Willard D. Bleick

ORBITAL TRANSFER IN MINIMUM TIME
ORBITAL TRANSFER IN MINIMUM TIME

by

Willard E. Bleick

Professor of Mathematics and Mechanics

RESEARCH PAPER NO. 34
August 1962
ORBITAL TRANSFER IN MINIMUM TIME

by

WILLARD E. BLEICK
Professor of Mathematics and Mechanics

Research Paper No. 34
UNITED STATES NAVAL POSTGRADUATE SCHOOL
Monterey, California
August 1962
ORBITAL TRANSFER IN MINIMUM TIME

W. E. BLEICK,* U. S. Naval Postgraduate School

1. Introduction. The problem of orbital transfer discussed here is that of scheduling the direction $\mathbf{p}$ of constant momentum thrust of a rocket, which loses mass at a constant rate, so that it transfers to an earth satellite orbit, with known elements of time, position and velocity, in a minimum time $T$ after launching of the rocket. The launching conditions are assumed to be fixed. This situation is illustrated in Figure 1 for the case of a circular orbit. The sector angle $B$ at which the rocket enters orbit will be called the rendezvous angle. To aid the discussion imaginary physical rendezvous of the rocket and satellite is assumed to occur at this angle. The time of rocket launch to achieve actual physical rendezvous can be determined, of course, only after both of the unknowns $T$ and $B$ have been found. The problem is set up as a calculus of variations problem of the Lagrange type, and is solved by an iterative process in which an initial approximation to the angle $B$ is estimated.

A non-rotating Oxy rectangular coordinate system with origin at the earth's center is used. The coordinates and velocity components of the rocket and target satellite are denoted by $x, y, u, v$ and $X, Y, U, V$ respectively. For simplicity the equations of motion of rocket and target will be written in a "non-dimensional" form by the use of suitable units. The unit of length is taken as the

---

* Supported by the Office of Naval Research.
Fig. 1. Orbital transfer.
earth's equatorial radius, $R_e = 20,925,000$ feet. The unit taken for time $t$ is the time required by a hypothetical earth satellite, in equatorial, circular, vacuum, sea level orbit, to traverse a sector of one radian. This unit of time is $\sqrt{R_e/g} = 13.459$ minutes, where $g = 32.086$ ft./sec.$^2$ is the acceleration of gravity at the equator. The unit of velocity is then the speed of this hypothetical satellite. These units of length and time will always be understood, unless other, more conventional units are specifically mentioned.

2. Statement of the problem. The equations of motion of the rocket, in terms of the specified units of length and time, are

\begin{align*}
\varphi_1 &= \dot{u} - g_1 - a \cos p = 0 \\
\varphi_2 &= \dot{v} - g_2 - a \sin p = 0 \\
\varphi_3 &= \dot{x} - u = 0 \\
\varphi_4 &= \dot{y} - v = 0
\end{align*}

where $g_1 = -x/r^3$, $g_2 = -y/r^3$, $r^2 = x^2 + y^2$, $a = c\dot{m}/g(1 - \dot{m}t)$, where $\dot{m}$ is the constant fraction of initial gross rocket mass lost per unit of time, $c$ is the constant speed of the emitted rocket gases, and $g$ is the acceleration of gravity at the equator. The fixed initial conditions of the rocket trajectory are taken as

\begin{align*}
x(0) &= 0, \quad u(0) = V_1 = V_0 \cos \theta \\
y(0) &= 1, \quad v(0) = V_2 = V_0 \sin \theta.
\end{align*}
The terminal point of the rocket trajectory is variable with

\[ x(T) = X(T), \quad y(T) = Y(T), \quad u(T) = U(T), \quad v(T) = V(T). \]

It is also assumed that the rocket thrust is turned off abruptly at time \( T \). This discontinuity will lead to a trivial steering corner in the calculus of variations problem. Note that (1) and (3) imply that

\[ \dot{U}(T) = -\left[ X/R^3 \right]_T = g_1(T) \quad \text{and} \quad \dot{V}(T) = -\left[ Y/R^3 \right]_T = g_2(T) \]

where \( R = \sqrt{X^2 + Y^2} \).

The problem is to choose the control variable \( p \) to effect orbital transfer with \( \int_0^T dt \) minimized, and to determine the corresponding rocket trajectory. This problem is equivalent to the Lagrange calculus of variations problem of requiring the integral

\[ I = \int_0^T \left( 1 + \lambda \varphi_1 + \mu \varphi_2 + \pi \varphi_3 + \rho \varphi_4 \right) dt \]

to be stationary. In (4) the equations (1) are regarded as constraints with \( \lambda(t), \mu(t), \pi(t), \rho(t) \) introduced as continua of Lagrangian multipliers [1]. If the time coordinate of the varied terminal point is taken as \( T + \Delta T \), the vanishing first variation [2] of \( I \) is

\[ \delta I = \int_0^T \left[ \lambda(\delta \dot{u} - g_{1x} \delta x - g_{1y} \delta y + a \sin p \delta p) + \pi(\delta \dot{x} - \delta u) \right. \\
+ \mu(\delta \dot{v} - g_{2x} \delta x - g_{2y} \delta y - a \cos p \delta p) + \rho(\delta \dot{v} - \delta v) \left. \right] dt \]

\[ + \int_{T+\Delta T}^{T} \left[ 1 + \lambda \delta \dot{u} + \mu \delta \dot{v} - (\lambda \cos p + \mu \sin p) \delta a \right] dt = 0 \]
where the finite variation \(\delta a, \delta \hat{u}, \delta \hat{v}\) terms in the integral from \(T\) to \(T + \Delta T\), created by thrust termination at time \(T\) on the unvaried trajectory and at \(T + \Delta T\) on the varied trajectory, cancel.

On integrating by parts one obtains

\[
(6) \quad \delta I = [\lambda \delta u + \mu \delta v + \pi \delta x + \rho \delta y]_T - \int_0^T [(\dot{\lambda} + \pi) \delta u + (\dot{\mu} + \rho) \delta v
\]

\[
+ (\dot{\pi} + g_{1x} \lambda + g_{2x} \mu) \delta x + (\dot{\rho} + g_{1y} \lambda + g_{2y} \mu) \delta y
\]

\[
+ a(\mu \cos p - \lambda \sin p) \delta p] dt + \Delta T = 0.
\]

The variations of the dependent coordinates at the variable terminal point must be taken as

\[
(7) \quad \delta u(T) = (\dot{U} - \dot{u})_T \Delta T = -(a \cos p)_T \Delta T, \quad \delta x(T) = (\dot{X} - \dot{x})_T \Delta T = 0
\]

\[
\delta v(T) = (\dot{V} - \dot{v})_T \Delta T = -(a \sin p)_T \Delta T, \quad \delta y(T) = (\dot{Y} - \dot{y})_T \Delta T = 0.
\]

Substitution of (7) into (6), and application of the fundamental lemma of the calculus of variations and the theory of ordinary extrema, gives the Euler equations

\[
(8) \quad \dot{\lambda} + \pi = 0
\]

\[
\dot{\mu} + \rho = 0
\]

\[
\dot{\pi} + g_{1x} \lambda + g_{2x} \mu = 0
\]

\[
\dot{\rho} + g_{1y} \lambda + g_{2y} \mu = 0
\]

\[
\tan p = \mu/\lambda
\]

and the transversality condition

\[
[a(\lambda \cos p + \mu \sin p)]_T = 1.
\]
The first four homogeneous equations of the Euler equations (8) constitute the adjoint system [3] of the system of variation equations

\[ \delta \varphi_1 = \delta \varphi_2 = \delta \varphi_3 = \delta \varphi_4 = 0 \]

which are the coefficients of $\lambda$, $\mu$, $\pi$, $p$ in (5) equated to zero. The adjoint system has a matrix of coefficients which is the negative transpose of that of (10). The last of the Euler equations (8) requires that the control variable $p$ be adjusted so that $\vec{a} = a(\vec{i} \cos p + \vec{j} \sin p)$, which is proportional to the rocket thrust, is continually parallel to the adjoint vector $\vec{\lambda} = \vec{i} \lambda + \vec{j} \mu$. The transversality condition (9), which may be written

\[ (\vec{a} \cdot \vec{\lambda})_T = 1 > 0, \]

requires that $\vec{a}$ and $\vec{\lambda}$ have the same sense. Since it is only the ratio of $\mu$ to $\lambda$ which determines $p$, it is a trivial matter to scale them to satisfy the magnitude requirement of (9).

