X-641-70-453

NASA TM X- 65398

# THE GODDARD VERSION OF THE SCHUBART-STUMPFF N-BODY PROGRAM

P. A. COMELLA
B. E. LOWREY

DECEMBER 1970



GSFC

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

# THE GODDARD VERSION OF THE SCHUBART-STUMPFF N-BODY PROGRAM

P. A. Comella

B. E. Lowrey

Contents

THE GODDARD VERSION OF THE SCHUBART-STUMPFF N-BODY PROGRAM

P. A. Comella

B. E. Lowrey

Abstract

The Schubart-Stumpff N-Body Program computes solar system orbits of the planets and of bodies of zero mass. It can also be used to solve more general problems in mechanics. A Fortran IV adaptation of the Yale version of the program is available to users from the GSFC documentation center. Primary modifications to the Yale program include simplification of program input and the addition of an option to compute the osculating orbital elements. The present document briefly summarizes the major features of the program and the development of the Schubart-Stumpff initial values, discusses the new input requirements and modifications, and presents a sample case for clarification.

I. Method of Schubart-Stumpff Program

Schubart and Stumpff (1966) have chosen to regard the orbital computation problem as one of finding the solution to a system of simultaneous differential equations. Basic to this approach is the decision to treat all bodies alike rather than to use the special perturbation approach which distinguishes perturbed from perturbing bodies. Hence only initial values and an appropriately chosen step-size are needed to achieve solution. The method of integration used is that of Adams-Störmer: a difference method with constant step-size for the numerical integration.

By avoiding the special perturbation method it is unnecessary to input coordinates of perturbing bodies at other than the starting epoch, thus avoiding much of the data manipulation associated with reading in tables of perturbing bodies, which of itself can be a formidable problem, both logically and logistically.

The method of numerically integrating the massive bodies rather than using tabular input is not uncompetitive in machine time, depending upon the problem. If it is desired to compute the orbits of many massless bodies for the same period of time, the machine time required to compute the planetary orbits becomes a small fraction of the total time. But the N-Body Program also has the ability to study the orbits of the planets themselves, as for example, Lieske's (1967) preparation of JPL's Development Ephemeris Number 28. In another application, Marsden (1969) used the program (with some modifications for differential corrections) to determine the influence of non-gravitational forces on the orbits of short period comets.

It is important to note that there are no provisions made for the problems that arise in extremely close passages. The planets are taken as mass points. Also the experimenter must know at what epoch a close approach will occur since there is no automatic control of step-length. Then he may interrupt the computation, input a smaller step-size with which to compute for the duration of the approach, and then again interrupt after approach to input a larger step-size. Schubart and Stumpff chose this seemingly clumsy method of step-length control because in their own theoretical work they could predict the epoch of a close approach very easily. The authors of the present note have retained this method in the GSFC version of the program. They plan, however, to implement a Nordsieck-type of predictor-corrector method in order to permit dynamic internal calculation of optimum interval-size for each integration step.

## II. The Initial Conditions

The Schubart-Stumpff N-Body Program together with the proper initial conditions for the planets calculates to a high degree of accuracy orbits of bodies of (essentially) zero mass and simultaneously the orbits of the planets.

For solar system orbits Schubart and Stumpff have derived a set of starting conditions of the planets for epoch JD = 2430000.5 . These derivations are discussed in great detail in their paper. The following paragraphs summarize their work.

With their initial values and a step-size of 2 days, the program reproduces to 10 decimal places the ephemerides of the planets of the solar system under the following conditions:

(1) Relativistic effects are ignored;

(2) The perturbing effects of Mercury are only approximated: Mercury's mass is added to that of the sun, thus introducing an error of $10^{-7}$ A.U. in the location of the origin of a heliocentric system;

(3) The mass of the moon is added to that of the earth but the perturbations in the earth-moon orbit caused by the moon are not considered.

(4) Perturbations caused by Pluto are ignored, a decision motivated not only by economics but also by the fact that Pluto's mass is not known with sufficient accuracy.

The Eckert, Brouwer, Clemence ephemeris (1951) of the five outer planets (Jupiter to Pluto); Herget's computations from Newcomb's tables (1953, 1955) of the ephemeris for Venus and the center of mass of the earth-moon system; and the R. L. Duncombe-G. M. Clemence ephemeris (1960, 1964) for Mars were the standards used in the development of the initial conditions and in the comparison of the final results.

To facilitate conversions and comparisons 11th differences of the acceleration were used which correspond to the value of 5 for M in the Fortran code. The initial epoch, J.D. = 2430000.5 , was chosen because for periods of 400 days on either side of that date, the Eckert, Brouwer, Clemence observations of the 5 outer planets were especially good, a factor vital to deriving the starting velocities.

For any solar system problem the Schubart-Stumpff initial values can always be used. If the epoch for which the experimenter wishes to enter coordinates of zero mass bodies differs from their epoch, a negative step-size may be used to integrate back to an earlier epoch, a positive step-size to integrate forward to a later epoch. The calculations then proceed with the additional bodies.

## III. GSFC Version of the Schubart-Stumpff N-Body Program

### A. Modifications

The original Schubart-Stumpff N-Body Program was written in the mid-1960's in the Fortran II language for use on a smallish computer at the University of Heidelberg. Subsequently, Schubart and M. Cooke (1965) rewrote the program in the Fortran IV language for use on an IBM 7094 at Yale. It is the Yale version that the authors have modified for use on Goddard Space Flight Center's IBM S/360 computers.

These modifications are simplifications in the authors' opinion: the Yale version still reflected the idiocyncrasies of the Fortran II language in coping with double word computations and complicated data input. These peculiarities made it clumsy to input data to the program. The authors adopted the NAMELIST convention of the Fortran IV language to input all data, except for one initialization card.

The NAMELIST option, by allowing selective initialization of data without destruction of data in locations not specifically named on a given READ, enables the programmer to eliminate much coding of trigger recognition parameters and statements. Thus although the input is still flexible, only one read statement is required for the entire program to initialize all (but two) of the input parameters.

A second modification was the addition of an option to print the osculating elements. A parameter triggers a call to a subroutine written by Blanchard and Wolf (1967) which converts the position and velocity vectors at a given time into the orbital elements by means of the two-body formulas.

Finally, the addition of 3 parameters permits the positional co-ordinates, the velocity coordinates and the mass parameters to be input with incompatible dimensional systems, by converting all to a compatible system following input. Hopefully this will eliminate the tedious hand computations which might be required for some problems.

B. Subprograms of the Program

1. MAIN - determines size of the A, B, and D [large] arrays, which initial conditions are to be used, and calls CONTRL.

2. SUBROUTINE CONTRL - drives the program by processing input, calling integration routines and requesting output.

3. SUBROUTINE KOEFFZ - calculates the K-coefficients, (CAI), used in the starting iteration.

4. SUBROUTINE ANFITN - performs the starting iteration and converts, if necessary, positional and velocity beginning co-ordinates

from a system originating in body #1, to a barycentric system, used in the integration.

5. SUBROUTINE WW - computes the backward differences of the accelerations, (BESCHL).

6. SUBROUTINE SCHRTT - calculates from BESCHL, the backward differences of the co-ordinates (XNABLA) and the co-ordinates (X), thus integrating from time T to time T + DELTAT.

7. SUBROUTINE DRUCKE - controls output at constant time intervals.

8. SUBROUTINE ELEMNT - controls output at arbitrary time intervals.

9. SUBROUTINE ORBIT - converts position and velocity co-ordinates (for output purposes only) into osculating elements at time intervals specified by DRUCKE or ELEMNT.

C. Input Parameters

Immediately following the IBM S/360 JCL card, //GØ.DATA5 DD* there is one card required. Then the NAMELIST data set(s) with parameters initialized (in arbitrary order) follows.

1. Card No. 1: This first card sets values of 2 parameters: NSIZE, occupying columns 1-5 of the card, right-adjusted, and INIT, occupying columns 6-10, right-adjusted.

NSIZE determines the sizes of 3 large arrays--A,B,D-- these sizes being dependent upon the number of bodies in a run of the program: $50 \geq$ NSIZE $\geq$ N. By setting this parameter himself, the programmer is able to reduce the memory requirements of the program, which may mean a higher priority in an MVT environment, hence a faster turn-around time.

INIT. The program initializes the coordinates and mass parameters for the Sun-Mercury system and the planets Venus to Pluto to those values that Schubart and Stumpff derived for J.D. = 2430000.5 . It also sets the integration parameters to the values that Schubart and Stumpff considered optimum.

By setting INIT = 0, the programmer can use these initials conditions. The NAMELIST data set is then used to initialize print-punch parameters, enter coordinates of massless bodies and alter any pre-initialized parameters (if he wishes).

By setting INIT = 1, the Schubart-Stumpff conditions are overridden. Hence all pertinent initializations must be made in the NAMELIST data set.

2. <u>NAMELIST parameters</u>: The first card of the NAMELIST DATA SET must contain & INPUT, with the & in column 2. Subsequent cards start in column 2; & END terminates the data set. Those users unfamiliar with the NAMELIST convention of the FORTRAN IV language are referred to the IBM manual on the FORTRAN IV language.

N — number of bodies being integrated. As new bodies are added during a computation at an epoch different from the starting epoch, N must be increased accordingly.

$$1 \leq N \leq NSIZE$$

EM(I) — mass of the ith body in units of m. If a relative system is used, the origin must be EM(1):

KQ — $= k^2 =$ G*EM(1) where G is the universal gravitational constant in units of $L'^3/S'^2m$, where L' is the unit of length, S' is the unit of time, EM(1) is the mass of body #1 in units of m.

W — is the conversion factor such that W*W expresses KQ in units of $L^3/S^2m$, where L & S are the units of integration.

