIAL AND ONE FINAL IMPULSE               00000160
C   = 4 FINAL IMPULSE                                 00000170
C WHEN IHI=2 OR 3, ZLO(5) AND ZERF(5) ARE DUMMIES.     00000180
C                                                       00000190
C                                                       00000200
C SUBROUTINE TRAJ                                    00000210
C                                                       00000220
C IMPLICIT REAL*8(A-H,O-S), INTEGER (I-N)             00000230
C                                                       00000240
C COMMON /XMMH/ZLO(5), DUMMY(5), ZERF(6)              00000250
C COMMON /TRA/TMAX, DT0, UER, EW(10)                  00000260
C COMMON /Z/Z(10), DERZ(10)                           00000270
C COMMON /INT/ITF, IPR, IDIM, IDIM2, NIMAX            00000280
C COMMON /T/TF, SD, TO, TMIN                          00000290
C COMMON /ELEM/ZPO(5), ZPF(5)                         00000300
C COMMON /DY/DYDT(6)                                  00000310
C COMMON /TC/NOP                                       00000320
C COMMON /POWER/PKW,CC,AMD,A0,A1,A2,A3,IPDW          00000330
C COMMON /A/A,AMU,PI                                   00000340
C COMMON /HIGH/DV11,DV12,IHI,ITR                     00000350
C COMMON /F/FLIM,KSTEP                                00000360
C                                                       00000370
C EXTERNAL FUNCT, OUTP                                00000380
C DIMENSION PRMT(5), AUX(16,10), DP(5), ZS(6)        00000390
C                                                       00000400
C IF ((IHI.EQ.1).OR.(IHI.FO.4)) GO TO 9              00000410
C IF (ITR.NE.1) GO TO 4                               00000420
C DO 3 I=1,6                                          00000430
C   ZS(I)=ZERF(I)                                     00000440
C 3 IF (ITR.EQ.4) GO TO 70                            00000450
C 4 IF (ITR.EQ.5) GO TO 40                            00000460
C                                                       00000470
C HIGH THRUST                                        00000480
C                                                       00000490
C UP= (PI/2.00)*ZLO(1)/DSQRT(1.00+ZLO(1)**2)         00000500
C CUP= DCOS(UP)                                       00000510
C XK= CUP*(.7500+.2500*ZLO(2)/DSQRT(1.00+ZLO(2)**2)) 00000520
C DUM= (1.00+CUP*XK)*DSQRT((CUP-XK)/(CUP+XK))         00000530
C XJ= DUM*ZLO(3)/DSQRT(1.00+ZLO(3)**2)              00000540
C JM= 2                                               00000550
C CALL MAINE(0.00,0.00,XK,UP,XJ,1.00,1,JM,DP,DV11)    00000560
C CALL OUTHI(JM,PI,ZPO(1),ZPO(2),0.00,IPR,Z,IDIM2)  00000570

```

| | |
|---|----------|
| DO 5 I= 1, IDIM2 | 00000580 |
| 5 Z(I+IDIM2)= ZLO(4)*Z(I+IDIM2)*1.D4 | 00000590 |
| C | 00000600 |
| C NEP | 00000610 |
| C | 00000620 |
| 9 DT= (TF-T0)/(IDINI((TF-T0)/DT0)+1.D0) | 00000630 |
| PRMT(1)= T0 | 00000640 |
| PRMT(2)= TF | 00000650 |
| PRMT(3)= DT | 00000660 |
| PRMT(4)= UEB | 00000670 |
| C | 00000680 |
| C Z IS A VECTOR OF STATE AND COSTATE | 00000690 |
| C | 00000700 |
| IF ((IMI.EQ.2).OR.(IMI.FQ.3)) GO TO 15 | 00000710 |
| DO 10 I=1, IDIM2 | 00000720 |
| Z(I)=ZPO(I) | 00000730 |
| 10 Z(I+IDIM2)= ZLO(I) | 00000740 |
| C | 00000750 |
| C EW ARE ERROR WEIGHTS--INPUT TO THE INTEGRATOR | 00000760 |
| C | 00000770 |
| 15 DO 20 I=1, IDIM | 00000780 |
| 20 DERZ(I)=EW(I) | 00000790 |
| C | 00000800 |
| C FOR ITF LT 3 USE NORM CUTOFF CONDITION | 00000810 |
| C | 00000820 |
| IF (ITF.LT.3) TF=TFMAX | 00000830 |
| C | 00000840 |
| C CALL THE PREDICTOR-CORRECTOR INTEGRATOR | 00000850 |
| C | 00000860 |
| CALL DHPCG(PRMT,Z,DERZ, IDIM, IMLF, FUNCT, OUTP, AUX) | 00000870 |
