JPL Ephemeris Implemented on a DG NOVA Computer

by

Randall L. Ricklefs
Department of Astronomy
University of Texas
Austin, TX 78712

ABSTRACT

The standard JPL export ephemeris package has been implemented on a Data General NOVA minicomputer operating under RDOS. Particular problems in conversion include the incompatibility of 24000 character blocks with the local tape drive, the maximum size of integer that can be accommodated in a 16 bit word, the unavailability of DECODE/ENCODE statements in D.G.'s FORTRAN and differences in binary READ's and WRITE's.

1 Introduction

The standard JPL ephemeris software package provides a means of reformatting, accessing, and otherwise manipulating the JPL ephemerides. This package, written in FORTRAN, has been implemented on many large main-frames in the past, but our requirements for the McDonald Laser Ranging System (MLRS) necessitated its implementation on a minicomputer, specifically a Data General NOVA 4 running under RDOS. This paper outlines the conversion steps required. The reader should consult JPL's document on this package (Newhall,1976) and D.G.'s FORTRAN manual (Data General, 1978) for more information. The specific ephemeris used in our conversion was DE-111/LE-55, although any of JPL's export ephemerides should work as well.

In short, the ephemeris package contains the programs BCDEPH, XSHORT, TDUMP, as well as the READE subroutine set. BCDEPH converts the BCD ephemeris tape into a binary file; XSHORT creates a restricted time-span binary file for easy access; TDUMP prints operator-specified parts of the binary file; and TSTRDE exercises the READE package and prints the results. The READE package accesses and interpolates the ephemeris to obtain the position and velocity of a

2 Conversion Problems

The first problem encountered was that the unpacked ASCII ephemeris file was blocked by 24000 characters (300 80-character lines). Our Digi-Data tape drive cannot handle physical records larger than 8192 characters. Hence, the tape had to be reformatted in 8000-character records on the University of Texas' dual CDC Cyber 170/750 system.

Integers with magnitudes larger than 32767 cannot be handled in D.G. 16 bit integer words. Larger integers must be handled as double precision reals to retain accuracy. This problem appeared with the data "KEY1" header 999999 which was changed to 9999. The integer group with "KEY1"=1050 contains integers greater than the acceptable value. Hence, this was converted to a double precision group by our version of BCDEPH. Many large integer variables and constants in various of the routines were also changed to double precision real.

Since NOVA words are 2 characters long rather than 6 as on the JPL UNIVAC, the variable "C" which indicates the length of BCD records had to be tripled by BCDEPH for output to the binary ephemeris file.

NOVA FORTRAN IV does not support some of the options assumed by JPL's software. DECODE/ENCODE, NAMELIST, and BACKSPACE are not available. DECODE, used in BCDEPH, was replaced by a parser (SCAN) that pulls numbers one at a time from a line in A1 format. ENCODE, which is used only in XSHORT was replaced by a write to a temporary file which is subsequently rewound and read with a different format. The BACKSPACE used by XSTATE to back up the ephemeris a certain number of records was replaced by a call to LH to rewind the tape and search again for the appropriate ephemeris record.

Since the ephemeris manipulation programs such as XSHORT and TDUMP will be used in an interactive rather than batch mode on the NOVA, the NAMELIST variables are now accepted from the operator via the console.

The JPL software assumes that an unformatted read or write such as

```
READ(IUNIT) N
```

is a binary read. Under NOVA FORTRAN one must replace these with READ BINARY or WRITE BINARY to get the same results. Also, under binary I/O, one must read an entire record at a time, as the next read starts where the last read ended. To read an entire binary record from a JPL ephemeris one must replace the example above with

```
READ BINARY(IUNIT) N,(A(K),K=1,N)
```

where N is the record length.

The routine XSTATE contains local variables which are assumed to be zero initially and are assumed to retain their values between calls. Since that is definitely not the case on the NOVA (all variables are put onto a run-time stack for the sake of reentrancy) these variables were put in common storage and explicitly initialized.

The NOVA interprets a variable dimensioned X(N1,N2,1) to have a third dimension of "1". Hence, referring to X(1,1,2), for example, results in an "out of bounds" error. This caused problems in the READE routines.

Other changes necessary to implement this ephemeris package were calls to system routines to open and close disk and tape files being used. The UNIVAC and CYBER, for example, automatically open and close these files.

A page of output from the TSTRDE program has been included. The entire output and a dump of part of the binary ephemeris tape comes with the ephemeris tape from JPL.

3 Conclusion

In conclusion, the ephemeris package was not difficult to implement once the FORTRAN language incompatibilities were isolated and resolved. The entire conversion required a month or so of part-time effort. Although this conversion is specifically oriented toward a NOVA, many of the problems will probably be similar for other brands of minicomputers.

4 References

Newhall,X.X., JPL Export Ephemeris DE-96 Users Guide, JPL, Pasadena, Jan. 21, 1976.

NOVA-LINE FORTRAN IV User's Manual, 093-000053-09, Southboro, Data General Corporation, 1978.