XP(1,I)<br>XP(2,I)<br>XP(3,I) — The x,y,z components of the position of ith body in units of, L'', in the appropriate coordinate system (barycentric or relative to body #1).

DIST — is the conversion factor necessary to express XP in units of L. DIST. = L/L''

XDOT(1,I)<br>XDOT(2,I)<br>XDOT(3,I) — the u,v,w components of the velocity of the ith body in units of L'''/S''', in the appropriate coordinate system.

VEL — is the conversion factor necessary to express XDOT in units of L/S. VEL = L/S/L'''/S'''.

H — is the integration step length in units of time S.

T — is the starting epoch for the problem in arbitrary time units. This is the time that appears in the output.

DELTAT is the integration step-length in the same units as T. If converted to units of S, DELTAT is equal to H. At the start of each integration step T is incremented by DELTAT.

M order of the integration (M≤5) in the initial iteration. This corresponds to 2*M differences; in the extrapolation components to 2*M+1. M=5 is the value used by Schubart and Stumpff in their calculations.

IEG=1 Input coordinate system has its origin in body # 1.

IEG=0 Input coordinate system is barycentric, the coordinate system used in the integration. Whenever IEG = 1, the position and velocity components are converted from a relative to a barycentric system.

IEXP 10**(-IEXP) is the limit of accuracy for the initial iteration.

Note: The Schubart-Stumpff initial conditions for solar system orbital computations are coded into the program. These involve the initialization of the following parameters:

EM(1), EM(2), ..., EM(9),

XP(K,1), ........, XP(K,9), K=1,2,3

XDOT(K,1),......., XDOT(K,9) K=1,2,3,

the parameters for the Sun-Mercury System, Venus, Earth-Moon System, Mars, Jupiter, Saturn, Uranus, Neptune and Pluto, respectively; as well as the parameters

N,W,KQ,DIST,VEL,H,M,IEG,IEXP,T,DELTAT

IORB activates calls to SUBROUTINE ORBIT, which computes osculating orbital elements, from subroutines CONTRL,ELEMNT,DRUCKE and prints them as specified by values of IORBIT.

IORB=0 do not compute any orbital elements, i.e. do not call Orbit.

IORB=1 compute orbital elements for selected bodies. If IORB=1, then it is necessary to specify IORBIT(J), J=1,...,N.

IORBIT(J)=0 Do not compute or print osculating orbital elements for body #J.

IORBIT(J)=1 Compute for body #J, and print the values.

N7 determines whether or not SUBROUTINE CONTROL initiates calls to SUBROUTINE ELEMNT, a print-punch control routine. SUBROUTINE ELEMNT is used when output at arbitrary integration intervals is desired.

N7=0 No calls to SUBROUTINE ELEMNT

N7=k k calls to SUBROUTINE ELEMNT

ILEM(I),I=1,...,N7 Associated with each integration step is a step number. the ILEM array specifies in increasing order at which step numbers calls to SUBROUTINE ELEMNT are to be made.

Note: ILEM(1)≥M, else SUBROUTINE ELEMNT is never called even though N7>0.

KD is a parameter which is utilized in SUBROUTINE ELEMNT to determine which coordinate system to use in the output and the number of significant figures to print. For each body in the problem the coordinates are printed. Error bound information is also supplied as well as time, T .

KD=1 Coordinate system relative to body #1.
Single precision.

KD=2 Coordinate system: barycentric.
Single precision.

KD=3 Coordinate system relative to body #1.
Double precision.

KD=4 Barycentric system
Double precision.

IED A control parameter used in SUBROUTINE ELEMNT to specify type of output.

IED=KD Print output form using ELEMNT in form determined by value of KD.

IED=KD+4 Punch or write-on tape, as well as print. The Job Control Language (JCL) Statements will direct whether to use the punch or tape.

IZA,IDELTZ,IZN initiate calls from SUBROUTINE CONTRL to SUBROUTINE DRUCKE.

IZA first step number at which DRUCKE called.

IDELTZ Every IDELTZ step after IZA SUBROUTINE DRUCKE is called until step IZN reached.

IZN SUBROUTINE DRUCKE called. This is the last iteration step as well: the program, depending upon the value of IGØ, does one of the following: terminates the run, begins a new case, or continues the case by adding bodies of zero mass or changing the integration step-size.

IPUNCH(J) controls information written concerning jth body in SUBROUTINE DRUCKE. This information is as follows: step number, time, co-ordinates as well as error growth information. DRUCKE allows selection of bodies for which to print/punch whereas ELEMNT prints/punch for all bodies.

IPUNCH(J)= 1 Barycentric system, Double Precision Print.

= 2 Barycentric system, D.P., Punch

= 3 Barycentric system, D.P., Print and Punch

=4 Barycentric System, S.P., Print

=5 Barycentric, S.P., Punch

=6 Barycentric, S.P., Print and Punch

=7 Relative, D.P., Print

=8 Relative D.P., Punch

=9 Relative D.P. Print and Punch

=10 Relative S.P., Print

=11 Relative S.P., Punch

=12 Relative S.P., Print and Punch

=13 No output for Body #J.

When IPUNCH(J)≥14 it signals to the CONTRL Program that Body #J is of zero mass and has been added to the program at an epoch different from the starting epoch.

The output options for Body #J then are such that 14 ≤ IPUNCH(J) ≤ 26 and such that IPUNCH(J)-13 gives the correct option.

IGO is a SUBROUTINE CONTROL parameter which tells the program what to do when Step IZN is reached (IGO=1 or IGO=2) or what tests to make for incorrect data on a continuation case (IGO=3).

IGO=1 when IZ=IZN go to 2 and continue case by

(1) adding new bodies of zero mass and incrementing N accordingly

and/or

(2) decreasing H and DELTAT for a close approach and modifying IZA, IZN and IDELTZ if necessary or

(3) increasing H and DELTAT when close approach calculations completed and modifying IZA, IZN and IDELTZ if necessary.

IGO=2 when IZ=IZN, go to 100: Initialize the NAMELIST DATA SET and begin new case.

IGO=3 is used in a continuation case, where if new bodies are added, they are checked to see that they are of zero mass with proper coordinates, relative to body #1.

D. Internal Conversion of Input

1. Units

The position and/or velocity components may be converted internally to the units required for computation using the DIST and/or VEL parameters, respectively. W may be used to convert the force, KQ.

If KQ=$k^2$, where k is the Gaussian constant for solar system bodies, k=.017202 098 950, then the units are distance in a.u., time in years, mass of sun = 1. Schubart and Stumpff have used a unit system where velocities are in a.u./40 ephemeris days; hence they set KQ=$40^2k^2$ = .474 345 961 216 687, EM(1) = 1. (The starting values for INIT=0 reflect this). If dimensionless mass units are not desired, the force unit, KQ, may be adjusted instead. Whenever it is not necessary to convert distance units, set DIST=1; velocity units, set VEL=1; mass units, set W=1.

2. Coordinate System

If IEG=1, the program assumes that the origin of the relative coordinate system used to input the positional and velocity components is body #1. It then converts to a barycentric system which is used for computation. The position and velocity coordinates of body #1 must be set equal to zero if the relative system is used.

## E. Program Notes

### 1. Output Parameters

Associated with each integration step, which is of length, H, are two parameters, T and IZ. T is the time in arbitrary units and is initialized in the NAMELIST DATA SET to the starting epoch. It is incremented by DELTAT, also initialized in the NAMELIST set. DELTAT, if converted to units of time, S, used in the integration, is equal to H. T is the parameter which is printed/punched in the output whenever time is specified. IZ is the step-number. It is initialized internally and is incremented by 1 at the start of each integration step. The ILEM array is compared to IZ and whenever an element of ILEM equals IZ, a call to SUBROUTINE ELEMNT is made for printed or punched output according to the values of KD and IED for all N bodies. This comparison enables output at arbitrary step numbers. The IZA, IDELTZ, and IZN parameters are used to get output at regular step intervals using SUBROUTINE DRUCKE. In this subroutine the IPUNCH array specifies the type of output desired for each individual body.

Whenever IORB=1, SUBROUTINE ELEMNT and SUBROUTINE DRUCKE call SUBROUTINE ORBIT to compute and print the osculating elements for bodies as specified in the IORBIT array.

2. Machine Time: The running time is proportional to the number of distances which must computed at every integration step. There are $n_1(n_1 - 1)/2 + n_1 \cdot n_0$ such distances where $n_1$ is the number of bodies with mass and $n_0$ is the number of bodies of zero mass. Thus, when one computation has been obtained for a particular computer, the machine time can be predicted for other computations on that computer.

## IV. Sample Case - Orbit of Lost City Meteorite

The Lost City Meteorite struck the earth on January 3, 1970. The Prairie Network observed its entry into the earth's atmosphere and provided the data which were used in the N-Body program to compute its (probable) heliocentric orbits.

The computations were done in two parts. Using the Schubart-Stumpff initial conditions for Julian Day 2430000.5, the integration was first carried forward (H > 0 and DELTAT > 0) to Julian Day 2440590, (January 3, 1970), in order to provide the planetary coordinates at the time the meteor was observed. These coordinates plus the observed coordinates of the meteor then provided the initial conditions for an integration backward (H < 0 and DELTAT < 0) through time for 300 years.

The input data for the first part are shown in Figure 1. The first line sets the dimensions (20) of the A, B, and D matrices to A(20,3,12), B(20,3,12), D(20,20), respectively, in the subroutines in which they appear corresponding to their initializations to the same sizes in the REAL*8 statement of the MAIN program.

The second number (1) instructs CONTRL to override the programmed initializations.

The following discussion will use the left most numbers (line numbers) in referencing parameters.