| IF (IMLF.GT.10) GO TO 100 | 00000880 |
| C | 00000890 |
| C Z IS NOW THE FINAL O.E., | 00000900 |
| C ZERF THE ERROR IN THE FINAL CONDITIONS | 00000910 |
| C | 00000920 |
| H=0.D0 | 00000930 |
| DO 30 I=1, IDIM2 | 00000940 |
| 30 H= H + Z(I+5)*DERZ(I) | 00000950 |
| ZERF(6)= H -1.D0 | 00000960 |
| IF (IMI.LT.3) GO TO 35 | 00000970 |
| C | 00000980 |
| C FINAL IMPULSE | 00000990 |
| C | 0001000 |
| PSI= (Z(3)*Z(7)-Z(2)*Z(8))*1.D-3 | 0001010 |
| PSIDOT= DERZ(3)*Z(7)+Z(3)*DERZ(7)-Z(2)*DERZ(8)-DERZ(2)*Z(8) | 0001020 |
| PSIDOT= PSIDOT*1.D-3 | 0001030 |
| C | 0001040 |
| CALL IMPLS(DVI2, IPR, ITR, Z, DERZ, IDIM2) | 0001050 |
| C | 0001060 |
| C FINAL CONDITION OPTION BRANCH | 0001070 |
| C | 0001080 |
| 35 GO TO (40,50), NOP | 0001090 |
| C | 0001100 |
| 40 DO 45 I=1, IDIM2 | 0001110 |
| ZERF(I)= Z(I) -ZPF(I) | 0001120 |
| 45 UYDT(I)= DERZ(I) | 0001130 |
| GO TO 55 | 0001140 |

| | | |
|----|---|----------|
| C | | 00001150 |
| 50 | ZERF(1)= Z(1) - 7PF(1) | 00001160 |
| | DUM1= DSQRT(Z(2)**2 + Z(3)**2) | 00001170 |
| | ZERF(2)= DUM1 - 7PF(2) | 00001180 |
| | DUM2= DSQRT(Z(4)**2 + Z(5)**2) | 00001190 |
| | ZERF(3)= DUM2 - 7PF(3) | 00001200 |
| | ZERF(4)= (Z(3)*Z(7) - Z(2)*Z(8))*1.0-3 | 00001210 |
| | ZERF(5)= (Z(5)*Z(9) - Z(4)*Z(10))*1.0-3 | 00001220 |
| | DYDT(1)= DERZ(1) | 00001230 |
| | DYDT(2)= 0.00 | 00001240 |
| | DYDT(3)= 0.00 | 00001250 |
| | IF (DUM1.GT.1.0-40) DYDT(2)= (Z(2)*DERZ(2) + Z(3)*DERZ(3))/DUM1 | 00001260 |
| | IF (DUM2.GT.1.0-40) DYDT(3)= (Z(4)*DERZ(4) + Z(5)*DERZ(5))/DUM2 | 00001270 |
| | DYDT(4)= DERZ(3)*Z(7)+Z(3)*DERZ(7)-DERZ(2)*Z(8)-Z(2)*DERZ(8) | 00001280 |
| | DYDT(4)=DYDT(4)*1.0-3 | 00001290 |
| | DYDT(5)= DERZ(5)*Z(9)+Z(5)*DERZ(9)-DERZ(4)*Z(10)-Z(4)*DERZ(10) | 00001300 |
| | DYDT(5)=DYDT(5)*1.0-3 | 00001310 |
| | IF (IMI.EQ.1) GO TO 55 | 00001320 |
| | IF (IMI.EQ.4) GO TO 52 | 00001330 |
| C | | 00001340 |
| C | DUMMY | 00001350 |
| C | | 00001360 |
| | ZERF(5)= ZLO(5)-1.00 | 00001370 |
| | DYDT(5)= 0.00 | 00001380 |
| | IF (IMI.LT.3) GO TO 53 | 00001390 |
| C | | 00001400 |
| 52 | ZERF(4)= PSI | 00001410 |
| | DYDT(4)= PSIDOT | 00001420 |
| 53 | IF (ITR.GT.0) RETURN | 00001430 |
| C | | 00001440 |
| | DO 54 I= 1,6 | 00001450 |
| 54 | ZS(I)= ZERF(I) | 00001460 |
| C | | 00001470 |
| 55 | IF (IPOW.EQ.1) GO TO 60 | 00001480 |
| | A4= 1.00-A1*TF | 00001490 |
| | DYDT(6)= A1*H/A4 | 00001500 |
| | RETURN | 00001510 |
| C | | 00001520 |
| 60 | B1= DEXP(-A2*TF) | 00001530 |
| | A4= 1.00+A1*(B1-1.00+A2*A3*TF) | 00001540 |
| | DYDT(6)= A2*H*(-B1/(B1-A3)+A1*(B1-A3)/A4) | 00001550 |
| | RETURN | 00001560 |
| C | | 00001570 |
| C | | 00001580 |
| C | | 00001590 |
| 70 | DO 75 I= 1,3 | 00001600 |
| 75 | ZERF(I)= ZS(I) | 00001610 |
| | DUM= DUMMY(4) | 00001620 |
| | IF (KSTEP.EQ.0) DUM= DUM*ZLO(4) | 00001630 |
| | ZERF(4)= (1.00+DUM)*ZS(4) | 00001640 |
| | ZERF(5)= ZS(5) | 00001650 |
| | ZERF(6)= (1.00+DUM)*(ZS(6)+1.00)-1.00 | 00001660 |
| | RETURN | 00001670 |
| C | | 00001680 |
| 80 | DO 85 I= 1,6 | 00001690 |
| 85 | ZERF(I)= ZS(I) | 00001700 |
| | ZERF(5)= ZS(5)+DUMMY(5) | 00001710 |

| | | |
|------|---|----------|
| | RETURN | 00001720 |
| C | | 00001730 |
| 100 | IF (IMLF.EQ.11) WRITE (6,1000) | 00001740 |
| | IF (IMLF.EQ.12) WRITE (6,1001) | 00001750 |
| | IF (IMLF.EQ.13) WRITE (6,1002) | 00001760 |
| | STOP | 00001770 |
| C | | 00001780 |
| 1000 | FORMAT (60H0 THE NUMBER OF RESECTIONS OF THE ORIGINAL INCREMENT HAS | 00001790 |
| | EXCEEDED 10) | 00001800 |
| 1001 | FORMAT (27H0 INITIAL INCREMENT IS ZERO) | 00001810 |
| 1002 | FORMAT (54H0 INITIAL INCREMENT HAS WRONG SIGN OR ROUNDS ARE WRONG) | 00001820 |
| | END | 00001830 |

