NASA TECHNICAL NOTE

NASA TN D-4470

C. 1

NASA TN D-4470

A FAST METHOD OF ORBIT COMPUTATION

by

Karl Stumpff

Goddard Space Flight Center
Greenbelt, Md.

and

E. H. Weiss

IBM Federal Systems Division
Greenbelt, Md.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION • WASHINGTON, D. C. • APRIL 1968

NASA TN D-4470

A FAST METHOD OF ORBIT COMPUTATION

By Karl Stumpff

Goddard Space Flight Center
Greenbelt, Md.

and

E. H. Weiss

IBM Federal Systems Division
Greenbelt, Md.

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

For sale by the Clearinghouse for Federal Scientific and Technical Information
Springfield, Virginia 22151 – CFSTI price $3.00

ABSTRACT

The problem of rapidly computing trajectories of spacecrafts from their initial conditions has become very important. Classical methods rely almost exclusively on precise integration techniques, but results thus obtained are too slow over extended arcs, even on high-speed computers. Moreover, great accuracy is often unnecessary. Here we present a new method of computing approximate ephemerides of a small body (minor planet or artificial satellite). This method is 10 to 15 times faster than the well-known methods of Encke or Cowell. The errors are small (e.g., of the order of one part in a thousand) and the results converge to the N-body point-mass solution for small time steps. It is also possible to account for non-point-mass effects; this, however, has not yet been implemented.

CONTENTS

A FAST METHOD OF ORBIT COMPUTATION

by
K. Stumpff* and E. H. Weiss**
Goddard Space Flight Center

INTRODUCTION

The new method of computing perturbations described below has been developed in response to a need for a means of quickly approximating a spacecraft ephemeris. In particular, a quick computation of the orbit of Explorer 33 (described in Example 3 of "Results") was required. The results need not be exact; small errors, of the order of one part in a thousand, are permitted. The special perturbation method yields:

a. The fast exact solution, or an even faster approximation, to the N-body point-mass problem.

b. Fast approximations to many non-point-mass problems.

The exact solution to the non-point-mass problem can be obtained by numerical integration. This solution, however, has not yet been implemented.

Reference 1 describes a forerunner of the present technique, used as early as 1942 to compute heliocentric orbits of minor planets under the influence of Jupiter and other major planets. The proof in this report is shorter, more direct, and more convenient for modern applications than its earlier counterpart. Both methods are variants of the well known Encke special perturbation technique. Encke's perturbations are defined as the deviations of the planet's coordinates from those of an osculating Kepler ellipse and are of the order h^2, where $h = t - t_0$ is the intermediate time beyond t_0, the epoch of osculation. The present technique (as in Reference 1) combines several Keplerian orbits to form an intermediate orbit that includes essential parts of Encke's perturbations. The deviations of the actual from the intermediate orbit are termed "rest perturbations." They are of the order h^4 and therefore very small for small h. Encke's perturbations and Stumpff's rest perturbations cannot be solved in closed form; they are solved by classical numerical integration.

*Prof. Emeritus, Univ. of Goettingen and Senior Post-Doctoral Resident Research Associate, U. S. National Academy of Science-National Research Council.

**Advisory Mathematician IBM Federal Systems Division, P. O. Box 67, Greenbelt, Maryland.

NOTATION AND EQUATIONS

The results about to be derived are valid for N point-mass bodies. For simplicity the formulas are given for the 4-body case (e.g., earth, moon, sun and spacecraft), but the proof is readily extended to N bodies. The formulas exhibit remarkable symmetry.

Let each of the subscripts i, j, k, and l assume the value 0, 1, 2, or 3 with the proviso that different subscripts be distinct. Denote the four bodies by Q_i and their masses by m_i. The vectors from Q_i to Q_j are q_{ij}, and their magnitudes are $r_{ij} = |q_{ij}|$. Without loss of generality, we can use canonical units; then the Gaussian constant equals unity. Assume that the bodies act as point masses. Then the vectors q_{ij} satisfy the differential equations

$$\ddot{q}_{ij} = -(m_i + m_j)\, q_{ij} r_{ij}^{-3} - m_k (q_{ik} r_{ik}^{-3} + q_{kj} r_{kj}^{-3}) - m_l (q_{il} r_{il}^{-3} + q_{lj} r_{lj}^{-3}). \qquad (1)$$

The 12 combinations i, j from 0, 1, 2, 3, contain only three linearly independent vectors q_{ij} (for instance q_{10}, q_{12}, q_{13}), as there exist six identities $q_{ij} = -q_{ji}$, and three independent equations of the form

$$q_{ij} + q_{jk} + q_{ki} = 0.$$

Here and elsewhere, $q_{ij}(0) = q_{ij}(t_0)$ refers to the time t_0; $q_{ij} = q_{ij}(t)$ refers to the time $t = t_0 + h$. A special solution of Equation 1 is determined by the initial values

$$q_{ij}(0) \text{ and } \dot{q}_{ij}(0). \qquad (2)$$

Use square brackets to denote Keplerian orbits that osculate q_{ij} at $t = t_0$. Then

$$[q_{ij}(0)] = q_{ij}(0); \; [\dot{q}_{ij}(0)] = \dot{q}_{ij}(0) \qquad (3)$$

are the conditions of osculation.

The osculating Keplerian orbits satisfy the differential equations

$$[\ddot{q}_{ij}] = -(m_i + m_j)\, [q_{ij}]\, [r_{ij}]^{-3}, \qquad (4)$$

where

$$[r_{ij}] = |\, [q_{ij}] \,|.$$

Equation 4 can be applied to any combination of two different subscripts i, j, k, and l. Now define the vectors s_{ij} $(= -s_{ji})$ by

$$s_{ij} = [q_{ij}]\, [r_{ij}]^{-3} - q_{ij} r_{ij}^{-3}. \qquad (5)$$

Substituting Equation 5 in Equation 4 and rearranging, one obtains

$$-q_{ij} r_{ij}^{-3} = \frac{1}{m_i + m_j} [\ddot{q}_{ij}] + s_{ij}. \tag{6}$$

Using Equation 6 to eliminate all expressions of this form from the right side of Equation 1 gives:

$$\ddot{q}_{ij} = [\ddot{q}_{ij}] + \frac{m_k}{m_i + m_k} [\ddot{q}_{ik}] + \frac{m_k}{m_k + m_j} [\ddot{q}_{kj}] + \frac{m_l}{m_i + m_l} [\ddot{q}_{il}]$$

$$+ \frac{m_l}{m_l + m_j} [\ddot{q}_{lj}] + (m_i + m_j) s_{ij} + m_k (s_{ik} + s_{kj}) + m_l (s_{il} + s_{lj}).$$

This can be written

$$\ddot{q}_{ij} = [\ddot{q}_{ij}] + \ddot{S}_{ij} + \ddot{R}_{ij}, \tag{7a}$$

where

$$\ddot{S}_{ij} = \frac{m_k}{m_i + m_k} [\ddot{q}_{ik}] + \frac{m_k}{m_k + m_j} [\ddot{q}_{kj}] + \frac{m_l}{m_i + m_l} [\ddot{q}_{il}] + \frac{m_l}{m_l + m_j} [\ddot{q}_{lj}] \tag{7b}$$

$$\ddot{R}_{ij} = (m_i + m_j) s_{ij} + m_k (s_{ik} + s_{kj}) + m_l (s_{il} + s_{lj}). \tag{7c}$$

The first and second integrals of $[\ddot{q}_{ij}]$ and $\ddot{S}_{ij}$ can be obtained from the well known properties of Keplerian conic sections. See Reference 2, or Chapter V of Reference 3. The integration of $\ddot{R}_{ij}$, however, can be effected only by numerical methods. The first and second integrals of Equation 7a are

$$\left.\begin{aligned} \dot{q}_{ij} &= [\dot{q}_{ij}] + \dot{S}_{ij} + \dot{R}_{ij} + a_{ij}, \\ q_{ij} &= [q_{ij}] + S_{ij} + R_{ij} + a_{ij} h + b_{ij}. \end{aligned}\right\} \tag{8}$$

To determine the constant vectors a_{ij} and b_{ij}, evaluate Equation 8 at t_0, from which, by Equation 3,

$$\dot{S}_{ij}(0) + \dot{R}_{ij}(0) + a_{ij} = 0,$$

$$S_{ij}(0) + R_{ij}(0) + b_{ij} = 0.$$

Without loss of generality, $\dot{R}_{ij}(0)$ and $R_{ij}(0)$ can be equated to zero, since non-zero values can be absorbed by a_{ij} and b_{ij}. Thus

$$a_{ij} = -\dot{S}_{ij}(0), \; b_{ij} = -S_{ij}(0). \tag{9}$$

Substituting Equation 9 in Equation 8 gives

$$\left.\begin{aligned} \dot{q}_{ij} &= [\dot{q}_{ij}] + \dot{P}_{ij} + \dot{R}_{ij} \\ q_{ij} &= [q_{ij}] + P_{ij} + R_{ij}, \end{aligned}\right\} \tag{10}$$

where

$$\dot{P}_{ij} = \dot{S}_{ij} - \dot{S}_{ij}(0), \; P_{ij} = S_{ij} - h\,\dot{S}_{ij}(0) - S_{ij}(0). \tag{11}$$

P_{ij} is termed the approximate perturbation, and R_{ij} the rest perturbation. Equation 5 shows that $s_{ij}(0) = 0$, whence $\ddot{R}_{ij}(0) = 0$ in view of Equation 7c. Similarly $\dddot{R}_{ij}(0) = 0$, as can be seen by differentiation, and hence R_{ij} is of the order h^4. For sufficiently small values of h, $\dot{R}_{ij}$ and R_{ij} in Equation 10 can be ignored, if desired. The resulting approximations are

$$\left.\begin{aligned} \dot{q}^*_{ij} &= [\dot{q}_{ij}] + \dot{P}_{ij}, \\ q^*_{ij} &= [q_{ij}] + P_{ij}. \end{aligned}\right\} \tag{12}$$

Exact results are given by the formulas

$$\left.\begin{aligned} \dot{q}_{ij} &= \dot{q}^*_{ij} + \dot{R}_{ij} \\ q_{ij} &= q^*_{ij} + R_{ij}, \end{aligned}\right\} \tag{13}$$

where $\dot{q}^*_{ij}$ and q^*_{ij} are obtained by Equation 12, and $\dot{R}_{ij}$ and R_{ij} are the first and second integrals of $\ddot{R}_{ij}$, defined by Equation 7c.

USAGE

There are three *different* ways to compute (accurately or approximately) the ephemerides of celestial bodies by the set of formulas just given. For computational simplicity, one of the four bodies is placed at the center of the coordinate system. This body is denoted by Q_c and is called the central body. Let r be a dummy subscript that can assume the values 0, 1, 2, or 3 as long as $c \neq r$. Then the initial conditions are given by the known values $q_{cr}(0)$ and $\dot{q}_{cr}(0)$, and our problem is to find q_{cr} and $\dot{q}_{cr}$. By vector addition any other vector between the four bodies can be determined since

$$q_{ij} = q_{ic} + q_{cj},$$

where

$$q_{ic} = -q_{ci}.$$

Method A

Method A, described in Reference 1, is a variant of Encke's method of computing special perturbations. This method is especially effective when the numerical integration uses difference tables; this mode of integration was especially popular before the advent of high-speed computers.

Compute the six osculating Keplerian orbits $[q_{ij}]$ and their velocity vectors $[\dot{q}_{ij}]$ at $t = t_0 + nh$ ($n = 1, 2 \ldots, N$), where h is a conveniently chosen constant step length. Then compute $S_{cr}(t)$ and $\dot{S}_{cr}(t)$, the two-body solutions of Equation 7b satisfying the initial conditions given by Equation 3. Compute $P_{cr}(t)$ and $\dot{P}_{cr}(t)$ by Equation 11, and obtain the coordinates of the intermediary orbits $q^*_{cr}(t)$ and $\dot{q}^*_{cr}(t)$ by Equation 12. The rest perturbation $R_{cr}(t)$ can be obtained by integrating Equation 7c twice, using the classical method of numerical integration by differences, with $R(0) = \dot{R}(0) = 0$ as initial conditions. The start of the integrating scheme of differences can be achieved without iteration: Calculate $s_{ij}(t)$ from

$$s^*_{ij} = [q_{ij}]\,[r_{ij}]^{-3} - q^*_{ij}\, r^*_{ij}{}^{-3} \tag{14}$$

instead of Equation 5 for small nh. Since $s_{ij} - s^*_{ij}$ is of order h^4 it follows from Equation 7c that the error of $\ddot{R}_{ij}$ is also of order h^4. Therefore the error in R_{ij} is of order h^6 and can be neglected for small intermediate times. This is a remarkable advantage over Encke's method, where iterations at the start cannot be avoided.

Method A is effective if (1) none of the bodies closely approach any other body, (2) the magnitudes of R remain substantially smaller than the magnitudes of the complete perturbations, $P+R$. Thus the calculations are restricted to relatively small time intervals.