A. Line 100:

1. IEG = 1: The Schubart-Stumpff planetary coordinates are relative to Body #1, (Sun-Mercury system).

B. Line 200:

1. KD = 3: when in SUBROUTINE ELEMNT print coordinates in relative system in double precision.

2. IED = 3=KD: suppresses punching of the coordinates when they are printed.

C. Lines 300-2000:

These lines contain the Schubart-Stumpff planetary coordinates for J.D. 2430000.5 with XP(1,K), XP(2,K), XP(3,K), K = 1,2,.........9, the positional coordinates relative to body #1 of the $K^{th}$ body in a.u. and XDOT(1,K), XDOT(2,K), XDOT(3,K), K = 1,2,.........9, the velocity coordinates relative to body #1 of the $K^{th}$ body in a.u./40 ephemeris days.

K = 1 Sun-Mercury system

K = 2 Venus

K = 3 Earth-Moon system

K = 4 Mars

K = 5 Jupiter

K = 6 Saturn

K = 7 Uranus

K = 8 Neptune

K = 9 Pluto

D. Lines 2100-2200:

Mass parameters, normalized to the sun of bodies, K = 1,2.......9.

E. Lines 2300-2800:

1. N = 9 the number of bodies integrated in this first part.

2. KQ = .47345961216687, reflecting the fact that the velocity units are a.u./40 days.

3. W = 1 because the computation of KQ was done externally; had it been done internally, one could have used KQ =.017202098950 and W = 40 to the same end.

4. Because the input units for the coordinates and the integration units were compatible, the program initialized values of DIST and VEL did not have to be changed.

5. T is set arbitrarily to zero although is could have been initialized to a (more) meaningful value, e.g., T = 2430000.5 .

6. Since the unit of time is 40 ephemeris days, the integration step-size, H, of 2 days is set to .05 (of 40 days) while the printing parameter, DELTAT, is correspondingly set to 2 days.

7. M = 5 determines the orders of the predictor-corrector method.

8. N7 = 5 determines that CONTRL will CALL ELEMNT for output five times during the run at steps numbers given by the ILEM array: the $10^{th}$, $30^{th}$, $500^{th}$, $1000^{th}$, and $1059^{th}$ steps. ILEM(1) must always by greater than or equal to M .

9. IZA, IDELTZ, IZN signal CONTRL to call the output subroutine, DRUCKE, starting at step number 1059, every $1059^{th}$ step thereafter, ending at step number 5295, which is also the last

integration step. At this time IGO = 1 tells CONTRL to return to statement number 2 for additional information to continue the case.

10. IORB = 1 instructs ELEMNT and DRUCKE to CALL ORBIT at each output step with orbital elements being printed for bodies 2 through 9, IORBIT(K) = 1, K = 2,3,4,.........,9 but not for body #1, IORBIT (1) = 0 .

11. IPUNCH = 9*7, 41*13 instructs DRUCKE to print in double precision the planetary coordinates relative to body #1 of bodies 1 through 9.

F. Note that the namelist begins at line 100 with &INPUT, starting in column 2 or greater and concludes at line 2800 with &END.