```

C QUAD                                00000010
C                                     00000020
C THIS IS A MODIFIED QUADRATURE PROGRAM FOR VECTOR VALUED FUNCTIONS. 00000030
C COMPUTES INTEGRAL OF THE FUNCTION G (OR H) OVER X FROM XL TO XU. THE 00000040
C RESULT IS Y. ROUTINE USES A 4 POINT GAUSS QUADRATURE. 00000050
C                                     00000060
C                                     00000070
C                                     00000080
C SUBROUTINE QUAD(XL,XU,FCT,Y,Z,G,H,N) 00000090
C                                     00000100
C IMPLICIT REAL*8(A-H,O-S), INTEGER (I-N) 00000105
C DIMENSION Y(1),H(1),G(1),Z(1) 00000110
C                                     00000130
C A= .500*(XU+XL) 00000140
C B= XU-XL 00000150
C C= .4305681557970262900*B 00000160
C K=1 00000170
C GO TO 50 00000180
10 DO 20 I=1,N 00000190
20 Y(I)= .17392742256*7269300*G(I) 00000200
C= .1699905217924281300*B 00000210
C K=2 00000220
C GO TO 50 00000230
30 DO 40 I=1,N 00000240
40 Y(I)= B*(Y(I) + .3260725774312730700*G(I)) 00000250
C RETURN 00000260
50 CALL FCT(A-C,A+C,Z,H,G) 00000270
C DO 60 I=1,N 00000280
60 G(I)=G(I) + H(I) 00000290
C GO TO (10,30), K 00000300
C END 00000310

```

| | |
|--|----------|
| C QUAD/QUADR | 00000010 |
| C | 00000020 |
| C THIS IS A MODIFIED QUADRATURE PROGRAM FOR VECTOR VALUED FUNCTIONS. | 00000030 |
| C COMPUTES THE INTEGRAL OF THE FUNCTION G (OF H) OVER X FROM XL TO XU. | 00000040 |
| C THE RESULT IS Y. ROUTINE USES A 8 POINT GAUSS QUADRATURE. | 00000050 |
| C | 00000060 |
| C | 00000070 |
| C | 00000080 |
| SUBROUTINE QUAD(XL,XU,FCT,Y,Z,G,H,N) | 00000090 |
| C | 00000100 |
| IMPLICIT REAL*(A-H,O-S) | 00000110 |
| DIMENSION Y(1),H(1),G(1),Z(1) | 00000120 |
| C | 00000130 |
| A=.5D0*(XU+XL) | 00000140 |
| B=XU-XL | 00000150 |
| C=.48014492R24R76R12H0*B | 00000160 |
| K=1 | 00000170 |
| GO TO 500 | 00000180 |
| 10 DO 20 I=1,N | 00000190 |
| 20 Y(I)=.5061426R1451**130D-1*G(I) | 00000200 |
| C=.39R33323R706R1337H0*B | 00000210 |
| K=2 | 00000220 |
| GO TO 500 | 00000230 |
| 30 DO 40 I=1,N | 00000240 |
| 40 Y(I)=Y(I)+.11119051**2266R724D0*G(I) | 00000250 |
| C=.26276620495R16444D0*B | 00000260 |
| K=3 | 00000270 |
| GO TO 500 | 00000280 |
| 50 DO 60 I=1,N | 00000290 |
| 60 Y(I)=Y(I)+.156R5332293R94364D0*G(I) | 00000300 |
| C=.91717321247R24900-1*B | 00000310 |
| K=4 | 00000320 |
| GO TO 500 | 00000330 |
| 70 DO 80 I=1,N | 00000340 |
| 80 Y(I)=B*(Y(I)+.181341R9168918099D0*G(I)) | 00000350 |
| RETURN | 00000360 |
| 500 CALL FCT(A-C,A+C,Z,H,G) | 00000370 |
| DO 510 I=1,N | 00000380 |
| 510 G(I)=G(I) + H(I) | 00000390 |
| GO TO (10,30,50,70).K | 00000400 |
| END | 00000410 |

| | | |
|-----|--|----------|
| C | QUAD32 | 00000010 |
| C | | 00000020 |
| C | THIS IS A MODIFIED QUADRATURE INTEGRATION PROGRAM FOR | 00000030 |
| C | VECTOR VALUED FUNCTIONS OF ONE VARIABLE. IT INTEGRATES | 00000040 |
| C | G (OR H) OVER X FROM XL TO XU. THE RESULT IS Y. | 00000050 |
| C | EVALUATION IS BY A 32 POINT GAUSS QUADRATURE FORMULA. | 00000060 |
| C | | 00000070 |
| | SUBROUTINE QUAD(XL,XU,FCT,Y,Z,G,H,N) | 00000080 |
| C | | 00000090 |
| | IMPLICIT REAL*8(A-H,I-S) | 00000100 |
| | DIMENSION Y(1),Z(1),G(1),H(1) | 00000110 |
| C | | 00000120 |
| | A= .500*(XU-XL) | 00000130 |
| | B= XU-XL | 00000140 |