A solution of the problem obtained from (1) and (8) guarantees a stationary time of transfer. The nature of the problem is such that this stationary time is a minimum time.

3. Numerical solution. There is a constructive aspect of a modification of equation (6), first used by Bliss [4] in his work on differential corrections in ballistics, and applied recently by Faulkner [5] in an iterative fashion in optimum control problems. To find the desired modification of (6), assume that a solution of the systems (1) and (8) has been obtained, which does not necessarily satisfy the terminal conditions (3). Using this
solution and holding $T$ constant, consider the variation of the vanishing integral

$$
\int_0^T (\lambda \varphi_1 + \mu \varphi_2 + \pi \varphi_3 + \rho \varphi_4) \, dt = 0
$$

with the terminal constraints (3) removed, so that the terminal variations $\delta u(T), \delta v(T), \delta x(T)$ and $\delta(T)$ become free. Since $\lambda, \mu, \pi, \rho$ satisfy the adjoint system, and since there is now no steering corner due to thrust termination, one obtains

$$
[\lambda \delta u + \mu \delta v + \pi \delta x + \rho \delta y]_T = \int_0^T a(-\lambda \sin \rho + \mu \cos \rho) \delta \rho \, dt
$$

$$
= \int_0^T \vec{\lambda} \cdot (\delta \vec{a} / \delta \rho) \delta \rho \, dt
$$

where $\vec{a} = a(\cos \rho + j \sin \rho)$. Equation (12), which is the desired modification of (6), is called the fundamental formula by Bliss [4], but is also known under the generic name of Green's formula [3]. By the use of (12) it is possible to generate the control parameters of a varied trajectory which, hopefully, comes closer to satisfying the desired terminal conditions (3). To do this, assume that the adjoint system has been solved to obtain a fundamental set of four linearly independent solutions given by the rows of

$$
\text{Transpose of } B(t) = [\lambda_i(t) \mu_i(t) \pi_i(t) \rho_i(t)] \quad i = 1, 2, 3, 4
$$

where $B(0) = I$ is the identity matrix. The solution $\vec{\lambda} = \vec{I} \lambda + \vec{J} \mu$ of the adjoint system, required to satisfy the last of the Euler
equations (8), is taken as the linear combination

\begin{align*}
\lambda &= \lambda_1 + l\lambda_2 + m\lambda_3 + n\lambda_4 \\
\mu &= \mu_1 + l\mu_2 + m\mu_3 + n\mu_4
\end{align*}

so that the control angle \(\varphi\) is determined by

\begin{equation}
\tan \varphi = (\mu_1 + l\mu_2 + m\mu_3 + n\mu_4)/(\lambda_1 + l\lambda_2 + m\lambda_3 + n\lambda_4)
\end{equation}

and its variation by

\begin{equation}
\delta \varphi = \frac{[(\lambda\mu_2 - \mu\lambda_2)\delta l + (\lambda\mu_3 - \mu\lambda_3)\delta m + (\lambda\mu_4 - \mu\lambda_4)\delta n]/(\lambda^2 + \mu^2)}
\end{equation}

When (13) and (16) are substituted into (12) there results the system of Green's formulae

\begin{equation}
[\delta u \, \delta v \, \delta x \, \delta y]_T B(T) = [0 \, 0 \, 0 \, 0]
\end{equation}

where the elements of the matrix \(A\) are

\begin{equation}
a_{ij} = \int_0^T a(\lambda\mu_i - \mu\lambda_i)(\lambda\mu_j - \mu\lambda_j)dt/(\lambda^2 + \mu^2)^{3/2}.
\end{equation}

The coordinates of the terminal point of the varied trajectory at time \(T + \Delta T\) may be taken as \([u + \Delta u \, v + \Delta v \, x + \Delta x \, y + \Delta y]_T\) where

\begin{equation}
[\Delta u \, \Delta v \, \Delta x \, \Delta y]_T = [\delta u \, \delta v \, \delta x \, \delta y]_T + [\dot{u} \, \dot{v} \, \dot{x} \, \dot{y}]_T \Delta T.
\end{equation}

In an effort to make this new terminal point come closer to satisfying the terminal conditions (3), one may take

\begin{equation}
[\Delta u \, \Delta v \, \Delta x \, \Delta y]_T = [U - u \, V - v \, X - x \, Y - y]_T + [\dot{U} \, \dot{V} \, \dot{X} \, \dot{Y}]_T \Delta T.
\end{equation}
Substitute (19) and (20) into (17) to obtain

\[(21) \left[\dot{U} - \dot{u} \dot{V} - \dot{v} \dot{X} - \dot{x} \dot{Y} - \dot{y}\right]_T \Delta T + [0 \delta l \delta m \delta n]AB^{-1}(T) = [U - u \ V - v \ X - x \ Y - y]_T\]

as the system of equations for the determination of \(\Delta T, \delta l, \delta m, \delta n\) on the varied trajectory. The Faulkner [5] scheme for the numerical solution of optimum control problems may now be stated: Make an initial guess for the values of \(T, l, m, n\); carry out a simultaneous numerical integration of the systems (1), (8) and (18) using the control variable of (15); solve the system (21) for \(\Delta T\) and the changes in the control parameters; iterate until convergence is obtained. This program may be carried out in a matter of seconds on modern digital computers.

To give an example of this control optimization in the present problem a circular satellite orbit was assumed of radius \(R = \|\bar{r} X + \bar{j} Y\| = 1.075699\), corresponding to an altitude of 300 statute miles above sea level. For this orbit \(\|\bar{r} U + \bar{j} V\| = 1/\sqrt{R}\) and \(\|\bar{r} \dot{U} + \bar{j} \dot{V}\| = 1/R^2\). The assumed rocket launching velocity was \(V_0 = |\bar{r} V_1 + \bar{j} V_2| = 0.585402\), launch angle \(\theta = 0.928084\) and rendezvous sector \(B = 0.153840\) as in Figure 1. Also assumed were \(c = 10000.9 \text{ ft./sec.}\) and \(\dot{m} = 0.00360583 \text{ sec}^{-1}\). The odd appearance of these figures is related to the difficulty, explained in the next section, of obtaining the initial "guesses" \(T = 0.289725\), \(l = -0.223125\), \(m = -29.9875\) and \(n = 19.0847\). With this input the process converged in four iterations to the seven significant
Fig. 2. Trajectory and thrust directions.
figure results $T = 0.2894592, l = 0.1840054, m = -108.94383, n = 67.95886$ and $B = 0.1536015$. Figure 2 shows the resulting trajectory and rocket thrust directions for equal time intervals, excepting the interval terminating in transfer, which is reduced by one half.

4. The initial guesses. The domain of convergence of the iteration scheme of the last section appears to be rather limited in the present control problem, requiring initial guesses for $T, l, m, n$ which make $[U-u \ V-v \ X-x \ Y-y]_T$ small in (21). In the numerical example given here, where the angle $B$ is small and radius $R$ nearly unity, the equations (1) can be linearized and an exact solution of the linearized transfer problem used to supply the input guesses required to solve the non-linear problem. The gravitational terms $g_1$ and $g_2$ of (1) were replaced by the linear terms in their Taylor expansions at $x = 0, y = 1$. The systems (1) and (8) then become uncoupled. Their solutions, using (2), are

\begin{align}
    x &= \int_0^t a \cos p \sin (t-w)dw + V_1 \sin t \\
    u &= \int_0^t a \cos p \cos (t-w)dw + V_1 \cos t \\
    \sqrt{2}y &= \int_0^t a \sin p \sinh \sqrt{2}(t-w)dw + V_2 \sinh \sqrt{2}t - (1/\sqrt{2}) \cosh \sqrt{2}t + 3/\sqrt{2} \\
    v &= \int_0^t a \sin p \cosh \sqrt{2}(t-w)dw + V_2 \cosh \sqrt{2}t - (1/\sqrt{2}) \sinh \sqrt{2}t \\
    \tan p &= \left[ 1 \cosh \sqrt{2}t - (n/\sqrt{2}) \sinh \sqrt{2}t \right]/\left( \cos t - m \sin t \right).
\end{align}
It will be of no avail to attempt to solve (22) and (3) for \( T, l, m, n \) by the Newton-Raphson method, since the Newton-Raphson equations are in fact the system (21) of the poorly convergent iterative routine of the last section. A substitute procedure of solving (22) and (3) for \( V_1, V_2, \dot{m}, B \) was used. When the transfer problem has been solved for a sufficient number of such sets \((V_1, V_2, \dot{m}, B)\), a basis will be at hand for obtaining desired rocket launching conditions by interpolation or extrapolation.