G. The second part (Figure 2) of the run begins with the input of the position and velocity coordinates of the Lost City Meteorite [XP(j,10),j = 1,3 and XDOT(j,10),j = 1,3, respectively], and the corresponding change of N to 10. To integrate backward, H becomes -0.5 and DELTAT,-2 . For printing T is re-initialized to 2440590. In order to be consistent with the astronomical convention of standard days, IZA, IDELTZ, and IZN are re-set so that printing (in DRUCKE) occurs in multiples of 400 days, starting at T = 2430000.5 , with IZN such that the computations continue back through 300 years. IORBIT(1) = 1 and IPUNCH(10) = 7, enable output of the coordinates (DRUCKE) and orbital elements (ORBIT) of the Lost City Meteorite. IGO = 3 indicates that the ensuing computations are a continuation of the run and not the start of a new case.

```
C     MAIN PROGRAM SETS  DIMENSION SIZES ON A, B, AND D MATRICES.
C       NSIZE......SETS MAXIMUM NUMBER OF BODIES FOR A RUN.
C       INIT.......INDICATES WHETHER USER WILL INPUT HIS OWN INITIAL
C                  CONDITIONS FOR THE MAJOR PLANETS OR USE THE SCHU-
C                  BART-STUMPFF CONDITONS FOR THE SOLAR SYSTEM AT
C                  EPOCH J.D.=2430000.5.
C       INIT=0.....USE STUMPFF-SCHUBART INITIAL CONDITIONS.
C       INIT=1.....USER SUBMITS OWN INITIAL CONDITIONS
        IMPLICIT REAL*8 (A-H,O-Z)
        REAL*8 A(20,3,12),B(20,3,12),D(20,20)
        COMMON /INOUT/ IN,IOUT
        IN=5
        IOUT=6
        READ(IN,1) NSIZE,INIT
1       FORMAT(2I5)
        CALL CONTRL(A,B,D,NSIZE,INIT)
        RETURN
        END
         SUBROUTINE CONTRL(A,B,D,NSIZE,INIT)
        IMPLICIT REAL*8 (A-H,O-Z)
C       ***  HEIDELBERG N-BODY PROGRAM  ***
C                                BY J.SCHUBART AND P.STUMPFF
C
C       ***  IBM S/360 FORTRAN 4 VERSION  ***
C                                BY P. COMELLA  & B. LOWREY
C                                   GODDARD SPACE FLIGHT CENTER
C                                   GREENBELT,MARYLAND 20771
C       ADAPTED FROM
C       ***  YALE FORTRAN 4 VERSION  ***
C                                BY M. COOKE AND J. SCHUBART
C
C
C
C
C       EXPLANATION OF INPUT
C
C       M              ORDER OF THE INTEGRATION (M.LE.5).IN THE INITIAL
C                      ITERATION THIS CORRESPONDS TO 2*M DIFFERENCES, IN
C                      THE EXTRAPOLATION COMPONENTS TO 2*M+1 DIFFERENCES.
C       IEXP           10**(-IEXP) LIMIT OF ACCURACY FOR THE INITIAL ITERATION
C       N              NUMBER OF BODIES     N.LE.50
C       KQ        =G*EM(1),WHERE G IS THE UNIVERSAL GRAVITATIONAL CONSTANT
C                  IN UNITS OF L'**3/((S'**)*M)
C       EM(I)     =MASS OF THE ITH BODY IN UNITS OF M.
C                  NOTE: IF A BARYCENTRIC SYSTEM IS NOT USED FOR INPUT,
C                        THE RELATIVE SYSTEM MUST ORIGINATE IN BODY #1.
C       XP(K,I)   =SPATIAL COMPONENTS OF THE ITH BODY IN UNITS OF LENGTH,L'',
C                  (IN THE APPROPRIATE CO-ORDINATE SYSTEM).
C       XDOT(K,I)
C                 =VELOCITY COMPONENTS OF THE ITH BODY IN UNITS OF L'''/S',
C                  WHERE S IS THE UNIT OF TIME.
C       W         =THE CONVERSION FACT OR NECESSARY TO EXPRESS KQ IN UNITS
C                  OF (L**3/S**2)**.5
C       DIST       =THE CONVERSION FACTOR NECESSARY TO EXPRESS XP IN
C                  UNITS OF L
C       VEL       =THE CONVERSION FACTOR NECESSARY TO EXPRESS XDOT IN
C                  UNITS OF L/S.
C       H         =THE INTEGRATION STEP-LENGTH IN UNITS OF S
C       T         =THE STARTING EPOCH IN ARBITRARY TIME UNITS.
C       DELTAT    =THE INTEGRATION STEP-LENGTH IN THE SAME TIME UNITS AS T.
```

```
C       (IPUNCH(I),
C          I=1,N)
C       IPUNCH(J)=1          BARY,D.P.,PRINT
C                =2          BARY,D.P. PUNCH
C                =3          BARY,D.P.,PRINT AND PUNCH
C                =4          BARY,S.P.,PRINT
C                =5          BARY,S.P.,PUNCH
C                =6          BARY,S.P.,PRINT AND PUNCH
C                =7          RELA,D.P.,PRINT
C                =8          RELA,D.P.,PUNCH
C                =9          RELA,D.P.,PRINT AND PUNCH
C                =10         RELA,S.P.,PRINT
C                =11         RELA,S.P.,PUNCH
C                =12         RELA,S.P.,PRINT AND PUNCH
C                =13             NO PRINT OUT
C             .GE. 14        NEW BODY BEING ADDED
C        IEG=0           INPUT IN BARYCENTRIC SYSTEM.THIS SYSTEM USED FOR INTEGRATION
C        IEG.NE. 0       INPUT RELATIVE TO SYSTEM ORIGINATING IN BODY(1)
C        KD=0 OR KD=2    OUTPUT IN BARYCENTRIC SYSTEM,S.P. OR D.P. RESPECTIVELY
C        KD=1 OR KD=3    OUTPUT IN RELATIVE SYSTEM, S.P. OR. D.P. RESPECTIVELY
C        IED=KD          PRINT CO-ORDINATES AND VELOCITIES FOR STEP NUMBER
C                        AS GIVEN BY ILEM(I),I=N7,N7-1,.....,1
C        IED=KD+4        PUNCH THIS OUTPUT AS WELL.
C        ILEM(I),I=1,N7    STEP NO. AT WHICH TO PRINT/PUNCH USING ELEMNT.
C        N7              NO. OF TIMES TO PRINT OUT CO-ORD. AND VEL. DURING
C                        INTEGRATION
C        IZA             FIRST STEP AT WHICH TO PRINT OUT STEP NUMBER,T,
C                        .1*H*H*MAX(LAST USED DIFF)
C        IDELTZ          EVERY IDELTZ-TH STEP CALL DRUCKE TO PRINT/PUNCH
C                        COORDINATES OF BODIES ACCORDING TO VALUES OF IPUNCH
C        IZN             LAST  STEP AT WHICH TO PRINT THIS.
C                        WHEN STEP IZN IS COMPLETED,
C                        NEW DATA ARE READ IN,EITHER NEW BODIES OF MASS ZERO OR
C                        MODIFYING CONDITIONS, THE PRESENT CALCULATIONS BEING
C                        CONTINUED. OR ENTIRELY NEW CALCULATIONS ARE BEGUN.
C        IF PRESENT CALCULATIONS ARE TO BE CONTINUED AFTER NEW BODY IS ADDED
C        THE FOLLOWING MUST BE DONE (AT THE MINIMUM)....
C        A.              SET MASS, EM(J)=0, FOR EACH NEW BODY
C        B.              IF INPUT COORDINATES AND VELOCITIES RELATIVE,
C                        FOR EACH NEW BODY
C                        SET IPUNCH(J)=X+13, WHERE 1. LE. X. LE. 13.
C                        ACCORDING TO DESCRIPTION ABOVE.
C                        INPUT VELOCITIES AND COORDINATES.
C        C.
C        IGO=1           WHEN IZ=IZN GO TO 2:ADD NEW BODIES OF ZERO MASS
C                        AND CONTINUE CASE.
C        IGO=2           WHEN IZ=IZN GO TO 100:INITIALIZE MASS PARAMETERS
C                        AND CO-ORDINATES FOR START OF NEW CASE.
C        IGO=3           CONTINUATION CASE. CHECK THAT ALL NEW BODIES OF
C                        ZERO MASS AND INPUT THEIR CO-ORDINATES.
C        IORB=0    DO NOT COMPUTE ORBITAL ELEMENTS
C            =1    COMPUTE ORBITAL ELEMENTS
C        IORBIT(J),J=1,N
C            =0    DO NOT COMPUTE ELEMENTS FOR BODY J
C            =1    COMPUTE THEM
C
C
C
 1011 FORMAT(34H0 COORDINATES OR MASSES INCOMPLETE)
 1017 FORMAT(' ILEM:CALL ELEMNT AT FOLLOWING STEPS'/(8I10))
1020  FORMAT('1',5X,'M=',I2/3X,'IEXP=',I2/3X,'IORB=',I2,
```

```
     1      ' CALL ORBIT PARAMETER'/7X,'J',5X,'IORBIT(J):'
     2      'ORBIT I/O CONTROL PARAMETERS'/(I8,8X,I2))
1021  FORMAT(' CALL DRUCKE PARAMETERS:'/4X,'IZA=',I6/
     1      ' IDELTZ=',I6/4X,'IZN=',I6/7X,'J',5X,'IPUNCH(J)'
     2      ':DRUCKE I/O CONTROL PARAMETER'/(I8,8X,I2))
1022  FORMAT('OELEMNT I/O PARAMETERS:'/4X,'IED=',I2/5X,
     1      'KD=',I2/' CALL ELEMNT PARAMETERS:'/5X,'N7=',I4)
1023  FORMAT('0',5X,'H=',D16.8/6X,'T=',D16.8/' DELTAT=',
     1      D16.8/5X,'KQ=',D20.12/6X,'W=',D16.8/3X,
     2   'DIST=',D16.8/4X,'VEL=',D16.8/6X,'N=',I3/
     3   4X,'IEG=',I3,5X,20A4/'0',2X,'J',T29,'EM(J)',
     4   T53,'X(J)',T77,'XDOT(J)'/(I4,5X,D24.16,5X,D24.16,5X,
     5   D24.16/38X,D24.16,5X,D24.16/38X,D24.16,5X,D24.16))
1024  FORMAT('0',2X,'J',T29,'EM(J)',T53,'X(J)',T77,'XPUNKT(J)'/
     1   (I4,5X,D24.16,5X,D24.16,5X,D24.16/38X,D24.16,5X,
     2   D24.16/38X,D24.16,5X,D24.16))
      DIMENSION IPUNCH(50),ILEM(100),DIFMAX(50),DIFJUL(50),KOODIZ(50),DI
     *SMIN(50),DISJUL(50),KPDIS(50)
      DOUBLE PRECISION A(NSIZE,3,12),B(NSIZE,3,12),D(NSIZE,NSIZE)
C     SCHUBART-STUMPFF INITIAL VALUES FOR EPOCH,J.D.=2430000.5. ORDER
C     OF INITIALIZATION AS FOLLOWS: SUN-MERCURY SYSTEM, VENUS,
C     EARTH-MOON SYSTEM,MARS,JUPITER,SATURN,URANUS,NEPTUNE,PLUTO.
C     POSITIONAL C-ORDINATES RELATIVE TO CENTER OF SUN-MERCURY SYSTEM.
C     IN A.U.'S.
      REAL*8 XP(3,50) /3*0.D0,
     A      -.5113942959,-.4780976854,-.18030874810,                  VENUS
     B      -.2614989917,+.8696237687,+.3771652157,                   E-M
     C      -1.295477589,-.8414136141,-.3513513446,                   MARS
     D      +3.429472643,+3.353869719,+1.354948917,                   JUPITER
     E      +6.641453441D0,+5.971569844D0,+2.182315015D0,             SATURN
     F      +11.26304125D0,+14.69525888D0,+6.279605833D0,             URANUS
     G      -30.15522934D0,+1.657000860D0,+1.437858110D0,             NEPTUNE
     1      -21.12383780D0,+28.44651101D0,+15.38826655D0,             PLUTO
     H      123*-1.23456789101112130-2/
C     VELOCITY COMPONENTS   IN A.U./40 EPHEMERIS DAYS.
      REAL*8 XDOT(3,50)/3*0.D0,
     J      +.566376818200,-.512087158900,-.266497874500,             VENUS
     K      -.674657318300,-.170094800800,-.0737743192000,            E-M
     L      +.344004260500,-.369667484300,-.178937395200,             MARS
     M      -.222864773900,+.202276882600,+.0922305178000,            JUPITER
     N      -.166228859000,+.146271235000,+.0676563647000,            SATURN
     O      -.130130816500,+.0758805177900,+.0350896779200,           URANUS
     P      -.0096195989840D0,-.115065704000,-.0468887522600,         NEPTUNE
     2      -.07074485007D0,-.08655927220D0,-.005946850713D0,         PLUTO
     Q      123*-1.23456789101112130-2/
      COMMON /TIME/T,H,HQ,HD,H10,H100,HQ1
      COMMON /BLOCK1/ CAI(5,11),EM(50),X(50,3),XPUNKT(50,3),BESCHL(50,3)
     1  ,XNABLA(50,3)