5 Appendix A

```
C     THIS PROGRAM READS THE BCD TAPE CONTAINING THE EXPORT
C     JPL PLANETARY EPHEMERIS AND RECONSTRUCTS THE ABSOLUTE
C     FORTRAN-READABLE BINARY TAPE EXPECTED BY THE INTERPOLATION
C     ROUTINE.  THE FORMAT OF THE BCD TAPE EXPECTED BY THIS
C     PROGRAM IS A FOLLOWS --
C
C        CHARACTERS (300 CARD IMAGES OF 80 CHARACTERS EACH).
C        THE MAIN BODY OF THIS PROGRAM WILL PROCESS ONE CARD
C        IMAGE AT A TIME AND USE THE FORTRAN 'DECODE' STATEMENT
C        TO FORM THE BINARY REPRESENTATION OF THE DATA.
C
C                         ***************
C                         *             *
C                         *  IMPORTANT  *
C                         *             *
C                         ***************
C
C        THE ABILITY TO READ THE BLOCKED, 300-CARD RECORDS
C        IS MACHINE-DEPENDENT.  THE FORTRAN 'READ' STATEMENTS
C        CANNOT DO THE TASK.  THEREFORE, YOU, THE USER, MUST
C        SUPPLY A FORTRAN-CALLABLE SUBROUTINE CALLED RCI,
C        WHOSE JOB WILL BE TO READ THIS BLOCKED TAPE (ONE RECORD
C        AT A TIME) AND TRANSMIT ONE CARD IMAGE PER CALL.
C        CALLING SEQUENCE --
C
C           CALL RCI(IMAGE,EOF)
C
C              IMAGE  IS AN ARRAY LARGE ENOUGH TO HOLD
C                     ONE CRD IMAGE(80 CHRACTERS)
C                     RCI IS TO FILL THIS ARRAY WITH
C                     THE NEXT SEQUENTIAL CARD IMAGE
C                     ON THE FILE.
C
C                EOF  IS A LOGICAL FLAG TO BE SET BY
C                     RCI.  IT IS TO BE SET TO .TRUE. IF
C                     AN ATTEMPT TO FETCH A CARD IMAGE
C                     RESULTS IN AN END-OF-FILE BEING
C                     READ.  OTHERWISE, IT SHOULD BE
C                     SET TO .FALSE. .
C
C
C     THE BINARY FILE WHICH THIS PROGRAM WRITES WILL BE ON
C     UNIT 12.  IT IS SUGGESTED THAT RCI BE CODED TO READ
C     THE BCD TAPE ON UNIT 9.
C
C           (O/P EPH ON UNIT 2, I/P ON 1. RLR 3/81)
C
C
C     IF YOU HAVE TROUBLE OR QUESTIONS ABOUT THIS EPHEMERIS
C     PACKAGE, CALL OR WRITE
```

```
C                                    SKIP NEWHALL
C                                    JET PROPULSION LABORATORY
C                                    4800 OAK GROVE DRIVE
C                                    PASADENA, CALIFORNIA  91103
C
C                                    PHONE --
C                                      COMMERICAL --
C                                         (213) 354-7000
C                                      FTS --
C                                         792-7000
C
C
C       MODIFIED FOR NOVA4 BY R. RICKLEFS       3/81
C       UNIVERSITY OF TX, MCDONALD OBS
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
C
        COMMON /DIALOG/ IDUM,IPTR,IMAGE(80),DW(2),DPNUM
        INTEGER IPTR,IMAGE,DW,SCAN
        DOUBLE PRECISION DPNUM

        COMMON /GARB/ A(6000)
        INTEGER A

      INTEGER EPHNAM(12)
      INTEGER C,H(5),H2
      INTEGER NDP1(3)
C
      DOUBLE PRECISION NDP2(3),NDP3(3)
      DOUBLE PRECISION PW2(201)
      DOUBLE PRECISION B(1500)
      EQUIVALENCE(A,B)
C
      LOGICAL EOF
C
C
C       SET UP DOUBLE PRECISION TABLE OF POWERS OF 2 FROM
C       2**(-100) THROUGH 2**(100).  THIS TABLE IS NEEDED
C       TO DECODE DP DATA ON THE FILE.
C
      PW2(100)=.5D0
      PW2(101)=1.D0
      PW2(102)=2.D0
      DO 50 K=1,99
      PW2(100-K)=PW2(101-K)*PW2(100)
      PW2(102+K)=PW2(101+K)*PW2(102)
   50 CONTINUE

C  NOVA-DEPENDENT CODE TO OPEN INPUT& OUTPUT FILES FOLLOWS
        CALL INIT("MT0",0,IERR)
        IF (IERR.EQ.1) GO TO 51
            TYPE "INIT MT0 ERROR",IERR
            STOP
51      CALL MTOPD(1,"MT0:0",0,IERR)
```

```
        IF (IERR.EQ.1) GO TO 52
            TYPE "OPEN MTO ERROR",IERR
            STOP
52      WRITE(10,1000)
1000    FORMAT(" ENTER OUTPUT EPHEMERIS FILE NAME: ",Z)
        READ(11,1001) EPHNAM
1001    FORMAT(12A2)
        CALL OPEN(2,EPHNAM,2,IERR)
        IF (IERR.EQ.1) GO TO 53
            TYPE "OUTPUT EPHEMERIS OPEN ERROR",IERR
            STOP

53      REWIND 2
C
C       READ 5-WORD INTEGER HEADER RECORD
C
    1 CONTINUE
      CALL RCI(IMAGE,EOF)
      IF(EOF) GO TO 99
C
C       DECODE AND WRITE ONTO OUTPUT UNIT
C
C     DECODE(103,IMAGE)C,(H(K),K=1,5)
        C=5
        IPTR=1
        DO 55 K=1,5
            II=SCAN(IPTR,IMAGE,DW,DPNUM)
            IF (DPNUM.LE.36535.D0) GO TO 55
            WRITE(10,1010) IMAGE
1010        FORMAT(" INTEGER TOO LARGE"/80A1)
            DW(2)=9999
55          H(K)=DW(2)
C  IF THIS IS GROUP 1050, IT WILL BE D.P., NOT INTEGER, ON OUTPUT
        H2=H(2)
        IF (H(4).EQ.1050) H2=2
      WRITE BINARY(2) H(1),H2,H(3),H(4),H(5)
      WRITE(12,105)H
  105 FORMAT('0 HEADER:',6I12)
C
C       IF THIS IS THE LAST HEADER ON THE FILE, QUIT.
C
      IF(H(2).NE.5) GO TO 2
    3 ENDFILE 2
      REWIND 2
      WRITE(12,106)
  106 FORMAT('0 ALL DONE.')
        CALL FCLOS(1)
        CALL FCLOS(2)
      STOP
C
C       OTHERWISE, WORD 2 OF HEADER INDICATES TYPE OF
C       DATA THAT FOLLOWS IN THIS GROUP.  READ FIRST
C       CARD IMAGE
C
```

```
    2 CALL RCI(IMAGE,EOF)
      IF(EOF) GO TO 99
      IT=H(2)-1
      GO TO (11,21,31),IT
C
C        THIS BLOCK HANDLES DOUBLE PRECISION DATA.  EACH CARD IMAGE
C        HAS UP TO 3 DP NUMBERS, EACH NUMBER BEING ENCODED AS A
C        SERIES OF 3 INTEGERS.  THE INTERPRETATION OF THESE 3 INTEGERS
C        IS --    DP NO. = N2*(2**(N1-30))+N3*(2**(N1-60)).
C
C  11 DECODE(102,IMAGE)C,((NDP(K1,K2),K1=1,3),K2=1,3)
C 102 FORMAT(I5,3(I3,2I11))
11       IPTR=1
         II=SCAN(IPTR,IMAGE,DW,DPNUM)
         C=DW(2)
         CALL DECDP(NDP1,NDP2,NDP3)
      NWB=0
      DO 12 I=1,C,3
      IF(I.EQ.1) GO TO 13
      CALL RCI(IMAGE,EOF)
C     DECODE(102,IMAGE)NDUM,((NDP(K1,K2),K1=1,3),K2=1,3)
         IPTR=1
         CALL DECDP(NDP1,NDP2,NDP3)
   13 NW=MIN0(3,C-I+1)
      DO 14 K=1,NW
      NX=71+NDP1(K)
      NWB=NWB+1
      IF(NDP2(K).NE.0.D0) GO TO 16
      B(NWB)=0.D0
X        WRITE(12,1010) NWB,B(NWB)
      GO TO 14
   16 IF(NDP1(K).GT.-41) GO TO 15
      B(NWB)=NDP2(K)*(2.D0**(NDP1(K)-30))
     1      +NDP3(K)*(2.D0**(NDP1(K)-60))
X        WRITE(12,1010) NWB,B(NWB)
      GO TO 14
   15 B(NWB)=NDP2(K)*PW2(NX)+NDP3(K)*PW2(NX-30)
         NNN=NX-30
X        WRITE(12,1010) NWB,B(NWB),NX,NDP2(K),PW2(NX),NDP3(K),PW2(NNN)
X1010    FORMAT(5X,I5,E25.18/5X,I5,4E25.18)
   14 CONTINUE
   12 CONTINUE
C
C        WRITE RECORD AND CHECK FOR END-OF-GROUP TRAILER
C
      WRITE BINARY(2)C,(B(K),K=1,C)
      IF((C.EQ.1).AND.(B(1).EQ.0)) GO TO 1
      GO TO 2
C
C
C        THIS BLOCK HANDLES INTEGER DATA
C
C  NOTE: INTEGER DATA CAN BE >36535 ALLOWED IN 16 BIT D.G. WORD.
C        HENCE, THIS INTEGER DATA IS OUTPUT AS DOUBLE PRECISION REAL.
```

```
C        RLR 3/81
C
C  21 DECODE(103,IMAGE)C,(A(K),K=1,6)
C 103 FORMAT(I5,1X,6I12)
21       IPTR=1
         II=SCAN(IPTR,IMAGE,DW,DPNUM)
         C=DW(2)
         DO 210 K=1,6
             II=SCAN(IPTR,IMAGE,DW,B(K))
             IF (II.LT.0) GO TO 211
210      CONTINUE
211    DO 22 I=1,C,6
       IF(I.EQ.1) GO TO 23
       CALL RCI(IMAGE,EOF)
C      DECODE(103,IMAGE)NDUM,(A(I+K-1),K=1,6)
         IPTR=1
         DO 215 K=1,6
             II=SCAN(IPTR,IMAGE,DW,B(I+K-1))
             IF (II.LT.0) GO TO 23
215      CONTINUE
   23 CONTINUE
   22 CONTINUE
C
C        WRITE RECORD AND CHECK FOR TRAILER
C
X        WRITE(12,1010) (B(K),K=1,C)
X1010    FORMAT(5X,10F10.0)
       WRITE BINARY(2)C,(B(K),K=1,C)
       IF(C.EQ.1) GO TO 1
       GO TO 2
C
C
C        THIS BLOCK HANDLES BCD DATA
C
C  31 DECODE(104,IMAGE)C,(A(K),K=1,12)
C 104 FORMAT(I5,1X,12A6)
31       IPTR=1
         II=SCAN(IPTR,IMAGE,DW,DPNUM)
         C=DW(2)
         DO 310 K=1,36
         KK=2*(K-1)+7
310      A(K)=(IMAGE(KK).AND.177400K).OR.(IMAGE(KK+1)/400K)
       DO 32 I=1,C,12
       IF(I.EQ.1) GO TO 33
       CALL RCI(IMAGE,EOF)
C      DECODE(104,IMAGE)NDUM,(A(I+K-1),K=1,12)
         DO 315 K=1,36
         KK=2*(K-1)+7
315      A(3*I+K-3)=(IMAGE(KK).AND.177400K).OR.(IMAGE(KK+1)/400K)
   33 CONTINUE
   32 CONTINUE
C      GO TO 25
C
C  UNIVAC WORD IS 6 BYTES, D.G. WORD IS 2, HENCE WE MULTIPLY BY 3:
```

```
        C=3*C
        WRITE BINARY(2) C,(A(K),K=1,C)
        IF (C.EQ.3) GO TO 1
        GO TO 2
C
C       ERROR EXIT FOR END-OF-FILE READ
C
   99 WRITE(12,110)
  110 FORMAT('0 UNEXPECTED END OF FILE ENCOUNTERED.')
      GO TO 3
C
C
      END

        SUBROUTINE RCI(IMAGE,EOF)
C  RCI READ'S A CARD IMAGE FROM THE INPUT JPL BCD EPHEMERIS AND
C  RETURNS IT TO BCDEPH.
C
C  INPUT (FROM TAPE):
C       BLK   -- 4000  WORD (100 80 CHAR LINE) RECORDS (ONE AT A TIME)
C
C  OUTPUT:
C       IMAGE -- 80 CHARACTER LINE IMAGE IN A1 FORMAT,
C       EOF   -- LOGICAL END OF FILE FLAG. TRUE IF EOF OR READ ERROR ON BCD
C                EPHEMERIS FILE. THIS WILL NOT BE SET FOR A NORMAL EOF WHICH IS
C                DETECTED IN BCDEPH.
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$ RLR 4/81

        LOGICAL EOF
        INTEGER IMAGE(80)

        COMMON /RCICMN/ II,BLK(4096),NREAD
        INTEGER II,BLK,NREAD
        LOGICAL FRSTIM

        DATA II,NREAD /0,0/

        IF (II.NE.NREAD) GO TO 20

C  READ A NEW RECORD FROM TAPE
        K=0
        NREAD=0

C  USE NOVA MAG TAPE I/O (NEEDED TO READ 4000 WORD BLOCKS RATHER THAN 256)
        CALL MTDIO(1,K,BLK,ISTAT,IERR,NREAD)
        IF (IERR.EQ.1) GO TO 15
            WRITE(10,1020) IERR,ISTAT,NREAD
1020        FORMAT(" READ ERROR: ",I3,", STATUS WORD: ",OI8,
     X              ", WORDS READ: ",I8)
            EOF= .TRUE.
            RETURN

15      II= 0
```

```
C  TRANSFER A LINE IMAGE FROM BLK TO "IMAGE" FOR CALLER