It can now be revealed that the numerical example of the last section really had its genesis in the assumptions \( B = \pi/16, \theta = \pi/4, V_0 = 0.65, \dot{m} = 0.0025 \text{ sec.}^{-1} \) and \( c = 10000 \text{ ft./sec.} \). It was estimated that \( T = 0.3, l = 1.25, m = -1.0 \) and \( n = 9.5 \) would satisfy (22) and (3). Using a cluster of five closely spaced points around \((T, l, m, n)\), a single application of regula falsi [6] was made in the hope of improving these values of \( T, l, m, n \). The particular cluster chosen gave the output values \( T = 0.289725, l = -0.223125, m = -29.9875, n = 19.0847 \), which will be recognized as the initial guesses of the last section. The output residuals were 
\[
\begin{bmatrix}
X-x, Y-y, U-u, V-v
\end{bmatrix}_T = [0.041389, -0.020160, 0.212093, -0.221912].
\]
These residuals were then processed by the linear system
\[
\begin{align*}
(X-x)_T &= \Delta V_1 \sin T - Y(T)dB + \Delta c \int_0^T a \cos p \sin (T-w)dw/c \\
(U-u)_T &= \Delta V_1 \cos T - V(T)dB + \Delta c \int_0^T a \cos p \cos (T-w)dw/c \\
\sqrt{2}(Y-y)_T &= \Delta V_2 \sinh \sqrt{2}T + \sqrt{2}X(T)dB + \Delta c \int_0^T a \sin p \sinh \sqrt{2}(T-w)dw/c \\
(V-v)_T &= \Delta V_2 \cosh \sqrt{2}T + U(T)dB + \Delta c \int_0^T a \sin p \cosh \sqrt{2}(T-w)dw/c,
\end{align*}
\]
derived from (22) and (3) for the case of a circular orbit, to obtain \( c + \Delta c \), launching conditions \( V_1 + \Delta V_1 \) and \( V_2 + \Delta V_2 \), and rendezvous sector \( B + dB \) which will give small residuals. The value of \( c + \Delta c \) turned out to be far from the desired 10000 ft./sec. Instead of changing the value of \( c \), the transformation

\[
10000 \ln \left[ 1 - \left( \dot{m} + \Delta \dot{m} \right) T \right] = (c + \Delta c) \ln (1 - \dot{m} T),
\]

which leaves the integral \( \int_0^T \dot{a} dt \) invariant, was used to change \( \dot{m} \) and cause \( c + \Delta c \) to approach 10000 ft./sec. on iterating (24), (22) and (23). Seven iterations produced the input values of \( V_1 \), \( V_2 \), \( c \), \( \dot{m} \) and \( B \) used in the last section.

5. The problem of two fixed end points. The problem in which the conditions at both ends of the rocket trajectory are fixed, or the problem in which rocket and target satellite achieve actual physical rendezvous at a specified angle \( B \), can also be solved by the methods presented here, but with greater convergence difficulties.

Acknowledgement. I am indebted to my colleague, Prof. Frank D. Faulkner, for suggesting the problem of this paper and for discussions of difficulties encountered.

References


CDC-1604 ASSEMBLY LANGUAGE COMPUTER PROGRAM

Set content of K100+1 = SLJ 0 L3440 ZRO 0 00000

REM COMPUTE LINEARIZED TRAJECTORY
REM (B1) = NUMBER OF TRAJECTORIES
REM (B4) = NUMBER OF FALSI ITERATIONS LESS ONE
REM (A) = FIRST DEPARTURE ANGLE
REM (Q) = DELTA DEPARTURE ANGLE

K0
STA 1 INT
STA 0 THETATEM
STQ 0 DELTHETA
LDA 0 UNITY
FDV 0 R
SLJ 4 SQROOT
STA 0 V
FMU 0 V
FDV 0 R
STA 0 VSQDR
LDA 0 B
SLJ 4 TRIG+70
STA 0 COSB
FMU 0 R
STA 0 CAPYFIN
LDA 0 B
SIU 4 K20+6
SLJ 4 TRIG
STA 0 SINB
FMU 0 R
STA 0 CAPXFIN
LDA 0 COSB
FMU 0 V
STA 0 CAPUFIN
LAC 0 SINB
FMU 0 V
STA 0 CAPYFIN
LAC 0 COSB
FMU 0 VSQDR
STA 0 CAPYDFN
LAC 0 SINB
FMU 0 VSQDR
STA 0 CAPUDFN
LDA 0 REARTH
FDV 0 GACCEL
SLJ 4 SQROOT
STA 0 OMEGA
LDA 0 C
FMU 0 MDOT
FDV 0 GACCEL
STA 0 A
LDA 0 THETATEM
STA 0 THEETA
ENI 4 0
SLJ 4 TRIG+70
FMU 0 VSTART
STA 0 V1
LDA 0 THEETA
SLJ 4 TRIG
FMU 0 VSTART
STA 0 V2
ENI 3 23
ENI 0 0
LDA 3 ELINIT
STA 3 EL1
IJP 3 -1
SLJ 4 00007
STA 0 EL1INIT

- 13 -
K40
ZRO 0 T5INIT
ENI 5 0 SET B5=0 FOR K40+3
SLJ 4 700C7 PRINT COORDINATES OCTAL
ZRO 0 EL1INIT PARAMETER WORD
ZRO 0 T5INIT SET FALSE FUNC. QUINTUPLE
SLJ 4 XYUV PRINT FUNCTIONS DECIMAL
ZRO 0 SLJ 4 70007
ZRO 0 STA 0 DELXFIN PARAMETER WORD
ZRO 0 DELYFIN
LDA 0 DELXFIN
STA 5 DELX1
LDA 0 DELYFIN
STA 5 DELY1
LDA 0 DELUFIN
STA 5 DELU1
LDA 0 DELVFIN
STA 5 DELV1
INI 5 5
INI 3 3
IKS 3 23
SLJ 0 K40
ENI 0 0 BEGIN INNER LOOP OR SLJ 0 K101 OR L3440
SLJ 4 MATRIX INVERT DELXYUV MATRIX
SLS 0 K50+2 SINGULAR MATRIX ALARM HALT
ZRO 0 0
ZRO 0 0 CODE TO INVERT = 5
ZRO 0 5 L = 5 ROWS IN DELXYUV MATRIX
ZRO 0 DELX1-1 ADD. OF DELXYUV MATRIX = DELX1-1
ZRO 0 0 N=0. WORD NOT USED
ZRO 0 0
ZRO 0 0 M=0
ZRO 0 DELX1+174 ADD. OF INVERSE = DELX1+174
ENI 0 0
SLJ 4 MATRIX INVERSE TIMES LMNT MATRIX
ZRO 0 K60 WORD NOT USED
ZRO 0 0
ZRO 0 0
ZRO 0 4 MATRIX MULT. CODE = 4
ZRO 0 5 L = 5 ROWS IN INVERSE
ZRO 0 DELX1+174 ADD. OF INVERSE = DELX1+174
ZRO 0 5 N=5 COLUMNS IN INVERSE
ZRO 0 EL1 ADDRESS OF LMNT MATRIX = EL1
ZRO 0 4 M=4 COLS. IN MATRIX PRODUCT
ZRO 0 EL1+24 ADDRESS OF PROD. = EL1+24
LDA 3 EL2 SHIFT COORS. AND FUNCS. B3=0
STA 3 EL1
LDA 3 DELX2-1
STA 3 DELX1-1
ISK 3 23
SLJ 0 /-2
ENI 3 3 FOR USE IN K70+2
SLJ 4 XYUV COMPUTE NEW DELXYUV FUNCS.
ZRO 0 0
LDA 5 DELXFIN STORE NEW FUNCS.
STA 5 DELX5
ISP 5 /-1
SLJ 4 70007 PRINT NEW COORS. DEC. WITH PANEL
SAU 0 EL5 PARAMETER WORD
ZRO 0 T5
EXF 0 70 CLEAR ARITHMETIC ERRORS
SLJ 4 70007 PRINT NEW COORS. OCTAL
ZRO 0 EL5 PARAMETER WORD
ZRO 0 T5
ENI 3 0 CLEAR B3
SLJ 4 70007 PRINT NEW FUNCS. DECIMAL
K100 STA 0 DELXFIN PARAMETER WORD