      COMMON /INDEX/N,M,M1,M2,MPM,MP1,IZ,IEG,IED,IEXP,IPUNCH
      REAL*4 IEGMSG(20,2)
      DATA IEGMSG/' INP','UT C','O-OR','DINA','TE S','YSTE','M: B',
     1 'ARYC','ENTR','IC  ',10*'    ',' INP','UT C','O-OR','DINA',
     2 'TE S','YSTE','M: O','RIGI','NATE','S IN',' BOD','Y #1',
     3 8*'    '/
      COMMON /CONBLO/ GG(11),DD(11)
      COMMON /INOUT/ IN,IOUT
      COMMON /ORB/  PI,WK,DEG,IORB,IORBIT(50)
      DATA SEVENS /-1.2345678910111213D-2/
      NAMELIST /INPUT/ N,EM,KQ,W,XP,DIST,XDOT,VEL,T,DELTAT,H,M,
     1      IEG,IEXP,IORB,IORBIT,N7,ILEM,KD,IED,IZA,IDELTZ,IZN,
```

```
     2       IPUNCH,IGO
       REAL*8 KO
C
C
      PI=DATAN2(0.D0,-1.D0)
      M=5
      IEG=1
      IEXP=14
      T=2430000.5
      DELTAT=2.D0
C      UNIT OF INTEGRATION STEP......40 EPHEMERIS DAYS
      H=.025D0
      W=1.D0
      VEL=1.D0
      DIST=1.D0
      KO=.4734596121668 7D0
      DEG=180.D0/PI
C
C     COEFFICIENTS
C
      DD(1)=0.0D0
      DD(2)=0.83333333333333333D-1
      DD(3)=0.83333333333333333D-1
      DD(4)=0.79166666666666667D-1
      DD(5)=0.75D-1
      DD(6)=0.71345899470899471D-1
      DD(7)=0.68204365079365079D-1
      DD(8)=0.65495756172839506D-1
      DD(9)=0.63140432098765432D-1
      DD(10)=.61072649861712362D-1
      DD(11)=.59240564123376623D-1
      GG(1)=0.16666666666666667
      GG(2)=0.41666666666666667D-1
      GG(3)=0.22222222222222222D-1
      GG(4)=0.14583333333333333D-1
      GG(5)=0.10615079365079365D-1
      GG(6)=0.82258597883597884D-2
      GG(7)=0.66479276895943563D-2
      GG(8)=0.55372299382716049D-2
      GG(9)=0.47180677475816365D-2
      GG(10)=.40919706740019240D-2
      GG(11)=.35997549720267974D-2
      IF(INIT.EQ.0) GO TO 2
100   DO 1 I=1,NSIZE
      EM(I)=SEVENS
      DO1 J=1,3
      XP(J,I)=SEVENS
1     XDOT(J,I)=SEVENS
2     READ(5,INPUT,END=1000)
      WRITE(IOUT,1020) M,IEXP,IORB,(J,IORBIT(J),J=1,N)
      WRITE(IOUT,1021) IZA,IDELTZ,IZN,(J,IPUNCH(J),J=1,N)
      WRITE(IOUT,1022) IED,KD,N7
      IF(N7.LE.0) GO TO 439
      WRITE(IOUT,1017) (ILEM(J),J=1,N7)
      N71=1
439   WRITE(IOUT,1023) H,T,DELTAT,KO,W,DIST,VEL,N,IEG,
     1       (IEGMSG(J,IEG+1),J=1,20),(J,EM(J),(XP(I,J),XDOT(I,J),
     2       I=1,3),J=1,N)
      MPM=M+M
      M1=MPM+1
```

```
      M2=MPM+2
      MP1=M+1
      HQ=H*H
      HQ1=.1D0*HQ
      HD=1.D0/H
      H10=.1D0*H
      H100=.01D0*H
      DO 4 I=1,N
      IF(EM(I).EQ.SEVENS) GO TO 39
      IF(IPUNCH(I).LT.14.AND.IGO.GT.2) GO TO 4
      DO 3 J=1,3
          IF(XP(J,I).EQ.SEVENS.OR.XDOT(J,I).EQ.SEVENS) GO TO 39
      X(I,J)=XP(J,I) *DIST
3     XPUNKT(I,J)=XDOT(J,I)  *VEL
4     CONTINUE
      IF(IGO.GT.2) IGO=IGO-2
      WK=KQ*W*W
33    IF(IZA.LT.M) IZA=M
35    ICHECK=0
      DO 405 I=2,N
      IF(EM(1).NE.WK) EM(I)=WK*EM(I)/EM(1)
400   IF(IPUNCH(I).LT.14) GO TO 405
401   IF(EM(I).NE.0.) GO TO 406
      IPUNCH(I)=IPUNCH(I)-13
      ICHECK=1
403   DO 404 K=1,3
      X(I,K)=X(1,K)+X(I,K)
  404 XPUNKT(I,K)=XPUNKT(1,K)+XPUNKT(I,K)
405   CONTINUE
      IF(ICHECK.EQ.1) IEG=0
      EM(1)=WK
      GO TO 40
  406 WRITE(6,1407)
 1407 FORMAT(44H ADDITION OF NEW BODY FORBIDDEN IN THIS CASE/
     1 20HO     READY      END
      RETURN
   39 WRITE(6,1011)
      GO TO 100
40    WRITE(IOUT,1024) (I,EM(I),(X(I,J),XPUNKT(I,J),J=1,3),I=1,N)
      IF(IORB.EQ.1) CALL ORBIT
      DO 440 J=2,N
      DIFMAX(J)=0.0D0
      DIFJUL(J)=0.0D0
      KOODIZ(J)=0
      DISMIN(J)=1.0D+35
      DISJUL(J)=0.0 D0
  440 KPDIS(J)=0
      DO 441 J=1,N
      DO 441 L=1,N
  441 D(J,L)=1.0D35
      CALL KOEFFZ
      IZ=M
            CALL ANFITN(A,B,D,NSIZE)
      DO 45 I=1,M
   45 T=T+DELTAT
      GO TO 47
46    CALL SCHRTT(A,B,D,NSIZE)
      T=T+DELTAT
      IZ=IZ+1
      DO 466 J=2,N
```

```
      DO 461 K=1,3
      ABB=DABS(B(J,K,M2))
      IF(ABB.LE.DIFMAX(J)) GO TO 461
  460 DIFMAX(J)=ABB
      DIFJUL(J)=T
      KOODIZ(J)=K
  461 CONTINUE
      LF=J-1
      DO 463 L=1,LF
      IF(D(J,L).GE.DISMIN(J)) GO TO 463
  462 DISMIN(J)=D(J,L)
      DISJUL(J)=T
      KPDIS(J)=100*L+J
  463 CONTINUE
      IF(J.GE.N) GO TO 47
  464 LA=J+1
      DO 466 L=LA,N
      IF (D(L,J).GE.DISMIN(J)) GO TO 466
  465 DISMIN(J)=D(L,J)
      DISJUL(J)=T
      KPDIS(J)=100*L+J
  466 CONTINUE
   47 IF(IZ-IZA)51,50,48
   48 IF(IS-IDELTZ)49,50,50
   49 IS=IS+1
      GO TO 51
   50 IS=1
      CALL DRUCKE(A,B,NSIZE)
   51 IF(IZ.EQ.IZN) GO TO 53
      IF(N7.LE.0.OR.N71.GT.N7) GO TO 46
      IF(IZ.NE.ILEM(N71))   GO TO 46
53    CALL ELEMNT (B,NSIZE)
      WRITE(6,1530) (DIFMAX(J),DIFJUL(J),KOODIZ(J),DISMIN(J),DISJUL(J),
     1 KPDIS(J),J=2,N)
 1530 FORMAT(21H0    DIFMAX AND DISMIN/
     1        (1H ,1PE15.8,1X,0PF10.1,1X,I1,5X,1PE15.8,1X,0PF10.1,1X,I4))
      N71=N71+1
      DO 531 J=2,N
      DIFMAX(J)=0.0D0
      DIFJUL(J)=0.0D0
      KOODIZ(J)=0
      DISMIN(J)=1.0D+35
      DISJUL(J)=0.0D0
  531 KPDIS(J)=0
54    IF(IZ.LT.IZN) GO TO 46
      GO TO (2,100,1000),IGO
1000  RETURN
      END
      SUBROUTINE KOEFFZ
       IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 CAI(5,11),C(10,5),R(11,11),S1,S,T,T1
      REAL*8 FAC(11) /1.D0,1.D0,2.D0,6.D0,24.D0,1.2D2,7.2D2,5.04D3,
     1 4.032D4,3.6288D5,3.6288D6/,SIG(11)/1.D0,-1.D0,1.D0,-1.D0,
     2 1.D0,-1.D0,1.D0,-1.D0,1.D0,-1.D0,1.D0/
      COMMON /BLOCK1/ CAI
     1         /INDEX/ N,M,M1,M2
      S=1.D0
      DO 8 L=1,M
      L1=2*L
      L2=L-1
```

```
      S1=S**2
      DO 7 K=1,L1
      C(K,L)=0.D0
      K2=K-2
      IF(K2) 4,4,3
    3 C(K,L)=C(K,L)+C(K2,L2)
    4 IF(K+1-L1) 5,6,7
    5 C(K,L)=C(K,L)-S1*C(K,L2)
      GOTO 7
    6 C(K,L)=C(K,L)-S1
    7 CONTINUE
    8 S=S+1.D0
C     L1=2*M+1
      L1=M1
      L2=L1-1
      S=1.D0 - S
      DO 10 L=1,L1
      R(L1,L)=1.D0
      DO 9  K=1,L2
      K2=L1-K
    9 R(K2,L)=C(K2,M)+S*R(K2+1,L)
   10 S=S+1.D0
      S=1.D0
      DO 13 I=1,M
      DO 12 L=1,L1
      L2=M2-L
      S1=S**2
      T=1.D0
      CAI(I,L)=0.D0
      DO 11 K=1,L1
      T1=R(K,L)*S1/T
      T=T+1.D0
      CAI(I,L)=CAI(I,L)+T1/T
   11 S1=S1*S
   12 CAI(I,L)=CAI(I,L)*SIG(L2)/(FAC(L)*FAC(L2))
   13 S=S+1.D0
      RETURN
      END
      SUBROUTINE ANFITN(A,B,D,NSIZE)
C
C     ANFANGSITERATION - STARTING ITERATION
C      IMPLICIT REAL*8 (A-H,O-Z)
      DIMENSION EMS(50)
      DOUBLE PRECISION A(NSIZE,3,12),B(NSIZE,3,12),D(NSIZE,NSIZE)
      DOUBLE PRECISION CAI,EM,X,XPUNKT,XNABLA,BESCHL,H,HO,FF,FI,EMMA,
     1 CAID,H1,H2,DUMMY
      COMMON /BLOCK1/ CAI(5,11),EM(50),X(50,3),XPUNKT(50,3),BESCHL(50,3)
     1,   XNABLA(50,3)
      COMMON /INDEX/ N,M,M2,M3,M4,M1,IZ,IEG,IED,IEXP,IPUNCH
      COMMON/TIME/ DUMMY,H,HO
C     IEG = 0 , BARYCENTRIC INPUT ***  ELSE INPUT RELATIVE
C
      IF(IEG.EQ.0) GO TO 105
  100 EMMA=0.0D0
      DO 101 J=1,N
  101 EMMA=EMMA+EM(J)
      DO 104 K=1,3
      X(1,K)=0.0D0
      XPUNKT(1,K)=0.0D0
      DO 102 J=2,N
```

```
      X(1,K)=X(1,K)+EM(J)*X(J,K)
  102 XPUNKT(1,K)=XPUNKT(1,K)+EM(J)*XPUNKT(J,K)
      X(1,K)=-X(1,K)/EMMA
      XPUNKT(1,K)=-XPUNKT(1,K)/EMMA
      DO 103 J=2,N
      X(J,K)=X(J,K)+X(1,K)
  103 XPUNKT(J,K)=XPUNKT(J,K)+XPUNKT(1,K)
  104 CONTINUE
105   CONTINUE
      CALL WW(D,NSIZE)
      DO 1 I=1,3
      DO 1 J=2,N
      A(J,I,12)=X(J,I)
    1 A(J,I,M1)=BESCHL(J,I)
      DELTA=10.D0**(-IEXP)
      DO 3I=2,N
      EMS(I)=DELTA*(DABS(BESCHL(I,1))+DABS(BESCHL(I,2))+DABS(BESCHL(I,3)
     *))
      IF(EMS(I).GE.1.0D-28) GO TO 3
    2 EMS(I)=1.0D-28
    3 CONTINUE
      MU=0
C     M2=M+M1
C     M3=1+M2
C     M4=M+M
      FF=0.0D0
      DO 31 MM=1,M
   31 FF=FF-1.0D0
    4 FI=FF
      IT=0
      DO 20 I=1,M2
      I1=I-M1
      I2=I1
      IF(I1) 5,20,6
    5 I1=-I1
    6 H1=FI*H
      IF(MU.GT.0) GO TO 8
    7 H2=0.5D0*H1*H1
    8 DO 13 J=2,N
      DO 13 K=1,3
      IF(MU.EQ.0) GO TO 12
    9 X(J,K)=0.D0
      DO 11 L=1,M2
      L1=L
      IF(I2.GT.0) GO TO 11
   10 L1=M3-L1
   11 X(J,K)=X(J,K)+CAI(I1,L1)*B(J,K,L)
      X(J,K)=H0*X(J,K)
      GO TO 13
   12 X(J,K)=H2*A(J,K,M1)
   13 X(J,K)=A(J,K,12)+H1*XPUNKT(J,K)+X(J,K)
      CALL WW(D,NSIZE)
      IF(MU.EQ.0) GO TO 18
   14 DO 17 J=2,N
      IF(IT.GT.0) GO TO 18
   15 DEL=DABS(BESCHL(J,1)-B(J,1,I))+DABS(BESCHL(J,2)-B(J,2,I))+DABS
     *(BESCHL(J,3)-B(J,3,I))
      IF(DEL.LE.EMS(J)) GO TO 17
   16 IT=1
   17 CONTINUE
```

```
   18 DO 19 J=2,N
      DO 19 K=1,3
   19 A(J,K,I)=BESCHL(J,K)
   20 FI=FI+1.0D0
      IF(IT.EQ.0) GO TO 23
   21 DO 22 J=2,N
      DO 22 K=1,3
      DO 22 L=1,M2
   22 B(J,K,L)=A(J,K,L)
      MU=MU+1
      GO TO 4
   23 IF(MU.EQ.0) GO TO 21
   24 WRITE(6,25) MU
   25 FORMAT(40H1STARTING ITERATION. NO. OF ITERATIONS =,I3)
      I2=M-1
      DO 30 J=2,N
      DO 30 K=1,3
      XNABLA(J,K)=0.0D0
      DO 27 L=1,M2
      CAID=CAI(M,L)
      IF(I2.EQ.0) GO TO 27
26    CAID=CAID-CAI(I2,L)
27    XNABLA(J,K)=XNABLA(J,K)+CAID*A(J,K,L)
      XNABLA(J,K)=H*(XPUNKT(J,K)+H*XNABLA(J,K))
      DO 28 I=1,M4
      L1=M2-I
      DO 28 L=1,L1
28    A(J,K,L)=A(J,K,L+1)-A(J,K,L)
      DO 29 L=1,M2
      L1=M3-L
29    B(J,K,L)=A(J,K,L1)
30    B(J,K,M3)=0. D0
      RETURN
      END
      SUBROUTINE SCHRTT(A,B,D,NSIZE)
C
C     EXTRAPOLATIONSSCHRITT,ORDNUNG=2*M+1 - STEP OF INTEGRATION
       IMPLICIT REAL*8 (A-H,O-Z)
      DOUBLE PRECISION A(NSIZE,3,12),B(NSIZE,3,12),D(NSIZE,NSIZE)
      DOUBLE PRECISION CAI,EM,X,XPUNKT,BESCHL,XNABLA,DUMMY,H,HQ,S,DD,GG
      COMMON /BLOCK1/ CAI(5,11),EM(50),X(50,3),XPUNKT(50,3),BESCHL(50,3)
     1,   XNABLA(50,3)
      COMMON /INDEX/ N,M,M1,M2
      COMMON/TIME/ DUMMY,H,HQ
      COMMON /CONBLO/ GG(11),DD(11)
      DO 2 J=2,N
      DO 2 K=1,3
      S=0.0D0
      DO 1 L=2,M1
    1 S=S+DD(L)*B(J,K,L+1)
      XNABLA(J,K)=XNABLA(J,K)+HQ*(B(J,K,1)+S)
    2 X(J,K)=X(J,K)+XNABLA(J,K)
      CALL WW(D,NSIZE)
      DO 4 J=2,N
      DO 4 K=1,3
      A(J,K,1)=BESCHL(J,K)
      DO 3 L=1,M1
    3 A(J,K,L+1)=A(J,K,L)-B(J,K,L)
      DO 4 I=1,M2
    4 B(J,K,I)=A(J,K,I)
```

```
      RETURN
      END
      SUBROUTINE WW(D,NSIZE)
      IMPLICIT REAL*8 (A-H,O-Z)
      REAL*8 D(NSIZE,NSIZE),DX(3)
      COMMON /BLOCK1/ CAI(5,11),EM(50),X(50,3),XPUNKT(50,3),BESCHL(50,3)
     1,   XNABLA(50,3)