C  (CONVERTS FROM A2 TO A1 FORMAT.)
20      DO 25 K=1,80,2
        II= II+ 1
        IMAGE(K)= BLK(II).AND.177400K.OR.40K
        IMAGE(K+1)= (BLK(II).AND.377K)*400K.OR.40K
25      CONTINUE
        EOF = .FALSE.

X       WRITE(10,1000) II, IMAGE, IMAGE
X1000   FORMAT(I4/80A1/8(10(1X,0I6)/))
        RETURN

        END


        SUBROUTINE DECDP(NDP1,NDP2,NDP3)
C
C  ROUTINE DECODES 3 SETS OF D.P. NUMBER COMPONENTS
C
        INTEGER NDP1(3)
        DOUBLE PRECISION NDP2(3),NDP3(3)

        COMMON /DIALOG/ IDUM,IPTR,IMAGE(80),DW(2),DPNUM
        INTEGER IPTR,IMAGE,DW,SCAN
        DOUBLE PRECISION DPNUM

        DO 10 K=1,3
        II=SCAN(IPTR,IMAGE,DW,DPNUM)
        IF (II.LT.0) RETURN
        NDP1(K)=DW(2)
        II=SCAN(IPTR,IMAGE,DW,NDP2(K))
        II=SCAN(IPTR,IMAGE,DW,NDP3(K))
 10     CONTINUE

        RETURN
        END


        INTEGER FUNCTION SCAN(INDEX,CHARAY,DINUM,DPNUM)

C  THIS FUNCTION CONVERTS AN ASCII-CHARACTER ARRAY IN A1 FORMAT TO A
C  SIGNED 2-WORD INTEGER & A DOUBLE-PRECISION REAL.  THE USER PASSES
C  THE CHARACTER ARRAY OR IT CAN BE READ FORM THE CONSOLE DEVICE.  WITH
C  THE FORMER AN INDEX TO THE 1ST ARRAY MEMBER TO BE CONVERTED IS ALSO
C  PASSED.
C
C  ANY LEADING BLANKS ARE IGNORED.  THE NUMBER MAY BE PRECEDED BY AN
C  ASCII PLUS OR MINUS SIGN.  CONVERSION IS TERMINATED BY A BLANK OR
C  WHEN THE INDEX REACHES 80, WHICHEVER COMES 1ST.  IF THE CHARACTER
C  IMMEDIATELY FOLLOWING THE NUMBER IS NOT A BLANK NO CONVERSION IS
```

```
C  PERFORMED.  IF THE NUMBER HAS A FRACTIONAL PART (INDICATED BY A
C  DECIMAL POINT) THE 2-WORD INTEGER WILL CONTAIN ONLY THE INTEGER
C  PORTION, WHILE THE DOUBLE-PRECISION REAL WILL REFLECT BOTH PARTS.
C
C  ON INPUT:
C       CHARAY - (INTEGER ARRAY) CHARACTER STRING IN A1 FORMAT (CHARACTER
C                IN LEFT BYTE, BLANK IN RIGHT BYTE OF EACH WORD).  NUMBER
C                MUST BE FOLLOWED BY AT LEAST ONE NON-DIGIT.
C       INDEX  - (INTEGER) POSITION IN CHARAY TO BEGIN CONVERSION,
C                IF = -1, CHARAY WILL BE READ FROM $TTI
C
C  ON OUTPUT:
C       CHARAY - IF INDEX WAS = -1 WILL CONTAIN LINE READ FROM $TTI,
C                OTHERWISE UNCHANGED
C       SCAN   - (INTEGER) RETURN INFORMATION SWITCH
C                =-1 IF LINE CONTAINED ONLY BLANKS
C                = 0 IF NUMBER WAS CONVERTED
C                = INDEX OF NON-DIGIT CHARACTER OTHERWISE
C       INDEX  - POSITION IN CHARAY FOLLOWING LAST DIGIT CONVERTED
C       DINUM  - (2-WORD INTEGER) CONTAINS NUMBER IF CONVERSION
C                WAS PERFORMED
C       DPNUM  - (DOUBLE-PRECISION REAL) CONTAINS NUMBER IF
C                CONVERSION WAS PERFORMED
C
C  CALLS: DWMPY,DWADD,DWSUB,DWFL
C     NOTE: THE ABOVE ROUTINES ARE FOUND IN THE NOVA FORTRAN 4
C       COMMERCIAL SUBROUTINE PACKAGE. THEY PERFORM DOUBLE WORD INTEGER ARITH-
C       METIC. DWMPY=MULTIPLY,  DWADD=ADD,  DWSUB=SUBTRACT,
C       DWFL=DOUBLE WORD INTEGER TO DOUBLE PRECISION FLOATING POINT CONVERSION.
C                                                                   RWH
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

        INTEGER CHARAY(1),NCHK,SIGNSW,DWDGT(2),DW10(2),INDEX,DINUM(2)
        DOUBLE PRECISION DPNUM,DPTMP

C  INITIALIZATIONS
        NCHK= 0           ;STATUS RETURNED FROM DOUBLE-INTEGER ROUTINES
        SIGNSW= 0         ;SET NON-ZERO IF MINUS SIGN FOUND
        DWDGT(1)= 0       ;UPPER HALF OF CONVERTED DIGIT
        DPNUM= 0          ;DOUBLE-PRECISION RUNNING TOTAL
        DINUM(1)= 0       ;DOUBLE-INTEGER RUNNING TOTAL
        DINUM(2)= 0
        DW10(1)= 0        ;DOUBLE-INTEGER 10 FOR MULTIPLICATION
        DW10(2)= 10

C  IF INDEX = -1, READ LINE FROM $TTI
        IF(INDEX.NE.-1)GO TO 2
 888    READ(11,777,ERR=888,END=888)(CHARAY(I),I=1,80)
 777    FORMAT(80A1)
        INDEX= 1

C  SKIP TO 1ST NON-BLANK CHARACTER
 2      IF(CHARAY(INDEX).NE."  ")GO TO 3
        INDEX= INDEX+1
```

```
      IF(INDEX.LE.80)GO TO 2

C  HERE LINE CONTAINED ONLY BLANKS
      SCAN= -1
      GO TO 12

C  IF 1ST NON-BLANK IS PLUS OR MINUS, SET SIGN FLAG & BUMP PTR
 3    IF((CHARAY(INDEX).NE."- ").AND.(CHARAY(INDEX).NE."+ "))GO TO 4
      IF(CHARAY(INDEX).EQ."- ")SIGNSW= 1
      INDEX= INDEX+1

C  CONVERT CHAR DIGITS OF INTEGER PART TO DOUBLE-WORD INTEGER
 4    IF((CHARAY(INDEX).LT."0 ").OR.(CHARAY(INDEX).GT."9 "))GO TO 5
      DWDGT(2)= ISHFT(CHARAY(INDEX)-"0 ",-8)
      CALL DWMPY(DW10,DINUM,NCHK)
      CALL DWADD(DWDGT,DINUM,NCHK)
      IF(NCHK.NE.0)GO TO 11
      INDEX= INDEX+1
      IF(INDEX.GT.80)GO TO 9
      GO TO 4

C  IF NEXT CHAR IS DECIMAL PT, CONVERT FRACTIONAL PART STORING IN DPNUM
 5    IF(CHARAY(INDEX).NE.". ")GO TO 8
      DPTMP= 10D0
 6    INDEX= INDEX+1
      IF(INDEX.GT.80)GO TO 9
      IF((CHARAY(INDEX).LT."0 ").OR.(CHARAY(INDEX).GT."9 "))GO TO 8
      IF(CHARAY(INDEX).EQ."0 ")GO TO 7
      DPNUM= DPNUM+DFLOAT(ISHFT(CHARAY(INDEX)-"0 ",-8))/DPTMP
 7    DPTMP= DPTMP*10D0
      GO TO 6

C  IF NEXT CHAR ISN'T BLANK ERROR HAS OCCURRED
 8    IF(CHARAY(INDEX).NE."  ".AND.CHARAY(INDEX).NE."- ")GO TO 11

C  HERE CONVERSION FINISHED, CONVERT TO NEGATIVE IF MINUS SIGN WAS FOUND
 9    SCAN= 0
      IF(SIGNSW.EQ.0)GO TO 10
      DWDGT(2)= 0
      CALL DWSUB(DINUM,DWDGT,NCHK)
      DINUM(1)= DWDGT(1)
      DINUM(2)= DWDGT(2)

C  ADD INTEGER PART TO DPNUM
 10   CALL DWFL(DINUM,DPTMP)
      IF(SIGNSW.EQ.0)DPNUM= DPNUM+DPTMP
      IF(SIGNSW.NE.0)DPNUM= DPTMP-DPNUM
      GO TO 12

C  HERE CHAR WHICH WASN'T PLUS, MINUS, DIGIT, OR BLANK WAS ENCOUNTERED
 11   SCAN= INDEX

 12   RETURN
```

```
      END



C     THIS PROGRAM READS A PLANETARY EPHEMERIS FILE AND CREATES A
C    SHORTENED VERSION.  THE INPUT FILE SHOULD BE ON FORTRAN LOGICAL
C     UNIT 9, AND THE OUTPUT FILE IS ON UNIT 2.
C
C     THE USER MUST INPUT START AND/OR STOP EPOCHS VIA THE CONSOLE.
C     IF EITHER EPOCH IS NOT INPUT, THE
C     EPOCH ON THE INPUT FILE IS USED AS A DEFAULT.
C
C     THE USER HAS A CHOICE OF TWO FORMATS TO EXPRESS EITHER EPOCH:
C
C       (1) VIGESIMAL, AN ENCODED FORM OF GREGORIAN CALENDAR DATE.
C           THE GENERAL FORM OF A VIGESIMAL DATE IS 2 INTEGERS:
C
C                        YYYYMM00DD,HHNNSSFFFF
C
C           WHERE YYYY = YEAR
C                   MM = MONTH NUMBER
C                   DD = DAY NUMBER IN MONTH
C                   HH = HOUR IN DAY
C                   NN = MINUTES
C                   SS = SECONDS
C                 FFFF = TENTHS OF MILLISECONDS
C
C            EXAMPLE:  THE EPOCH JULY 4, 1976, 13:21:17 WOULD BE
C            EXPRESSED AS  1976070004,1321170000
C
C
C       (2) JULIAN EPHEMERIS DATE
C
C
C     IF VIGESIMAL EPOCH IS INPUT, USE THE INPUT SYMBOL VEP.  THE START
C     EPOCH DESIRED ON THE OUTPUT FILE SHOULD BE IN VEP(1) AND VEP(2).
C     THE STOP EPOCH SHOULD BE IN VEP(3) AND VEP(4).  IF JULIAN DATE
C     FORMAT IS WANTED, USE THE DOUBLE PRECISION INPUT SYMBOL EP(1) FOR
C     START ND EP(2) FOR STOP.  EITHER EPOCH MAY BE IN EITHER FORMAT.
C
C       MODIFIED FOR D.G. NOVA BY R. RICKLEFS   3/81
C       UNIVERSITY OF TEXAS, MCDONALD OBS.
C
      DOUBLE PRECISION EP(2),VSEC
      DOUBLE PRECISION SS(6),D1
      DOUBLE PRECISION D(1000)
C
      INTEGER DW(2),ANS(80),SCAN
C     INTEGER VEP(4)
      INTEGER H(5),H4(8),EL(6),N1,EOF(5)
      INTEGER A(4000)
      INTEGER FILNAM(12)
```

```
      COMMON /SECTOR/ H, IFILL(2)
      COMMON /XSHCMN/ D,EOF,EL,N1,D1,H4

      EQUIVALENCE(A,D)
      DATA EOF/2,5,0,1,0/
      DATA EL/' EARLI   LAT'/,N1/1/,D1/1.D0/
      DATA H4/9999,1010,1020,1030,1040,1041,1050,1070/
C
C
      M=2
      J=9
      WRITE(10,110)
110   FORMAT(10X,"EPHEMERIS SHORTENING PROGRAM"/
     X          " ENTER INPUT (LONG) EPHEMERIS FILE NAME: ",Z)
      READ(11,105) FILNAM
      CALL OPEN(J,FILNAM,2,IERR)
        IF (IERR.NE.1) CALL ERROR(1,IERR)

      WRITE(10,115)
115   FORMAT(" ENTER OUTPUT (SHORT) EPHEMERIS FILE NAME: ",Z)
      READ(11,105) FILNAM
      CALL OPEN(M,FILNAM,2,IERR)
        IF (IERR.NE.1) CALL ERROR(2,IERR)
      CALL OPEN(3,"XTEMPX",2,IERR)
        IF (IERR.NE.1) CALL ERROR(3,IERR)
      REWIND J
C
C       SPACE INPUT FILE TO GROUP 1010 TO GET EXISTING LIMITS
C       FOR PRINTING AND THEN TO 1030 TO READ DP LIMITS
C
      CALL LH(J,1010)
      DO 1 I=1,3
      READ BINARY(J)N,(A(K),K=1,N)
      WRITE(12,101)(A(K),K=1,N)
  101 FORMAT(5X,42A2)
    1 CONTINUE
      READ BINARY(J)N,(A(K),K=1,N)
      CALL LH(J,-1030)
      READ BINARY(J)N,(SS(K),K=1,N)
      REWIND J
C
C       PRINT MESSAGE AND READ INPUTS
C
19    WRITE(10,102)
  102 FORMAT('0 ENTER START AND STOP EPOCH (JULIAN DATE OR YYYYMMOODD.HHMMSS).')
C     READ(5,IN)
      IPTR=-1
      IF (SCAN(IPTR,ANS,DW,EP(1))) 22,20,19
20    IF (SCAN(IPTR,ANS,DW,EP(2))) 22,22,19
22    DO 2 I=1,2
C     VEP(I*2)=0
C     IF(VEP(I*2-1).EQ.0) GO TO 3
C     CALL VIGSEC(VEP(I*2-1),EP(I))
C     EP(I)=EP(I)/86400.D0+2433282.5D0
```

```
      IF (EP(I).LT.2500000.D0) GO TO 3
          CALL VIGSEC(EP(I),VSEC)
          EP(I) = VSEC/86400.D0 + 2433282.5D0
    3 CONTINUE
      IF(EP(I).EQ.0.D0) EP(I)=SS(I)
    2 CONTINUE
X       TYPE "EPOCHS:",EP
C
C       BE SURE INPUT EPOCHS ARE WITHIN SPAN ON TAPE, AND SET LIMITS
C
      EP(1)=DMAX1(EP(1),SS(1))
      EP(2)=DMIN1(EP(2),SS(2))
      SS(1)=SS(1)+DINT((EP(1)-SS(1))/SS(4))*SS(4)
      SS(2)=SS(2)-DINT((SS(2)-EP(2))/SS(4))*SS(4)
C
C       NOW READ FROM INUT FILE TO OUTPUT FILE, MAKING
C       CHANGES AS WE GO
C
C       COPY HEADER RECORD
C
      REWIND M
    4 READ BINARY(J)H
      WRITE BINARY(M)H
      WRITE(12,103)H
  103 FORMAT('0 HEADER ',6I12)
C
C       WORD 4 OF HEADER IS KEY1 -- TELLS US WHAT DATA IS IN GROUP
C
      DO 5 I=1,8
      IP=I
      IF(H(4).EQ.H4(I)) GO TO 6