Method B

Let $h_0 = t_1 - t_0$ be a small step length, for which $R(t_1)$ is very small. Compute $q_{cr}(t_1) \approx q^*_{cr}(t_1)$, $\dot{q}_{cr}(t_1) \approx \dot{q}^*_{cr}(t_1)$ by Method A, neglecting the rest terms. Use these approximations of $q(t_1)$ and $\dot{q}(t_1)$ as new initial conditions, and compute $q(t_2)$, $\dot{q}(t_2)$ at $t_2 = t_1 + h_1$, again neglecting the rest terms, etc. The results will be accurate as long as the neglected rest terms do not accumulate beyond a specified tolerance. Thus Method B, which is simple and efficient, can be used to approximate orbits when great accuracy is not needed.

Method C

Use Method B to compute $\dot{q}^*_{cr}(t_0 + 4h_0)$ as well as $q^*_{cr}(t_0 + nh_0)$ for $n = 1, 2, 3, 4$. Then, using s^*_{ij} defined by Equation 14 instead of s_{ij}, compute $\ddot{R}_{cr}(t_0 + nh_0)$ by Equation 7c, and obtain $\dot{R}(t_0 + 4h_0)$ and $R(t_0 + 4h_0)$ by integration. For example, Stirling's five-point formula with $\ddot{R}(0) = 0$, and with the initial conditions $R(t_0) = \dot{R}(t_0) = 0$, yields

$$\left.\begin{aligned} \dot{R}(t_0 + 4h_0) &= \frac{h_0}{45}\,[64\ddot{R}(t_0 + h_0) + 24\ddot{R}(t_0 + 2h_0) + 64\ddot{R}(t_0 + 3h_0) + 14\ddot{R}(t_0 + 4h_0)] \\ R(t_0 + 4h_0) &= \frac{h_0^2}{45}\,[192\ddot{R}(t_0 + h_0) + 48\ddot{R}(t_0 + 2h_0) + 64\ddot{R}(t_0 + 3h_0)]\,. \end{aligned}\right\} \tag{15}$$

Then use Equation 13 to obtain $\dot{q}(t_1) = \dot{q}^*(t_1) + \dot{R}(t_1)$, $q(t_1) = q^*(t_1) + R(t_1)$ at $t_1 = t_0 + 4h_0$. With these vectors as new initial conditions proceed to $t_2 = t_1 + 4h_1$, $t_3 = t_1 + 4h_2$, etc.

This method delivers accurate ephemerides for an extended range of time as long as the step lengths h_n are sufficiently small, though they may be noticeably larger than those used in Method B. If the h_n surpass a certain limit, errors will occur because:

1. The supposition $s \approx s^*$ is no longer valid. This may be cancelled by iteration, if desired.

2. Integration Equations 15 use interpolating polynomials of the fourth degree. The formulas lose accuracy for large h, but this error can be avoided by the use of integration formulas of a higher degree.

The relative smallness of the rest perturbations permits the use of rather large time steps. Proper time-step control significantly reduces the computing time; considerable effort was spent in selecting the "best" time-step. The four time-step criteria given below were specialized to the following bodies and units:

1. Q_0 : spacecraft
   Q_1 : earth
   Q_2 : moon
   Q_3 : sun

2. Length is measured in earth radii, time in canonical units of 806.813645 seconds, and the mass of the earth is taken as unity. The four time-step criteria are:

   a. h is constant. This is best avoided unless the orbit is nearly circular.

   b. $h = (K/Q)^{1/4}$, where K is an input parameter, and Q (a theoretical overestimate of the rest perturbations) is given by

$$Q = \frac{m_2}{r_{10}^2\, r_{20}^2}\left(\frac{1}{r_{10}} + \frac{1}{r_{20}}\right) + \frac{m_2}{r_{12}^2}\left(\frac{1}{r_{10}^3} + \frac{1}{r_{20}^3}\right) + \frac{m_3}{r_{13}^3}\left(\frac{m_2}{r_{20}^2} + \frac{m_1}{r_{10}^2}\right).$$

3. $h = (A + W)/B$, where $W = \mathrm{Min}\,(r_{10}\ , Cr_{20})$, and A, B, C are input parameters.

4. $h_{new} = (1/2)(\,|D/\nabla^4 \ddot{R}_{10}| + 1)\, h_{old}$, where D is an input parameter, and
$$\nabla^4 \ddot{R}_{10} = \ddot{R}_{10}(t_0 + 4h) - 4\ddot{R}_{10}(t_0 + 3h) + 6\ddot{R}_{10}(t_0 + 2h) - 4\ddot{R}_{10}(t_0 + h).$$
Large changes in step size are avoided by the condition $0.5\, h_{old} \le h_{new} \le 3.5\, h_{old}$.

The following section gives numerical examples. In Tables 3, 4, and 5, the column headed "time-step criterion" contains either: the fixed value of h; or K; or A, B, C; or D. In general, best results are obtained with the A, B, C, or D criteria.

Test results were computed not only with large step sizes but also with very small ones. It was always possible to converge to results that are invariant under further reduction of the time-step. These converged results may rightly be considered the accurate ephemerides of the 4-body point-mass problem.

RESULTS

The following five examples illustrate the results.

Example 1

Method A was first used by K. Stumpff (Reference 1) in 1942 to compute special perturbations of the minor planet (931) Whittemora by Jupiter. Some numerical results are taken, abbreviated, from Reference 1, to illustrate the difference between this method and the classical method of Encke (see Reference 4, p. 378ff). The formulas of Section 1 may be used with masses

m_0 = 0 (minor planet),
m_1 = 1 (sun),
m_2 = 1/1047.35 (Jupiter),
m_3 = 0 (as no other perturbing planet has been taken in account).

The initial epoch is $t_0 = 1920$, April 29.0; and h, the constant step of integration, equals 40 days = 40k canonical units (k = 0.0172021).

Table 1 gives the heliocentric equatorial x-coordinates of the minor planet for $t = t_0 + nh$ ($n = 0, 1, 2, \ldots, 8$). It lists

P = approximate perturbation (Equation 11),
R = rest perturbation obtained by numerical integration of Equation 7c,
P+R = complete perturbation,
σ = complete perturbation derived by Encke's method.

Table 1 illustrates:

1. The very slow increase of R for small intermediate times, compared with the rapid increase of P and σ,

2. $\sigma = P + R$, except for small deviations due to rounding.

Table 1

Perturbations of the x-coordinate of (931) Whittemora for t_0 + 40 n Days.

n	P	R	P+R	σ
0	0.0	0.0	0.0	0
1	8.6	0.0	8.6	8
2	38.5	0.4	38.9	39
3	95.5	3.9	99.4	99
4	182.9	17.9	200.8	201
5	301.4	55.0	356.4	356
6	448.8	133.4	582.2	581
7	620.8	276.6	897.4	896
8	807.8	513.6	1321.4	1320

NOTES: 1. t_0 = 1920 April 29.0.

2. Results expressed in 10^{-7} A.U.

Table 2

Coordinates of (931) Whittemora at t_0 + 1600 Days, in A.U.

h in days	x	y	z
20	0.6061220	2.299933	-0.2915877
40	0.6061201	2.299943	-0.2915872
50	0.6061180	2.299950	-0.2915866
80	0.6061054	2.299979	-0.2915840
100	0.6060915	2.300004	-0.2915813
160	0.6060304	2.300097	-0.2915702
Exact Value	0.6061222	2.299931	-0.2915879

NOTES: 1. t_0 = 1920 April 29.0.

2. Results expressed in A.U.

Example 2

Method B is used to compute the orbit of the same minor planet, (931) Whittemora. The computations extend to 1600 days (approximately 80 percent of one revolution) beyond the epoch 1920, April 29.0. Table 2 lists the terminal heliocentric rectangular coordinates of the planet in A.U. for several constant values of h, as well as the true values for comparison. The true values are obtained almost perfectly when h = 20 days, and the error barely exceeds 10^{-4} A.U., when the computation is performed in 10 steps of 160 days each.

The remaining examples involve the applications of Methods B and C to a spacecraft in the gravitational field of earth, moon and sun. The orbits are highly eccentric, closely approaching the earth or the moon. Hence, variable step lengths should be chosen.

Example 3

Compute the orbit of Explorer 33, launched July 1, 1966. Integrate for 180 days beyond t_0, the epoch of computation, where t_0 = 1966, July 31. The spacecraft describes over 12 highly eccentric trajectories around the earth and several times closely approaches the surfaces of the earth and the moon (to within approximately 3 and 5 earth radii, respectively). The earth-spacecraft separation is the criterion that measures the effectiveness of the method. The exact 4-body point mass separation is 138801.68 km at t_0 + 170 days and 437108.24 km at t_0 + 180 days. Table 3 gives the deviations of the earth-spacecraft separations at these times, for several runs involving different time-step controls. The deviation always attains its maximum near t_0 + 170 days. Table 3 also gives the machine execution time and the number of steps used in the computation. The last line gives the equivalent information for the JPL-Holdridge program (Reference 5) which accounts for planetary and non-point-mass perturbations. The agreement, it will be noted, is quite satisfactory.

Table 3

Explorer 33: Deviations in Earth Spacecraft Separation.

Deviation of earth-spacecraft distance at t_0 + 170 days (km)	Deviation of earth-spacecraft distance at t_0 + 180 days (km)	Time (seconds)	No. of computing steps	Method	Time-step criterion
2495	915	50	Not Available	B	$K = 10^{-5}$
766	279	88	Not Available	B	$K = 10^{-6}$
245	90	157	Not Available	B	$K = 10^{-7}$
41	11	421	2440	C	4h = 3 hrs
8146	3052	38	218	C	A = 0 B = 1.5 C = 1.5
940	286	53	310	C	A = 0 B = 2.5 C = 3.5
722	220	61	356	C	A = 0 B = 2.5 C = 1.5
616	186	70	408	C	A = 0 B = 2.5 C = 1
86	27	86	502	C	A = 0 B = 4 C = 2.5
42	13	122	711	C	A = 0 B = 4 C = 1
1	1	190	1104	C	A = 0 B = 8 C = 1.5
0	0	363	2108	C	A = 5 B = 16 C = 1
930	436	about 15 min	Not Available	JPL-Holdridge*	

* - Accounts for planetary and non-point mass effects.

The inverse relation between speed and accuracy can be seen best from Figure 1. It plots maximum deviation of the earth-spacecraft distance versus machine time for Explorer 33, computed 180 days beyond the epoch. Note that Method C is superior to Method B in this example and probably in others.

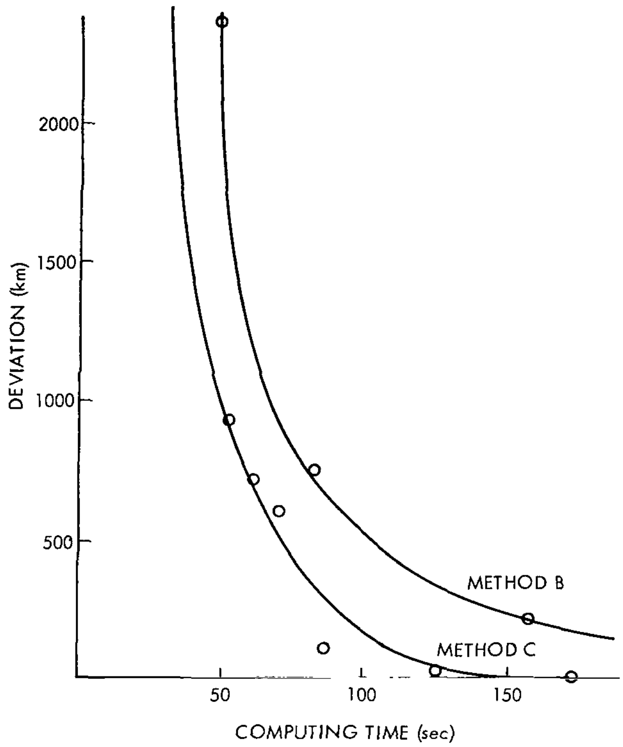

Figure 1—Maximum deviation in earth-spacecraft separation for Explorer 33.

NOTES

1. All machine times refer to the IBM 7094 Model 1.

2. The number of time-steps is available for Method C only. A time-step extends from t_i to $t_{i+1} = t_i + 4h_i$.

Example 4

Consider a typical earth-moon trajectory integrated for 2.5 days beyond the epoch. The trajectory starts less than 450 km from the surface of the earth, i.e., at third-stage cutoff. The exact earth-spacecraft separation for the point-mass earth, moon, sun problem equals 362,141.35 km at t_0 + 2.5 days. Table 4 lists the effects of different time-steps and includes the results of the JPL-Holdridge program, which accounts for planetary perturbations. The results agree very well with the point-mass JPL-Holdridge program.

Example 5

Consider a typical lunar orbit with the following initial characteristics: aposelenium = 8634 km, periselenium = 3443 km, period = 702 min, e = 0.429. The integration extends for 180 days beyond the epoch July 4, 1966. Table 5 gives the terminal discrepancy in moon-spacecraft separation for different time-steps. The exact value is 8562.37 km. There is close agreement between the simple 4-body solution and the JPL program, even though the latter accounts for harmonics and planetary perturbations.

COMPUTER PROGRAMS

Four double precision computer programs have been written in FORTRAN IV. They are entitled STUMPFF1, STUMPFF2, STUMPFF3, and STUMPFF4, and are reproduced in Appendix A. This section briefly describes points of interest to users of the programs.