- 14 -
ZRO 0 DELVFIN
IJP 4 K50+1 END INNER LOOP ( OR SLJ 0 L3440 )
LDQ 0 ELS
LDA 0 UNITY
SLJ 4 POLAR+130 DEPARTURE THRUST ANGLE
STQ 0 QQ MOVE ELS5, EM5 AND ENS5
ENI 1 2
LDA 1 ELS
STA 1 EL
IJP 1 1/-1
LAC 0 V
FMU 0 TAUFIN (-V)
FDV 0 R
FAD 0 R (VT/R)
STA 0 S
STA 0 NT PARAMETER WORD
ZRO 0 S
SLJ 4 70007 S = INITIAL TARGET SECTOR
ZRO 0 0 OCTAL DUMP
ZRO 0 LAM DECEIMAL DUMP
ZRO 0 0
STA 0 NT PARAMETER WORD
ZRO 0 S
SLJ 4 70007 OCTAL DUMP
ZRO 0 0
ZRO 0 NT
ZRO 0 DELTHETA
LDA 0 THEETA
FAD 0 DELTHETA
STA 0 THEETATEM SET NEW THETA
RSO 0 NT
K110
AJP 1 K20+5 END OUTER LOOP. (SEE K440)
SLS 0 BEGIN HALT, START RENDEVOUS PROG. 
XYUV
SLJ 0 0 ENTER DELXYUV SUBROUTINE
LDA 3 T1
STA 0 TAUFIN
LAC 0 OMEGA
FMU 0 TAUFIN (-W)
FAD 0 UNITY
STA 0 STOR1 (STOR1) = 1 - WT
FDV 0 OMEGA
STA 0 STOR2 (STOR2) = (1-WT)/W
FMU 0 STOR2
STA 0 STOR2 SQUARE
FMU 0 STOR2 CUBE
FMU 0 STOR2 FOURTH
FMU 0 STOR2 FIFTH
FMU 0 ERRRCOF MULT. BY -15.B-12
K120
FDV 0 TAUFIN
STA 0 STOR3
LDA 0 STOR1
SLJ 4 LOG+66 1-WT
FMU 0 STOR3 
SLJ 4 SQROOT
SLJ 4 SQROOT
ZRO 0 0
STA 0 DELTAU
STA 0 DELTI
STA 0 DELTI
STA 0 RUNKT1
STA 0 RUNKT1
STA 1 RUNKT1
CLEAR RUNKT1 AREA
IJP 1 1/-1
ENI 0 0
LDA 0 TAUFIN
SLJ 4 TRIG
STA 0 SINT
FMU 0 V1
FSB 0 CAPXFINT V1 SINT
ENI 0 0
LDA 0 TAUFIN SINT
| STA 0 | DELXFIN        | COST T |
| LDA 0 | TAUFIN        | V1 COST |
| SLJ 4 | TRIG+70       | V1 COST - U |
| STA 0 | COST          | T TIMES SQ. ROOT 2 |
| FMU 0 | V1           | STORE HYP. COSINE |
| FSB 0 | CAPUFIN      | COSHTR2/SQ.ROOT 2 |
| STA 0 | DELUFIN      | |
| LDA 0 | TAUFIN        | |
| FMU 0 | ROOT2         | |
| SLJ 4 | EXP+107       | |
| ZR0 0 | 0             | |
| STA 0 | COSHTR2      | STORE HYP. SINE |
| FMU 0 | ROOPTHALF    | V2 SINHTR2 |
| STA 0 | STOR1        | ADD (3/SQ.ROOT 2) |
| LDA 0 | TAUFIN        | (-Y) |
| FMU 0 | ROOT2         | |
| SLJ 4 | EXP+71        | |
| STA 0 | SINHTR2      | (STOR1) = V2 COSHTR2 |
| FMU 0 | V2           | (-SINHTR2) |
| FSB 0 | STOR1        | |
| FAD 0 | THREDR2      | |
| STA 0 | STOR1        | |
| LAC 0 | CAPFIN       | EL1 SQUARED |
| FMU 0 | ROOT2         | 1 + EL1 SQ. |
| LDA 0 | 0             | |
| STA 0 | HYP           | |
| LDA 0 | A             | |
| FDV 0 | HYP           | |
| STA 0 | STOR1        | (STOR1) = A/HYP = A COS P |
| FMU 0 | SINT          | |
| STA 0 | SINDOT       | A COSP SINT |
| LDA 0 | STOR1        | A COSP |
| FMU 0 | COST          | |
| STA 0 | COSDOT       | A COSP COST |
| LDA 0 | STOR1        | A/HYP |
| FMU 0 | 3 EL1        | A SINP |
| LDA 0 | 0             | |
| STA 0 | HYP           | |
| LDA 0 | A             | |
| STA 0 | STOR1        | (STOR1) = A/HYP = A COS P |
| FAD 0 | 0             | |
| STA 0 | SINDOT       | |
| LDA 0 | STOR1        | |
| FMU 0 | 0             | |
| STA 0 | COSHTR2      | |
| LDA 0 | STOR1        | |
| FMU 0 | COSHDOT      | |
| SLJ 4 | ALPHA+1      | |
| ZR0 0 | RUNKT1       | |
| ZR0 0 | DERIVI       | SET UP GILL ROUTINE |
| TJP 2 | +/-1         | |
| SLJ 0 | +/-3         | |
| ENI 0 | 0            | |
| SLJ 4 | ALPHA+1      | |
| ENI 0 | 0            | |
| SLJ 0 | +/-2         | |
| LDA 0 | DELXFIN      | |
| FAD 0 | SIN          | |
| STA 0 | DELXFIN      | |
| LDA 0 | DELUFIN      | |
| FAD 0 | COS          | |

- 16 -
K210  
STA 0 DELUFIN  
LDA 0 DELYFIN  
FAD 0 SINTH  
FDV 0 ROOT2  
STA 0 DELYFIN  
LDA 0 DELVFIN  
FAD 0 COSH  
STA 0 DELVFIN  
SLJ 0 XUVY  
TO EXIT OF SUBROUTINE  
DERIV1  
LAC 0 TAU1  
FMU 0 OMEGA  
FAD 0 UNITY  
STA 0 UNMINWT  
LDA 0 A  
FDV 0 UNMINWT  
STA 0 ADOMINWT  
LDA 0 TAUFIN  
FSB 0 TAU1  
STA 0 STOR1  
(STER1) = TAUFIN - TAU1  
K220  
ENI 0 0  
SLJ 4 TRIG  
STA 0 SINT  
LDA 0 STOR1  
SLJ 4 TRIG+70  
ZRO 0 0  
STA 0 COST  
LDA 0 STOR1  
FMU 0 ROOT2  
SLJ 4 EXP+71  
STA 0 SINHTR2  
LDA 0 STOR1  
FMU 0 ROOT2  
SLJ 4 EXP+107  
STA 0 COSHTR2  
LDA 0 TAU1  
K230  
SLJ 4 TRIG  
ZRO 0 0  
FMU 3 EM1  
STA 0 STOR2  
(LSTOR2) = - EM1 LAM3  
LDA 0 TAU1  
SLJ 4 TRIG+70  
FBS 0 STOR2  
LAM 0 LAM  
FMU 0 LAM  
STA 0 STOR3  
(LSTOR3) = LAM SQURED  
LDA 0 TAU1  
FMU 0 ROOT2  
SLJ 4 EXP+71  
ZRO 0 0  
FMU 3 EN1  
FDV 0 ROOT2  
STA 0 STOR2  
(LSTOR2) = - EN1 MU4  
LDA 0 TAU1  
K240  
FMU 0 ROOT2  
SLJ 4 EXP+107  
FMS 3 EL1  
FBS 0 STOR2  
EL1 MU2  
STA 0 MU  
FMU 0 MU  
FAD 0 STOR3  
LAM SQ. + MU SQ.  
SLJ 4 SQROOT  
STA 0 HYP  
LDA 0 MU  
FMU 0 HYP  
STA 0 STOR3  
(LSTOR3) = SIND  
K250  
LDA 0 HYP  
DMA 0 STOR4  
FMS 0 SINT  
FMS 0 ADOMINWT  
(STOR4) = COSD  
- 17 -
STA 0 SINDOT
LDA 0 STOR4  
FMU 0 COST  
FMU 0 ADOMINWT  
STA 0 COSDOT  
LDA 0 STOR3  
FMU 0 SINHTR2  
FMU 0 ADOMINWT  
STA 0 SINHDOT  
LDA 0 STOR3  
FMU 0 COSHTR2  
FMU 0 ADOMINWT  
STA 0 COSHDOT  
SLJ 0 ALPHA+2  
EXIT FROM DERIV1
ZRO 0 0