| | C= .498631930924740/H00*B | 00000150 |
| | K=1 | 00000160 |
| | GO TO 500 | 00000170 |
| 10 | DO 20 I=1,N | 00000180 |
| 20 | Y(I)= .350930500473504R3D-2*G(I) | 00000190 |
| | C= .49280575577263417D0*B | 00000200 |
| | K=2 | 00000210 |
| | GO TO 500 | 00000220 |
| 30 | DO 40 I=1,N | 00000230 |
| 40 | Y(I)= Y(I) + .8137197365452R35D-2*G(I) | 00000240 |
| | C= .4823R112779375322D0*B | 00000250 |
| | K=3 | 00000260 |
| | GO TO 500 | 00000270 |
| 50 | DO 60 I=1,N | 00000280 |
| 60 | Y(I)= Y(I) + .12696032654631030D-1*G(I) | 00000290 |
| | C= .467453037968R69R4D0*B | 00000300 |
| | K=4 | 00000310 |
| | GO TO 500 | 00000320 |
| 70 | DO 80 I=1,N | 00000330 |
| 80 | Y(I)= Y(I) + .17136931456510717D-1*G(I) | 00000340 |
| | C= .4481605778R302606D0*B | 00000350 |
| | K=5 | 00000360 |
| | GO TO 500 | 00000370 |
| 90 | DO 100 I=1,N | 00000380 |
| 100 | Y(I)= Y(I) + .21417949011113340D-1*G(I) | 00000390 |
| | C= .4246838068662R499D0*B | 00000400 |
| | K=6 | 00000410 |
| | GO TO 500 | 00000420 |
| 110 | DO 120 I=1,N | 00000430 |
| 120 | Y(I)= Y(I) + .2549902963118R08R0D-1*G(I) | 00000440 |
| | C= .3972418979R397120D0*B | 00000450 |
| | K=7 | 00000460 |
| | GO TO 500 | 00000470 |
| 130 | DO 140 I=1,N | 00000480 |
| 140 | Y(I)= Y(I) + .29342046739267774D-1*G(I) | 00000490 |
| | C= .366091059370144R4D0*B | 00000500 |
| | K=8 | 00000510 |
| | GO TO 500 | 00000520 |
| 150 | DO 160 I=1,N | 00000530 |
| 160 | Y(I)= Y(I) + .329111113881R0923D-1*G(I) | 00000540 |
| | C= .331522133465107R0D0*B | 00000550 |
| | K=9 | 00000560 |
| | GO TO 500 | 00000570 |

| | | |
|-----|--|----------|
| 170 | DO 180 I=1,N | 00000580 |
| 180 | Y(I)= Y(I) + .361728970544242530-1*G(I) | 00000590 |
| | C= .2938578786203811600*B | 00000600 |
| | K=10 | 00000610 |
| | GO TO 500 | 00000620 |
| 190 | DO 200 I=1,N | 00000630 |
| 200 | Y(I)= Y(I) + .390964478935351530-1*G(I) | 00000640 |
| | C= .2534499544661147000*B | 00000650 |
| | K=11 | 00000660 |
| | GO TO 500 | 00000670 |
| 210 | DO 220 I=1,N | 00000680 |
| 220 | Y(I)= Y(I) + .416559621134733780-1*G(I) | 00000690 |
| | C= .2106756380653176700*B | 00000700 |
| | K=12 | 00000710 |
| | GO TO 500 | 00000720 |
| 230 | DO 240 I=1,N | 00000730 |
| 240 | Y(I)= Y(I) + .438260465022019060-1*G(I) | 00000740 |
| | C= .1659343011410638200*B | 00000750 |
| | K=13 | 00000760 |
| | GO TO 500 | 00000770 |
| 250 | DO 260 I=1,N | 00000780 |
| 260 | Y(I)= Y(I) + .4558649393478819420-1*G(I) | 00000790 |
| | C= .1196436811260685400*B | 00000800 |
| | K=14 | 00000810 |
| | GO TO 500 | 00000820 |
| 270 | DO 280 I=1,N | 00000830 |
| 280 | Y(I)= Y(I) + .469221995404022830-1*G(I) | 00000840 |
| | C= .72235980791398250-1*B | 00000850 |
| | K=15 | 00000860 |
| | GO TO 500 | 00000870 |
| 290 | DO 300 I=1,N | 00000880 |
| 300 | Y(I)= Y(I) + .478193600396374300-1*G(I) | 00000890 |
| | C= .241538328438691580-1*B | 00000900 |
| | K=16 | 00000910 |
| | GO TO 500 | 00000920 |
| 310 | DO 320 I=1,N | 00000930 |
| 320 | Y(I)= B*(Y(I) + .482700442573639000-1*G(I)) | 00000940 |
| | RETURN | 00000950 |
| 500 | CALL FCT(A-C,A+C,Z,H,G) | 00000960 |
| | DO 510 I=1,N | 00000970 |
| 510 | G(I)= G(I) + H(I) | 00000980 |
| | GO TO (10,30,50,70,90,110,130,150,170,190,210,230,250,270,290,310) | 00000990 |
| | I ,K | 00001000 |
| | END | 00001010 |