The programs generate their own ephemerides; this facilitates programming and saves memory locations.

Table 4

Typical Earth-Moon Trajectory: Deviations in Earth-Spacecraft Separation at $t_0 + 2.5$ Days.

Deviation of earth-spacecraft distance (km)	Time (seconds)	No. of steps	Method	Time-step criterion	Comments
0.73	1.44	7	C	$D = 5 \times 10^{-6}$	
0.37	1.63	8	C	$D = 2 \times 10^{-6}$	
0.06	1.99	10	C	$D = 9 \times 10^{-7}$	
0.03	2.18	11	C	$D = 7 \times 10^{-7}$	
0	3.26	17	C	$D = 2 \times 10^{-7}$	
0.51	about one minute		JPL-Holdridge		Planetary perturbations included. Point-mass bodies assumed.
2791.98	about one minute		JPL-Holdridge		Planetary perturbations included. Earth and moon harmonics included.

STUMPFF1

The program is set up to compute Example 2 by Method B. The central body is the sun; locations X0, Y0, Z0, XD0, YD0, and ZD0 contain the initial position and velocity values of (931) Whittemora; X10, Y10, Z10, XD10, YD10, and ZD10 contain the initial values for Jupiter. The unit of mass is the mass of the sun; the mass of Jupiter = 1/1047.35; the mass of the minor planet equals zero. The unit of length is the A.U. and the unit of time $58^d.13244$. The step size, in days, is in location DIFF. The program prints the number of days beyond the epoch, and the coordinates of the minor planet and Jupiter. The program stops when TAU ≥ TAUMAX, where TAU = 0.0172021 × DIFF.

STUMPFF2

The program is set up to compute Example 3 by Method B. Q_0 is the spacecraft; Q_1 is the earth, which is the central body; Q_2 is the moon; and Q_3 is the sun. Locations Y10(I), Y12(I), and Y13(I), (I = 1, 2, 3, 4, 5, 6), contain the initial values of $q_{10}(0)$, $\dot{q}_{10}(0)$, $q_{12}(0)$, $\dot{q}_{12}(0)$, $q_{13}(0)$, $\dot{q}_{13}(0)$ in canonical units. The unit of length is the mean earth radius of 6378.165 km, the unit of mass is the mass of the earth, and the unit of time equals 806.813645 seconds. The program prints the initial conditions. Then it prints four lines every N^{th} day, namely:

(Line 1) q_{10}, $\dot{q}_{10}$, number of days since epoch,

(Line 2) q_{12}, $\dot{q}_{12}$,

(Line 3) q_{13}, $\dot{q}_{13}$,

(Line 4) Q, r_{10}, r_{20}, number of days since epoch.

Table 5

Typical Lunar Orbit: Deviations in Moon-Spacecraft Separation at t_0 + 180 Days.

Deviation of moon-spacecraft distance (km)	Time (minutes)	No. of steps	Method	Time-step criterion	Comments
1068	6.9	2416	C	A = 2 B = 1.5 C = 1	
43	13.8	4814	C	A = 2 B = 3 C = 1	
51	17.5	6104	C	A = 5 B = 5 C = -1	
14	23.0	8020	C	A = 2 B = 5 C = 1	
22	25.2	8762	C	A = 6 B = 9 C = -1	
0	87.5	30459	C	A = 2 B = 19 C = 1	
28	29.8	9230	C	D = 3×10^{-7}	
16	35.8	11108	C	D = 2×10^{-7}	
0	68.1	21108	C	D = 6×10^{-8}	
28	about 120		JPL-Holdridge		Includes harmonics and planetary effects. Encke mode used.
22	about 175		JPL-Holdridge		Includes harmonics and planetary effects. Cowell mode used.

The results are printed in km and km/sec. At present, the program prints every tenth day. This can be changed by altering "10.D00" in the two consecutive instructions:

```
IJI = IDINT (TIMED/10.D00),
IJ2 = IDINT (TIMEDN/10.D00).
```

The program stops TIMEMX days beyond the epoch.

STUMPFF3

The program computes Example 5 by Method C, using the A, B, C time-step criterion. The bodies are numbered as in STUMPFF2 and the units of length, mass, and time are defined as in STUMPFF2. The first printout gives the initial conditions followed by A, B, C, and m_1. Subsequent printouts are four lines each:

(Line 1) q_{10}, $\dot{q}_{10}$, number of days since epoch

(Line 2) q_{12}, $\dot{q}_{12}$

(Line 3) q_{13}, $\dot{q}_{13}$

(Line 4) Contains five words. The first is immaterial; the others are r_{10}, r_{20}, number of days since epoch, and number of computing steps.

There are eight input cards per case, and two or more cases may be stacked. The fields of the input cards end in columns 16, 32, 48, and 64, and contain the following floating point input:

(Card 1) $q_{10}(0)$

(Card 2) $\dot{q}_{10}(0)$

(Card 3) $q_{12}(0)$

(Card 4) $\dot{q}_{12}(0)$

(Card 5) $q_{13}(0)$

(Card 6) $\dot{q}_{13}(0)$

Card 7 has four fields that specify A, B, ONEDAY, and TIMEMX. The only field of Card 8 specifies C. The time-step criterion uses A, B, and C; ONEDAY governs the frequency of the printing; and a case is terminated after TIMEMX days of computation. If column 72 of Card 7 equals 1, a new case will be processed after the present case; the last case must contain a blank in column 72 of Card 7.

STUMPFF4

The program computes Example 5 by Method C, using the D time-step criterion. Everything is as in STUMPFF3 except:

1. Input Card 8 does not exist. Card 7 contains D, h_{init}, ONEDAY, TIMEMX. D governs the time step; h_{init} gives the initial value of h; and ONEDAY and TIMEMX are as in STUMPFF3.

2. The line printed after the initial conditions contains D, h_{min}, one immaterial word, and m_1.

Goddard Space Flight Center
National Aeronautics and Space Administration
Greenbelt, Maryland, October 2, 1967
311-02-01-01-51

REFERENCES

1. Stumpff, K., "Untersuchungen ueber das Problem der speziellen Stoerungen in den rechtwinkligen Koordinaten," Astron. Nachr. 273: 105, 1942.

2. Stumpff, K., "Neue Formeln und Hilfstafeln zur Ephemeridenrechnung," Astron. Nachr. 275: 108, 1947.

3. Stumpff, K., "Himmelsmechanik," Vol. 1, Berlin: Deutscher Verlag der Wissenschaften, 1959.

4. Stumpff, K., "Himmelsmechanik," Vol. 2, Berlin: Deutscher Verlag der Wissenschaften, 1965.

5. Holdridge, D. B., "Space Trajectories Program for the IBM 7090 Computer," NASA JPL Tech. Report No. 32-223, March 2, 1962.