    5 CONTINUE
      IP=0
    6 GO TO (11,12,11,13,11,14,14,15),IP
C
C       THIS BLOCK COPIES A BCD OR INTEGER GROUP
C
   11 READ BINARY(J)N,(A(K),K=1,N)
      WRITE BINARY(M)N,(A(K),K=1,N)
      N=N/3
      IF(N.EQ.1) GO TO 4
      GO TO 11
C
C       THIS BLOCK ENCODES AND WRITES A NEW TITLE GROUP (1010)
C
   12 READ BINARY(J)N,(A(K),K=1,N)
      WRITE BINARY(M)N,(A(K),K=1,N)
      WRITE(12,106)(A(K),K=1,N)
  106 FORMAT('0 LABEL GROUP ON NEW EPHEMERIS:',/,5X,42A2)
      DO 21 I=1,2
      REWIND 3
      READ BINARY(J)N,(A(K),K=1,N)
      D(101)=(SS(I)-2433282.5D0)*86400.D0
      CALL SECCAL(D(101),A(51))
```

```
      A(53)=A(53)+1900
C
C  ENCODE IS REPLACED BY WRITE TO & READ FROM TEMPORARY FILE.
C     ENCODE(104,A(1))EL(I),SS(I),(A(50+K),K=1,3),D(101)
      IIL=3*(I-1)+1
      IIH=3*I
      WRITE(3,104) (EL(II),II=IIL,IIH),SS(I),(A(K),K=51,53),D(101)
  104 FORMAT(1X,3A2,'EST EPOCH: JED=',F10.1,',',I3,'/',I2,'/',I4,
     1F13.0,' SECS PAST 1950',15X)
      REWIND 3
      READ(3,105) (A(K),K=1,N)
  105 FORMAT(42A2)
      WRITE BINARY(M)N,(A(K),K=1,N)
      WRITE(12,101)(A(K),K=1,N)
   21 CONTINUE
      GO TO 11
C
C        HERE WHEN WE ARE READY TO WRITE LIMITS GROUP (1030)
C
   13 READ BINARY(J)N,(D(K),K=1,N)
      WRITE BINARY(M)N,(SS(K),K=1,N)
      READ BINARY(J)N,D(1)
      WRITE BINARY(M)N,D(1)
      GO TO 4
C
C        THIS BLOCK COPIES DP RECORDS
C
   14 READ BINARY(J)N,(D(K),K=1,N)
      WRITE BINARY(M)N,(D(K),K=1,N)
      IF(N.GT.1) GO TO 14
      GO TO 4
C
C        THIS BLOCK HANDLES THE DP EPHEMERIS DATA RECORDS
C
   15 READ BINARY(J)N,(D(K),K=1,N)
      IF(N.EQ.1) GO TO 16
      IF(SS(1).LE.D(1)) WRITE BINARY(M)N,(D(K),K=1,N)
      IF(SS(2).GT.D(2)) GO TO 15
   16 WRITE BINARY(M)N1,D1
      WRITE BINARY(M)EOF
      END FILE M
      REWIND M
      CALL FCLOS(J)
      CALL FCLOS(M)
      CALL DELETE("XTEMPX")
      WRITE(12,107)
  107 FORMAT('0 ALL DONE.')
      STOP
      END


C******************************
C
```

```
      SUBROUTINE LH(NU,KEY1)

C
C  READ TO THE BEGINNING OF RECORD TYPE KEY1 ON I/O UNIT NU.
C        KEY1 > 0 - READ FORWARD,
C        KEY1 < 0 - REWIND AND READ
C
C******************************
C
      INTEGER IH(5),NSEC,NRS,IBUF(7200)
      DOUBLE PRECISION BUF
C
      COMMON/SECTOR/IH,NSEC,NRS
      COMMON/CETB3/ BUF(1800)

      EQUIVALENCE (BUF,IBUF)
C
      KY=IABS(KEY1)
      IF(KEY1.GT.0) REWIND NU
      NSEC=0
 1001 READ BINARY(NU)IH
      NSEC=NSEC+1
X       TYPE " LH:",NSEC,IH
      IF(IH(4).EQ.KY) RETURN
 1002 READ BINARY(NU)N
      IF(IH(2).EQ.2) GO TO 10
C  BCD
      READ BINARY(NU) (IBUF(K),K=1,N)
      N=N/3
      GO TO 20
C  DOUBLE PRECISION
10    READ BINARY(NU) (BUF(K),K=1,N)
20    NSEC=NSEC+1
C       TYPE " ...",NSEC,N,BUF(1,1),BUF(2,1)
      IF(IH(3).EQ.1) GO TO 1001
      IF(N.EQ.1) GO TO 1001
      GO TO 1002
C
      END


C*************************
C
      SUBROUTINE SECCAL(SECS,N)
C
C*************************
C
C     THIS SUBROUTINE CONVERTS DOUBLE PRECISION FLOATING-POINT SECONDS
C     PAST OH, JAN. 1, 1950 TO CALENDAR DATE.
C
      DOUBLE PRECISION T,SECS
      DOUBLE PRECISION AL,ANC4,ANC,ANLY,ANY
      INTEGER N(7)
```

```
C
      T=DINT((SECS+61530883200.D0)*1.D4+.5D0)
      N(4)=IDINT(DMOD(T/36.D6,24.D0))
      N(5)=IDINT(DMOD(T/60.D4,60.D0))
      N(6)=IDINT(DMOD(T/1.D4,60.D0))
      N(7)=IDINT(DMOD(T,1.D4))
C
      AL=DINT(T/864.D6)
      ANC4=DINT(AL/146097.D0)
      ANC=DMIN1(DMOD(AL,146097.D0)/36524.D0,3.D0)
      ANLY=DMOD(DMOD(AL,146097.D0),36524.D0)/1461.D0
      ANY=DMIN1(DMOD(DMOD(DMOD(AL,146097.D0),36524.D0),1461.D0)/365.D0,3.D0)
      ANC=DINT(ANC)
      ANLY=DINT(ANLY)
      ANY=DINT(ANY)
C
      N(2)=IDINT(AL-(ANC4*146097.D0+ANC*36524.D0+ANLY*1461.D0+ANY*365.D0))+1
      DO 8 I=1,12
      IF(N(2)-1.LT.MOD(I+(I-1)/5,2)+30) GO TO 9
    8 N(2)=N(2)-MOD(I+(I-1)/5,2)-30
    9 N(1)=MOD(I+1,12)+1
      N(3)=DINT(ANC4*400.D0+ANC*100.D0+ANLY*4.D0+ANY)+I/11-1900
C
C     END OF CALENDAR DATE CONVERSION
C
      RETURN
      END


C*****************************
C
      SUBROUTINE VIGSEC(V,S)
C
C*****************************
C
C     THIS SUBROUTINE CONVERTS A VIGESIMAL DATE TO
C     D.P. SECONDS PAST OH, JAN. 1, 1950
C
C     VIGESIMAL FORMAT:  D.P. REAL  ..  YYYYMMOODD.HHNNSSXXXX
C       WHERE  YYYY = CALENDAR YEAR (E.G. 1973)
C                MM = CALENDAR MONTH (E.G. 04 = APRIL)
C                DD = CALENDAR DAY (E.G. 15)
C                HH = HOUR OF DAY (0 .LE. HH .LE. 23)
C                NN = MINUTES (0 .LE. NN .LE. 59)
C                SS = SECONDS (0 .LE. SS .LE. 59)
C              XXXX = TENTHS OF MILLISECONDS (0 .LE. XXXX .LE. 9999)
C
C     CALLING SEQUENCE:     CALL VIGSEC(V,S)
C        V IS THE INPUT VIG. DATE
C        S IS THE OUTPUT D.P. EQUIVALENT OF V IN SECONDS PAST OH, 1/1/50
C
C
C
```

```
      DOUBLE PRECISION S, V, V2, NYMD
C
C
      IF (V.LT.30001D0) RETURN
C
C
C
C       CALCULATE THE NUMBER OF FULL YEARS, MONTHS, AND DAYS ALREADY
C       ELAPSED SINCE OH, MARCH 1, -1 (UP TO MOST RECENT MIDNIGHT).
C       (THE '91' IS COMPLEMENT ARITHMETIC FOR ADDING 9 TO 'MM'.)
C
      NYMD = DINT(V-910001D0)
C
C       CONVERT 2ND WORD OF VIGESIMAL DATE AND ELAPSED DAYS TO
C       D.P. SECONDS
C
      V2 = (V-DINT(V))*1.D10
      S = DMOD(DINT(V2/1.D4),1.D2)
     X  + DMOD(DINT(V2/1.D6),1.D2)*60D0
     X  + DINT(V2/1.D8)*3.6D3
     X  + DMOD(NYMD,1.D2)*8.64D4
C
C
C       CALCULATE THE NUMBER OF FULL MONTHS ELAPSED SINCE
C       THE CURRENT OR MOST RECENT MARCH
C
      MTH = MOD(IDINT(DMOD(DINT(NYMD/1.D4),1.D2)),12)
C
C
C       CALCULATE NUMBER OF FULL 12-MONTH YEARS SINCE 3/1/-1
C
      IY = IDINT(NYMD/1.D6) + IDINT(DMOD(DINT(NYMD/1.D4),1.D2))/12
C
C
C       STARTING WITH MARCH AS MONTH 0 AND ENDING WITH THE FOLLOWING
C       FEBRUARY AS MONTH 11, THE CALCULATION OF THE NUMBER OF DAYS
C       PER MONTH REDUCES TO A SIMPLE FORMULA. THE FOLLOWING STATEMENT
C       DETERMINES THE NUMBER OF WHOLE DAYS ELAPSED SINCE 3/1/-1 AND
C       THEN SUBTRACTS THE 712163 DAYS BETWEEN THEN AND 1/1/50.  IT CONVERTS
C       THE RESULT TO SECONDS AND ADDS THE ACCUMULATED SECONDS ABOVE.
C
      S = DINT(DMOD(V2,1.D4))*1.D-4
     X  + S
     X  +(DFLOAT(IY)*365D0
     X  + DFLOAT((MTH+1+MTH/6+MTH/11)/2 * 31
     X  +        (MTH-MTH/6-MTH/11)/2 * 30
     X  +        IY/4
     X  -        IY/100
     X  +        IY/400)
     X  - 712163D0) * 8.64D4
C
      RETURN
      END
```

```
C     THIS PROGRAM IS A TEST CASE FOR READE.  IT CALLS READE
C     AT 3 SEPARATE POINTS 32 DAYS APART.
C
C       MODIFIED FOR D.G. NOVA 4 BY R. RICKLEFS          3/81
C       UNIVERSITY OF TX, MCDONALD OBS.
C
      DOUBLE PRECISION AU,RE,TPD,EMRAT,TABOUT,NUT
      DOUBLE PRECISION JED,TSEC
      DOUBLE PRECISION XAU(2),D,E
      INTEGER CAL(7)
C
      COMMON/CETB1/AU,RE,TPD,EMRAT
      COMMON/CETB2/ICW,ICENT,IREQ(13)
      COMMON/CETB4/TABOUT(6,12),NUT(4)
      COMMON/TSTCMN/XAU
C
      DATA XAU(1),XAU(2) /0.D0,1.D0/
      DATA IREQ /13*2/
C
      CALL OPEN(2,"XJPLEPH",2,IERR)
      IF (IERR.NE.1) STOP 1
      CALL OPEN(12,"TROP",2,IERR)
      IF (IERR.NE.1) STOP 2

      ICW=1
      TSEC=86400.D0*.25D0

C  DO FOR AU & KM
      DO 1 I=1,2
          JED=2441001.5D0
          AU=XAU(I)
          TPD=XAU(I)

C  DO FOR 3 JULIAN DATES 32 DAYS APART
          DO 2 J=1,3
              D=JED+TSEC/86400.D0
              E=(D-2433282.5D0)*86400.D0
              CALL SECCAL(E,CAL)
                  CAL(3)=CAL(3)+1900
              WRITE(12,101)D,E,(CAL(K),K=1,6),AU
  101         FORMAT('1 JD=',F12.3,F14.0,' SECS PAST 1950',I4,'/',I2,'/',I4,
     1        I3,2(':',I2),3X,'AU FAC =',F3.0)

C  DO FOR EACH BODY AS CENTER OF SYSTEM
              DO 3 K=1,11
                  ICENT=K
                  CALL READE(JED,TSEC,IERR)
                  IF(IERR.NE.0) GO TO 99

C  WRITE HEADER
                  WRITE(12,102)ICENT
```

```
 102             FORMAT('0',40X,'ICENT =',I3,/)
                 DO 4 K1=1,12
                    DO 5 K2=1,2
                       K3=3*K2-3
                       L1=K3+1
                       L2=K3+3

C  WRITE POSITION COMPONENTS ON ONE LINE & VELOCITY ON NEXT
                       WRITE(12,103) K1,K2,(TABOUT(KX,K1),KX=L1,L2)
 103                   FORMAT(2I3,3(1PD25.17))
   5                CONTINUE
   4             CONTINUE

C  WRITE NUTATION AND ITS DERIVATIVES
                 WRITE(12,104)(NUT(IK),IK=1,4)
 104             FORMAT('0 NUTATIONS:',2(/,2X,2(1PD25.17)))
   3          CONTINUE
              JED=JED+32.D0
   2       CONTINUE
   1    CONTINUE

        CALL FCLOS(2)
        CALL FCLOS(12)
        STOP

  99    WRITE(12,105)IERR
 105    FORMAT('0 ERROR RETURN FROM READE. ERROR FLAG=',I4)
        STOP
        END



        OVERLAY RDEPKG
        SUBROUTINE READE(JED,TSEC,IERR)
C
        DOUBLE PRECISION AU,RE,TPD,EMRAT,TABOUT,NUT,JED,TSEC
        DOUBLE PRECISION JD(2),BIVECT,TP,EM,EMR,PF,VF,BUF,PVSUN(6)
C
        INTEGER LIST(11)
C
        LOGICAL KM,BARY
C
        COMMON/CETB1/AU,RE,TPD,EMRAT
        COMMON/CETB2/ICW,ICENT,IREQ(13)
        COMMON/CETB4/TABOUT(6,12),NUT(4)
        COMMON/STCOMM/KM,BARY,PVSUN
        COMMON/CETB3/BUF(900,2)
        COMMON/CETB5/BIVECT(6,13)
C
C
C
C  KM OR AU?
        KM=AU.LE.0.D0
```

```
C  SET KM TO AU CONVERSIONS
      PF=1.D0
      TP=TPD
      IF(TP.EQ.0.D0) TP=86400.D0
      IF(.NOT.KM) PF=AU
      VF=PF/TP
      IF(KM) VF=VF*86400.D0

C  SET EARTH-MOON MASS RATIO TO INTERNAL CONSTANT IF NOT SET BY USER
      EM=EMRAT
      IF(EM.EQ.0.D0) EM=81.3007D0

C  IF EPHEMERIS TO BE RE-READ FROM BEGINNING, DO APPROPRIATE INITIALIZATION
      IF(ICW.EQ.2) GO TO 1
      ICW=2