```

C EARTH2 00000010
C 00000020
C 00000030
C THIS SUBPROGRAM SETS THE VALUES FOR EARTH'S ORBITAL ELEMENTS 00000040
C AND CALCULATES THE MEAN ANOMALY AT THE INITIAL TIME 00000050
C O.E. TAKEN FROM BATTIN, 1964. EPOCH 1960 JAN. 1.5, JD=2436935. 00000060
C INPUT 00000070
C TL--INITIAL TIME, BEGINNING OF LOW THRUST TRAJ 00000080
C OUTPUT 00000090
C C(1)--EARTH'S SEMIMAJOR AXIS 00000100
C C(2)--EARTH'S ECCENTRICITY 00000110
C C(3)--ARGUMENT OF PERI. 00000120
C C(4)--MEAN ORBITAL MOTION 00000130
C C(5)--MEAN ANOMALY AT TL 00000140
C C(6)--COSINE OF ANGLE OF OBLIQUITY 00000150
C C(7)--SINE OF ANGLE OF OBLIQUITY 00000160
C C(17)--LONG. OF N MAG. POLE AT TIME TL 00000170
C C(18)--LAT. OF N. MAG. POLE 00000180
C C(19)--EARTH ROT. PERIOD. 00000190
C C(20)--2*PI 00000200
C 00000210
C 00000220
C 00000230
C SUBROUTINE EARTH 00000240
C 00000250
C IMPLICIT REAL*8(A-H,O-S) 00000260
C 00000270
C COMMON /JD/ TL 00000280
C COMMON /TERRA/ C(20) 00000290
C COMMON /UNITS/UTS,UTM,UTH,UTD,UTKM,DTR,UTKG,UTKW,UTMS2 00000300
C 00000310
C 00000320
C C(1) = 1.00 00000330
C C(2) = .01672600 00000340
C C(3) = 102.2525300 00000350
C C(4) = .98560900 00000360
C 00000370
C MEAN ANOMALY AT EPOCH 00000380
C AN = 100.1581500 - C(3) 00000390
C BO = AN 00000400
C MEAN ANOMALY AT TIME TL 00000410
C AN = AN + C(4) * (TL - 2436935.00) 00000420
C AN = AN / 360.00 00000430
C AN = AN - IDINT(AN) 00000440
C 00000450
C C(5) = AN * 360.00 * DTR 00000460
C C(3) = C(3) * DTR 00000470
C C(4) = C(4) * DTR * UTD 00000480
C 00000490
C DUM = 23.45 * DTR 00000500
C C(6) = DCOS(DUM) 00000510
C C(7) = DSIN(DUM) 00000520
C 00000530
C CALCULATE THE ROTATION MATRIX FROM EQUATORIAL TO GMT COORD. 00000540
C 00000550
C BO = BO + (2.00 * C(2) - .7500 * C(2) ** 3) * DSIN(BO) + 1.2500 * C(2) ** 2 00000560
C 1 * DSIN(2.00 * BO) + 1.08333333333333300 * C(2) ** 3 * DSIN(3.00 * BO) 00000570

```

BO= BO+C(3)
RX= -DCOS(A0)
RY= -DSIN(A0)*C(6)
RHO= DSQRT(RX**2+RY**2)
RX= RY/RHO
RY= RY/RHO
C(8)= RX
C(9)= -RY
C(10)= 0.00
C(11)= RY
C(12)= RX
C(13)= 0.00
C(14)= 0.00
C(15)= 0.00
C(16)= 1.00

C

C(17)= 289.900*DTR
C(18)= 78.6*DTR
C(19)= 359.017041600*DTR*UTD
C(20)= 360.00*DTR
C(17)= C(17)+C(19)*(TL-2436935.00)/UTD

C

RETURN
END

00000580
00000590
00000600
00000610
00000620
00000630
00000640
00000650
00000660
00000670
00000680
00000690
00000700
00000710
00000720
00000730
00000740
00000750
00000760
00000770
00000780
00000790
00000800
00000810

```

C ORBIT
C
C THIS PROGRAM IS USED TO PRINT POSITION AND THRUST DIRECTION
C FOR A SINGLE ORBIT.
C INPUT: Z, 10 DIM O.E. AND THEIR ADJOINTS
C         A, ACCELERATION LEVEL. MAY BE SET TO ONE
C         N, ONE HALF NUMBER OF POINTS ON ORBIT PRINTED
C         NCONT, FLAG TO READ DATA FOR ANOTHER ORBIT
C
C
C         IMPLICIT REAL*(A-H,O-Z)
C         DIMENSION Z(10),H(10),G(10)
C         COMMON/A/A,AMU,PI
C         NAMELIST/IN/Z,A,N
C
C         PI= DARCOS(-1.00)
C         AMU=1.00
10      READ (5,1000) Z,A
        READ (5,1001) N
        DF= PI/DFLOAT(N)
        WRITE (6,1N)
        DO 20 I=1,N
          CALL FCT(F1,F2,Z,H,G)
          F1=F1+DF*2.00
          F2=F2+DF*2.00
20      CONTINUE
        READ (5,1001) NCONT
        IF (NCONT.GT.0) GO TO 10
        STOP
1000  FORMAT (F25.15)
1001  FORMAT (I2)
        END

```

```

00000010
00000020
00000030
00000040
00000050
00000060
00000070
00000080
00000090
00000100
00000110
00000120
00000130
)0000140
00000150
00000160
00000170
00000180
00000190
00000200
00000210
00000220
00000250
00000270
00000280
00000290
00000300
00000310
00000320
00000330
00000340
00000350
00000360

```

| | | |
|----|---|----------|
| C | FCT | 00000010 |
| C | | 00000020 |
| C | CONTAINS PRINT STATEMENTS | |
| C | | 00000050 |
| C | | 00000060 |
| C | | 00000070 |
| C | SUBROUTINE FCT(F1,F2,7,M,G) | 00000080 |
| C | | 00000092 |
| C | IMPLICIT REAL*(A-H,O-S), INTEGER (I-N) | 000000R4 |
| C | | 00000090 |
| C | NAMelist/PRFCT/VEC | 00000092 |
| C | | 00000094 |
| C | COMMON /A/A,AMU,PI | 00000100 |
| C | | 00000110 |
| C | DIMENSION Z(10),G(10),H(10),AM(5,3),PAM(5,3,5),VEC(3),X(5),PRA(5) | 00000120 |
| C | | 00000140 |
| C | M=0 | 00000150 |
| C | F=F1 | 00000160 |
| C | | 00000170 |
| C | EVALUATE M AND PARTIAL OF M WRT STATE | 00000180 |
| C | | 00000190 |
| | DO 5 I=1,5 | 00000192 |
| 5 | X(I)=Z(I) | 00000194 |
| 10 | CALL EVALMP(X,F,AMU,AM,PAM,I) | 00000200 |
| C | | 00000220 |
| C | EVALUATE THE COMMON SCALAR FACTOR | 00000230 |
| C | | 00000240 |
| | CT=DCOS(F) | 00000242 |
| | ST=DSIN(F) | 00000244 |
| | RA= 1.000-Z(3)*CT -7(2)*ST | 00000250 |
| | FAC= A*RA/(2.000*PI) | 00000260 |
| C | | 00000270 |
| C | EVALUATE THE UNIT OF M TRANSPOSE LAMBDA | 00000280 |
| C | | 00000290 |
| | ABVEC= 0.00 | 00000300 |
| | DO 30 I=1,3 | 00000310 |
| | VEC(I)=0.000 | 00000320 |
| | DO 20 J=1,5 | 00000330 |
| 20 | VEC(I)= VEC(I) + AM(J,I)*Z(J+5) | 00000340 |
| 30 | ABVEC= ABVEC + VEC(I)**2 | 00000350 |
| | ABVEC= DSQRT(ABVEC) | 00000360 |
| | DO 40 I=1,3 | 00000370 |
| 40 | VEC(I)=VEC(I)/ABVEC | 00000380 |
| C | | 00000382 |
| | WRITE (6,PRFCT) | 00000384 |
| C | | 00000386 |
| C | | 00000390 |
| C | EVALUATE THE PARTIAL OF RA WRT X | 00000391 |
| C | | 00000392 |
| | PRA(1)=0.00 | 00000393 |
| | PRA(2)=-ST | 00000394 |
| | PRA(3)=-CT | 00000395 |
| | PRA(4)=0.00 | 00000396 |
| | PRA(5)=0.00 | 00000397 |
| | ABVEC=ABVEC/RA | 00000398 |
| C | | 00000399 |
| C | EVALUATE THE FUNCTION | 00000400 |

C