Appendix

Symbolic Listing of FORTRAN IV Programs STUMPFF1, STUMPFF2, STUMPFF3, and STUMPFF4

```
$JOB  0510      STUMPFF1  BY E.H. WEISS          RIGGS BLDG   X 7319
$PAUSE
$EXECUTE        IBJOB
$IBJOB          GO,LOGIC
C      STUMPFF 1 BY E.H.WEISS  RIGGS BLDG  X 7319
$IBFTC MAIN     LIST,REF,DECK
       DOUBLE PRECISION
      DQK,DIFF,TAUMAX,TAU,TAU1,XO,YO,ZO,XDO,YDO,ZDO,
      DX10,Y10,Z10,XD10,YD10,ZD10,QKAP0,QKAP1,QKAP2,
      DTE1,TE2,TE3,TE4,TE5,TE6,TE7,TE8,TE9,TE10,TE11,TE12,
      DSX,SY,SZ,SDX,SDY,SDZ,SIGX,SIGY,SIGZ
       QK= 0.0172021D00
       DIFF=100.D00
       WRITE (3,30)
    30 FORMAT (1H1)
       TAUMAX=28.D00
       TAU= 0.D00
       TAU1=QK*DIFF
       XO=-3.244464D00
       YO=.288260D00
       ZO=.572613D00
       XDO=-.164671D00
       YDO=-.506418D00
       ZDO=.069948D00
       X10=-4.096193D00
       Y10=3.11893D00
       Z10=1.43980D00
       XD10=-.19726D00
       YD10=-.20136D00
       ZD10=-.08162D00
       QKAP0= 1.D00
       QKAP1= 1.D00/1047.35D00
       QKAP2= 1.D00+QKAP1
       TE1=DSQRT(QKAP2)/(QK*40.D00)
       XD10=XD10*TE1
       YD10=YD10*TE1
       ZD10=ZD10*TE1
    31 TAU= TAU+ TAU1
       CALL SUB1(TAU1,XO,YO,ZO,XDO,YDO,ZDO,QKAP0,
      1 SX,SY,SZ,SDX,SDY,SDZ,SIGX,SIGY,SIGZ)
       TE1= XO+SX
       TE2= YO+SY
       TE3= ZO+SZ
       TE4= XDO+SDX
       TE5= YDO+SDY
       TE6= ZDO+SDZ
       TE7=XO-X10
       TE8=YO-Y10
       TE9=ZO-Z10
       TE10=XDO-XD10
       TE11=YDO-YD10
       TE12=ZDO-ZD10
       CALL SUB1(TAU1,TE7,TE8,TE9,TE10,TE11,TE12,QKAP1,
      1 SX,SY,SZ,SDX,SDY,SDZ,SIGX,SIGY,SIGZ)
      TE1=TE1+SIGX
      TE2=TE2+SIGY
      TE3=TE3+SIGZ
      TE4=TE4+ SDX
      TE5=TE5+ SDY
      TE6=TE6+ SDZ
      CALL SUB1(TAU1,X10,Y10,Z10,XD10,YD10,ZD10,QKAP2,
     1 SX,SY,SZ,SDX,SDY,SDZ,SIGX,SIGY,SIGZ)
      X0= TE1+ QKAP1*SIGX
      Y0= TE2+ QKAP1*SIGY
      Z0=TE3+QKAP1*SIGZ
      XD0=TE4+ QKAP1*SDX
      YD0=TE5+ QKAP1*SDY
      ZD0=TE6+ QKAP1*SDZ
      X10= X10+SX
      Y10= Y10+SY
      Z10= Z10+SZ
      XD10=XD10+SDX
      YD10=YD10+SDY
      ZD10=ZD10+SDZ
      WRITE (3,35)  TAU,X0,Y0,Z0,XD0,YD0,ZD0
   35 FORMAT(1X6H TAU =,D15.7,13H SMALL BODY =,6(D15.7,1X))
      WRITE (3,36) X10,Y10,Z10,XD10,YD10,ZD10
   36 FORMAT(1X34H                 JUPITER=         ,6(D15.7,1X))
      IF(TAU.LE.TAUMAX)GO TO 31
      RETURN
      END
$IBFTC SUB1     LIST,REF,DECK
      SUBROUTINE SUB1 (TAU,X0,Y0,Z0,XD0,YD0,ZD0,QKAP,
     1    SX,SY,SZ,SDX,SDY,SDZ,SIGX,SIGY,SIGZ)
      DOUBLE PRECISION
     D TAU,X0,Y0,Z0,XD0,YD0,ZD0,QKAP,
     D SX,SY,SZ,SDX,SDY,SDZ,SIGX,SIGY,SIGZ,
     D R0,QMU0,QKSI0,SIG0,ETA0,OME0,EPS0,ZETA0,RHO0,CHI0,
     DTEMP1,TEMP2,
     DZ,DELTA,C1,C2,C3,
     D FM,G,FD,GDM
      R0=DSQRT(X0**2+Y0**2+Z0**2)
      QMU0= QKAP/R0**3
      QKSI0= QMU0*TAU**2
      SIG0= (X0*XD0+ Y0*YD0+ Z0*ZD0)/R0**2
      ETA0= SIG0*TAU
      OME0=(XD0**2+YD0**2+ZD0**2)/R0**2
      EPS0= OME0- QMU0
      ZETA0=EPS0*TAU**2
      RHO0= QMU0- EPS0
      CHI0= RHO0*TAU**2
      CALL SUB2(ETA0,ZETA0,CHI0,Z,DELTA,C1,C2,C3)
      FM= C2*QKSI0*Z**2
      G=TAU*(1.D00-(C3*QKSI0*Z**3))
      FD=-(C1*QKSI0*Z)/(DELTA*TAU)
      GDM= C2*QKSI0*Z**2/DELTA
      SX= G*XD0- FM*X0
      SY= G*YD0- FM*Y0
      SZ= G*ZD0- FM*Z0
      SDX= FD* X0- GDM*XD0
      SDY= FD* Y0- GDM*YD0
      SDZ= FD* Z0- GDM*ZD0
      TEMP1=-QKSI0*Z**2
      TEMP2= C3*Z*TAU
      SIGX= TEMP1*(C2*X0+TEMP2*XD0)
      SIGY= TEMP1*(C2*Y0+TEMP2*YD0)
      SIGZ=TEMP1*(C2*Z0+TEMP2*ZD0)
      RETURN
      END
```

```
$IBFTC SUB2     LIST,REF,DECK
      SUBROUTINE SUB2 (ETA,ZETA,CHI,Z,DELTA,C1,C2,C3)
      DOUBLE PRECISION
     D QLA, Z, C1, C2, C3, ETA, ZETA, CHI,
     DDELTA,TOL1,QH
      ITER=0
      Z=1.0D00
      TOL=1.D-08
  201 ITER=ITER+1
      QLA=CHI*Z*Z
      C2=.5D00*(1.DC0-(QLA/12.D00)*(1.D00-(QLA/30.DC0)*
     1    (1.DC0-(QLA/56.D00)*(1.DC0-(QLA/90.D00)))))
      C3= (1.DC0/6.D00)*(1.D00-(QLA/20.DC0)*(1.D00-(QLA/42.D00)*
     1   (1.D00-(QLA/72.D00)*(1.DC0-(QLA/110.D00)))))
      C1=1.DC0-(QLA*C3)
      DELTA=1.DC0+C1*ETA*Z +C2*ZETA*Z**2
      QH= C2*ETA*Z**2+C3*ZETA*Z**3+Z-1.DC0
      Z=Z- QH/DELTA
      IF(DABS(QH).LE.TOL1)RETURN
      IF(ITER.LE.10)GO TO 201
      RETURN
```

```
$JOB  0510      STUMPFF2  BY E.H. WEISS       RIGGS BLDG  X7319
$PAUSE
$EXECUTE        IBJOB
$IBJOB          GO,LOGIC
$IBFTC MAIN     LIST,REF,NODECK
      DOUBLE PRECISION
     DERR,TIMECN,ITDN,ITD,
     DX10,X12,X13,XTEMP,XKEP,XDEL,
     DM1,M2,M3,M12,M13,M23,M2F12,M2F23,M3F13,M3F23,
     DCML,CMV,CMT,Y10,Y12,Y13,TIMED,TIMEMX,
     DTAUD,TAU,R10,R20,TEMP,Q,R10E2,R2CE2,
     DZ1,Z2,Z3,Z4,Z5,Z6,
     DL,C1,C2,C3,DELTA,H,
     DRE2,R,KSI,ETA,ZETA,CHI
      COMMON X10(12),X12(12),X13(12),XTEMP(6),XKEP(6),XDEL(6),
     CM1,M2,M3,M12,M13,M23, M2F12, M2F23, M3F13,M3F23,
     CCML,CMV,CMT,Y10(6),Y12(6),Y13(6),TIMED,TIMEMX,
     CTAUD,TAU,R10,R20,TEMP,Q,R10E2,R2CE2,
     CZ1,72,Z3,Z4,Z5,Z6,
     CL,C1,C2,C3,DELTA,H,
     CRE2,R,KSI,ETA,ZETA,CHI,DUM(1000)
C     STUMPFF2 BY EH WEISS RIGGS
      Y10(1)=.18352640D06
      Y10(2)=-.24094338D06
      Y10(3)=-.36452764D05
      Y10(4)=.10044146D01
      Y10(5)=-.32081304D00
      Y10(6)=-.15168001D00
      Y12(1)= .21384734D06
      Y12(2)=-.29619053D06
      Y12(3)=-.16430464D06
      Y12(4)= .84600696D00
      Y12(5)= .47626001D00
      Y12(6)= .17218044D00
      TIMFMX=181.D00
      Y13(1)=-.93856134D08
      Y13(2)= .10949798D09
      Y13(3)= .47486129D08
      Y13(4)=-.22920520D02
      Y13(5)=-.16789843D02
      Y13(6)=-.72817681D01
      CML=6378.165D00
      CMT=806.813645D00
      CMV=CML/CMT
C      INPUT IS IN  KILOMETERS AND SECONDS.
C     MULTIPLY CANONICAL BY CML,CMT OR CMV, TC OBTAIN METRIC .
C     TAUD IS IN DAYS. TAU IS CANONICAL.
      ERR=1.D-06
      Q=1.D-07
      M1=1.D00
      M2=1.D00/81.30150052D00
      M3=332951.2658D00
      M12=M1+M2
      M13=M1+M3
      M23=M2+M3
      M2F12=M2/M12
      M2F23=M2/M23
      M3F13=M3/M13
      M3F23=M3/M23
      Z1=.9D00
      Z2=.9D00
      Z3=.9D00
      Z4=.9D00
      Z5=.9D00
      Z6=.9D00
      DO 25 I=1,3
      X10(I)=Y10(I)/CML
      X12(I)=Y12(I)/CML
      X13(I)=Y13(I)/CML
      X10(I+3)=Y10(I+3)/CMV
      X12(I+3)=Y12(I+3)/CMV
      X13(I+3)=Y13(I+3)/CMV
   25 CONTINUE
      TIMED =0.D00
      WRITE(3,26)(Y10(I),I=1,6),TIMED
      WRITE(3,19)(Y12(I),I=1,6)
      WRITE(3,19)(Y13(I),I=1,6)
   26 FORMAT(1H1,6D16.8,D16.8)
    9 TAU=DSQRT(DSQRT(ERR/Q))
      TAU=DMIN1(TAU,100.D00)
      TAUD=TAU*CMT/86400.D00
      TIMEDN=TIMED+TAUD
      IJ1=IDINT(TIMED/10.D00)
      IJ2=IDINT(TIMEDN/10.D00)
      IF(IJ1.EQ.IJ2)GO TO 60
      IJ2=IDINT(TIMEDN)
      TIMEDN=IJ2
      TAUD=TIMEDN-TIMED
      TAU=TAUD*86400.D00/CMT
      TIMED=TIMEDN
      IPR=1
   61 CONTINUE
      CALL SUB1(X10,M1,Z1)
      DO 10 I=1,6
      X10(I+6)= XKEP(I)
   10 CONTINUE
      CALL SUB1(X12,M12,Z2)
      DO 11 I=1,6
      X10(I+6)=X10(I+6)+M2F12*XDEL(I)
      X12(I+6)=XKEP(I)
      X13(I+6)=M2F12*XDEL(I)
   11 CONTINUE
      CALL SUB1(X13,M13,Z3)
      DO 12 I=1,6
      X10(I+6)= X10(I+6)+M3F13*XDEL(I)
      X12(I+6)= X12(I+6)+M3F13*XDEL(I)
      X13(I+6)= X13(I+6)+XKEP(I)
   12 CONTINUE
      DO 13 I=1,6
      XTEMP(I)=X10(I)-X12(I)
   13 CONTINUE
      CALL SUB1(XTEMP,M2,Z4)
      DO 14 J=1,6
      X10(J+6)=X10(J+6)+XDEL(J)
   14 CONTINUE
      DO 15 J=1,6
      XTEMP(J)=X10(J)-X13(J)
   15 CONTINUE
      CALL SUB1(XTEMP,M3,Z5)
      DO 16 J=1,6
      X10(J+6)=X10(J+6)+XDEL(J)
   16 CONTINUE
      DO 17 J=1,6
      XTEMP(J)=X13(J)-X12(J)
   17 CONTINUE
      CALL SUB1(XTEMP,M23,Z6)
      DO 29 J=1,6
      X12(J)=X12(J+6)-M3F23*XDEL(J)
      X13(J)=X13(J+6)+M2F23*XDEL(J)
      X10(J)=X10(J+6)
   29 CONTINUE
      DO 18 J=1,3
      Y10(J)=X10(J)*CML
      Y12(J)=X12(J)*CML
      Y13(J)=X13(J)*CML
      Y10(J+3)=X10(J+3)*CMV
      Y12(J+3)=X12(J+3)*CMV
      Y13(J+3)=X13(J+3)*CMV
   18 CONTINUE
      R10E2=X10(1)**2+X10(2)**2+X10(3)**2
      R20E2=(X10(1)-X12(1))**2+(X10(2)-X12(2))**2+(X10(3)-X12(3))**2
      R10=DSQRT(R10E2)
      R20=DSQRT(R20E2)
      Q=(M2/(R10E2*R20E2))*(M1/R10+M1/R20)+
     1 (M2/3600.D00)*(M1/R10**3+M1/R20**3)+
     2 (M3/23400.D00**3)*(M2/R20E2+M1/R10E2)
      R10=R10*CML
      R20=R20*CML
      IF(IPR.EQ.0)GO TO 65
      WRITE(3,20)(Y10(I),I=1,6),TIMED
      WRITE(3,19)(Y12(I),I=1,6)
      WRITE(3,19)(Y13(I),I=1,6)
   19 FORMAT(1H ,6D16.8)
   20 FORMAT(1H0,7D16.8)
      WRITE(3,28)Q,R10,R20,TIMED
   28 FORMAT(1H ,4D16.8)
   65 CONTINUE
      IF(TIMED .LE.TIMEMX)GO TO 9
      STOP
   60 TIMED=TIMEDN
      IPR=0
      GO TO 61
      END
$IBFTC SUB1     LIST,REF,NODECK
      SUBROUTINE SUB1 (X,M,Z)
      DOUBLE PRECISION
     DX(6),M,Z,
     DX10,X12,X13,XTEMP,XKEP,XDEL,
     DM1,M2,M3,M12,M13,M23,M2F12,M2F23,M3F13,M3F23,
     DCML,CMV,CMT,Y10,Y12,Y13,TIMED,TIMEMX,
     DTAUD,TAU,R10,R20,TEMP,Q,R10E2,R20E2,
     DZ1,Z2,Z3,Z4,Z5,Z6,
     DL,C1,C2,C3,DELTA,H,
     DRE2,R,KSI,ETA,ZETA,CHI
      COMMON X10(12),X12(12),X13(12),XTEMP(6),XKEP(6),XDEL(6),
     CM1,M2,M3,M12,M13,M23, M2F12, M2F23, M3F13,M3F23,
     CCML,CMV,CMT,Y10(6),Y12(6),Y13(6),TIMED,TIMEMX,
     CTAUD,TAU,R10,R20,TEMP,Q,R10E2,R20E2,
     CZ1,Z2,Z3,Z4,Z5,Z6,
     CL,C1,C2,C3,DELTA,H,
     CRE2,R,KSI,ETA,ZETA,CHI,DUM(1COO)
      RE2=X(1)**2+X(2)**2+X(3)**2
      R=DSQRT(RE2)
      KSI=M*(TAU**2)/R**3
      ETA=(X(1)*X(4)+X(2)*X(5)+X(3)*X(6))*TAU/RE2
      ZETA=(X(4)**2+X(5)**2+X(6)**2)*(TAU**2)/RE2-KSI
      CHI=KSI-ZETA
      CALL SUB2 (Z)
      DO 40 I=1,3
      XDEL(I)=(-KSI*Z**2)*(C2*X(I)+C3*Z*TAU*X(I+3))
      XDEL(I+3)=(-KSI*Z)*(C1*X(I)+C2*Z*TAU*X(I+3))/(DELTA*TAU)
      XKEP(I)=X(I)+TAU*X(I+3)+XDEL(I)
      XKEP(I+3)=X(I+3)+XDEL(I+3)
   40 CONTINUE
      RETURN
      END
$IBFTC SUB2     LIST,REF,NODECK
      SUBROUTINE SUB2(Z)
      DOUPLE PRECISION
     DZ,
     DX10,X12,X13,XTEMP,XKEP,XDEL,
     DM1,M2,M3,M12,M13,M23,M2F12,M2F23,M3F13,M3F23,
     DCML,CMV,CMT,Y10,Y12,Y13,TIMEC,TIMEMX,
     DTAUC,TAU,R10,R20,TEMP,Q,R1OE2,R2CE2,
     DZ1,72,Z3,Z4,Z5,Z6,
     DL,C1,C2,C3,DELTA,H,
     CRE2,R,KSI,ETA,ZETA,CHI