SLJ 0 0  
S.R. TO EQUALIZE DELTAF
LDA 0 TAUFIN  
FDV 0 DELTAF
SLJ 4 FIXIT  
CONVERT DELTAF TO FIXED P.T.
STA 0 INTSIGN  
STORE INTEGRAL PART
STQ 0 FRACSIGN  
STORE FRACTIONAL PART
AJP 1 /+5  
JUMP IF A NOT = ZERO
QJP 0 /+3  
A=0. JUMP IF Q = 0
LDA 0 TAUFIN  
STA 0 DELTFLSG
ENI 2 1  
SLJ 0 /+12
ENA 0 0

STA 0 DELTFLSG
K270
ENI 2 0
SLJ 0 /+10
AJP 2 /+1  
A NOT 0. JUMP IF A GREATER THAN 0
INH 0 1  
A LESS THAN 0. ABSO. INTEGER TO A
SAU 0 /+1  
ABSQ. INTEGER + 1 TO B2
SCA 1 2057
SSA 0 57
LLS 0 44
STA 0 FLOIPLON
LDA 0 TAUFIN  
FDV 0 FLOIPLON
STA 0 DELTFLSG
ENI 2 0
SLJ 2 FIXIPLON  
USED IN FINAL PROG.
LDA 0 DELTFLSG
SLJ 0 EQUAL  
TO EXIT OF S.R.
ZRO 0 0

FIXIT BSS 44  
B1=5 B6=65302 P=460
R DEC 1.075698925 NON-DIMEN. ORBIT RADIUS
V BSS 1  
NON-DIMEN. TARGET SPEED
VSQDR BSS 1  
NON-DIMEN. TARGET ACCEL.
B OCT 1775622077325042 TARGET SECTOR AT RENDEVOUS = PI/16
COSB BSS 1
SINB BSS 1
REARTH DEC 20925000. EARTH EQUATOR RAD. IN FT.
GACCEL DEC 32.086 EQUATOR GRAVITY IN FT/SEC/SEC
RMS BSS 1  
NON-DIMEN. MASS LOSS RATE
OMEGA BSS 1
THETATEM BSS 1  
NON-DIMEN. INIT. HORIZONTAL VEL.
V1 BSS 1
V2 BSS 1  
NON-DIMEN. INIT. VERTICAL VEL.
EERRCOF DEC -15.B-12 (-15/4096 DEC.) USED TO FIND DELTAF
ROOTHALF OCT 2000552023631500
SINT BSS 1
COST BSS 1
SINHTR2 BSS 1
COSHTR2 BSS 1
UNMINWT BSS 1  
STORE FOR (1-WT)

---
BSS

EL1INIT OCT 2001500400000000 DEC. VALUE = +1.251953125
EM1INIT OCT 5776377777777777 DEC. VALUE = -1.000000000
EN1INIT OCT 2004460000000000 DEC. VALUE = +9.500000000
T1INIT OCT 1776463146314631 DEC. VALUE = +0.300000000
EL2INIT OCT 2001500300000010 DEC. VALUE = +1.251464844
EM2INIT OCT 5776377777777777 DEC. VALUE = -1.000000000
EN2INIT OCT 2004460100000000 DEC. VALUE = +9.503906250
T2INIT OCT 1776461700000000 DEC. VALUE = +0.300000000
EL3INIT OCT 2001500200000000 DEC. VALUE = +9.500000000
EM3INIT OCT 5776377777777777 DEC. VALUE = -1.000000000
EN3INIT OCT 2004460020000000 DEC. VALUE = +9.507812500
T3INIT OCT 1776463146314631 DEC. VALUE = +0.298706054
EL2INIT OCT 2001500300000010 DEC. VALUE = +9.500000000
EM2INIT OCT 5776377777777777 DEC. VALUE = -1.000000000
EN2INIT OCT 2004460100000000 DEC. VALUE = +9.511718750
T4INIT OCT 1776464300000000 DEC. VALUE = +0.301147460
EL4INIT OCT 2001500400000000 DEC. VALUE = +9.515625000
EM4INIT OCT 5776377777777777 DEC. VALUE = -1.000000000
EN4INIT OCT 2004460030000000 DEC. VALUE = +9.515625000
T5INIT OCT 1776465000000000 DEC. VALUE = +0.301757812

RESET LMNT INIT. (SEE K120)

HALT. START RENDEVOUS PROG.

BEGIN
REM RENDEVOUS PROG. STARTS HERE
SET NUMBER OF ITERATIONS

V/R = W
STORE TARGET ANGULAR VEL.

SET GILL S.R. FOR 29 DIFF. EQU.