```
DO 60 I=1,5
  G(I)= 0.000
  G(I+5)=0.000
  DO 50 J=1,3
    G(I)= G(I) + AM(I,J)*VEC(J)
    DO 50 L=1,5
      50   G(I+5)= G(I+5) - (L+5)*PAM(L,J,I)*VEC(J)
    G(I)= FAC*G(I)
  60   G(I+5)=FAC*(G(I+5)-ARVEC*PRA(I))
  IF (M.EQ.1) RETURN
  DO 70 I=1,10
    70   H(I)= G(I)
  F=F2
  M=1
  GO TO 10
END
```

```
00000410
00000420
00000430
00000440
00000450
00000460
00000470
00000480
00000490
00000500
00000510
00000520
00000530
00000540
00000545
00000550
00000560
```

```

C      SUBROUTINE EVALMP                                00000010
C      THIS SUBROUTINE EVALUATES THE 5X3 MATRIX M AND THE  00000020
C      5X3X5 PARTIAL OF M WRT X                          00000030
C      CONTAINS PRINT STATEMENTS
C
C      IF IMFLAG=1, BOTH M (AM) AND ITS PARTIAL (PAM) ARE EVALUATED 00000040
C      IF IMFLAG=2, ONLY M (AM) IS EVALUATED              00000050
C      IF IMFLAG=3, ONLY THE PARTIAL OF M (PAM) IS EVALUATED 00000060
C
C      SUBROUTINE EVALMPIX, THETA, AMU, AM, PAM, IMFLAG) 00000070
C      IMPLICIT REAL*(A-H,O-S), INTEGER (I-N)            00000080
C
C      NAMELIST/PRINT/ F,X1,Y1                          00000090
C
C      DIMENSION X(5), AM(5,3), PAM(5,3,5)             00000100
C      EN=DSORT(AMU/X(1)**3)                             00000110
C      P=O= DSORT(1.00- X(2)**2- X(3)**2)               00000120
C      BETA= 1.00/(1.00 +RHO)                           00000130
C      CT= DCOS(THETA)                                   00000140
C      ST= DSIN(THETA)                                   00000150
C      RA= 1.00-X(3)*CT -X(2)*ST                       00000160
C      ZETA= X(3)*ST-X(2)*CT                            00000170
C      BETA3= BETA**3/(1.00 -BETA)                      00000180
C      X1= X(1)*((1.00 -X(2)**2*BETA)*CT +X(2)*X(3)*BETA*ST -X(3)) 00000190
C      Y1= X(1)*((1.00 -X(3)**2*BETA)*ST -X(2)*X(3)*BETA*CT -X(2)) 00000200
C      X1DOT= -(1.00 -X(2)**2*BETA)*ST -X(2)*X(3)*BETA*CT *EN*X(1)/RA 00000210
C      Y1DOT= ((1.00 -X(3)**2*BETA)*CT -X(2)*X(3)*BETA*ST)*EN*X(1)/RA 00000220
C      PZ1= X(1)*(ZETA*(BETA+X(2)**2*BETA3) -(X(2)*BETA -ST)*CT/RA) 00000230
C      PZ2= -X(1)*(-ZETA*X(2)*X(3)*BETA3 +1.00 +(ST -X(2)*BETA)*ST/RA) 00000240
C      PZ3= X(1)*(-ZETA*X(2)*X(3)*BETA3-1.00 +(X(3)*BETA -CT)*CT/RA) 00000250
C      PZ4= X(1)*(-ZETA*(BETA +X(3)**2*BETA3) +(CT -X(3)*BETA)*ST/RA) 00000260
C      IF (IMFLAG .EQ. 3) GO TO 10                      00000270
C
C      F=THETA                                           00000280
C      WRITE (6,PRINT)                                   00000290
C
C      IF DO NOT WANT TO EVALUATE PART AL OF M, BRANCH TO 10 00000293
C
C      AM(1,1)= 2.00*X1DOT/(EN**2*X(1))                 00000294
C      AM(1,2)= 2.00*Y1DOT/(EN**2*X(1))                 00000296
C      AM(1,3)=0.00                                     00000300
C      DUM= RHO/(EN*X(1)**2)                             00000310
C      AM(2,1)= DUM*(PZ2- X(2)*BETA*X1DOT/EN)           00000320
C      AM(2,2)= DUM*(PZ4 -X(2)*BETA*Y1DOT/EN)           00000330
C      AM(2,3)= DUM*(X(3)*(X(5)*Y1 -X(4)*X1))/RHO**2   00000340
C      AM(3,1)= -DUM*(PZ1 +X(3)*BETA*X1DOT/EN)          00000350
C      AM(3,2)= -DUM*(PZ3 +X(3)*BETA*Y1DOT/EN)          00000360
C      AM(3,3)= -DUM*(X(2)*(X(5)*Y1 -X(4)*X1))/RHO**2   00000370
C      AM(4,1)=0.00                                     00000380
C      AM(4,2)=0.00                                     00000390
C      DUM= (1.00 +X(4)**2 +X(5)**2)/(2.00 +X(1)**2*RHO) 00000400
C      AM(4,3)= DUM*Y1                                   00000410
C      AM(5,1)=0.00                                     00000420
C      AM(5,2)=0.00                                     00000430
C      AM(5,3)= DUM*X1                                   00000440
C      IF (IMFLAG .EQ. 2) RETURN                        00000450
C      IF WE ONLY WISH TO EVALUATE M THEN PROGRAM RETURNS HERE 00000460
C
C      IF WE ONLY WISH TO EVALUATE M THEN PROGRAM RETURNS HERE 00000470
C
C      IF WE ONLY WISH TO EVALUATE M THEN PROGRAM RETURNS HERE 00000480
C
C      IF WE ONLY WISH TO EVALUATE M THEN PROGRAM RETURNS HERE 00000490

```