      COMMON X10(12),X12(12),X13(12),XTEMP(6),XKEP(6),XDEL(6),
     CM1,M2,M3,M12,M13,M23, M2F12, M2F23, M3F13,M3F23,
     CCML,CMV,CMT,Y10(6),Y12(6),Y13(6),TIMED,TIMEMX,
     CTAUC,TAU,R10,R20,TEMP,Q,R1OE2,R2CE2,
     CZ1,Z2,Z3,Z4,Z5,Z6,
     CL,C1,C2,C3,DELTA,H,
     CRE2,R,KSI,ETA,ZETA,ChI,DUM(1CCO)
      ITER=0
   30 ITER=ITER+1
      L=CHI*Z**2
      C2=.5D00*(M1-(L/12.D00)*(M1-(L/3C.D00)*
     1    (M1-(L/56.D00)*(M1-(L/90.D00)))))
      C3= (M1/6.D00)*(M1-(L/2C.D00)*(M1-(L/42.D00)*
     1    (M1-(L/72.D00)*(M1-(L/11C.DCC)))))
      C1=M1-L*C3
      DELTA=M1+C1*ETA*Z+C2*ZETA*Z**2
      H=C2*ETA*Z**2+C3*ZETA*Z**3+Z-M1
      Z=Z-H/DELTA
      IF(DABS(H).LE.1.D-07)RETURN
      IF(ITER.LE.10)GC TC 30
      RETURN
      END
```

```
$IBSYS
$JOB  051C      STUMPFF3 RH HILLIARC ATT. EHW RIGGS BLDG X-7267
$DATE           102766
$EXECUTE        IBJOB
$IBJOB          GO,MAP,LOGIC,SCURCE
$IBFTC MAIN     LIST,REF,NODECK
       DOUBLE PRECISION
     DERR,TIMEDN,ITDN,ITD,A,CX10,
     DX10,X12,X13,XTEMP,XKEP,XCEL,
     DST2X10,ST2X12,ST2X13,ST2X20,ST2X30,ST2X23,
     DM1,M2,M3,M12,M13,M23,M2F12,M2F23,M3F13,M3F23,
     DCML,CMV,CMT,Y10,Y12,Y13,TIMED,TIMEMX,
     DTAUD,TAU,R10,R20,TEMP,Q,R10E2,R2CE2,
     DL,C1,C2,C3,DELTA,H,XFR1C,XAFR10,
     DXFR12,XFR13,XFR20,XFR23,XFR30,XAFR12,XAFR13,XAFR20,XAFR23,XAFR30,
     DRE2,R,KSI,ETA,ZETA,CHI,R12,R13,R23,R30,Z,
     DRD1C,RD12,RD13,RDD10,RDD12,RDD13,
     DR10F2C,R20E2C,R10C,R20C
      DOUBLE PRECISION WW10,WW12,WW13,WW20,WW30,WW23,
     DFF1C,FF12,FF13,FF20,FF3C,FF23
      DOUBLE PRECISION C
      DOUBLE PRECISION
     DW,TOTDAY,TDAYCU,ONEDAY,SWT,TAUN,COUNT,AT,BT,
     DDEL10,DEL12,DEL13,MONE,X20,DEL20,X30,DEL30,X23,DEL23,
     DEPS10,XAST1C,EPS12,XAST12,EPS13,XAST13,XAST20,XAST23,XAST30,
     DRAST10,RAST12,RAST13,RAST20,RAST23,RAST30,S10,S12,S13,S20,
     DS23,S3C,RDOT10,RDOT12,RDOT13,RD1CF,RD12F,RD13F,RDD10F,RDD12F,
     DRDD13F,RWGT,RDWGT,TSQ240,R4F10,R4F12,R4F13,TAU45,RD4F10,
     DBX1C,BX12,BX13,
     DRD4F12,RD4F13,X4F10,X4F12,X4F13
      DOUBLE PRECISION EPS20,EPS30,EPS23
      COMMON X1C(12),X12(12),X13(12),XTEMP(6),XKEP(6),XDEL(6),
     CM1,M2,M3,M12,M13,M23, M2F12, M2F23, M3F13,M3F23,
     CST2X10(6),ST2X12(6),ST2X13(6),ST2X20(6),ST2X3C(6),ST2X23(6),
     CCML,CMV,CMT,Y10(6),Y12(6),Y13(6),TIMED,TIMEMX,
     CTAUD,TAU,R1C,R20,TEMP,Q,R1CE2,R2CE2,
     CL,C1,C2,C3,DELTA,H,XFR1C(3),XAFR1C(3),
     CRE2,R,KSI,ETA,ZETA,CHI,R12(6),R13(6),R23(6),R30(6),Z,
     CR10F2C,R20E2C,R10C,R20C
      COMMON WW10(6),WW12(6),WW13(6),WW20(6),WW30(6),WW23(6),
     CFF10(6),FF12(6),FF13(6),FF20(6),FF30(6),FF23(6)
      COMMON
     CW,TOTDAY,TDAYCU,ONEDAY,SWT,TAUN,COUNT,AT,BT,
     CTSQ240,BX10(6),BX12(6),BX13(6),
     CRDD12F(6),RDD13F(6),RWGT(4),RDWGT(4),R4F10(3),R4F12(3),
     CRD1D(3),RD12(3),RD13(3),RDD1C(3),RDD12(3),RDD13(3),
     CDEL10(6),DEL12(6),DEL13(6),DEL20(6),DEL23(6),DEL30(6),
     CMONE,X20(6),X23(6),X30(6),EPS10(6),EPS12(6),EPS13(6),
     CXAST1C(6),XAST12(6),XAST13(6),XAST20(6),XAST23(6),XAST30(6),
     CRAST1C(6),RAST12(6),RAST13(6),RAST2C(6),RAST23(6),RAST30(6),
     CS10(6),S12(6),S13(6),S20(6),S23(6),S30(6),RDOT10(6),
     CRDOT12(6),RDOT13(6),RD1CF(6),RD12F(6),RD13F(6),RDD1CF(6),
     CR4F13(3),TAU45,RD4F1C(3),RD4F12(3),RD4F13(3),X4F10(6),X4F12(6),
     CX4F13(6),XFR12(3),XFR13(3),XFR20(3),XFR23(3),XFR30(3),XAFR12(3),
     CXAFR13(3),XAFR20(3),XAFR23(3),XAFR3C(3)
      COMMON EPS20(6),EPS3C(6),EPS23(6)
C     STUMPFF3 BY EH WEISS AND RH HILLIARD RIGGS BLDG
 900  READ(2,8CC)(Y10(I),I=1,6)
      READ(2,8CC)(Y12(I),I=1,6)
      READ(2,8CC)(Y13(I),I=1,6)
 800   FORMAT(3D16.8)
       READ(2,802)AT,BT,ONEDAY,TIMEMX,MORE
 802   FORMAT(4D16.8,7X,I1)
       READ(2,803)C
 803   FORMAT(D16.8)
       CML=6378.165D00
       CMT=((6378.165D00**3)/.3986032D06)
       CMT=DSQRT(CMT)
       CMV=CML/CMT
C        INPUT IS IN  KILOMETERS AND SECONDS.
C      MULTIPLY CANONICAL BY CML,CMT OR CMV, TO OBTAIN METRIC .
C      TAUD IS IN DAYS. TAU IS CANONICAL.
       MONF=1.D00
       M1=1.D00
       M2=1.D00/81.3015D052D00
       M3=332951.2658D00
       M12=M1+M2
       M13=M1+M3
       M23=M2+M3
       M2F12=M2/M12
       M2F23=M2/M23
       M3F13=M3/M13
       M3F23=M3/M23
       TIMED =0.D00
       COUNT=0.D00
       TAUN=0.D00
       ONEDAY=(ONEDAY*86400.D00)/CMT
       TOTDAY=0.D00
       SWT=0.D00
       WRITE(3,26)(Y10(I),I=1,6),TIMED
       WRITE(3,19)(Y12(I),I=1,6)
       WRITE(3,19)(Y13(I),I=1,6)
  26   FORMAT(1H1,6D16.8,D16.8)
       WRITE(3,801)AT,BT,C,M1
 801   FORMAT(4H AT=,D16.8,2X,4H BT=,D16.8,2X,3H C=,D16.8,2X,
      15H M1 =,D16.8)
       RWGT(1)=721.D00
       RWGT(2)=476.D00
       RWGT(3)=245.D00
       RWGT(4)=18.D00
       RDWGT(1)=64.D00
       RDWGT(2)=24.D00
       RDWGT(3)=64.D00
       RDWGT(4)=14.D00
       W=6.D00
       TAU=(W+AT)/BT
 9     CONTINUE
       COUNT=COUNT+1.D00
       TSQ240=(TAU**2)/240.D00
       TAU45=TAU/45.D00
       DO 350 I=1,6
       RD10F(I)=0.D00
       RD12F(I)=0.D00
       RD13F(I)=0.D00
       RDD10F(I)=0.D00
       RDD12F(I)=0.D00
       RDD13F(I)=0.D00
 350   CONTINUE
       DO 341 J=1,4
       TAUN=TAUN+TAU
      DO 25 I=1,3
      BX10(I)=Y10(I)/CML
      BX12(I)=Y12(I)/CML
      BX13(I)=Y13(I)/CML
      BX10(I+3)=Y10(I+3)/CMV
      BX12(I+3)=Y12(I+3)/CMV
      BX13(I+3)=Y13(I+3)/CMV
   25 CONTINUE
      A=J
      TAU=A*TAL
      IPR=1
   61 CONTINUE
C
C     INPUT/M1,X10       OUTPUT/ X10,DEL10
C
      CALL SUB1(BX10,M1)
      DO 100 I=1,6
      X10(I)=XKEP(I)
      DEL10(I)=XDEL(I)
  100 CONTINUE
C
C     INPUT/M1+M2,X12    OUTPUT/ X12,DEL12
C
      CALL SUB1(BX12,M12)
      DO 110 I=1,6
      X12(I)=XKEP(I)
      DEL12(I)=XDEL(I)
  110 CONTINUE
C
C     INPUT/M1+M3,X13    OUTPUT/ X13,DEL13
C
      CALL SUB1(BX13,M13)
      DO 120 I=1,6
      X13(I)=XKEP(I)
      DEL13(I)=XDEL(I)
  120 CONTINUE
      DO 13 I=1,6
      XTEMP(I)=BX10(I)-BX12(I)
   13 CONTINUE
C
C     INPUT/M2,X10-X12   OUTPUT/ X20,DEL20
C
      CALL SUB1(XTEMP,M2)
      DO 130 I=1,6
      X20(I)=XKEP(I)
      DEL20(I)=XDEL(I)
  130 CONTINUE
      DO 15 M=1,6
      XTEMP(M)=BX10(M)-BX13(M)
   15 CONTINUE
C
C     INPUT/M3,X10-X13   OUTPUT/ X30,DEL30
C
      CALL SUB1(XTEMP,M3)
      DO 140 I=1,6
      X30(I)=XKEP(I)
      DEL30(I)=XDEL(I)
  140 CONTINUE
      DO 17 M=1,6
      XTEMP(M)=BX13(M)-BX12(M)
   17   CONTINUE
C
C       INPUT/M2+M3,X13-X12OUTPUT/ X23,DEL23
C
        CALL SUB1(XTEMP,M23)
        DO 150 I=1,6
        X23(I)=XKEP(I)
        DEL23(I)=XDEL(I)
  150   CONTINUE
C
C       COMPUTE EPS10        E10=DEL20+DEL30+M2/(M1+M2)*DEL12+(M3/(M1+M3)
C                                 *DEL13
C
        DO 190 I=1,6
        EPS10(I)=DEL20(I)+DEL30(I)+M2F12*DEL12(I)+M3F13*DEL13(I)
        XAST10(I)=X10(I)+EPS10(I)
C
C       COMPUTE EPS12        E12=(M3/(M1+M3))*DEL13-(M3/(M2+M3))*DEL23
C
        EPS12(I)=M3F13*DEL13(I)-M3F23*DEL23(I)
        XAST12(I)=X12(I)+EPS12(I)
C
C       COMPUTE EPS13        E13=(M2/(M2+M3))*DEL23+(M2/(M1+M2))*DEL12
C
        EPS13(I)=M2F23*DEL23(I)+M2F12*DEL12(I)
        XAST13(I)=X13(I)+EPS13(I)
        EPS20(I)=EPS10(I)-EPS12(I)
        EPS30(I)=EPS10(I)-EPS13(I)
        EPS23(I)=EPS13(I)-EPS12(I)
        XAST20(I)=XAST10(I)-XAST12(I)
        XAST30(I)=XAST10(I)-XAST13(I)
        XAST23(I)=XAST13(I)-XAST12(I)
  190   CONTINUE
C
C       COMPUTE R(IJ) SQUARED AND RAST(IJ) SQUARED
C                            R(IJ)**2= X(IJ)(1)**2+X(IJ)(2)**2+X(IJ)(3)**2
C
        R10=DSQRT(X10(1)*X10(1)+X10(2)*X10(2)+X10(3)*X10(3))
        R12=DSQRT(X12(1)*X12(1)+X12(2)*X12(2)+X12(3)*X12(3))
        R13=DSQRT(X13(1)*X13(1)+X13(2)*X13(2)+X13(3)*X13(3))
        R20=DSQRT(X20(1)*X20(1)+X20(2)*X20(2)+X20(3)*X20(3))
        R23=DSQRT(X23(1)*X23(1)+X23(2)*X23(2)+X23(3)*X23(3))
        R30=DSQRT(X30(1)*X30(1)+X30(2)*X30(2)+X30(3)*X30(3))
        RAST10=DSQRT(XAST10(1)**2+XAST10(2)**2+XAST10(3)**2)
        RAST12=DSQRT(XAST12(1)**2+XAST12(2)**2+XAST12(3)**2)
        RAST13=DSQRT(XAST13(1)**2+XAST13(2)**2+XAST13(3)**2)
        RAST20=DSQRT(XAST20(1)**2+XAST20(2)**2+XAST20(3)**2)
        RAST23=DSQRT(XAST23(1)**2+XAST23(2)**2+XAST23(3)**2)
        RAST30=DSQRT(XAST30(1)**2+XAST30(2)**2+XAST30(3)**2)
C
C       COMPUTE S10,S12,S13,S20,S23,S30 WHERE IN GENERAL
C                            S(IJ)=X(IJ)/R(IJ)**3*XAST(IJ)/RAST(IJ)**3
C
        DO 250 I=1,3
        S10(I)=X10(I)/(R10*R10*R10)-XAST10(I)/RAST10**3
        S12(I)=X12(I)/(R12*R12*R12)-XAST12(I)/RAST12**3
        S13(I)=X13(I)/(R13*R13*R13)-XAST13(I)/RAST13**3
        S20(I)=X20(I)/(R20*R20*R20)-XAST20(I)/RAST20**3
        S23(I)=X23(I)/(R23*R23*R23)-XAST23(I)/RAST23**3
        S30(I)=X30(I)/(R30*R30*R30)-XAST30(I)/RAST30**3
  250 CONTINUE
C
C
C     COMPUTE RDOT (IJ)
C                             RDOT(IJ)=(M(I)+M(J))*S(IJ)+M(K)*(S(IK)+S(KJ))
C                                    +M(L)*(S(IL)+S(LJ))
C                             WHERE  I .NE. J .NE. K .NE. L
C                             AND (I,J,K,L)=(0,1,2,3)
C
C
      DO 260 I=1,3
      RDOT10(I)=M1*S10(I)+M2*(S12(I)+S20(I))+M3*(S13(I)+S30(I))
      RDOT12(I)=M12*S12(I)+M3*(S13(I)-S23(I))
      RDOT13(I)=M13*S13(I)+M2*(S12(I)+S23(I))
  260 CONTINUE
      TAU=TAU/A
      DO 340 I=1,3
      RD10(I)=RWGT(J)*RDOT10(I)
      RD10F(I)=RD10F(I)+RD10(I)
      RD12(I)=RWGT(J)*RDOT12(I)
      RD12F(I)=RD12F(I)+RD12(I)
      RD13(I)=RWGT(J)*RDOT13(I)
      RD13F(I)=RD13F(I)+RD13(I)
      RDD10(I)=RDWGT(J)*RDOT10(I)
      RDD10F(I)=RDD10F(I)+RDD10(I)
      RDD12(I)=RDWGT(J)*RDOT12(I)
      RDD12F(I)=RDD12F(I)+RDD12(I)
      RDD13(I)=RDWGT(J)*RDOT13(I)
      RDD13F(I)=RDD13F(I)+RDD13(I)
  340 CONTINUE
  341 CONTINUE
      DO 370 I=1,3
      R4F10(I)=TSQ240*RD10F(I)
      R4F12(I)=TSQ240*RD12F(I)
      R4F13(I)=TSQ240*RD13F(I)
      RD4F10(I)=TAU45*RDD10F(I)
      RD4F12(I)=TAU45*RDD12F(I)
      RD4F13(I)=TAU45*RDD13F(I)
  370 CONTINUE
      CX10=DSQRT((R4F10(1)**2)+(R4F10(2)**2)+(R4F10(3)**2))
  380 DO 390 I=1,3
      X4F10(I)=XAST10(I)+R4F10(I)
      X4F12(I)=XAST12(I)+R4F12(I)
      X4F13(I)=XAST13(I)+R4F13(I)
      X4F10(I+3)=XAST10(I+3)+RD4F10(I)