CLEAR RUNKT+17 AREA

- 19 -
<table>
<thead>
<tr>
<th>STA</th>
<th>RUNKT+20</th>
</tr>
</thead>
<tbody>
<tr>
<td>INI</td>
<td>1</td>
</tr>
<tr>
<td>ISK</td>
<td>2</td>
</tr>
<tr>
<td>SLJ</td>
<td>0 /-2</td>
</tr>
<tr>
<td>LDA</td>
<td>0 UNITY</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+20</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+37</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+56</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+75</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+25</td>
</tr>
<tr>
<td>LAC</td>
<td>0 UNITY</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+47</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+66</td>
</tr>
<tr>
<td>LAC</td>
<td>0 TWO</td>
</tr>
<tr>
<td>STA</td>
<td>0 RUNKT+44</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L20</th>
</tr>
</thead>
<tbody>
<tr>
<td>STA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>LDA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>LDA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>LDA</td>
</tr>
<tr>
<td>STA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L30</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLJ</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>LDA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>IJP</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>SLJ</td>
</tr>
<tr>
<td>ZRO</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>SLJ</td>
</tr>
<tr>
<td>SLJ</td>
</tr>
<tr>
<td>ZRO</td>
</tr>
<tr>
<td>SLJ</td>
</tr>
<tr>
<td>SIL</td>
</tr>
<tr>
<td>SIU</td>
</tr>
<tr>
<td>LDA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>LOA</td>
</tr>
<tr>
<td>FMU</td>
</tr>
<tr>
<td>FAD</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>INT</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>ISK</td>
</tr>
<tr>
<td>SLJ</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>L50</th>
</tr>
</thead>
<tbody>
<tr>
<td>STA</td>
</tr>
<tr>
<td>FMU</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>ENI</td>
</tr>
<tr>
<td>LDA</td>
</tr>
<tr>
<td>FMU</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LMU</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLJ</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>STA</td>
</tr>
<tr>
<td>ENI</td>
</tr>
</tbody>
</table>

```
- 20 -
```
FA0 0 STOR3
STA 0 STOR3
INIT 3 14
ENI 0 0
ISK 1 2
SLJ 0 /-3
STA 0 MU
FMU 0 MU
FA0 0 STOR1
STA 0 HYP2
SLJ 4 SQROOT
ZRO 0 0
STA 0 HYP
FMU 0 HYP2
STA 0 HYP3
ENI 1 0
ENI 3 0
SLJ 0 LAMMU
BSS 3
LDA 0 A
FMU 0 LAM
FDV 0 HYP
STA 0 RUNKT+11
SET UD0T(0) = A(LAM)/HYP
LDA 0 A
FMU 0 MU
FDV 0 HYP
FSB 0 UNITY
STA 0 RUNKT+14
SET Vue(0)
ENA 0 */+2
SAL 0 L220-4
SLJ 0 L150-3
ENA 0 ALPHA+2
SAL 0 L220-4
ENI 0 0
SLJ 0 L220-3
LAC 0 OMEGA
FMU 0 RUNKT+2
FA0 0 UNITY
STA 0 UNMINWT
LDA 0 A
FDV 0 UNMINWT
STA 0 ADOMINWT
ENI 0 0
LDA 0 RUNKT+12
STA 0 RUNKT+3
LDA 0 RUNKT+15
STA 0 RUNKT+6
LDA 0 RUNKT+4
FMU 0 RUNKT+4
STA 0 STOR2
LDA 0 RUNKT+7
FMU 0 RUNKT+7
STA 0 STOR1
SLJ 4 SQROOT
FMU 0 STOR1
STA 0 R32
FMU 0 STOR1
STA 0 R52
ENI 1 0
SLJ 4 LAMMU
LAC 1 RUNKT+4
FDV 0 R32
STA 0 STOR1
LDA 0 ADOMINWT
FMU 1 LAM
FDV 0 HYP
FA0 0 STOR1
STA 1 RUNKT+11
SET UD0T(T)
INI 1 2
- 21 -
ENI 0 0
ISK 1 5
SLJ 0 L120-3
ENI 0 0
ENI 3 0
LAC 3 RUNKT+26 (-P11)
STA 3 RUNKT+17 SET LAM1DOT = -P11
LAC 3 RUNKT+31 (-RHO1)
STA 3 RUNKT+22 SET MU1DOT = -RHO1
LDA 0 RUNKT+7 Y
FMU 0 RUNKT+7 Y SQUARED
FSB 0 STOR2 Y SQ - X SQ
FSB 0 STOR2 Y SQ - 2(X SQ)
FMU 0 RUNKT+20 (Y SQ - 2 X SQ)LAM1
STA 0 STOR1
LAC 0 THREE MINUS THREE
FMU 0 RUNKT+4 (-3X)
FMU 0 RUNKT+7 (-3XY)
FMU 3 RUNKT+23 (-3XY MU1)
FAD 0 STOR1
FDV 0 R52
STA 3 RUNKT+25 SET PI1DOT
LAC 0 RUNKT+7 (-Y)
FMU 0 RUNKT+7 (-Y SQUARED)
FMU 0 TWO (-2 Y SQUARED)
FAD 0 STOR2 X SQ - 2 Y SQ
FMU 3 RUNKT+23 (X SQ - 2 Y SQ)MU1
STA 0 STOR1
LAC 0 THREE MINUS THREE
FMU 0 RUNKT+4 (-3X)
FMU 0 RUNKT+7 (-3XY)
FMU 3 RUNKT+20 (-3XY LAM1)
FAD 0 STOR1
FDV 0 R52
STA 3 RUNKT+30 SET RH01DOT
INI 3 13
ENI 0 0
ISK 3 57
SLJ 0 L13C-4
LAC 0 OMEGA (-W), ENTER FROM L70+5
FMU 0 RUNKT+2 (-WT)
FAD 0 UNITY 1-WT
STA 0 UNMINWT
LDA 0 A LOAD INITIAL ACCEL.
FDV 0 UNMINWT A/(1-WT)
STA 0 ADOMINWT
ENI 0 0
LAC 0 MU
FMU 0 RUNKT+20 (-MU LAM1)
STA 0 STOR1
LDA 0 LAM
FMU 0 RUNKT+23 LAM MU1
FAD 0 STOR1
STA 0 A1
LAC 0 MU
FMU 0 RUNKT+34 (-MU LAM2)
STA 0 STOR1
LDA 0 LAM
FMU 0 RUNKT+37 LAM MU2
FAD 0 STOR1
STA 0 A2
LAC 0 MU
FMU 0 A1 TIMES A2
FMU 0 ADOMINWT (A/1-WT)A1A2
FDV 0 HYP3
STA 0 RUNKT+77 SET A12DOT
LAC 0 MU
FMU 0 RUNKT+50 (-MU LAM3)
STA 0 STOR1
LDA 0 LAM
FMU 0 RUNKT+53 LAM MU3
- 22 -
FAD 0 STOR1
STA 0 A3  A3 = LAM MU3 - MU LAM3
FMU 0 A1  A1 TIMES A3
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+102 SET A13DOT
LAC 0 MU
FMU 0 RUNKT+64 (-MU LAM4)
STA 0 STOR1
LDA 0 LAM
FMU 0 RUNKT+67  LAM MU4
FAD 0 STOR1
STA 0 A4  A4 = LAM MU4 - MU LAM4
FMU 0 A1  A1 TIMES A4
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+105 SET A14DOT
LDA 0 A2
FMU 0 A2  A2 SQUARED
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+110 SET A22DOT
LDA 0 A2
FMU 0 A3  A2 TIMES A3
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+113 SET A23DOT
LDA 0 A3
FMU 0 A3  A3 SQUARED
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+116 SET A24DOT
LDA 0 A4
FMU 0 A4  A4 SQUARED
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+121 SET A33DOT
LDA 0 A3
FMU 0 A4  A3 TIMES A4
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+124 SET A34DOT
LDA 0 A4
FMU 0 A4  A4 SQUARED
FMU 0 ADOMINWT
FDV 0 HYP3
STA 0 RUNKT+127 SET A44DOT
ENI 0 0
SLJ 0 ALPHA+2 END OF DERIV PROG.
ENI 0 0 ENTER FROM L70+7
ENI 1 147
LDA 1 RUNKT STORE INIT. CONDITIONS
STA 1 SAVE FOR NEXT ITERATE
IJP 1 /-1
SLJ 4 ALPHA SET UP GILL ROUTINE
ZRO 0 RUNKT PARAMETER WORD
ZRO 0 DERIV
IJP 2 /+1 JUMP IF B2 NOT = ZERO
SLJ 0 /+3 B2=0. JUMP TO RMS COMP.
ENI 0 0 B2 NOT = ZERO
SLJ 4 ALPHA+1 INTEGRATE AGAIN
ENI 0 0
SLJ 0 /-2 JUMP TO TEST B2
LDA 0 CAPXFIN BEGIN R.M.S. COMPUTATION
FSB 0 RUNKT+4 CAPXFIN - XFIN
STA 0 C3
FMU 0 C3
STA 0 STOR2
LDA 0 CAPYFIN
FSB 0 RUNKT+7 CAPYFIN - YFIN

- 23 -
L230
STA 0 C4
FMU 0 C4
FAD 0 STOR2
STA 0 STOR2
LDA 0 CAPUFIN
FSB 0 RUNKT+12 CAPUFIN - UFIN
STA 0 C1
FMU 0 C1
FAD 0 STOR2
STA 0 STOR2
LDA 0 CAPVFIN
FSB 0 RUNKT+15 CAPVFIN - VFIN
STA 0 C2
FMU 0 C2
FAD 0 STOR2
SLJ 4 SQROOT
ZRO 0 0
FDV 0 TWO
STA 0 RMS
ENI 1 17
ENI 3 55
LDA 3 RUNKT+20 SET 4 BY 4 MATRIX
STA 1 BB11
INI 3 -3
IJP 1 7-1
SLJ 4 MATRIX INVERT THE MATRIX
ZRO 0 0
SLS 0 L250-3 SINGULAR MATRIX ALARM HALT
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 5
ZRO 0 4
ZRO 0 BB11 BB11 = ADDRESS OF 8 MATRIX
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 BB11+120 BB11+120 = ADDRESS OF INVERSE
LDA 0 RUNKT+100 A12 TO SET 4 BY 3 D MATRIX
STA 0 BB11+100 SET D11 = A12
LDA 0 RUNKT+103 A13
STA 0 BB11+101 SET D12 = A13
LDA 0 RUNKT+106 A14
STA 0 BB11+102 SET D13 = A14
LDA 0 RUNKT+111 A22
STA 0 BB11+103 SET D21 = A22
LDA 0 RUNKT+114 A23
STA 0 BB11+104 SET D22 = A23
STA 0 BB11+106 SET D31 = A32
LDA 0 RUNKT+117 A24
STA 0 BB11+105 SET D23 = A24
STA 0 BB11+111 SET D41 = A42
LDA 0 RUNKT+122 A33
STA 0 BB11+111 SET D41 = A42
LDA 0 RUNKT+107 SET D32 = A33
STA 0 BB11+112 SET D42 = A43
LDA 0 RUNKT+125 A34
STA 0 BB11+110 SET D33 = A34
STA 0 BB11+112 SET D42 = A43
LDA 0 RUNKT+130 A44
STA 0 BB11+113 SET D43 = A44
ENI 0 0
ENI 3 2
SLJ 4 MATRIX FOR USE 6TH WORD HENCE
ZRO 0 L270-2 MULT. B INVERSE BY D
ZRO 0 0
ZRO 0 0
ZRO 0 4
ZRO 0 0
ZRO 0 0
ZRO 0 0
ZRO 0 4
ZRO 0 BB11+120 BB11+120 = ADDRESS OF B INVERSE
ZRO 0 0
ZRO 0 BB11+100 BB11+100 = ADDRESS OF D MATRIX
ZRO 0 3
M = 3 COLS. IN THE MATRIX PROD.
- 24 -
L300

L310

L320

L330

ZRO 0 BB11+1

BB11+1 = ADDRESS OF MATRIX PROD.