      BUF(1,1)=0.D0
      BUF(2,1)=0.D0
      BUF(1,2)=0.D0
      BUF(2,2)=0.D0
    1 CONTINUE
      EMR=1.D0/(1.D0+EM)

C  VERIFY THAT CENTRAL-BODY NUMBER IS VALID
      IF(ICENT.GE.1.AND.ICENT.LE.11) GO TO 7
      IERR=4
      RETURN
    7 CONTINUE

C  CHECK TYPE OF INTERP REQUESTED (0=NONE,1=POSITION,2=POSITION&VELOCITY)
      DO 2 I=1,9
          IF(IREQ(I).GE.0.AND.IREQ(I).LE.2) GO TO 2
          IERR=3
          RETURN
    2     LIST(I)=IREQ(I)

C  MOON:
      LIST(10)=IREQ(11)

      IF(ICENT.EQ.10) GO TO 8
      LST=0

C  IF ANY BODY REQUIRES POSITION OR VELOCITY, CENTRAL BODY MUST ALSO
      DO 3 I=1,10
    3     LST=MAX0(LST,LIST(I))
      IC=ICENT-ICENT/11
      LIST(IC)=MAX0(LST,IREQ(10))
    8 CONTINUE

C  EARTH:
      LIST(3)=MAX0(LIST(3),LIST(10))

C  EARTH-MOON BARYCENTER:
      LIST(3)=MAX0(LIST(3),IREQ(12))
```

```
C   SUN:
       LIST(10)=LIST(3)

C   NUTATION:
       LIST(11)=IREQ(13)
C

C   SET UP JULIAN DATES & INTERPOLATE
       JD(1)=DINT(JED-.5D0)+.5D0
       JD(2)=(JED-JD(1))+TSEC/86400.D0
       CALL XSTATE(JD,LIST,BIVECT,NUT,IERR)
       IF(IERR.NE.0) RETURN
       DO 4 I=1,6
C   MOON:
             BIVECT(I,11)=BIVECT(I,3)+EM*EMR*BIVECT(I,10)
C   NUTATION:
             BIVECT(I,12)=BIVECT(I,3)
C   EARTH:
             BIVECT(I,3)=BIVECT(I,3)-EMR*BIVECT(I,10)
C   SUN:
             BIVECT(I,10)=0.D0
     4 CONTINUE
C
C   DO FOR EACH BODY & NUTATION
       DO 5 I=1,12
C   DO FOR POSITION/VELOCITY ELEMENT
       DO 5 J=1,3
C   SUBTRACT CENTRAL-BODY CONTRIBUTION
             IF(IREQ(I).EQ.0) GO TO 5
             TABOUT(J,I)=(BIVECT(J,I)-BIVECT(J,ICENT))*PF
             IF(IREQ(I).EQ.1) GO TO 5
             TABOUT(J+3,I)=(BIVECT(J+3,I)-BIVECT(J+3,ICENT))*VF
     5 CONTINUE
C
       RETURN
C
       END




       SUBROUTINE XSTATE(JD,LIST,PV,NUT,IFL)
       DOUBLE PRECISION JD(2),PV(3,2,10),NUT(4),SS(6),BUF(900,2),T(2)
       DOUBLE PRECISION TFAC,BF
       DOUBLE PRECISION AUFAC,PVSUN(3,2)
       INTEGER LIST(11),LOC(3,12),NRL
       LOGICAL FLAG,KM,BARY
C
       COMMON/STCOMM/KM,BARY,PVSUN
       COMMON/SECTOR/IH(5),NSEC,NRS
       COMMON/CETB3/BUF
       COMMON/XSTCMN/TFAC,AUFAC,FLAG,SS,BF,LOC,NRL
C
C   DATA FOR LOCAL VARIABLES. FLAG MUST BE INITIALIZED TO .FALSE.
```

```
C  IN THE CALLING PROGRAM TO AVOID ITS RESETTING TO .FALSE.
C  WHEN READE IS IN AN OVERLAY.
      DATA TFAC/1.D0/,AUFAC/1.D0/
      DATA KM/.FALSE./,BARY/.FALSE./
      DATA NRL/0/
C
      IF(FLAG) GO TO 1
      CALL LH(2,1030)
      READ BINARY(2)N,(BUF(K,1),K=1,N)
      DO 180 I=1,6
180   SS(I)=BUF(I,1)
      READ BINARY(2)N,(BUF(K,1),K=1,N)
      CALL LH(2,-1041)
      DO 8 I=1,4
    8 READ BINARY(2)N,(BUF(K,1),K=1,N)
      READ BINARY(2)N,(BUF(K,1),K=1,N)
      BF=BUF(1,1)
      READ BINARY(2)N,(BUF(K,1),K=1,N)
      CALL LH(2,-1050)
      READ BINARY(2)N,(BUF(K,1),K=1,N)
      READ BINARY(2)N,(BUF(K,1),K=1,N)
      DO 190 II=1,12
      KK=3*(II-1)
      DO 190 JJ=1,3
190   LOC(JJ,II)=IDINT(BUF(KK+JJ))
      READ BINARY(2)N,(BUF(K,1),K=1,N)
      CALL LH(2,-1070)
      FLAG=.TRUE.

    1 CONTINUE
      IFL=0
      T(2)=SS(3)
      AUFAC=1.D0
      TFAC=1.D0
      IF(KM) T(2)=SS(3)*86400.D0
      IF(KM) TFAC=86400.D0
      IF(.NOT.KM) AUFAC=1.D0/BF
      IF(JD(1).LT.SS(1)) GO TO 99
      IF(JD(1)+JD(2).GT.SS(2)) GO TO 98
   13 IF((JD(1).GE.BUF(1,1)).AND.(JD(1)+JD(2).LE.BUF(2,1))) GO TO 11
      IF((JD(1).GE.BUF(1,2)).AND.(JD(1)+JD(2).LE.BUF(2,2))) GO TO 14
      NR=IDINT((JD(1)-SS(1))/SS(4))+1
C     ********************************************************
      IF(JD(1)+JD(2).GE.SS(2)-SS(3)) NR=NR-1
      IF(JD(1)+JD(2).EQ.SS(2)) NR=NR-1
      NSK=NR-NRL-1
      IF(NSK)7,5,12
   12 DO 9 I=1,NSK
    9 READ BINARY(2,END=98)N,(BUF(K,1),K=1,N)
      GO TO 5
    7 CALL LH(2,1070)
      NSK=NR-1
      GO TO 12
C   7 NSK=-NSK
```

```
C       DO 10 I=1,NSK
C    10 BACKSPACE 2
C       ****************************************************
C       NS=NSEC+NR
C       CALL FSEEK(2,NS)
C         TYPE " CALLING FSEEK ",NR,NSEC
      5 READ BINARY(2,END=98)N,(BUF(I,1),I=1,N)
        READ BINARY(2,END=15)N,(BUF(I,2),I=1,N)
X         TYPE "READ: ",JD, BUF(1,1), BUF(1,2), BUF(2,1), BUF(2,2)
15      NRL=NR+1
        GO TO 13
     11 NB=1
        GO TO 2
     14 NB=2
      2 T(1)=((JD(1)-BUF(1,NB))+JD(2))*TFAC/T(2)
        LL=LOC(1,11)
        CALL XINTRP(BUF(LL,NB),T,LOC(2,11),3,LOC(3,11),2,PVSUN)
      6 DO 3 I=1,10
        IF(LIST(I).EQ.0) GO TO 3
        LL=LOC(1,I)
        CALL XINTRP(BUF(LL,NB),T,LOC(2,I),3,LOC(3,I),LIST(I),PV(1,1,I))
        NM=LIST(I)*3
        DO 4 J=1,NM
        IF(I.LE.9.AND.(.NOT.BARY))PV(J,1,I)=(PV(J,1,I)-PVSUN(J,1))*AUFAC
        IF(I.LE.9.AND.BARY)PV(J,1,I)=PV(J,1,I)*AUFAC
        IF(I.EQ.10) PV(J,1,I)=PV(J,1,I)*AUFAC
      4 CONTINUE
      3 CONTINUE
        IF(LIST(11).EQ.0) RETURN
        LL=LOC(1,12)
        CALL XINTRP(BUF(LL,NB),T,LOC(2,12),2,LOC(3,12),LIST(11),NUT)
        RETURN
C
     99 CONTINUE
        IFL=1
        RETURN
     98 IFL=2
        RETURN
C
        END



C***********************************************
C
        SUBROUTINE XINTRP(BUF,T,NCF,NCM,NA,FL,PVA)
C
C***********************************************
C
C
C       THIS SUBROUTINE DIFFERENTIATES AND INTERPOLATES A
C       SET OF CHEBYSHEV COEFFICIENTS TO GIVE POSITION, VELOCITY,
C       AND ACCELERATION.
C
```

```
C     CALLING SEQUENCE PARAMETERS --
C
C       INPUT --
C
C         BUF   1ST LOCATION OF ARRAY OF DP CHEBYSHEV COEFFICIENTS OF POSITION
C
C           T   T(1) IS DP FRACTIONAL TIME IN INTERVAL COVERED BY
C               COEFFICIENTS AT WHICH INTERPOLATION IS WANTED
C               (0 .LE. T(1) .LE. 1).  T(2) IS DP LENGTH OF WHOLE
C               INTERVAL IN INPUT TIME UNITS.
C
C         NCF   NO. OF COEFFICIENTS PER COMPONENT
C
C         NCM   NO. OF COMPONENTS PER SET OF COEFFICIENTS
C
C          NA   NO. OF SETS OF COEFFICIENTS IN FULL ARRAY
C               (I.E., NO. OF SUB-INTERVALS IN FULL INTERVAL)
C
C          FL   INTEGER FLAG  =1 FOR POSITIONS ONLY
C                             =2 FOR P,V ONLY
C                             =3 FOR P,V,A
C
C
C       OUTPUT --
C
C         PVA   INTERPOLATED QUANTITIES REQUESTED.  DIMENSION
C               EXPECTED IS PVA(NCM,FL), DP.
C
C
      DOUBLE PRECISION BUF(NCF,NCM,15)
      DOUBLE PRECISION T(2)
      DOUBLE PRECISION PVA(NCM,FL)
      DOUBLE PRECISION BMA,BMA2
      DOUBLE PRECISION TEMP
      DOUBLE PRECISION TC
      DOUBLE PRECISION TCL
      DOUBLE PRECISION PC(18),VC(18),AC(18)
      DOUBLE PRECISION TWOT

      INTEGER FL

      COMMON /XINCMN/ PC,VC,AC

      EQUIVALENCE (TCL,PC(2))

      DATA PC(1),PC(2)/1.D0,2.D0/
      DATA VC(2)/1.D0/
      DATA AC(3)/4.D0/
C
C       CALCULATE LOCATION OF CORRECT COMPONENT ARRAYS
C       AND SCALED CHEBYSHEV TIME IN THAT INTERVAL
C
X       TYPE " ENTERING XINTRP"
      TEMP=T(1)*DFLOAT(NA)
```

```
      L=IDINT(TEMP-DINT(T(1)))+1
      TC=2.D0*((TEMP-DINT(TEMP))+DINT(T(1)))-1.D0
C
C       CHECK TO SEE WHETHER CHEBYSHEV TIME HAS CHANGED,
C       AND COMPUTE NEW POLYNOMIAL VALUES IF IT HAS
C
      IF(TC.EQ.TCL) GO TO 1
      NP=2
      NV=3
      NAC=4
      TCL=TC
      TWOT=TC+TC
    1 CONTINUE
      IF(NP.GE.NCF) GO TO 2
      M=NP+1
      NP=NCF
      DO 3 I=M,NCF
    3 PC(I)=TWOT*PC(I-1)-PC(I-2)
    2 CONTINUE
C
C       INTERPOLATE TO GET POSITION FOR EACH COMPONENT
C
X       TYPE "INTERP",NCM,NCF,FL
      DO 4 I=1,NCM
      PVA(I,1)=0.D0
      DO 4 J=1,NCF
      JJ=NCF-J+1
      PVA(I,1)=PVA(I,1)+PC(JJ)*BUF(JJ,I,L)
    4 CONTINUE
      IF(FL.LE.1) RETURN
C
C       CHECK VELOCITY POLYNOMIAL VALUES, AND GENERATE
C       NEW ONES IF REQUIRED
C
C
X       TYPE "CHECK VELOCITY"
      BMA=2.D0*DFLOAT(NA)/T(2)
      VC(3)=TWOT+TWOT
      IF(NV.GE.NCF) GO TO 5
      M=NV+1
      NV=NCF
      DO 6 I=M,NCF
    6 VC(I)=TWOT*VC(I-1)+PC(I-1)+PC(I-1)-VC(I-2)
C
C       EVALUATE VELOCITY FOR EACH COMPONENT
C
    5 CONTINUE
X       TYPE "EVAL VELOC (LABEL 5)"
      DO 7 I=1,NCM
      PVA(I,2)=0.D0
      DO 11 J=2,NCF
      JJ=NCF-J+2
X       TYPE I,J,JJ,BMA,NCM,NCF
      PVA(I,2)=PVA(I,2)+VC(JJ)*BUF(JJ,I,L)
```

```
   11 CONTINUE
      PVA(I,2)=PVA(I,2)*BMA
    7 CONTINUE
      IF(FL.EQ.2) RETURN
C
C        CHECK ACCELERATION POLYNOMIAL VALUES, AND
C        RE-DO IF NECESSARY
C
X        TYPE "CHECK ACCEL"
      BMA2=BMA*BMA
      AC(4)=24.D0*PC(2)
      IF(NAC.GE.NCF) GO TO 8
      M=NAC+1
      NAC=NCF
      DO 9 I=M,NCF
    9 AC(I)=TWOT*AC(I-1)+4.D0*VC(I-1)-AC(I-2)
    8 CONTINUE
C
C        GET ACCELERATION FOR EACH COMPONENT
C
X        TYPE "GET ACCEL"
      DO 10 I=1,NCM
      PVA(I,3)=0.D0
      DO 12 J=3,NCF
      JJ=NCF-J+3
      PVA(I,3)=PVA(I,3)+AC(JJ)*BUF(JJ,I,L)
   12 CONTINUE
      PVA(I,3)=PVA(I,3)*BMA2
   10 CONTINUE
C
X        TYPE " XINTRP RETURN"
      RETURN
C
      END




C****************************************
C
      SUBROUTINE XCONST(NAM,VAL,SS,N)
C
C****************************************
C
C     THIS SUBROUTINE READS THE EXPORT JPL PLANETARY EPHEMERIS
C     AND OBTAINS THE NAMES AND VALUES OF THE ASTRONOMICAL CONSTANTS
C     THAT WERE USED TO GENERATE THE EPHEMERIS.
C
C     CALLING SEQUENCE PARAMETERS:
C
C         NAM   INTEGER ARRAY THE N 6-CHARACTER NAMES OF THE CONSTANTS
C
C         VAL   DOUBLE PRECISION ARRAY THAT WILL CONTAIN THE N VALUES
C
```

```
C         SS   6-WORD DOUBLE-PRECISION ARRAY THAT WILL BE FILLED
C              WITH THE CONTENTS OF GROUP 1030 (START AND STOP EPOCHS, ETC.)
C
C          N   INTEGER NUMBER OF VALUES READ FROM FILE
C
C
C     *** NOTE ***   BECAUSE OF TAPE POSITIONING, THIS ROUTINE, IF IT