      X4F12(I+3)=XAST12(I+3)+RD4F12(I)
      X4F13(I+3)=XAST13(I+3)+RD4F13(I)
  390 CONTINUE
   69 DO 18 J=1,3
      Y10(J)=X4F10(J)*CML
      Y12(J)=X4F12(J)*CML
      Y13(J)=X4F13(J)*CML
      Y10(J+3)=X4F10(J+3)*CMV
      Y12(J+3)=X4F12(J+3)*CMV
      Y13(J+3)=X4F13(J+3)*CMV
   18 CONTINUE
      R10E2=Y10(1)**2+Y10(2)**2+Y10(3)**2
      R20E2=(Y10(1)-Y12(1))**2+(Y10(2)-Y12(2))**2+(Y10(3)-Y12(3))**2
      R10=DSQRT(R10E2)
      R20=DSQRT(R20E2)
      R1CF2C=X4F10(1)**2+X4F10(2)**2+X4F10(3)**2
      R20F2C=(X4F10(1)-X4F12(1))**2+(X4F10(2)-X4F12(2))**2+(X4F10(3)-X4F
     112(3))**2
      R10C=DSQRT(R10E2C)
      R20C=DSQRT(R20E2C)
      W=DMIN1(R10C,C*R20C)
      TAU=(W+AT)/BT
      IF(SWT-1.CCC)1CC6,1C01,1C06
 1006 IF(ONEDAY-(TAUN+A*TAL))1CCC,1C05,9
 1005 SWT=1.COC
      GC TO 9
 10C0 TAU=(ONEDAY-TAUN)/A
      SWT=1.DOC
      GO TO 9
 1001 TOTCAY=TCTDAY+ONECAY
      TDAYCL=(TCTCAY*CMT)/864CC.CCC
      SWT=0.COC
      TIMEC=TDAYCU
      IF(IPR.EQ.0)GC TO 65
      WRITE(3,2C)(Y10(I),I=1,6),TIMED
      WRITE(3,19)(Y12(I),I=1,6)
      WRITE(3,19)(Y13(I),I=1,6)
  19  FORMAT(1H ,6D16.8)
  20  FORMAT(1HC,7D16.8)
 2050 FORMAT(1H 3D24.16)
      WRITE(3,28)CX10,R1C,R20,TIMEC,COUNT
  28  FORMAT(1H ,5D16.8)
      IF(IDINT(TIMEMX)-IDINT(TDAYCL)) 1C03,10C3,1002
 1002 CONTINUE
      TAUN=0.DCC
      GO TO 9
 1003 IF(MORE.EQ.1)GO TC 9C0
  65  CONTINLE
      STOP
      END
$IBFTC SUB1     LIST,REF,NODECK
      SUBROUTINE SUB1 (X,M)
      DOUBLE PRECISION
     DX(6),M,
     DERR,TIMECN,ITDN,ITC,A,CX10,
     DX10,X12,X13,XTEMP,XKEP,XDEL,
     DST2X10,ST2X12,ST2X13,ST2X20,ST2X3C,ST2X23,
     DM1,M2,M3,M12,M13,M23,M2F12,M2F23,M3F13,M3F23,
     DCML,CMV,CMT,Y10,Y12,Y13,TIMEC,TIMEMX,
     DTAUD,TAU,R1C,R20,TEMP,Q,R1CE2,R2CE2,
     DL,C1,C2,C3,DELTA,H,XFR1C,XAFR10,
     DXFR12,XFR13,XFR20,XFR23,XFR3C,XAFR12,XAFR13,XAFR20,XAFR23,XAFR30,
     DRE2,R,KSI,ETA,ZETA,CHI,R12,R13,R23,R30,Z,
     DR10F2C,R20E2C,R10C,R20C
      DOUBLE PRECISION WW1C,WW12,WW13,WW2C,WW30,WW23,
     DFF10,FF12,FF13,FF20,FF3C,FF23
      DOUBLE PRECISION
     DW,TCTDAY,TDAYCU,ONEDAY,SWT,TAUN,CCUNT,AT,BT,
     DRD1C,RD12,RD13,RDD10,RDD12,RDD13,
     DDEL10,DEL12,DEL13,MONE,X20,DEL2C,X3C,DEL30,X23,DEL23,
     DEPS10,XAST10,EPS12,XAST12,EPS13,XAST13,XAST20,XAST23,XAST30,
     DRAST10,RAST12,RAST13,RAST2C,RAST23,RAST30,S10,S12,S13,S20,
     DS23,S30,RDOT10,RDOT12,RDOT13,RD1CF,RD12F,RD13F,RDD10F,RDD12F,
     DRDD13F,RWGT,RDWGT,TSQ24C,R4F10,R4F12,R4F13,TAU45,RD4F10,
     DBX1C,BX12,BX13,
     DRD4F12,RD4F13,X4F1C,X4F12,X4F13
      DOUBLE PRECISION EPS2C,EPS30,EPS23
      COMMON X1C(12),X12(12),X13(12),XTEMP(6),XKEP(6),XDEL(6),
     CM1,M2,M3,M12,M13,M23, M2F12, M2F23, M3F13,M3F23,
     CST2X10(6),ST2X12(6),ST2X13(6),ST2X20(6),ST2X3C(6),ST2X23(6),
     CCML,CMV,CMT,Y10(6),Y12(6),Y13(6),TIMED,TIMEMX,
     CTAUC,TAU,R10,R20,TEMP,Q,R10E2,R2CE2,
     CL,C1,C2,C3,DELTA,H,XFR1C(3),XAFR10(3),
     CRE2,R,KSI,ETA,ZETA,CHI,R12(6),R13(6),R23(6),R30(6),Z,
     CR10E2C,R2CE2C,R10C,R20C
      COMMON WW1C(6),WW12(6),WW13(6),WW20(6),WW3C(6),WW23(6),
     CFF1C(6),FF12(6),FF13(6),FF2C(6),FF30(6),FF23(6)
      COMMON
     CW,TCTCAY,TCAYCU,ONEDAY,SWT,TAUN,CCUNT,AT,BT,
     CTSQ240,BX10(6),BX12(6),BX13(6),
     CRCD12F(6),RCD13F(6),RWGT(4),RDWGT(4),R4F10(3),R4F12(3),
     CRD10(3),RC12(3),RC13(3),RCC10(3),RCD12(3),RCD13(3),
     CDEL10(6),CEL12(6),DEL13(6),DEL20(6),DEL23(6),CEL3C(6),
     CMONF,X20(6),X23(6),X30(6),EPS10(6),EPS12(6),EPS13(6),
     CXAST10(6),XAST12(6),XAST13(6),XAST20(6),XAST23(6),XAST30(6),
     CRAST10(6),RAST12(6),RAST13(6),RAST20(6),RAST23(6),RAST30(6),
     CS10(6),S12(6),S13(6),S20(6),S23(6),S30(6),RDOT10(6),
     CRDOT12(6),RCOT13(6),RD1CF(6),RD12F(6),RC13F(6),RCC10F(6),
     CR4F13(3),TAU45,RD4F10(3),RC4F12(3),RD4F13(3),X4F1C(6),X4F12(6),
     CX4F13(6),XFR12(3),XFR13(3),XFR20(3),XFR23(3),XFR3C(3),XAFR12(3),
     CXAFR13(3),XAFR2C(3),XAFR23(3),XAFR30(3)
      CCMMON EPS20(6),EPS30(6),EPS23(6)
      RE2=X(1)**2+X(2)**2+X(3)**2
      R=DSQRT(RE2)
      KSI=M*(TAU**2)/R**3
      ETA=(X(1)*X(4)+X(2)*X(5)+X(3)*X(6))*TAU/RE2
      ZETA=(X(4)**2+X(5)**2+X(6)**2)*(TAU**2)/RE2-KSI
      CHI=KSI-ZETA
      Z=MCNE-.5CCO*ETA-(1.DOC/ 6.DOO)*ZETA+.5COO*ETA**2+(5.DOO/12.DOO)
     1*ETA*ZETA
      ITER=0
   30 ITER=ITER+1
      L=CHI*Z**2
      C2=.5DOO*(MONE-(L/12.DCO)*(MONE-(L/30.DCO)*
     1   (MONE-(L/56.DOO)*(MONE-(L/90.DCO)))))
      C3=(MONE/ 6.DCO)*(MONE-(L/20.DOO)*(MONE-(L/42.DOO)*
     1   (MCNE-(L/72.DOO)*(MCNE-(L/110.DCO)))))
      C1=MONE-L*C3
      DELTA=MONE+C1*ETA*Z+C2*ZETA*Z**2
      H=C2*ETA*Z**2+C3*ZETA*Z**3+Z-MONE
      Z=Z-H/CELTA
      IF(DABS(H).LE.1.D-07) GC TC 31
      IF(ITER.LE.10)GO TO 30
 31   CONTINUE
      DO 40 I=1,3
      XDEL(I)=(-KSI*Z**2)*(C2*X(I)+C3*Z*TAU*X(I+3))
      XDEL(I+3)=(-KSI*Z)*(C1*X(I)+C2*Z*TAU*X(I+3))/(DELTA*TAU)
      XKEP(I)=X(I)+TAU*X(I+3)+XDEL(I)
      XKEP(I+3)=X(I+3)+XDEL(I+3)
  40  CONTINUE
      RETURN
      END
$DATA
    .23621551D06   -.28407815CC6   -.15944798C06
    .84415966DCC   -.71504167C-1   -.40781976C-1
```

```
           .22841895D06   -.28612319DC6   -.16078232C06
           .82437821DCC    .50846563CC0    .18637202CC0
          -.31957216D08    .13642131CC9    .59157903C08
          -.28628154D02   -.56407282DC1   -.24463687D01
                2.0DC0         19.0DC0         10.0DC0         179.0D00
                1.0DCC

      $FMSYS
      $PAUSE
```

```
$IBSYS
$JOB  0510        STUMPFF4 E.H.WEISS  RIGGS BLDG   X-7266
$DATE            102766
$EXECUTE         IBJOB
$IBJOB           GO,MAP,LOGIC,SCURCE
$IBFTC MAIN      LIST,REF,NODECK
      DOUBLE PRECISION
     DERR,TIMECN,ITDN,ITD,A,CX10,
     DX10,X12,X13,XTEMP,XKEP,XDEL,
     DST2X10,ST2X12,ST2X13,ST2X20,ST2X30,ST2X23,
     DM1,M2,M3,M12,M13,M23,M2F12,M2F23,M3F13,M3F23,
     DCML,CMV,CMT,Y10,Y12,Y13,TIMED,TIMEMX,
     DTAUD,TAU,R10,R20,TEMP,Q,R10E2,R2CE2,
     DL,C1,C2,C3,DELTA,H,XFR10,XAFR10,
     DXFR12,XFR13,XFR20,XFR23,XFR30,XAFR12,XAFR13,XAFR20,XAFR23,XAFR30,
     DRE2,R,KSI,ETA,ZETA,CHI,R12,R13,R23,R30,Z,
     DRD10,RD12,RD13,RDD10,RDD12,RDD13,
     DR10E2C,R20E2C,R10C,R20C
      DOUBLE PRECISION WW10,WW12,WW13,WW20,WW30,WW23,
     DFF10,FF12,FF13,FF20,FF30,FF23
      DOUBLE PRECISION C
      DOUBLE PRECISION
     DW,TOTDAY,TDAYCU,ONEDAY,SWT,TAUN,COUNT,AT,BT,
     DDEL10,DEL12,DEL13,MONE,X20,DEL20,X30,DEL30,X23,DEL23,
     DEPS10,XAST10,EPS12,XAST12,EPS13,XAST13,XAST20,XAST23,XAST30,
     DRAST10,RAST12,RAST13,RAST20,RAST23,RAST30,S10,S12,S13,S20,
     DS23,S30,RDOT10,RDOT12,RDOT13,RD10F,RD12F,RD13F,RDD10F,RDD12F,
     DRDD13F,RWGT,RDWGT,TSQ240,R4F10,R4F12,R4F13,TAU45,RD4F10,
     DBX10,BX12,BX13,
     DDELT,DEL(4),
     DRD4F12,RD4F13,X4F10,X4F12,X4F13
      DOUBLE PRECISION TAUP,R10F4T,R10FT4
      COMMON X10(12),X12(12),X13(12),XTEMP(6),XKEP(6),XDEL(6),
     CM1,M2,M3,M12,M13,M23, M2F12, M2F23, M3F13,M3F23,
     CST2X10(6),ST2X12(6),ST2X13(6),ST2X20(6),ST2X30(6),ST2X23(6),
     CCML,CMV,CMT,Y10(6),Y12(6),Y13(6),TIMED,TIMEMX,
     CTAUD,TAU,R10,R20,TEMP,Q,R10E2,R2CE2,
     CL,C1,C2,C3,DELTA,H,XFR10(3),XAFR10(3),
     CRE2,R,KSI,ETA,ZETA,CHI,R12(6),R13(6),R23(6),R30(6),Z,
     CR10E2C,R20E2C,R10C,R20C
      COMMON WW10(6),WW12(6),WW13(6),WW20(6),WW30(6),WW23(6),
     CFF10(6),FF12(6),FF13(6),FF20(6),FF30(6),FF23(6)
      COMMON
     CW,TCTCAY,TCAYCU,ONEDAY,SWT,TAUN,CCUNT,AT,BT,
     CTSQ240,BX10(6),BX12(6),BX13(6),
     CRDD12F(6),RDD13F(6),RWGT(4),RDWGT(4),R4F10(3),R4F12(3),
     CRD10(3),RD12(3),RD13(3),RDD10(3),RDD12(3),RDD13(3),
     CDEL10(6),DEL12(6),DEL13(6),DEL20(6),DEL23(6),DEL30(6),
     CMONE,X20(6),X23(6),X30(6),EPS10(6),EPS12(6),EPS13(6),
     CXAST10(6),XAST12(6),XAST13(6),XAST20(6),XAST23(6),XAST30(6),
     CRAST10(6),RAST12(6),RAST13(6),RAST20(6),RAST23(6),RAST30(6),
     CS10(6),S12(6),S13(6),S20(6),S23(6),S30(6),RDOT10(6),
     CRDOT12(6),RDOT13(6),RD10F(6),RD12F(6),RD13F(6),RDD10F(6),
     CR4F13(3),TAU45,RD4F10(3),RD4F12(3),RD4F13(3),X4F10(6),X4F12(6),
     CX4F13(6),XFR12(3),XFR13(3),XFR20(3),XFR23(3),XFR30(3),XAFR12(3),
     CXAFR13(3),XAFR20(3),XAFR23(3),XAFR30(3)
C     STUMPFF BY E.H. WEISS  RIGGS BLDG
 900  READ(2,800)(Y10(I),I=1,6)
      READ(2,800)(Y12(I),I=1,6)
      READ(2,800)(Y13(I),I=1,6)
```

```
  800 FORMAT(3D16.8)
      READ(2,802)AT,BT,ONEDAY,TIMEMX,MORE
  802 FORMAT(4D16.8,7X,I1)
  803 FORMAT(D16.8)
      CML=6378.165D00
      CMT=((6378.165D00**3)/.3986032D06)
      CMT=DSQRT(CMT)
      CMV=CML/CMT
C      INPUT IS IN  KILOMETERS AND SECONDS.
C     MULTIPLY CANONICAL BY CML,CMT OR CMV, TO OBTAIN METRIC .
C     TAUD IS IN DAYS. TAU IS CANONICAL.
      IPR=0
      MONE=1.D00
      M1=1.D00
      M2=1.D00/81.30150052D00
      M3=332951.2658D00
      M12=M1+M2
      M13=M1+M3
      M23=M2+M3
      M2F12=M2/M12
      M2F23=M2/M23
      M3F13=M3/M13
      M3F23=M3/M23
      TIMED =0.D00
      COUNT=0.D00
      TAUN=0.D00
      ONEDAY=(ONEDAY*86400.D00)/CMT
      TOTDAY=0.D00
      SWT=0.D00
      WRITE(3,26)(Y10(I),I=1,6),TIMED
      WRITE(3,19)(Y12(I),I=1,6)
      WRITE(3,19)(Y13(I),I=1,6)
   26 FORMAT(1H1,6D16.8,D16.8)
      WRITE(3,801)AT,BT,C,M1
  801 FORMAT(4H0AT=,D16.8,2X,4H BT=,D16.8,2X,3H C=,D16.8,2X,
     15H M1 =,D16.8)
      RWGT(1)=721.D00
      RWGT(2)=476.D00
      RWGT(3)=245.D00
      RWGT(4)=18.D00
      RDWGT(1)=64.D00
      RDWGT(2)=24.D00
      RDWGT(3)=64.D00
      RDWGT(4)=14.D00
      DEL(1)=-4.D00
      DEL(2)=6.D00
      DEL(3)=-4.D00
      DEL(4)=1.D00
      W=6.D00
      TAU=BT
    9 CONTINUE
      COUNT=COUNT+1.D00
      TSQ240=(TAU**2)/240.D00
      TAU45=TAU/45.D00
      DO 350 I=1,6
      RD10F(I)=0.D00
      RD12F(I)=0.D00
      RD13F(I)=0.D00
      RDD10F(I)=0.D00
      RDD12F(I)=0.D00
      RDD13F(I)=0.D00
  350 CONTINUE
      DELT=0.D00
      DO 341 J=1,4
      TAUN=TAUN+TAU
      DO 25 I=1,3
      BX10(I)=Y10(I)/CML
      BX12(I)=Y12(I)/CML
      BX13(I)=Y13(I)/CML
      BX10(I+3)=Y10(I+3)/CMV
      BX12(I+3)=Y12(I+3)/CMV
      BX13(I+3)=Y13(I+3)/CMV
   25 CONTINUE
      A=J
      TAU=A*TAU
   61 CONTINUE
C
C     INPUT/M1,X10        OUTPUT/ X10,DEL10
C