      COMMON /INDEX/ N
      COMMON/TIME/ DUMMY,H
      DO 2 K=1,3
      X(1,K)=0.D0
      DO 1 J=2,N
    1 X(1,K)=X(1,K)+EM(J)*X(J,K)
    2 X(1,K)=-X(1,K)/EM(1)
      DO 14 J=1,N
      DO 3 K=1,3
3     BESCHL(J,K)=0. D0
      DO 14 I=1,N
      IF(I-J)4,14,7
    4 IF(EM(I))5,14,5
    5 A=EM(I)/D(I,J)
      DO 6 K=1,3
    6 BESCHL(J,K)=BESCHL(J,K)+A*(X(I,K)-X(J,K))
      GO TO 14
    7 IF(EM(I))9,8,9
    8 IF(EM(J))9,14,9
    9 DO 10 K=1,3
   10 DX(K)=X(I,K)-X(J,K)
      A=DX(1)*DX(1)+DX(2)*DX(2)+DX(3)*DX(3)
      D(I,J)=DSQRT(A)
      D(J,I)=A*D(I,J)
      IF(J-1)11,14,11
   11 IF(EM(I))12,14,12
   12 A=EM(I)/D(J,I)
      DO 13 K=1,3
   13 BESCHL(J,K)=BESCHL(J,K)+A*DX(K)
   14 CONTINUE
      RETURN
      END
       SUBROUTINE ELEMNT (B,NSIZE)
      IMPLICIT REAL*8  (A-H,O-Z)
C
C     COORDINATES AND VELOCITY, IED=KD PRINT ONLY, IED=KD+4 WITH PUNCH
C
999   FORMAT(//51H      COORDINATES AND VELOCITIES FOR STEP NUMBER
     1   I6/9H0    T =     1PD17.9)
1003  FORMAT(3(A4,I1,A4,I2,A4,1PD22.15,A4/),
     1 3(2A4,I1,A4,I2,A4,1PD22.15,A4/))
 1004 FORMAT(32H00.01*H*MAX(LAST USED DIFF.)  = ,1PE10.2  )
 1005 FORMAT(14H0 BODY NUMBER ,I2,6H  EM= ,1PD23.15)
 1006 FORMAT(9H COORD.= ,1PD23.15,6X,9HVELOCITY=,1PD23.15)
 1007 FORMAT(9H COORD.= ,1P3D16.7,11H VELOCITY= ,1P3D16.7)
1008  FORMAT(A4,1PD22.15,A4)
      DOUBLE PRECISION B(NSIZE,3,12)
      DOUBLE PRECISION CAI,EM,X,XPUNKT,BESCHL,XNABLA,GG,DD,V(6),S,T,
     1 H,H0,HD,H10,H100
      REAL*8 MU
      REAL*4 ALPHA(7)
      COMMON /BLOCK1/ CAI(5,11),EM(50),X(50,3),XPUNKT(50,3),BESCHL(50,3)
     1,   XNABLA(50,3)
```

```
      COMMON /INDEX/ N,M,M1,M2,MPM,MP1,IZ,IEG,IFD,IEXP,IPUNCH(50)
      COMMON/ TIME/ T,H,HQ,HD,H10,H100
      COMMON /CONBLO/ GG(11),DD(11)
      COMMON /ORB/ PI,MU,RAD,IORB,IORBIT(50)
      DATA ALPHA/' XP(,     )=       XDOT(   T=,     '/
      IF(IED.GE.4)WRITE(7,1008) ALPHA(6),T,ALPHA(7)
      DO 3 K=1,3
      XPUNKT(1,K)=0.0D0
      DO 2 J=2,N
      S=0.0D0
      DO 1 L=1,M1
    1 S=S+GG(L)*B(J,K,L+1)
      XPUNKT(J,K)=HD*XNABLA(J,K)+H*(B(J,K,1)*0.5D0-S)
    2 XPUNKT(1,K)=XPUNKT(1,K)+FM(J)*XPUNKT(J,K)
    3 XPUNKT(1,K)=-XPUNKT(1,K)/FM(1)
      WRITE(6,999)  IZ,T
      IF(MOD(IED,2).EQ.0) GO TO 200
      DO 100 J=2,N
      WRITE(6,1005) J,FM(J)
      DO 50 K=1,3
      K3 = K + 3
      V(K)=X(J,K)-X(1,K)
50    V(K3)=XPUNKT(J,K)-XPUNKT(1,K)
      IF(IED.GE.4) WRITE(7,1003) (ALPHA(1),K,ALPHA(2),J,ALPHA(3),V(K),
     1 ALPHA(7),K=1,3),(ALPHA(4),ALPHA(5),K,ALPHA(2),J,ALPHA(3),
     2      V(K+3),ALPHA(7),K=1,3)
      IF(KD.EQ.1) GO TO 75
      WRITE(6,1006) V(1),V(4),V(2),V(5),V(3),V(6)
      GO TO 100
75    WRITE(6,1007) V
100   CONTINUE
      GO TO 500
200   DO 300 J=1,N
      WRITE(6,1005) J,FM(J)
      IF(KD.EQ.1) GO TO 275
      IF(IED.GE.4) WRITE(7,1003) (ALPHA(1),K,ALPHA(2),J,ALPHA(3),X(J,K),
     1 ALPHA(7),K=1,3),(ALPHA(4),ALPHA(5),K,ALPHA(2),J,ALPHA(3),
     2XPUNKT(J,K),ALPHA(7),K=1,3)
      WRITE(6,1006) (X(J,K),XPUNKT(J,K),K=1,3)
      GO TO 300
275   WRITE(6,1007) (X(J,K),K=1,3),(XPUNKT(J,K),K=1,3)
300   CONTINUE
500   BM=0.
      DO 600 J=2,N
      DO 600 K=1,3
      BN=DABS(B(J,K,M2))
600   BM=DMAX1(BM,BN)
      BM = BM*H100
      WRITE(6,1004) BM
      IF (IORB.EQ.1) CALL ORBIT
      RETURN
      END
      SUBROUTINE ORBIT
C     EQUATIONS FOUND IN NASA DOCUMENT NO. X-643-67-198
C                            BY ROBERT BLANCHARD
      IMPLICIT REAL*8    (A-H,O-Z)
      REAL*8 MU,P(4),INCL
      COMMON /BLOCK1/ C(105),X(50,3),XDOT(50,3)
      COMMON /INOUT/ IN,IOUT
      COMMON /INDEX/ N
```

```
      COMMON /ORB/ PI,MU,RAD,IORB,IORBIT(50)
      WRITE(IOUT,1)
      DO 1000 J=1,N
      IF(IORBIT(J).EQ.0)GO TO 1000
      PERIOD=0.D0
      RA=0.D0
      P1= X(J,1)-X(1,1)
      P2= X(J,2)-X(1,2)
      P3= X(J,3)-X(1,3)
      P40=P1*P1+P2*P2+P3*P3
      P4=DSQRT(P40)
      V1= XDOT(J,1)-XDOT(1,1)
      V2= XDOT(J,2)-XDOT(1,2)
      V3= XDOT(J,3)-XDOT(1,3)
      V40=V1*V1+V2*V2+V3*V3
      V4= DSQRT(V40)
      SQRTMU=DSQRT(MU)
      DZ= P1*V1+P2*V2+P3*V3
      DZMU= DZ/SQRTMU
      AINV=2.D0/P4-V40/MU
      ABAINV=DABS(AINV)
      IF (ABAINV.LE.1.D-20) GO TO 1500
50    SQAINV=DSQRT(ABAINV)
      CZ=1.D0-AINV*P4
      SZ=DZMU*SQAINV
C
C     SEMI-MAJOR AXIS-A
C
      A=1./AINV
      SZ2=DZMU*DZMU*AINV
C
C     ECCENTRICITY -ECC
C
      ECC=DSQRT(SZ2+CZ*CZ)
C
C     ANGULAR MOMENTUM/UNIT MASS
C
      H1= P2*V3-P3*V2
      H2= V1*P3-P1*V3
      H3= P1*V2-P2*V1
      H4= DSQRT(H1*H1+H2*H2+H3*H3)
      TERMA=(1.D0/P4-AINV)/ECC
      TERMB=DZ/(ECC*MU)
      P(1)=TERMA*P1-TERMB*V1
      P(2)= TERMA*P2-TERMB*V2
      P(3)= TERMA*P3-TERMB*V3
C
C     INCLINATION
C
      COSI=H3/H4
      SINI =DSQRT(1.D0-COSI*COSI)
      IF(COSI.NE.0.D0) GO TO 75
      INCL=90. D0
      GO TO 100
75    INCL=DATAN(SINI/COSI)
C
C     RIGHT ASCENSION OF ASCENDING NODE
C
100   IF (SINI.EQ.0.D0)GO TO 150
      DUM = H4*SINI
```

```
      SOMG=H1/DUM
      COMG=-H2/DUM
      OMEGA= DATAN2(SOMG,COMG)
C
C     ARGUMENT OF PERIGEE
C
      ARGPER=DATAN2(P(3)/SINI,P(1)*COMG+P(2)*SOMG)
C
C     MEAN MOTION
C
  150 XMOT= SQRTMU*ABAINV*SQAINV
C
C     MEAN AND ECCENTRIC ANOMOLY
C
200   IF(A.LT.0.D0)GO TO 225
      EZ=DATAN2(SZ,CZ)
      GO TO 250
225   EZ=DLOG((CZ+SZ)/ECC)
250   XMA=EZ-SZ
      IF(A.LT.0.D0)XMA=-XMA
C
C     TRUE ANAMOLY
C
300   E2=.5D0*EZ
      EZ=EZ*RAD
      DUM=(1.D0+ECC)/(1.D0-ECC)
      IF(A.LT.0.D0)GO TO 325
      XP=DCOS(E2)
      Y=DSQRT(DUM)*DSIN(E2)
      GO TO 350
325   EX1=DEXP(E2)
      EX2=1.D0/EX1
      Y=DSQRT(-DUM)*.5D0*(EX1-EX2)
      XP=.5*(EX1+EX2)
350   TA =    DATAN2(Y,XP) *2.D0
C
C     TIME FROM PERIGEE
C
375   TPER=-XMA/XMOT
C
C     BYPASS PERIOD AND APOFOCUS CALC IF HYPERBOLIC
C
      IF (A.LT.0.D0) GO TO 900
C
C     PERIOD
C
      PERIOD= PI*A**1.5/SQRTMU
C
C     APOFOCUS
C
      RA=A*(1.D0+ECC)
900   INCL= INCL*RAD
      OMEGA= OMEGA*RAD
      ARGPER= ARGPER*RAD
      XMA= XMA*RAD
      EZ= EZ*RAD
      TA= TA*RAD
      WRITE(IOUT,2) J,A,ECC,INCL,OMEGA,TA,ARGPER,TPER
1000  CONTINUE
      RETURN
```

```
1500  WRITE(IOUT,3) J,AINV
      GO TO 1000
1     FORMAT('0BODY NO.',T13,'SEMI-MAJOR AXIS',T33,'ECCENTRICITY',
     1T53,'INCLINATION',T73,'R.A. OF ASCENDING NODE',
     2 T98,'TRUE ANAMOLY',T113,'ARG. OF PERIGEE')
2     FORMAT(I8,2X,6D20.12/T113,'TIME FROM PERIGEE'/110X,D20.12)
3     FORMAT('0SEMI-MAJOR AXIS OF BODY NO.',I3,' IS TOO LARGE. 1./A=',
     1D20.12)
      END
      SUBROUTINE DRUCKE(A,B,NSIZE)
      IMPLICIT REAL*8     (A-H,O-Z)
      REAL*8 MU
      REAL*4 ALPHA(7)
      DOUBLE PRECISION A(NSIZE,3,12),B(NSIZE,3,12)
      DOUBLE PRECISION CAI,EM,X,XPUNKT,S(6),T,H,HD,H10,H100,HQ1 ,HQ
     1,GG,DD
     1,BESCHL,XNABLA
      COMMON /BLOCK1/ CAI(5,11),EM(50),X(50,3),XPUNKT(50,3)
     1 ,BESCHL(50,3),XNABLA(50,3)
      COMMON /INDEX/ N,M,M2,M1,MPM,MP1,IZ,IEG,IED,IEXP,IPUNCH(50)
      COMMON /TIME/ T,H,HQ,HD,H10,H100,HQ1
      COMMON /CONBLO/ GG(11),DD(11)
      COMMON /ORB/ PI,MU,RAD,IORB,IORBIT(50)
 1000 FORMAT(1H0/14H0 STEP NUMBER ,I6,9H      T = ,1PD17.9)
 1002 FORMAT(14H0 BODY NUMBER ,I2)
1003  FORMAT(14H0 BODY NUMBER  ,I2/9H COORD.=  ,1P3D28.15/
     1 9H VEL.=    ,1P3D28.15)
1004  FORMAT(14H0 BODY NUMBER  ,I2/9H COORD.=  ,1P3D20.7/
     1 9H VEL.=    ,1P3D20.7)
1005  FORMAT(2(A4,I1,A4,I2,A4,1PD15.7,A4)/A4,I1,A4,I2,A4,1PD15.7,A4,2A4,
     1 I1,A4,I2,A4,1PD15.7,A4/2(2A4,I1,A4,I2,A4,1PD15.7,A4))
1006  FORMAT(3(A4,I1,A4,I2,A4,1PD22.15,A4/),
     1 3(2A4,I1,A4,I2,A4,1PD22.15,A4/))
 1007 FORMAT(34H0 0.1*H*H*MAX(LAST USED DIFF.)  = ,1PE10.2)
1008  FORMAT(A4,1PD22.15,A4)
      DATA ALPHA/' XP(,    )=      XDOT(   T=,    '/
      DO 3 K=1,3
      XPUNKT(1,K)=0.D0
      DO 2 J=2,N
      D=0.D0
      DO 1 L=1,M2
1     D=D+GG(L)*B(J,K,L+1)
      XPUNKT(J,K)=HD*XNABLA(J,K)+H*(B(J,K,1)*0.5D0-D)
    2 XPUNKT(1,K)=XPUNKT(1,K)+EM(J)*XPUNKT(J,K)
    3 XPUNKT(1,K)=-XPUNKT(1,K)/EM(1)
      IP=0
      DO 999 J=1,N
      I=IPUNCH(J)
      IF(I.EQ.13) GO TO 999
      IF(I.GE.7) GO TO 350
      IP=IP+1
      GO TO (50,100,150,200,250,300),I
50    IF(IP.EQ.1) WRITE(6,1000) IZ,T
       WRITE(6,1003) J,(X(J,K),K=1,3),(XPUNKT(J,K),K=1,3)
      GO TO 999
150   IF(IP.EQ.1) WRITE(6,1000) IZ,T
       WRITE(6,1003) J,(X(J,K),K=1,3),(XPUNKT(J,K),K=1,3)
100   IF(IP.EQ. 1)WRITE(7,1008) ALPHA(6),T,ALPHA(2)
      WRITE(7,1006) (ALPHA(1),K,ALPHA(2),J,ALPHA(3),X(J,K),ALPHA(7),
     1 K=1,3),(ALPHA(4),ALPHA(5),K,ALPHA(2),J,ALPHA(3),XPUNKT(J,K),
```

```
     2 ALPHA(7),K=1,3)
      GO TO 999
200   IF(IP.EQ.1) WRITE(6,1000) IZ,T
       WRITE(6,1004) J,(X(J,K),K=1,3),(XPUNKT(J,K),K=1,3)
      GO TO 999
300   IF(IP.EQ.1) WRITE(6,1000) IZ,T
       WRITE(6,1004) J,(X(J,K),K=1,3),(XPUNKT(J,K),K=1,3)
250   IF(IP.EQ.1)WRITE(7,1008)ALPHA(6),T,ALPHA(2)
      WRITE(7,1005) (ALPHA(1),K,ALPHA(2),J,ALPHA(3),X(J,K),ALPHA(7),
     1 K=1,3),(ALPHA(4),ALPHA(5),K,ALPHA(2),J,ALPHA(3),XPUNKT(J,K),
     2 ALPHA(7),K=1,3)
      GO TO 999
350   DO 400 K=1,3
      S(K+3)=XPUNKT(J,K)-XPUNKT(1,K)
400   S(K)=X(J,K)-X(1,K)
      IP=IP+1
      GO TO(999,999,999,999,999,999,450,500,550,600,650,700),I
450   IF(IP.EQ.1) WRITE(6,1000) IZ,T
      WRITE(6,1003) J,S
      GO TO 999
550   IF(IP.EQ.1) WRITE(6,1000) IZ,T
      WRITE(6,1003) J,S
500   IF(IP.EQ.1)WRITE(7,1008)ALPHA(6),T,ALPHA(2)
      WRITE(7,1006) (ALPHA(1),K,ALPHA(2),J,ALPHA(3),S(K),ALPHA(7),
     1 K=1,3),(ALPHA(4),ALPHA(5),K,ALPHA(2),J,ALPHA(3),S(K+3),
     2 ALPHA(7),K=1,3)
      GO TO 999
600   IF(IP.EQ.1) WRITE(6,1000) IZ,T
      WRITE(6,1004) J,S
      GO TO 999
700   IF(IP.EQ.1) WRITE(6,1000) IZ,T
      WRITE(6,1004) J,S
650   IF(IP.EQ.1)WRITE(7,1008)ALPHA(6),T,ALPHA(2)
      WRITE(7,1005) (ALPHA(1),K,ALPHA(2),J,ALPHA(3),S(K),ALPHA(7),
     1 K=1,3),(ALPHA(4),ALPHA(5),K,ALPHA(2),J,ALPHA(3),S(K+3),
     2 ALPHA(7),K=1,3)
999   CONTINUE
      IF (IP.EQ.0) GO TO 1200
      BM=0.D0
      DO 1100 J=2,N
      DO 1100 K=1,3
1100   BM=DMAX1(BM,DABS(B(J,K,M1)))
      BM=H01*BM
      WRITE(6,1007) BM
1200  IF (IORB .EQ.1) CALL ORBIT
      RETURN
      END
```