REM

L273-306 SETS UP MATRIX OF SYSTEM TO BE

REM

SOLVED FOR DELTAUFIN, VAREL, VAREM AND VAREN

LDA 3 BB11+12

STA 3 B42

IJP 3 /-1

ENI 3 2

LDA 3 BB11+7

STA 3 B32

IJP 3 /-1

ENI 3 2

LDA 3 BB11+4

STA 3 B22

IJP 3 /-1

LDA 0 RUNKT+11

FSB 0 CAPUDFN

STA 0 BB11

LDA 0 RUNKT+14

FSB 0 CAPVDFN

STA 0 B21

LDA 0 RUNKT+12

STA 0 B31

LDA 0 RUNKT+15

STA 0 B41

SLJ 0 L310

SLJ 0 BB11

ZRO 0 C4

ENI 0 0

SLJ 4 MATRIX

SLS 0 L310+1

ZRO 0 0

ZRO 0 0

ZRO 0 5

ZRO 0 4

ZRO 0 BB11

ZRO 0 0

ZRO 0 0

ZRO 0 0

ZRO 0 BB11+120

ENI 0 0

SLJ 4 MATRIX

ZRO 0 0

ZRO 0 L320-1

ZRO 0 0

ZRO 0 0

ZRO 0 4

ZRO 0 4

ZRO 0 BB11+120

ZRO 0 0

ZRO 0 C1

ZRO 0 0

ZRO 0 C1+4

ENI 0 0

SLJ 0 /+2

STA 0 BB11

ZRO 0 C1+7

ENI 0 0

ENI 0 0

ENI 0 0

ENI 0 0

ENI 0 0

ENI 0 0

IJP 6 /+1

SLJ 0 FINAL

STU 6 /+1

SLS 2 /+1

ENI 6 0

JUMP IF B6 NOT = ZERO

B6 = 0. JUMP TO FINAL PROG.

STOP SWITCH

RESET B6

- 25 -
ENI 1 3
LDA 1 C1+4
STA 1 VARTAU
JJP 1 /-1
FAD 0 TAUFIN
STA 0 SAVE+143
ENI
FMU 0 W
FAD 0 S
STA 0 SPWT
SLJ 4 TRIG+70
STA 0 COSSPWT
FMU 0 R
STA 0 SAVE+136
L340
LDA 0 SPWT
SLJ 4 TRIG
ZRO 0 0
STA 0 SINSPWT
FMU 0 R
STA 0 SAVE+135
L350
LDA 0 V
STA 0 SAVE+137
LAC 0 SPWT
FMU 0 V
STA 0 SAVE+140
L360
LAC 0 COSSPWT
FMU 0 VSQDR
STA 0 SAVE+141
LAC 0 SINSPWT
FMU 0 VSQDR
STA 0 SAVE+142
ENI 0 0
ENI 0 2
ENI 1 2
LDA 1 C1+5
FAD 1 EL
STA 1 EL
JJP 1 /-1
SLJ 4 70007
ZRO 0 0
SAX 0 MDOT
ZRO 0 S
EXP 0 70
SLJ 4 70007
STA 0 CAPXFIN
ZRO 0 LAM
SLJ 0 RECUR
ZRO 0 0
BSS 4 200
POLAR
EL1 1
EM1 1
EN1 1
T1 1
EL2 1
EM2 1
EN2 1
T2 1
EL3 1
EM3 1
EN3 1
T3 1
EL4 1
EM4 1
EN4 1
T4 1
EL5 1
EM5 1
EN5 1

STORE DELTAUFIN, VAREL, VAREM, VAREN
NEW TAUFIN FOR NEXT ITERATE
STORE NEW TAUFIN (NOTE L30+3)
OR SLJ 0 L350+5 IF REND. POINT FIXED
NEW CAPYFIN
NEW CAPXFIN
NEW CAPUFIN
NEW CAPVFIN
NEW CAPVDFN
NEW CAPUDFN
STORE NEW EL, EM AND EN
PRINT
DECIMAL DUMP WITH PANEL
CLEAR ARITH. ERRORS
PRINT
DEC. DUMP
TO NEXT ITERATION
B1=33     B6=66370 P=460
FINAL

T5

BSS 1
BSS 26
ENI 1 147
LIL 2 FIXIPLON
LDA 1 SAVE
STA 1 RUNKT
IJP 1 /-1
SLJ 4 ALPHA
ZRO 0 RUNKT
ZRO 0 DERIV
ENI 0
SLJ 4 /+4
IJP 2 /+1
SLJ 0 L700+1
ENI 0 0
SLJ 4 ALPHA+1
ENI 0 0
SLJ 0 /-3
SLJ 0 0
LDA 0 RUNKT+2
ENI 0 0
ENI 0 0
STA 0 DUMP
LDA 0 RUNKT+4
ENI 0 0
ENI 0 0
STA 0 DUMP+4
LDA 0 RUNKT+7
ENI 0 0
ENI 0 0
STA 0 DUMP+5
LDA 0 RUNKT+12
ENI 0 0
ENI 0 0
STA 0 DUMP+6
LDA 0 RUNKT+15
ENI 0 0
ENI 0 0
STA 0 DUMP+7
LDA 0 RUNKT+11
ENI 0 0
ENI 0 0
STA 0 DUMP+10
LDA 0 RUNKT+14
ENI 0 0
ENI 0 0
STA 0 DUMP+11
LDA 0 RUNKT+12
LDQ 0 RUNKT+15
SLJ 4 POLAR+130
ENI 0 0
ENI 0 0
STQ 0 DUMP+12
LDA 0 LAM
LDQ 0 MU
SLJ 4 POLAR+130
ENI 0 0
ENI 0 0
STQ 0 DUMP+13
SLJ 4 70007
STA 0 DUMP
ZRO 0 DUMP+13
PRINT
PARAMETER WORD

L650

TO EXIT OF DUMP S.R.

L660

L700

C3 = CAPXFIN - XFIN
C4 = CAPYFIN - YFIN
ENI 0 0 STA 0 DUMP+15 (DUMP+15) = CAPYFIN - YFIN
LDA 0 C1 C1 = CAPUFIn - UFIN
ENI 0 0
ENI 0 0 STA 0 DUMP+16 (DUMP+16) = CAPUFIn - UFIN
LDA 0 C2 C2 = CAPVFIN - VFIN
ENI 0 0
ENI 0 0 STA 0 DUMP+17 (DUMP+17) = CAPVFIN - VFIN
SLJ 4 70007 PRINT
ZRO 0 0 STA 0 DUMP+14 PARAMETER WORD
ZRO 0 DUMP+17 SLJ 0 L3570+5 JUMP TO PRINT B FINAL
STOR1 BSS 1
STOR2 BSS 1
STOR3 BSS 1
STOR4 BSS 1
L720 DEC 1.
DELY1 BSS 1
DELY1 BSS 1
DELU1 BSS 1
DELV1 BSS 1
DELY2 BSS 1
DELY2 BSS 1
DELU2 BSS 1
DELV2 BSS 1
DELY3 BSS 1
DELY3 BSS 1
DELU3 BSS 1
DELV3 BSS 1
DELY4 BSS 1
DELY4 BSS 1
DELU4 BSS 1
DELV4 BSS 1
DELY5 BSS 1
DELY5 BSS 1
DELU5 BSS 1
SAVE BSS 150 ALSO DELV5 STORE
RUNKT BSS 135
CAPXFIN BSS 1
CAPYFIN BSS 1
CAPUFIn BSS 1
CAPVFIN BSS 1
CAPUDFN BSS 1
CAPVDFN BSS 1
TaufIN BSS 1
LAM BSS 1
HYP BSS 1
HYP3 BSS 1
MU BSS 1
HYP2 BSS 1
UNITY DEC 1.
W BSS 1
DELTau BSS 1
SPWT BSS 1
COSSPWT BSS 0
SINSPWT BSS 1
R32 BSS 1
R52 BSS 1
INTSIGN BSS 1
FRACSIGN BSS 1
FLOIPLON BSS 1
FIXIPLON BSS 1