C                    IS USED AT ALL, MUST NOT BE CALLED AFTER READE
C                    OR STATE HAVE BEEN REFERENCED IN THE SAME PROGRAM.
C
C
      INTEGER NAM(600)
      INTEGER H(5)
C
      DOUBLE PRECISION VAL(200),SS(6)
C
      COMMON/SECTOR/H
C
C
C
C       IF OTHER EPH ROUTINES HAVE BEEN CALLED, DON'T DO ANYTHING
C
      IF(H(4).NE.0) RETURN
C
C       OTHERWISE, FIND LIMITS GROUP AND COPY INTO SS
C
      CALL LH(2,1030)
      READ BINARY(2)NDUM,(SS(K),K=1,NDUM)
C
C       FIND RECORD 4 IN GROUP 1040 (NAMES OF CONSTANTS)
C
      DO 1 I=1,5
      READ BINARY(2)NDUM,(NAM(K),K=1,NDUM)
    1 CONTINUE
      READ BINARY(2)NDUM,(NAM(K),K=1,NDUM)
C
C       FIND AND READ VALUES OF CONSTANTS
C
      DO 2 I=1,6
      READ BINARY(2)NDUM,(VAL(K),K=1,NDUM)
    2 CONTINUE
      READ BINARY(2)N,(VAL(K),K=1,N)
C
      REWIND 2
      RETURN
      END


C   THIS PROGRAM PRINTS THE CONTENTS OF A TYPE-66 TAPE.
C
C     INPUTS
C         TAPE - ONE TYPE 66 EPHEMERIS TAPE - ON UNIT -NUNIT-
C         CARDS - CIN NAMELIST
```

```
C           CONTENTS
C              IOPT     1-5         OPTION REQUESTED
C              KEY                   KEY1 OF SPECIAL GROUP ON TAPE
C              NREC                 COUNT OF RECORDS TO BE DUMPED FOR
C                                   OPTIONS 2 AND 4
C              NUNT                FORTRAN UNIT NUMBER OF INPUT TAPE
C              JD                   STARTING JD IOPT=4 OPTION
C
C
C     OUTPUT
C         PRINTED- A COMPLETE LIST OF HEADERS AND TRAILERS FROM THE TAPE
C                  AS WELL AS LIST OF ALL DATA WITH THE DELETIONS AS
C                  SPECIFIED BY THE IOPT FLAG.
C
C
C       MODIFIED FOR D.G. NOVA BY R.L. RICKLEFS         3/81
C       UNIVERSITY OF TX, MCDONALD OBS.
C
      DOUBLE PRECISION JD(10),TT
      DOUBLE PRECISION DBUF(2000)
C
      INTEGER NW(4)
      INTEGER IBUF(6000)
      INTEGER NPOS,NWD(2),IOPT
      INTEGER STAR
      INTEGER T
      INTEGER ATYPE(5)
      INTEGER FILNAM(12)
C
      LOGICAL EOG
C
      INTEGER SBUF(6000)
C
      COMMON ISNGL,NWORDS,KTYPE
      COMMON /DDDCMN/STAR,NW,ATYPE,DBUF, KEY,NPOS,NWD

      EQUIVALENCE (DBUF(1),IBUF(1)), (DBUF(1),SBUF(1))
C
      DATA STAR /"**"/
      DATA NW /7,4,9,54/
      DATA ATYPE /1HR,1HD,1HI,1HB,1HE/
      DATA KEY /1070/
      DATA NPOS /1/
      DATA NWD /1,10000/
C
 2000 FORMAT (1H1,8X,27HGENERAL EPHEMERIS TAPE DUMP,13X,15A2)
 2010 FORMAT (1H0,8X,30HDUMPING ALL OF ALL KEY1 GROUPS)
 2020 FORMAT (1H0,8X,49HDUMPING ALL OF ALL KEY1 GROUPS BUT ONLY THE FIRST,
     *       I3,9H AND LAST,I3,8H OF KEY1,I3,6H GROUP)
 2030 FORMAT (1H0,8X,47HDUMPING ALL OF ALL KEY1 GROUPS EXCEPT ONE LINE ,
     1       20HONLY FROM EACH KEY1 ,I4, 6H BLOCK)
 2040 FORMAT (28H0T TREC GR GREC  NWDS  DATA ,49A2,5H DATA/)
 2050 FORMAT (1H ,A1,I5,I3,I5,I6,2X,9I10)
 2060 FORMAT (1H ,A1,I5,I3,I5,I6,2X,54A2)
```

```
 2070 FORMAT (1H ,A1,I5,I3,I5,I6,2X,1P7E15.7)
 2080 FORMAT (1H ,A1,I5,I3,I5,I6,2X,1P4D25.17)
 2090 FORMAT (1H0,65A2////50X,17HNORMAL END OF JOB////,1H0,65A2)
 2100 FORMAT (1H0,8X, "DUMPING ALL OF ALL KEY1 GROUPS EXCEPT FOR GROUP "
     *"KEY1 OF",I2," WHERE WE ARE DUMPING",I4/" RECORDS FROM FIRST RECORD "
     *"WITH JD.GE.              ",7P7D24.7)
 2110 FORMAT ('0',8X,'DUMPING FIRST GROUP WITH KEY1=',I10)
C
C
C
C     ********************** BEGIN EXECUTION HERE ********************
C
C
      NUNIT=2
      WRITE(10,1000)
1000  FORMAT(" ENTER EPHEMERIS FILE NAME: ",Z)
      READ(11,1005) FILNAM
1005  FORMAT(12A2)
      CALL OPEN(NUNIT,FILNAM,2,IERR)
      IF (IERR.EQ.1) GO TO 3
        TYPE "OPEN ERROR: ",IERR
        STOP
   3  NPOS=1
      ACCEPT "OPTION: ",IOPT
      IF (IOPT.NE.2.AND.IOPT.NE.4) GO TO 5
      ACCEPT "KEY1: ",KEY
      ACCEPT "NUMBER OF RECORDS TO BE DUMPED: ",NREC
      IF (IOPT.EQ.4) ACCEPT "STARTING JULIAN DATES (10 OF THEM): ",JD
5     ACCEPT "POSITION OF TIME TAG: (1) ",NPOS
      ACCEPT "START DUMP W/ WHICH WORD: (1) ",NWD(1)
      ACCEPT "END DUMP W/ WHICH WORD: (10000) ",NWD(2)
      REWIND NUNIT
C
C
      IF (IOPT.EQ.1) WRITE (12,2010)
C
      IF (IOPT.EQ.2) WRITE (12,2020) NREC,NREC,KEY
C
      IF (IOPT.EQ.3) WRITE (12,2030) KEY
C
      IF (IOPT.EQ.4) WRITE (12,2100) KEY,NREC,JD
C
      IF (IOPT.EQ.5) WRITE (12,2110) KEY
C
C         PRINT OUT THE STARS AND INITIALIZE COUNTS
C
      WRITE (12,2040) (STAR,I=1,49)
      ITREC=0
      IGR=0
C
C     READ A HEADER
C
   10 READ BINARY(NUNIT) NMAX,KTYPE,ISNGL,KEY1,KEY2
C
```

```
C     STEP COUNTS AND LIST THE HEADER
C
      IFRONT=0
      ITREC=ITREC+1
      IGR=IGR+1
      IGREC=0
      NWORDS=0
      T=ATYPE(KTYPE)
      WRITE (12,2050) T,ITREC,IGR,IGREC,NWORDS,NMAX,KTYPE,ISNGL,KEY1,KEY2
C
C     IS THIS AN END OF FILE HEADER (Q)
C
      IF (KTYPE.EQ.5) GO TO 90
C
C     NO, STEP RECORD COUNTS
C
C
C         IF OPTION IS TWO, ARE WE AT RIGHT KEY1 GROUP, IF SO DO WE
C              NEED TO PRINT THIS RECORD OR START SPINNING DOWN TAPE.
C
   20 IF (IOPT.EQ.5.AND.KEY.NE.KEY1) GO TO 30
      IF (IOPT.NE.2) GO TO 50
C
      IF (KEY1.NE.KEY) GO TO 50
C
      IF (IGREC.LE.NREC) GO TO 50
C
C     READ ON THRU THE END OF THIS KEY1 GROUP.
C
   30 IF(KTYPE.EQ.2) READ BINARY(NUNIT)NWORDS,DBUF(1)
      IF(KTYPE.NE.2) READ BINARY(NUNIT)NWORDS,IBUF(1)
      IGREC=IGREC+1
      ITREC=ITREC+1
      IF(.NOT.EOG(1)) GO TO 30
C
C     BACK UP TO ALLOW DUMP OF NREC RECORDS FROM END OF GROUP
C
      GO TO 10
      IF (IOPT.GE.4) GO TO 10
      L= IGREC-NREC+1
      M= NREC+1
      N=MINO(L,M)
      IGREC=IGREC-N
      ITREC=ITREC-N
      DO 40 I=1,N
   40 BACKSPACE NUNIT
      IOPT=1
C
C     READ A DATA RECORD
C
   50 IF (KTYPE.EQ.2) READ BINARY(NUNIT) NWORDS,(DBUF(I),I=1,NWORDS)
      IF (KTYPE.NE.2) READ BINARY(NUNIT) NWORDS,(SBUF(I),I=1,NWORDS)
      ITREC=ITREC+1
      IGREC=IGREC+1
```

```
      TT=DBUF(NPOS)
      IF(KTYPE.NE.2)TT=DBLE(FLOAT(SBUF(NPOS)))
C
C     DECIDE WHETHER TO PRINT JUST ONE LINE FROM THIS RECORD.
C
   60 CONTINUE
      IF(NWD(1).NE.0) GO TO 61
      N1=1
      N2=NWORDS
      GO TO 62
   61 IF(NWD(1).GT.0) GO TO 63
      N2=NWORDS
      N1=MAX0(NWORDS+NWD(1)+1,1)
      GO TO 62
   63 IF(NWD(2).NE.0) GO TO 64
      N2=MIN0(NWD(1),NWORDS)
      N1=1
      GO TO 62
   64 N1=MIN0(NWD(1),NWORDS)
      N2=MIN0(NWD(2),NWORDS)
   62 NWRITE=MIN0(NW(KTYPE),N2-N1+1)
      IF (KEY1.NE.KEY) GO TO 70
      IF (IOPT.EQ.4) GO TO 80
      IF (IOPT.EQ.2.AND.IGREC.GT.NREC) GO TO 30
      IF(IOPT.EQ.5.AND .IGREC.GT.NREC.AND.NREC.GT.0) GO TO 90
C
C     LIST A DATA RECORD.
C
   70 L1=N1
      L2=N1+NWRITE-1
      IF (KTYPE.EQ.1) WRITE (12,2070) T,ITREC,IGR,IGREC,NWORDS,
     *                                (SBUF(I),I=L1,L2)
      IF (KTYPE.EQ.2) WRITE (12,2080) T,ITREC,IGR,IGREC,NWORDS,
     *                                (DBUF(I),I=L1,L2)
      IF (KTYPE.EQ.3) WRITE (12,2050) T,ITREC,IGR,IGREC,NWORDS,
     *                                (SBUF(I),I=L1,L2)
      IF (KTYPE.EQ.4) WRITE (12,2060) T,ITREC,IGR,IGREC,NWORDS,
     *                                (SBUF(I),I=L1,L2)
C
      N1=N1+NWRITE
      IF(IOPT.EQ.3.OR.N1.GT.N2) GO TO 75
      DO 76 K=N1,N2,NWRITE
      KW=MIN0(NWRITE-1,N2-K)
      L1=K
      L2=K+KW
      IF(KTYPE.EQ.1) WRITE(12,101)K,(SBUF(I),I=L1,L2)
  101 FORMAT(I21,'.',1X,7(1PE15.7))
      IF(KTYPE.EQ.2) WRITE(12,102)K,(DBUF(I),I=L1,L2)
  102 FORMAT(I21,'.',1X,4(1PD25.17))
      IF(KTYPE.EQ.3) WRITE(12,103)K,(SBUF(I),I=L1,L2)
  103 FORMAT(I21,'.',1X,9I10)
      IF(KTYPE.EQ.4) WRITE(12,104)K,(SBUF(I),I=L1,L2)
  104 FORMAT(I21,'.',1X,54A2)
   76 CONTINUE
```

```
C
C     IS THIS THE LAST DATA RECORD OF THE GROUP
C
   75 IF(EOG(1).AND.IOPT.EQ.5) GO TO 90
      IF(EOG(1)) GO TO 10
C
C     NO, GO READ ANOTHER DATA RECORD
C
      GO TO 50
C
C **** HERE  ON  IOPT=4 ****
C
80    IF(JD(2).NE.0.D0)GO  TO  81
      IF(TT.LT.JD(1))GO  TO  75
      IFRONT=IFRONT+1
      IF(IFRONT.LE.NREC)GO  TO  70
      GO  TO  90  ;FINISHED  WITH  TAPE
81    DO  811  K=1,5
      IF(JD(2*K-1).EQ.0.D0)GO  TO  75
      X=(TT-JD(2*K-1))*(JD(2*K)-TT)
      IF(X.GE.0.)GO  TO  70
811   CONTINUE
      GO  TO  75
C
C     WRITE END OF JOB MESSAGE.
C
   90 WRITE (12,2090) (STAR,I=1,130)
C
      STOP
      END
```

```
JD= 2441001.750    666943200. SECS PAST 1950   2/19/1971  6: 0: 0   AU FAC = 0.

                                  ICENT =  1

1  1  0.00000000000000000D  0  0.00000000000000000D  0  0.00000000000000000D  0
1  2  0.00000000000000000D  0  0.00000000000000000D  0  0.00000000000000000D  0
2  1 -1.15841653845160410D  8  2.51377345681265740D  6  1.39764531574209220D  7
2  2 -1.85668327698334960D  1 -4.55985271532626940D  1 -1.95042326863801700D  1
3  1 -1.50351954250855510D  8  1.23070576998692220D  8  6.12583073787269440D  7
3  2 -5.16081084967129120D  1 -4.19954671049660640D  1 -1.63736275160668360D  1
4  1 -2.03594504214149160D  8 -8.42087254995767880D  7 -2.72925205412061880D  7
4  2 -1.96060970246275160D  1 -3.32123296179223430D  1 -1.33842497575554090D  1
5  1 -4.95873472528967680D  8 -5.49328607710719700D  8 -2.15954755152517630D  8
5  2 -2.56183242574628980D  1 -2.47209031502363490D  1 -9.10801053288185040D  0
6  1  8.08383574218654630D  8  1.06698285095959220D  9  4.14228285811124860D  8
6  2 -4.42288037171842530D  1 -1.30240479853370660D  1 -3.56365473242831430D  0
7  1 -2.71841365066665730D  9 -4.25641995722163200D  8 -1.40764168373901400D  8
7  2 -3.48402332240523730D  1 -2.47379578391935060D  1 -8.93624404410176700D  0
8  1 -2.23124686181931040D  9 -3.63107322287867630D  9 -1.42265057028855480D  9
8  2 -3.13354169479585960D  1 -2.07013666992783460D  1 -7.20128599288886820D  0
9  1 -4.54510677664667700D  9 -3.08297081141257940D  8  1.28731420955798450D  9
```