0887 CARDS

```
 20      1
&INPUT      IEG=1,
 KD=3,IED=3,
 XP(1,1)=0.,XP(2,1)=0.0,XP(3,1)=0.,
 XDOT(1,1)=0.0,XDOT(2,1)=0.0,XDOT(3,1)=0.,
XP(1,2)=-0.5113942959,XP(2,2)=-.4780976854,XP(3,2)=-0.1830874810,
XDOT(1,2)=+0.5663768182,XDOT(2,2)=-0.5120871589,XDOT(3,2)=-.2664978745,
 XP(1,3)=-0.2614989917,XP(2,3)=+0.8696237687,XP(3,3)=+0.3771652157,
 XDOT(1,3)=-0.6746573183,XDOT(2,3)=-.1700948008,XDOT(3,3)=-0.0737743192,
 XP(1,4)=-1.295477589,XP(2,4)=-0.8414136141,XP(3,4)=-0.3513513446,
 XDOT(1,4)=+0.3440042605,XDOT(2,4)=-0.3696674843,XDOT(3,4)=-0.1789373952,
 XP(1,5)=+3.429472643,XP(2,5)=3.353869719,XP(3,5)=+1.354948917,
 XDOT(1,5)=-0.2228647739,XDOT(2,5)=+0.2022768826,XDOT(3,5)=+0.0922305178,
 XP(1,6)=+6.641453441,XP(2,6)=+5.971569844,XP(3,6)=+2.182315015,
 XDOT(1,6)=-0.1662288590,XDOT(2,6)=+0.1462712350,XDOT(3,6)=+0.0676563647,
 XP(1,7)=+11.26304125,XP(2,7)=+14.69525888,XP(3,7)=+6.279605833,
 XDOT(1,7)=-0.1301308165,XDOT(2,7)=+0.07588051779,XDOT(3,7)=+0.03508967792,
 XP(1,8)=-30.15522934,XP(2,8)=+1.657000860,XP(3,8)=+1.437858110,
 XDOT(1,8)=-0.009619598984,XDOT(2,8)=-0.1150657040,XDOT(3,8)=-0.04688875226,
 XP(1,9)=-21.12383780,XP(2,9)= +28.44651101,XP(3,9)=+15.38826655,
 XDOT(1,9)=-0.07074485007,XDOT(2,9)=-0.0865592722,XDOT(3,9)=-0.005946850713,
 EM(1)=1.,EM(2)=2.451E-06,EM(3)=3.03591E-06,EM(4)=3.2326E-07,EM(5)=9.54786E-04,
 EM(6)=2.8558E-04,EM(7)=4.3727E-05,EM(8)=5.17759E-05,EM(9)=2.78E-06,
 KO=.47345961216687,T=0.,DELTAT= 2.,H=.05,M=5,N7=5,W=1.,
 ILEM(1)=10,ILEM(2)=50,ILEM(3)=500,ILEM(4)=1000,ILEM(5)=1059,
 IORB=1,IORBIT(3)=1,IZA=1059,IDELTZ=1059,IZN=5295,N=9,IEXP=10,
  IORBIT(1)=0,IORBIT(2)=1,IORBIT(4)=1,IORBIT(5)=1,IORBIT(6)=1,IORBIT(7)=1,
 IORBIT(8)=1,IORBIT(9)=1,IGO=1,
  IPUNCH=9*7,41*13,&END
```