      CALL SUB1(BX10,M1)
      DO 100 I=1,6
      X10(I)=XKEP(I)
      DEL10(I)=XDEL(I)
  100 CONTINUE
C
C     INPUT/M1+M2,X12     OUTPUT/ X12,DEL12
C
      CALL SUB1(BX12,M12)
      DO 110 I=1,6
      X12(I)=XKEP(I)
      DEL12(I)=XDEL(I)
  110 CONTINUE
C
C     INPUT/M1+M3,X13     OUTPUT/ X13,DEL13
C
      CALL SUB1(BX13,M13)
      DO 120 I=1,6
      X13(I)=XKEP(I)
      DEL13(I)=XDEL(I)
  120 CONTINUE
      DO 13 I=1,6
      XTEMP(I)=BX10(I)-BX12(I)
   13 CONTINUE
C
C     INPUT/M2,X10-X12    OUTPUT/ X20,DEL20
C
      CALL SUB1(XTEMP,M2)
      DO 130 I=1,6
      X20(I)=XKEP(I)
      DEL20(I)=XDEL(I)
  130 CONTINUE
      DO 15 M=1,6
      XTEMP(M)=BX10(M)-BX13(M)
   15 CONTINUE
C
C     INPUT/M3,X10-X13    OUTPUT/ X30,DEL30
C
      CALL SUB1(XTEMP,M3)
      DO 140 I=1,6
      X30(I)=XKEP(I)
      DEL30(I)=XDEL(I)
```

```
 140  CONTINUE
      DO 17 M=1,6
      XTEMP(M)=BX13(M)-BX12(M)
  17  CONTINUE
C
C     INPUT/M2+M3,X13-X12OUTPUT/ X23,DEL23
C
      CALL SUB1(XTEMP,M23)
      DO 150 I=1,6
      X23(I)=XKEP(I)
      DEL?3(I)=XDEL(I)
 150  CONTINUE
C
C     COMPUTE EPS10       E10=DEL20+DEL30+M2/(M1+M2)*DEL12+(M3/(M1+M3)
C                                *DELU3
C
      DO 190 I=1,6
      EPS10(I)=CEL20(I)+DEL30(I)+M2F12*DEL12(I)+M3F13*DEL13(I)
      XAST10(I)=X10(I)+EPS10(I)
C
C     COMPUTE FPSU2       E12=(M3/(M1+M3))*DEL13-(M3/(M2+M3))*DEL23
C
      EPS12(I)=M3F13*DEL13(I)-M3F23*DEL23(I)
      XAST12(I)=X12(I)+EPS12(I)
C
C     COMPUTE EPSU3       E13=(M2/(M2+M3))*DEL23+(M2/(M1+M2))*DEL12
C
      EPS13(I)=M2F23*DEL23(I)+M2F12*DEL12(I)
      XAST13(I)=X13(I)+FPS13(I)
      XAST20(I)=XAST10(I)-XAST12(I)
      XAST30(I)=XAST10(I)-XAST13(I)
      XAST23(I)=XAST13(I)-XAST12(I)
 190  CONTINUE
C
C     COMPUTE R(IJ) SQUARED AND RAST(IJ) SQUARED
C                         R(IJ)**2= X(IJ)(1)**2+X(IJ)(2)**2+X(IJ)(3)**2
C
      R10=DSQRT(X10(1)*X10(1)+X10(2)*X10(2)+X10(3)*X10(3))
      R12=DSQRT(X12(1)*X12(1)+X12(2)*X12(2)+X12(3)*X12(3))
      R13=DSQRT(X13(1)*X13(1)+X13(2)*X13(2)+X13(3)*X13(3))
      R20=DSQRT(X20(1)*X20(1)+X20(2)*X20(2)+X20(3)*X20(3))
      R23=DSQRT(X23(1)*X23(1)+X23(2)*X23(2)+X23(3)*X23(3))
      R30=DSQRT(X30(1)*X30(1)+X30(2)*X30(2)+X30(3)*X30(3))
      RAST10=DSQRT(XAST10(1)**2+XAST10(2)**2+XAST10(3)**2)
      RAST12=DSQRT(XAST12(1)**2+XAST12(2)**2+XAST12(3)**2)
      RAST13=DSQRT(XAST13(1)**2+XAST13(2)**2+XAST13(3)**2)
      RAST20=DSQRT(XAST20(1)**2+XAST20(2)**2+XAST20(3)**2)
      RAST23=DSQRT(XAST23(1)**2+XAST23(2)**2+XAST23(3)**2)
      RAST30=DSQRT(XAST30(1)**2+XAST30(2)**2+XAST30(3)**2)
C
C     CCMPUTE S10,S12,S13,S20,S23,S30 WHERE IN GENERAL
C                         S(IJ)=X(IJ)/R(IJ)**3*XAST(IJ)/RAST(IJ)**3
C
      DO 250 I=1,3
      S10(I)=X10(I)/(R10*R10*R10)-XAST10(I)/RAST10**3
      S12(I)=X12(I)/(R12*R12*R12)-XAST12(I)/RAST12**3
      S13(I)=X13(I)/(R13*R13*R13)-XAST13(I)/RAST13**3
      S20(I)=X20(I)/(R20*R20*R20)-XAST20(I)/RAST20**3
      S23(I)=X23(I)/(R23*R23*R23)-XAST23(I)/RAST23**3
      S30(I)=X30(I)/(R30*R30*R30)-XAST30(I)/RAST30**3
 250   CONTINUE
C
C
C      COMPUTE RCOT (IJ)
C                              RDOT(IJ)=(M(I)+M(J))*S(IJ)+M(K)*(S(IK)+S(KJ))
C                                          +M(L)*(S(IL)+S(LJ))
C                              WHERE  I .NE. J .NE. K .NE. L
C                              AND (I,J,K,L)=(0,1,2,3)
C
C
       DO 260 I=1,3
       RDOT10(I)=M1*S10(I)+M2*(S12(I)+S20(I))+M3*(S13(I)+S30(I))
       RDOT12(I)=M12*S12(I)+M3*(S13(I)-S23(I))
       RDOT13(I)=M13*S13(I)+M2*(S12(I)+S23(I))
 260   CONTINUE
       DELT=RDOT10(1)**2+RDOT10(2)**2+RCCT10(3)**2
       DELT=DSQRT(CELT)*DEL(J)+CELT
       TAU=TAU/A
       DO 340 I=1,3
       RD1C(I)=RWGT(J)*RDOT10(I)
       RD1CF(I)=RC10F(I)+RD10(I)
       RD12(I)=RWGT(J)*RDOT12(I)
       RD12F(I)=RD12F(I)+RD12(I)
       RD13(I)=RWGT(J)*RDOT13(I)
       RD13F(I)=RC13F(I)+RC13(I)
       RCD10(I)=RDWGT(J)*RDOT10(I)
       RCD10F(I)=RCD10F(I)+RCD1C(I)
       RDD12(I)=RDWGT(J)*RDOT12(I)
       RCD12F(I)=RCD12F(I)+RCC12(I)
       RCD13(I)=RDWGT(J)*RDOT13(I)
       RDD13F(I)=RCD13F(I)+RCD13(I)
 340   CONTINUE
 341   CONTINUE
       DO 370 I=1,3
       R4F10(I)=TSQ240*RD10F(I)
       R4F12(I)=TSQ240*RD12F(I)
       R4F13(I)=TSQ240*RC13F(I)
       RD4F10(I)=TAU45*RCD10F(I)
       RD4F12(I)=TAU45*RCD12F(I)
       RD4F13(I)=TAU45*RCD13F(I)
 370   CONTINUE
       CX1C=DSQRT((R4F10(1)**2)+(R4F10(2)**2)+(R4F10(3)**2))
 380   DO 390 I=1,3
       X4F10(I)=XAST10(I)+R4F10(I)
       X4F12(I)=XAST12(I)+R4F12(I)
       X4F13(I)=XAST13(I)+R4F13(I)
       X4F10(I+3)=XAST10(I+3)+RC4F10(I)
       X4F12(I+3)=XAST12(I+3)+RC4F12(I)
       X4F13(I+3)=XAST13(I+3)+RC4F13(I)
 390   CONTINUE
  69   DO 18 J=1,3
       Y10(J)=X4F10(J)*CML
       Y12(J)=X4F12(J)*CML
       Y13(J)=X4F13(J)*CML
       Y10(J+3)=X4F10(J+3)*CMV
       Y12(J+3)=X4F12(J+3)*CMV
       Y13(J+3)=X4F13(J+3)*CMV
  18   CCNTINUE
       R10F2=Y1C(1)**2+Y10(2)**2+Y1C(3)**2
       R20F2=(Y1C(1)-Y12(1))**2+(Y1C(2)-Y12(2))**2+(Y10(3)-Y12(3))**2
```

```
      R10=DSQRT(R10E2)
      R20=DSQRT(R20E2)
      IF(TPR)6027,6026,6027
 6027 CONTINUE
      TAUP=4.0D00*TAU
      WRITE(3,6000)R20
 6000 FORMAT(11H R20(KM) = ,D16.8)
      WRITE(3,6001) TAUP
 6001 FORMAT(11H FOUR TAU =,D16.8)
      WRITE(3,6002) CX10
 6002 FORMAT(24H MAG OF R10 CORRECTION =,D16.8)
      WRITE(3,6005)DELT
 6005 FORMAT(6H DELT=,D16.8)
 6026 CONTINUE
      IF(SWT-1.D00)6024,1001,6024
 6024 R10F4T=DABS(DELT)
      IF(R10F4T)6021,6023,6021
 6021 R10F4T=AT/R10F4T
      IF(R10F4T - 6.D00)6022,6022,6023
 6022 TAU=TAU*.5D00*(R10F4T+1.D00)
 1006 IF(ONEDAY-(TAUN+A*TAU*1.1D00))1000,1000,9
 1000 TAU=(ONEDAY-TAUN)/A
      SWT=1.D00
      GO TO 9
 1001 TOTDAY=TOTDAY+ONEDAY
      TDAYCU=(TOTDAY*CMT)/86400.D00
      SWT=0.D00
      TIMED=TDAYCU
      WRITE(3,20)(Y10(I),I=1,6),TIMED
      WRITE(3,19)(Y12(I),I=1,6)
      WRITE(3,19)(Y13(I),I=1,6)
  19  FORMAT(1H ,6D16.8)
  20  FORMAT(1H0,7D16.8)
 2050 FORMAT(1H 3D24.16)
      WRITE(3,28)CX10,R10,R20,TIMED,COUNT
  28  FORMAT(1H ,5D16.8)
      IF(TIMEMX-TDAYCU)1003,1003,1002
 1002 CONTINUE
      TAUN=0.D00
      GO TO 6024
 6023 R10F4T=6.D00
      GO TO 6022
 1003 IF(MORE.EQ.1)GO TO 900
  65  CONTINUE
      STOP
      END
$IBFTC SUB1     LIST,REF,NODECK
      SUBROUTINE SUB1 (X,M)
      DOUBLE PRECISION
     DX(6),M,
     DERR,TIMEDN,ITDN,ITD,A,CX10,
     DX10,X12,X13,XTEMP,XKEP,XDEL,
     DST2X10,ST2X12,ST2X13,ST2X20,ST2X30,ST2X23,
     DM1,M2,M3,M12,M13,M23,M2F12,M2F23,M3F13,M3F23,
     DCML,CMV,CMT,Y10,Y12,Y13,TIMED,TIMEMX,
     DTAUD,TAU,R10,R20,TEMP,Q,R10E2,R20E2,
     DL,C1,C2,C3,DELTA,H,XFR10,XAFR10,
     DXFR12,XFR13,XFR20,XFR23,XFR30,XAFR12,XAFR13,XAFR20,XAFR23,XAFR30,
     DRE2,R,KSI,ETA,ZETA,CHI,R12,R13,R23,R30,Z,
     DR10E2C,R20E2C,R10C,R20C
```

```
      DOUPLE PRECISION WW1C,WW12,WW13,WW2C,WW30,WW23,
     DFF1C,FF12,FF13,FF2C,FF3C,FF23
      DOUPLE PRECISION
     DW,TCTDAY,TDAYCU,ONEDAY,SWT,TAUN,COUNT,AT,BT,
     DRD1C,RC12,RD13,RDC10,RDD12,RCD13,
     DDEL10,DEL12,DEL13,MCNE,X20,DEL20,X30,DEL30,X23,DEL23,
     DEPS10,XAST10,EPS12,XAST12,EPS13,XAST13,XAST20,XAST23,XAST30,
     DRAST10,RAST12,RAST13,RAST20,RAST23,RAST30,S10,S12,S13,S20,
     DS23,S30,RDOT10,RDOT12,RDCT13,RD1CF,RD12F,RD13F,RCD10F,RDD12F,
     DRCD13F,RWGT,RDWGT,TSQ240,R4F10,R4F12,R4F13,TAU45,RD4F10,
     DBX1C,BX12,BX13,
     DRD4F12,RD4F13,X4F10,X4F12,X4F13
      COMMON X1C(12),X12(12),X13(12),XTEMP(6),XKEP(6),XDEL(6),
     CM1,M2,M3,M12,M13,M23, M2F12, M2F23, M3F13,M3F23,
     CST2X10(6),ST2X12(6),ST2X13(6),ST2X20(6),ST2X3C(6),ST2X23(6),
     CCML,CMV,CMT,Y10(6),Y12(6),Y13(6),TIMED,TIMEMX,
     CTAUC,TAU,R10,R20,TEMP,Q,R1CE2,R2CE2,
     CL,C1,C2,C3,CELTA,H,XFR1C(3),XAFR10(3),
     CRE2,R,KSI,ETA,ZETA,CHI,R12(6),R13(6),R23(6),R30(6),Z,
     CR10E2C,R20E2C,R10C,R20C
      COMMON WW10(6),WW12(6),WW13(6),WW20(6),WW30(6),WW23(6),
     CFF1C(6),FF12(6),FF13(6),FF20(6),FF3C(6),FF23(6)
      COMMON
     CW,TCTDAY,TCAYCU,ONEDAY,SWT,TAUN,COUNT,AT,BT,
     CTSQ?40,BX10(6),BX12(6),BX13(6),
     CRDD12F(6),RCD13F(6),RWGT(4),RDWGT(4),R4F10(3),R4F12(3),
     CRD1C(3),RD12(3),RD13(3),RCD10(3),RCD12(3),RDD13(3),
     CDEL10(6),CEL12(6),DEL13(6),DEL20(6),DEL23(6),DEL30(6),
     CMONF,X20(6),X23(6),X3C(6),EPS10(6),EPS12(6),EPS13(6),
     CXAST10(6),XAST12(6),XAST13(6),XAST2C(6),XAST23(6),XAST30(6),
     CRAST10(6),RAST12(6),RAST13(6),RAST20(6),RAST23(6),RAST30(6),
     CS10(6),S12(6),S13(6),S20(6),S23(6),S3C(6),RDOT10(6),
     CRDOT12(6),RDOT13(6),RD1CF(6),RD12F(6),RD13F(6),RCD10F(6),
     CR4F13(3),TAU45,RD4F10(3),RD4F12(3),RD4F13(3),X4F10(6),X4F12(6),
     CX4F13(6),XFR12(3),XFR13(3),XFR20(3),XFR23(3),XFR30(3),XAFR12(3),
     CXAFR13(3),XAFR20(3),XAFR23(3),XAFR3C(3)
      RE2=X(1)**2+X(2)**2+X(3)**2
      R=DSQRT(RE2)
      KSI=M*(TAU**2)/R**3
      ETA=(X(1)*X(4)+X(2)*X(5)+X(3)*X(6))*TAU/RE2
      ZETA=(X(4)**2+X(5)**2+X(6)**2)*(TAU**2)/RE2-KSI
      CHI=KSI-ZETA
      Z=MCNE-.5CO0*ETA-(1.DC0/ 6.D00)*ZETA+.5C00*ETA**2+(5.D00/12.D00)
     1*ETA*ZETA
      ITER=0
   30 ITER=ITER+1
      L=CHI*Z**2
      C2=.5D00*(MONE-(L/12.D00)*(MONE-(L/30.DC0)*
     1   (MONE-(L/56.D00)*(MONE-(L/90.DC0)))))
      C3=(MONE/ 6.D00)*(MONE-(L/20.D00 )*(MONE-(L/42.D00)*
     1   (MONE-(L/72.D00)*(MONE-(L/110.DC0)))))
      C1=MONE-L*C3
      DELTA=MONE+C1*ETA*Z+C2*ZETA*Z**2
      H=C2*ETA*Z**2+C3*ZETA*Z**3+Z-MONE
      Z=Z-H/DELTA
      IF(DABS(H).LE.1.D-07) GC TC 31
      IF(ITER.LE.10)GO TO 30
  31  CCNTINUE
      DO 40 I=1,3
      XDEL(I)=(-KSI*Z**2)*(C2*X(I)+C3*Z*TAU*X(I+3))
      XDEL(I+3)=(-KSI*Z)*(C1*X(I)+C2*Z*TAU*X(I+3))/(DELTA*TAU)
      XKEP(I)=X(I)+TAU*X(I+3)+XDEL(I)
      XKEP(I+3)=X(I+3)+XDEL(I+3)
  40  CONTINUE
      RETURN
      END
$DATA
    .23621551D06    -.28407815DC6   -.15944798C06
    .84415966DC0    -.71504167D-1   -.40781976C-1
    .22841895D06    -.28612319D06   -.16078232C06
    .82437821DC0     .50846563CC0    .18637202C00
   -.31957216D08     .13642131D09    .59157903D08
   -.28628154D02    -.56407282CC1   -.24463687C01
         2.5D-08           1.D-C2          1.0C01          179.D00
```