```

10 CA= DSQRT(AMU/X(1))/RA          00000500
P75= X(2)*BETA3                   00000510
P76= X(3)*BETA3                   00000520
P24= CA*ST/RA                      00000530
P710= CA*CT/RA                    00000540
P220= X(1)*(-2.00*X(2)*BETA*CT +X(3)*BETA*ST +P25*ZETA*X(2)) 00000550
P226= X(1)*(X(2)*BETA*ST -1.00 +P26*X(2)*ZETA) 00000560
P229= X(1)*(X(3)*BETA*CT -1.00 -P25*X(3)*ZETA) 00000570
P235= X(1)*(-2.00*X(3)*BETA*ST +X(2)*BETA*CT -P26*X(3)*ZETA) 00000580
P211= -X1DOT/(2.00*X(1))          00000590
P212= -Y1DOT/(2.00*X(1))          00000600
DUM1= 1.00 -RA                    00000610
P213= -CA*(-2.00*X(2)*BETA*ST -X(3)*BETA*CT -P25*X(2)*DUM1)+P29 00000620
      *X1DOT/CA                    00000630
1 P214= -CA*(-X(2)*BETA*CT -P26*X(2)*DUM1) +P210*X1DOT/CA 00000631
P215= -CA*(X(3)*BETA*ST +P25*X(3)*DUM1) +P29*Y1DOT/CA 00000632
P216= -CA*(2.00*X(3)*BETA*CT +X(2)*BETA*ST +P26*DUM1*X(3)) 00000633
      +P210*Y1DOT/CA              00000634
DUM= BETA +X(1)*P25               00000640
P217= 1.00+ P25*X(2)*(3.00/BETA +1.00/(1.00-BETA)) 00000650
P218= (2.00 +P217)*P25           00000660
P219= P217*P26                   00000670
DUM2= X(2)*BETA -ST              00000680
P221= -X(1)*(CT*DUM -ZETA*P218 +CT*DUM/RA +CT*ST*(DUM2/RA**2)) 00000690
P222= X(1)*(ST*DUM +ZETA*P219 -CT*X(2)*P26/RA-CT**2*DUM2/RA**2) 00000700
P223= BETA3*(3.00/BETA +1.00/(1.00 -BETA)) 00000710
P224= P223*P25                   00000720
P225= P223*P26                   00000730
P227= X(1)*(-CT*X(2)*X(3)*BETA3 +ZETA*X(3)*(BETA3 +X(2)*P224) 00000740
      +(ST*(BETA +X(2)*P25))/RA +ST**2*DUM2/RA**2) 00000750
1 P228= X(1)*(ST*X(2)*X(3)*BETA3 +ZETA*X(2)*(BETA3 +X(3)*P225) 00000760
      +X(2)*ST*P26/RA +ST*CT*DUM2/RA**2) 00000770
1 DUM2= X(3)*BETA-CT              00000780
P230= X(1)*(CT*X(2)*X(3)*BETA3 -ZETA*X(3)*(BETA3 +X(2)*P224) 00000790
      +CT*X(3)*P25/RA +CT*ST*(DUM2/RA**2)) 00000800
1 P231= X(1)*(-ST*X(2)*X(3)*BETA3 -ZETA*X(2)*(BETA3 +X(3)*P225) 00000810
      +CT*(BETA +X(3)*P26)/RA +CT**2*DUM2/RA**2) 00000820
1 DUM= BETA +X(3)*P26             00000830
P232= 1.00 +P26*X(3)*(3.00/BETA +1.00/(1.00 -BETA)) 00000840
P233= P232*P25                   00000850
P234= P232*P26 +2.00*X(3)*BETA3 00000860
P236= X(1)*(CT*DUM -ZETA*P233 -ST*X(3)*P25/RA -ST**2*DUM2/RA**2) 00000870
P237= X(1)*(-ST*DUM -ZETA*P234 -ST*(BETA +X(3)*P26)/RA -ST*CT 00000880
      *DUM2/RA**2) 00000890
1 DO 20 J=1,2                    00000900
20 PAM(1,J,1)= 3.00*AM(1,J)/(2.00*X(1)) 00000910
DUM =2.00*X(1)**2/AMU             00000920
PAM(1,1,2)= P213*DUM             00000930
PAM(1,1,3)= P214*DUM             00000940
PAM(1,2,2)= P215*DUM             00000945
PAM(1,2,3)=P216*DUM             00000950
DUM= DSQRT(A J*X(1))             00000955
CB=RHO/DUM                        00000960
P238= -X(2)*CB/RHO**2            00000970
P239= -X(3)*CB/RHO**2            00000980
PAM(2,1,1)= AM(2,1)/(2.00*X(1)) 00000990
PAM(2,1,2)= -CB*BETA*X1DOT/EN +P238*AM(2,1)/CR +CB*(P227 00001000

```