**INPUT DATA - ORBIT OF LOST CITY METEORITE - PART I**

0029 CARDS

**FIGURE 1**

```
&INPUT  IGO=3,  T=2440590.,  DELTAT=-2.,  H=-.05,  N = 10,
N7=5,  ILEM(1)=10,  ILEM(2)=20,  ILEM(3)=30,  ILEM(4)=50,  ILEM(5)=100,
IZA=5295,  IDELTZ=400,  IZN=54750,
 EM(10)=0.,XP(1,10)=+.51198278,XP(2,10)=-2.0232007,XP(3,10)=-.87761397,
 XDOT(1,10)=+.32109876,XDOT(2,10)=+.11411804, XDOT(3,10)=+.13002863,
IPUNCH(10)=20,  &END
```

**INPUT DATA - ORBIT OF LOST CITY METEORITE, PART 2**

0006 CARDS

**FIGURE 2**

## References


1. R.C. Blanchard and H. Wolf, "A Method of Calculating Interplanetary Trajectories", GSFC X-643-67-198, May 1967.

2. R.L. Duncombe and G.M. Clemence, "Provisional Ephemeris of Mars 1950-2000", U. S. Naval Obs. Circ. No. 90, 1960.

3. R.L. Duncombe, "Provisional Ephemeris of Mars 1800-1950", U. S. Naval Obs Circ. No. 95, 1964.

4. W.J. Eckert, D. Brouwer, G.M. Clemence, "Coordinates of the Five Outer Planets 1653-2060", Astr. Papers Wash. Vol. XII, 1951.

5. P. Herget, "Solar Coordinates 1800-2000", Astr. Papers Wash. Vol. XIV, 1953.

6. P. Herget, "Coordinates of Venus 1800-2000", Astr. Papers Wash. Vol. XV, Part III, 1955.

7. J.H. Lieske, "Newtonian Planetary Ephemerides 1800-2000, Development Ephemeris Number 28", JPL Tech. Rep. 32-1206, 1967.

8. B.G. Marsden, "Comets and Nongravitational Forces", Astron. J., Vol. 73, p. 367, 1968.

9. J. Schubart and P. Stumpff, "On an N-Body Program of High Accuracy for the Computation of Ephemerides of Minor Planets and Comets", Veröff Astron. Rechen-Inst., Heidelberg No. 18, Verlag. G. Braun, Karlsruhe, 1966.

10. J. Schubart and M. Cooke, "Yale Fortran IV Version of Schubart-Stumpff N-Body Program", Yale University Observatory, 1965.