9	2	-3.52644116433816490D	1	-2.37583961112890840D	1	-8.06300569155179360D	0
10	1	-2.27388009771042430D	7	5.[illegible]5068932979396770D	7	3.15262937042456120D	7
10	2	-3.60564055822395880D	1	-1.83188932560449050D	1	-6.10643026840661900D	0
11	1	-1.50494265534759560D	8	1.22755219458875790D	8	6.10839193135507890D	7
11	2	-5.06471875940273400D	1	-4.22926238600450010D	1	-1.64696505141363940D	1
12	1	-1.50353683413352360D	8	1.23066745226425200D	8	6.12561884652161710D	7
12	2	-5.15964327648821310D	1	-4.19990777274161640D	1	-1.63747942497406630D	1

NUTATIONS:

5.37071582249820450D -5 3.66039299624601530D -5
4.02632824244921960D-12 -9.15520080485547150D-13

ICENT = 2

1	1	1.15841653845160410D	8	-2.51377345681265740D	6	-1.39764531574209220D	7
1	2	1.85668327698334960D	1	4.55985271532626940D	1	1.95042326863801700D	1
2	1	0.00000000000000000D	0	0.00000000000000000D	0	0.00000000000000000D	0
2	2	0.00000000000000000D	0	0.00000000000000000D	0	0.00000000000000000D	0
3	1	-3.45103004056951030D	7	1.20556803541879560D	8	4.72818542213060220D	7
3	2	-3.30412757268794120D	1	3.60306004829662970D	0	3.13060517031333400D	0
4	1	-8.77528503689887520D	7	-8.67224989563894460D	7	-4.12689736986271110D	7
4	2	-1.03926425479401960D	0	1.23861975353403490D	1	6.11998292882476540D	0
5	1	-3.80031818683807310D	8	-5.51842381167532380D	8	-2.29931208309938550D	8
5	2	-7.05149148762940520D	0	2.08776240030263410D	1	1.03962221534983240D	1
6	1	9.24225228063814990D	8	1.06446907750277950D	9	4.00251832653703930D	8
6	2	-2.56619709473507530D	1	3.25744791679256270D	1	1.59405779539518600D	1
7	1	-2.60257199682149690D	9	-4.28155769178975880D	8	-1.54740621531322330D	8
7	2	-1.62734004542188730D	1	2.08605693140691810D	1	1.05679886422784070D	1
8	1	-2.11540520797415010D	9	-3.63358699633548900D	9	-1.43662702344597570D	9
8	2	-1.27685841781250990D	1	2.48971604539843410D	1	1.23029466934913060D	1
9	1	-4.42926512280151650D	9	-3.10810854598070620D	8	1.27333775640056360D	9
9	2	-1.66975788735481490D	1	2.18401310419736030D	1	1.14412269948283810D	1
10	1	9.31028528680561630D	7	5.19931198411270190D	7	1.75498405468246900D	7
10	2	-1.74895728124060880D	1	2.72796338972177850D	1	1.33978024179735550D	1
11	1	-3.46526116895991530D	7	1.20241446002063130D	8	4.71074661561298670D	7
11	2	-3.20803548241938400D	1	3.30590329321769240D	0	3.03458217224377470D	0
12	1	-3.45120295681919490D	7	1.20552971769612540D	8	4.72797353077952490D	7
12	2	-3.30295999950486310D	1	3.59944942584652930D	0	3.12943843663950650D	0

NUTATIONS:

5.37071582249820450D -5 3.66039299624601530D -5
4.02632824244921960D-12 -9.15520080485547150D-13

On-Site Integration of Starlette in a Taylored Field

by

E. Vermaat
Working Group for Satellite Geodesy
Kootwijk
Delft University of Technology
Delft, The Netherlands.

ABSTRACT

The periodical transmission of updated orbital parameters by a prediction center to mobile laser ranging stations, can be limited to very small data sets if there is an on-site capacity to reproduce an approximating orbit, not deviating from the original orbit beyond the prediction accuracy. To meet this requirement the STARLETTE orbit is numerically integrated in a tailored force field, especially designed to allow for integration of the equations of motion in a small computer. Deviations from the reference orbit can be kept within typically 50 to 75 m for 7-day arcs with a gravity field comprising about 20 selected terms, together with auxiliary expressions for the effects of lunar gravity and air drag.