- 28 -

.........................
THREOR2 OCT 2002417416663160 THREE/ROOT2
TREHALF DEC 1.5
DELFLSG BSS 1
NT BSS 1
THETA BSS 1
VSTART DEC .65 NON-DIMEN. INITIAL SPEED
DELTTHETA BSS 1
DELYFIN BSS 1
DELYFIN BSS 1
DELYFIN BSS 1
DELYFIN BSS 1
MDOT DEC .0025 FRACTIONAL MASS LOSS PER SEC.
A BSS 1
QQ BSS 1
C BSS 1
10000. ROCKET NOZZLE VEL. IN FT./SEC.
EM BSS 1
EN BSS 1
S BSS 1
A1 BSS 1
A2 BSS 1
A3 BSS 1
A4 BSS 1
VARTAU BSS 1
VAREL BSS 1
VAREM BSS 1
VAREN BSS 1
ALPHA BSS 134 INIT. TARGET SECTOR ANGLE
DUMP BSS 20 B1=30 B6=67340 P=460
BB11 BSS 1
B12 BSS 1
B13 BSS 1
B14 BSS 1
B21 BSS 1
B22 BSS 1
B23 BSS 1
B24 BSS 1
B31 BSS 1
B32 BSS 1
B33 BSS 1
B34 BSS 1
B41 BSS 1
B42 BSS 1
B43 BSS 1
B44 BSS 1
C1 BSS 1
C2 BSS 1
C3 BSS 1
C4 BSS 1
LOG BSS 100 5 TIMES L SQUARED CELLS
REM STORE FOR MATRIX PRODUCT
L2000 BSS 1440 B1=12 B6=67700 P=460
L3440 BSS 4 LOG S.R. ALARM EXIT TO K110
REM (67703) = 750 65110 000 00000
REM USNPGS GEN. DUMP B1=32 P=21
REM AID TO LINEARIZED TRAJECTORY PROG.
REM CLEAR BB11 MATRIX AREA
ENA 0 0
ENI 117
STA 1 BB11
IJP 17
LDA 0 TAUFUN
SLJ 4 TRIG
STA 0 BB11
LDA 0 TAUFUN
SLJ 4 TRIG+70
ZRO 0 0
STA 0 B21
LDA 0 TAUFUN
FMU 0 ROOT2
(8B11) = SINT.
(8B21) = COST
SLJ 4 EXP + 71
STA 0 B32
LDA 0 TAU FIN
FMU 0 ROOT 2
SLJ 4 EXP + 107
STA 0 B42
LDA 0 SIN
FDV 0 C
STA 0 B13
LDA 0 COS
FDV 0 C
STA 0 B23
LDA 0 SINH
FDV 0 C
STA 0 B33
LDA 0 COSH
FDV 0 C
STA 0 B43
LAC 0 CAPYFIN
STA 0 B14
LAC 0 CAP VFIN
STA 0 B24
SLJ 0 L 3570 + 3
STA 0 B34
LDA 0 CAP UF IN
STA 0 B44
LAC 0 DEL XFIN
STA 0 C1
LAC 0 DEL UF IN
STA 0 C2
LAC 0 DEL YFIN
STA 0 C3
LAC 0 DEL VFIN
STA 0 C4
ENA 0 7 + 2
SAL 0 L 320 + 4
SLJ 0 L 310
ZRO 0 0
ENA 0 L 320 + 6
SAL 0 L 320 + 4
LDA 0 C1 + 4
FAD 0 V1
STA 0 STOR1
FMU 0 STOR1
STA 0 STOR2
LDA 0 C2 + 4
FAD 0 V2
STA 0 STOR3
FMU 0 STOR3
FAD 0 STOR2
SLJ 4 SQ ROOT
ZRO 0 0
STA 0 VSTART1
LDA 0 STOR1
FDV 0 VSTART1
SLJ 4 POLAR
STA 0 THE TAU 1
LDA 0 C3 + 4
FAD 0 C
STA 0 EJECT
LDA 0 C4 + 4
FAD 0 B
STA 0 SECTOR
SLJ 4 70 007
ZRO 0 SECTOR
SLJ 4 70 007
ZRO 0 VSTART1

(B32) = SINHTR2

(B42) = COSHTR2

SINE IMPULSE INTEGRAL

COSINE IMPULSE INTEGRAL

SINH IMPULSE INTEGRAL

COSH IMPULSE INTEGRAL

JUMP TC PATCH

C1 = ADD. OF MATRIX COL.

CREATE LINEAR EQU. ROUTINE EXIT

JUMP TO LINEAR EQU. ROUTINE

REBUILD RENDEVOUS PROGRAM

DEL V1

( STOR1 ) = NEW V1

( STOR2 ) = NEW V1 SQUARED

DEL V2

( STOR3 ) = NEW V2

NEW V1 SQ. + NEW V2 SQ.

NEW VTHETA

NEW C

NEW B

( SECTOR ) = NEW B

PRINT RESULTS DECIMAL

PARAMETER WORD

PRINT RESULTS OCTAL

PARAMETER WORD

- 30 -
ZRO 0 SECTOR
ENI 0 0
LDA 0 VSTART1 RESUME COMP. OF INIT. CONDITIONS
STA 0 VSTART SET NEW VSTART
LDA 0 Transformer
STA 0 THETA SET NEW THETA
LDA 0 EJECT SET NEW C
LDA 0 SECTOR SET NEW B
STA 0 B
SLJ 4 TRIG+70
ZRO 0 0
STA 0 COSB
FMU 0 R
STA 0 CAPYFIN TARGET COOR. AT RENDEVOUS
LDA 0 0
SLJ 4 TRIG
ZRO 0 0
STA 0 SINB
FMU 0 R
STA 0 CAPXFIN TARGET COOR. AT RENDEVOUS
LDA 0 0
FMU 0 V
STA 0 CAPUFVIN TARGET VEL. COMP. AT RENDEVOUS
LAC 0 0
FMU 0 V
STA 0 CAPVFIN TARGET VEL. COMP. AT RENDEVOUS
LAC 0 0
FMU 0 VSDQR
STA 0 CAPVDFN TARGET ACC. COMP. AT RENDEVOUS
LAC 0 0
FMU 0 VSDQR
STA 0 CAPUDFN TARGET ACC. COMP. AT RENDEVOUS
SLJ 4 TRIG+70 COSINE THETA
ZRO 0 0
FMU 0 VSTART INITIAL HORIZONTAL VEL.
STA 0 V1
LDA 0 0
SLJ 4 TRIG SINE THETA
FMU 0 VSTART
STA 0 V2
LDA 0 0
ROCKET NOZZLE VEL.
FSB 2 /+1 JUMP IF DIFF. IS POS.
A.A 0 0 0 0 0 0
SCM 0 0 ABSO. VALUE OF DIFF.
THS 0 UNITY EXIT IF ONE GREATER THAN DIFF.
SLJ 0 0 SECTOR+2 NO. JUMP TO ADJUST MDOT AND C
LDA 0 0
FMU 0 MDOT
FDV 0 GACCEL
STA 0 A
LDQ 0 0 EL5 NON-DIMEN. ACCELERATION
SLJ 0 0 K100+2 RETURN TO LIN. TRAJ. PROG.
VSTART1 BSS 1 1
THETA1 BSS 1 1
EJECT BSS 1 1
SECTOR BSS 1 1
SLS 0 0 SECTOR+1 HALT IF 1-WT IS NEGATIVE
ZRO 0 0
LAC 0 0 TAUFIN (-T)
FMU 0 OMEGA (+WT)
FAD 0 0 UNIITY (1-WT)
AJP 3 0 SECTOR+1 HALT IF 1-WT IS NEGATIVE
SLJ 4 0 LOG+66 LOG(1-WT) TO BASE E
ZRO 0 0
FMU 0 0 C
FDV 0 TENGRAND
SLJ 4 EXP
- 31 -
ZRO 0 0
SOM 0 MASK
FAD 0 UNITY
FDV 0 TAUFIN
STA 0 OMEGA
LDA 0 GACCEL
FDV 0 REARTH
SLJ 4 SQROOT
ZRO 0 0
FMU 0 OMEGA
STA 0 MDOT
LDA 0 TENGRAND
STA 0 C
ENI 0 0
ENI 0 0
ENI 0 0
ENI 0 0
ENI 0 0
ENI 0 0
ENI 0 0
LDA 0 MDOT
FMU 0 C
FDV 0 GACCEL
STA 0 A
SLJ 0 K70
LDA 0 CAPxFIN
FMU 0 RC012
SLJ 0 L3460+2
ZRO 0 0
SLJ 4 70007
ZRO 0 0
STA 0 UNITY
ZRO 0 SPWT
SLS 0 L3570+7
ZRO 0 0
EN D