```

1      -X(2)*HETA*P713/EN -X(2)*X1DOT*P25/EN          00001010
PAM(2,1,3)= P239*AM(2,1)/CR +CR*(P724 -P76*X(2)*X1DOT/EN 00001020
1      -X(2)*HETA*P714/EN          00001030
PAM(2,2,1)= AM(2,2)/(2.00*X(1))          00001040
PAM(2,2,2)= P238*AM(2,2)/CR +CR*(P236 -BETA*Y1DOT/EN -X(2) 00001050
1      *Y1DOT*P75/EN -X(2)*BETA*P715/EN          00001060
PAM(2,2,3)= P239*AM(2,2)/CR +CR*(P737 -X(2)*Y1DOT*P26/EN 00001070
1      -X(2)*HETA*P716/EN          00001080
PAM(2,3,1)= AM(2,3)/(2.00*X(1))          00001090
DUM1= X(5)*Y1 -X(4)*X1          00001100
PAM(2,3,2)= X(3)*(X(5)*P229 -X(4)*P220)/(RHO*DUM1) +X(2)*X(3) 00001120
1      *DUM1/(RHO**3*DUM1)          00001130
PAM(2,3,3)= DUM1/(RHO*DUM1) +X(3)*(X(5)*P235 -X(4)*P226)/(RHO 00001140
1      *DUM1) +X(3)**2*DUM1/(RHO**3*DUM1)          00001150
PAM(2,3,4)= -X(3)*X1/(RHO*DUM1)          00001160
PAM(2,3,5)= X(3)*Y1/(RHO*DUM1)          00001170
PAM(3,1,1)= AM(3,1)/(2.00*X(1))          00001180
PAM(3,1,2)= P238*AM(3,1)/CB -CR*(P221 +X(3)*X1DOT*P25/EN 00001190
1      +X(3)*BETA*P213/EN          00001200
PAM(3,1,3)= P239*AM(3,1)/CB -CR*(P222 +(BETA*X1DOT +X(3) 00001210
1      *X1DOT*P26 +X(3)*BETA*P214)/EN          00001220
PAM(3,2,1)= AM(3,2)/(2.00*X(1))          00001230
PAM(3,2,2)= P238*AM(3,2)/CB -CB*(P230 +X(3)*(Y1DOT*P25 00001240
1      +BETA*P215)/EN          00001250
PAM(3,2,3)= P239*AM(3,2)/CB -CB*(P231 +(BETA*Y1DOT +X(3) 00001260
1      *Y1DOT*P26 +X(3)*BETA*P216)/EN          00001270
PAM(3,3,1)= AM(3,3)/(2.00*X(1))          00001280
PAM(3,3,2)= -DUM1/(RHO*DUM1) -X(2)*(X(5)*P229 -X(4)*P220)/ 00001290
1      (RHO*DUM1) -X(2)**2*DUM1/(RHO**3*DUM1)          00001300
PAM(3,3,3)= -X(2)*(X(5)*P235 -X(4)*P226)/(RHO*DUM1) -X(2)*X(3) 00001310
1      *DUM1/(RHO**3*DUM1)          00001320
PAM(3,3,4)= X(2)*X1/(RHO*DUM1)          00001330
PAM(3,3,5)= -X(2)*Y1/(RHO*DUM1)          00001340
Z5= (1.00 +X(5)**2 +X(4)**2)/(2.00*DUM*RHO) 00001350
PZ40= -Z5/(2.00*X(1))          00001360
PZ41= X(2)*Z5/RHO**2          00001370
PZ42= X(3)*Z5/RHO**2          00001380
PZ43= X(4)/(DUM*RHO)          00001390
PZ44= X(5)/(DUM*RHO)          00001400
PAM(4,3,1)= AM(4,3)/(2.00*X(1))          00001410
PAM(4,3,2)= PZ41*Y1+Z5*P229          00001420
PAM(4,3,3)= PZ42*Y1 +Z5*P235          00001430
PAM(4,3,4)= PZ43*Y1          00001440
PAM(4,3,5)= PZ44*Y1          00001450
PAM(5,3,1)= AM(5,3)/(2.00*X(1))          00001460
PAM(5,3,2)= PZ41*X1 +Z5*P229          00001470
PAM(5,3,3)= PZ42*X1 +Z5*P235          00001480
PAM(5,3,4)= PZ43*X1          00001490
PAM(5,3,5)= PZ44*X1          00001500
DO 30 K=1,5          00001510
PAM(1,3,K)=0.00          00001520
DO 30 I=4,5          00001530
DO 30 J=1,2          00001540
30 PAM(I,J,K)=0.00          00001550
DO 40 I=1,3          00001560
DO 40 J=1,2          00001570
DO 40 K=4,5          00001580
40 PAM(I,J,K)=0.00          00001590
RETURN          00001600
END          00001610

```