"The aeronautical and space activities of the United States shall be conducted so as to contribute . . . to the expansion of human knowledge of phenomena in the atmosphere and space. The Administration shall provide for the widest practicable and appropriate dissemination of information concerning its activities and the results thereof."

—NATIONAL AERONAUTICS AND SPACE ACT OF 1958

NASA SCIENTIFIC AND TECHNICAL PUBLICATIONS

TECHNICAL REPORTS: Scientific and technical information considered important, complete, and a lasting contribution to existing knowledge.

TECHNICAL NOTES: Information less broad in scope but nevertheless of importance as a contribution to existing knowledge.

TECHNICAL MEMORANDUMS: Information receiving limited distribution because of preliminary data, security classification, or other reasons.

CONTRACTOR REPORTS: Scientific and technical information generated under a NASA contract or grant and considered an important contribution to existing knowledge.

TECHNICAL TRANSLATIONS: Information published in a foreign language considered to merit NASA distribution in English.

SPECIAL PUBLICATIONS: Information derived from or of value to NASA activities. Publications include conference proceedings, monographs, data compilations, handbooks, sourcebooks, and special bibliographies.

TECHNOLOGY UTILIZATION PUBLICATIONS: Information on technology used by NASA that may be of particular interest in commercial and other non-aerospace applications. Publications include Tech Briefs, Technology Utilization Reports and Notes, and Technology Surveys.

Details on the availability of these publications may be obtained from:

SCIENTIFIC AND TECHNICAL INFORMATION DIVISION
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Washington, D.C. 20546