1 Introduction

The reproduction of an orbit of STARLETTE from a given set of osculating elements (state vector) requires numerical integration of the equations of motion, to be derived from a dynamical model featuring the significant forces acting on the satellite. On-site this would require an unrealistically large computer capacity. For several SLR stations it is therefore common practice to truncate the dynamical model to acceptable sizes, thus obtaining a reproduced orbit which is tangential to the originally predicted orbit at the initial epoch. The reproduced orbit will diverge from the original one. The divergence occurring, being a function of the degree of simplification, can be controlled by limiting the arc length.

An alternative solution, which might allow for a considerably greater length of arc, would be to re-estimate the initial state vector, fitting the reproducing arc to the originally predicted orbit over a certain arc length, in an adjustment, utilizing a truncated dynamical model. The actual reproduction then proceeds as described above, but now starting from the re-estimated state vector, which could be considered to be a 'mean' state vector typical to the truncated model utilized. This model will have to be designed carefully, optimizing permissible arc length and prediction accuracy versus simplicity. It can be expected that the model will have to be tailored specifically to the characteristics of the orbit of each satellite involved.

This approach has been investigated to some extent, mainly for the orbit of STARLETTE and the results indicate that with relatively simple models, arc lengths can be covered of the order of the valid length of the predicted orbit itself.

2 Earth Gravity

As a first step it was attempted to locate those spherical harmonics coefficients of the earth's gravity field which significantly would contribute to the accuracy of the reproduced orbit in the approximation to the original orbit. Therefore a reference orbit of STARLETTE was generated, exclusively utilizing the complete GEM9 gravity field. From a 3-day arc (the 3-day arc length was more or less hand-picked, since by that time there was no clear insight into the permissible arc length to which these investigations would converge) out of this orbit a gravity field model up to degree and order 14 was estimated. To test the significance of the coefficients solved for, each estimated value was divided by its formal standard deviation. These 'significance parameters' as far as exceeding the value of 1.0 are displayed in figure 1.
This resulted in 34 individual C- or S-coefficients, pertaining to 29 different terms. With this tailored gravity model comprising the estimated values of the 'significant' terms, a 6-day arc out of the reference orbit was approximated, solving only for the initial state vector. From this re-estimated state vector a new 6-day arc was reproduced utilizing the same tailored model. The deviations of the reproduced orbit with respect to the reference orbit are displayed in figure 2 in along-track,radial and across-track components.

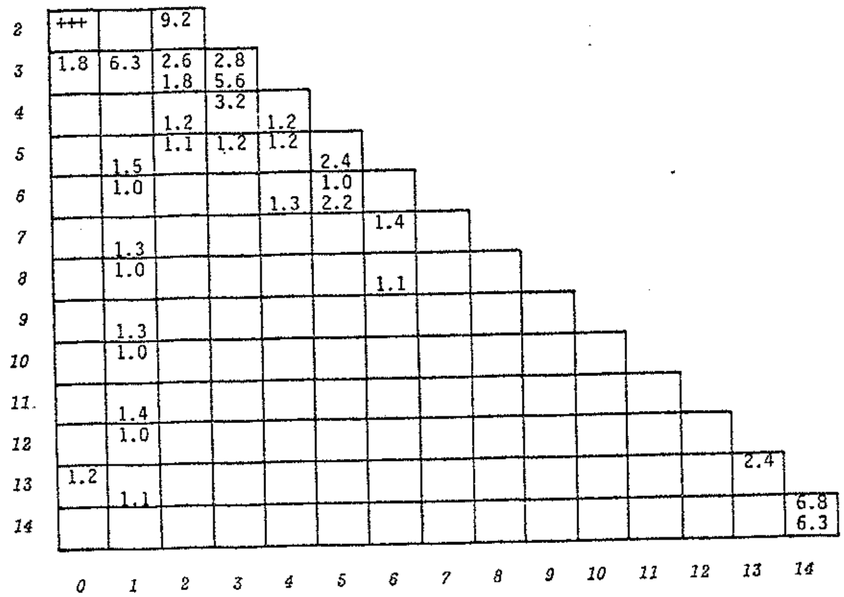
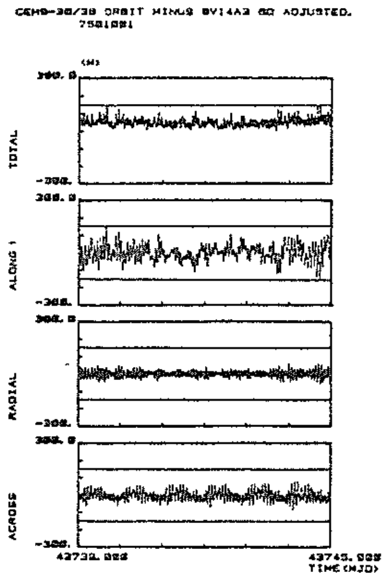


Figure 1. *Values of significance parameters from a gravity field solution up to degree and order 14 utilising a 3-day STARLETTE arc. The significance parameter is the estimated value devided by its standard error. For each term the upper value pertains to the C-coefficient and the lower value to the S-coefficient. Only the values greater than 1.0 are displayed.*

A dominant feature is the short-periodic oscillation with a typical wave length of one revolution. Further attempts have been mainly focussed on limiting the amplitude of these oscillations. After various attempts it was concluded that the 'significance parameters' were not very informative, probably mainly due to the neglected correlation. Therefore a more systematic approach was followed starting from all coefficients up to degree and order 4. Highlights of these attempts are depicted in the figures 3, 4 and 5.
The solution in figure 5 was concluded to be very satisfactory, yielding a maximum along-track error of 50 m.

3 Disturbing Force Fields

The next step was to investigate till what extent the influence of other force fields peculiar to the STARLETTE orbit, could be absorbed by the gravity coefficients deployed in the solution of figure 5. Therefore a new reference orbit was generated utilizing the GEM9 earth gravity field as well as the additional force models, outlined in table 1.
The result of an approximation of a 7-day arc out of this reference orbit, utilizing the tailored earth gravity model only, is displayed in figure 6.
It is remarkable that only the across-track component suffers notably from the addition of the disturbing forces and the amplitude is quite acceptable still. Further study revealed that this result does not always reproduce as can be seen from figure 7.
There a 7-day reference orbit was generated for a completely different period. This figure shows the result of an approximation to a GEM9 reference orbit and an orbit generated in a realistic force field discussed above, respectively. The solution for the realistic orbit now suffers from large effects up to 300 m in the along-track component. Figure 8 indicates which of the disturbing



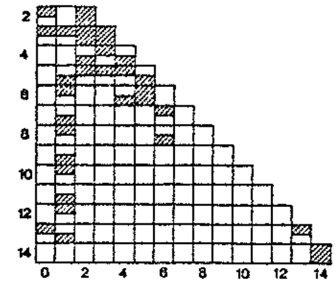
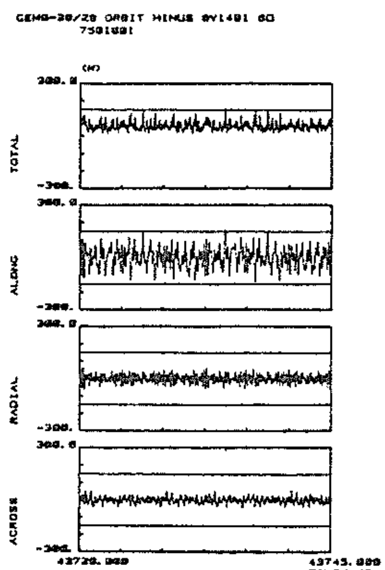
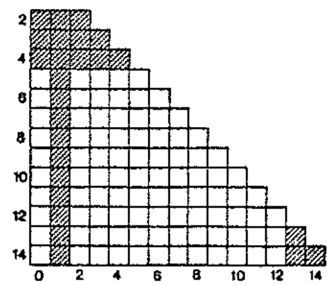


fig. 2

fig. 3





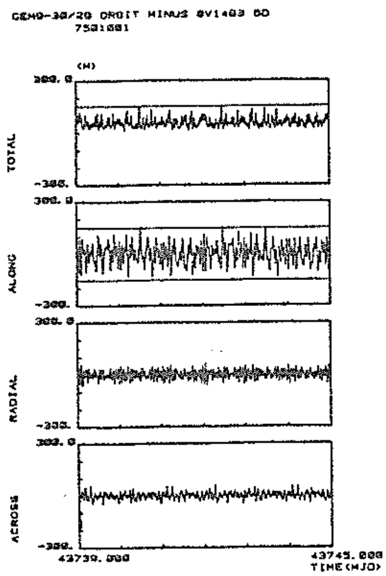
Figures 2 and 3.

Deviations of the approximated arc from the reference orbit. The reference orbit is created from the GEM9 gravity field and no other forces were accounted for. The selected gravity coefficients included in the taylored model are indicated in the right hand figures.

force fields are primarily responsible.
Both the effects of lunar gravity and of air drag cannot be sufficiently absorbed by the gravity coefficients in this case. The influence of all other individual effects was negligible. It is anticipated that simplified analytical expressions to accommodate the effect of lunar gravity and of air drag would in practice be sufficient to limit the approximation error in a 7-day orbit to 50-75 m.

4 LAGEOS

To have an impression on the applicability of this technique to the LAGEOS orbit, a reference orbit was generated for this satellite utilizing the



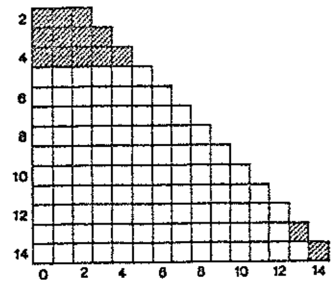
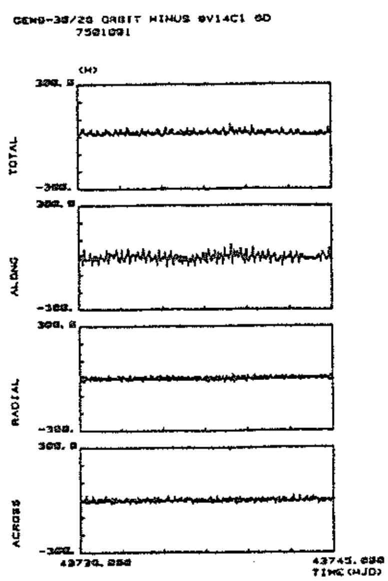
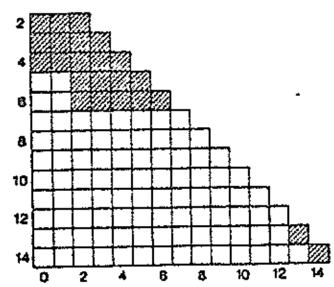


fig. 4

fig. 5





Figures 4 and 5.

Similar results as in figures 2 and 3. The taylored model indicated in figure 5 has been selected for the subsequent investigations.

models as outlined in table 1. Results of the approximation of a 30-day arc to this orbit are illustrated in figure 9.
At first the tailored model consisted of earth gravity coefficients up to degree and order 4 only. The quite fatal long-periodic effects illustrated, appeared to be mainly due to lunar and solar gravity, as is illustrated by the lower graph.

5 Conclusions

Although further investigations will be required, the results obtained so far, clearly indicate the feasibility of the approximation technique to represent a 7-day STARLETTE arc by one single state vector, going with a

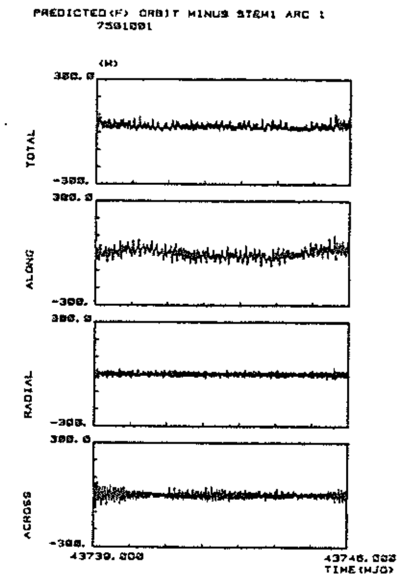


Figure 6. *Deviations from the reference orbit described in table 1. The taylored model used in the approximation is the one indicated in figure 5.*

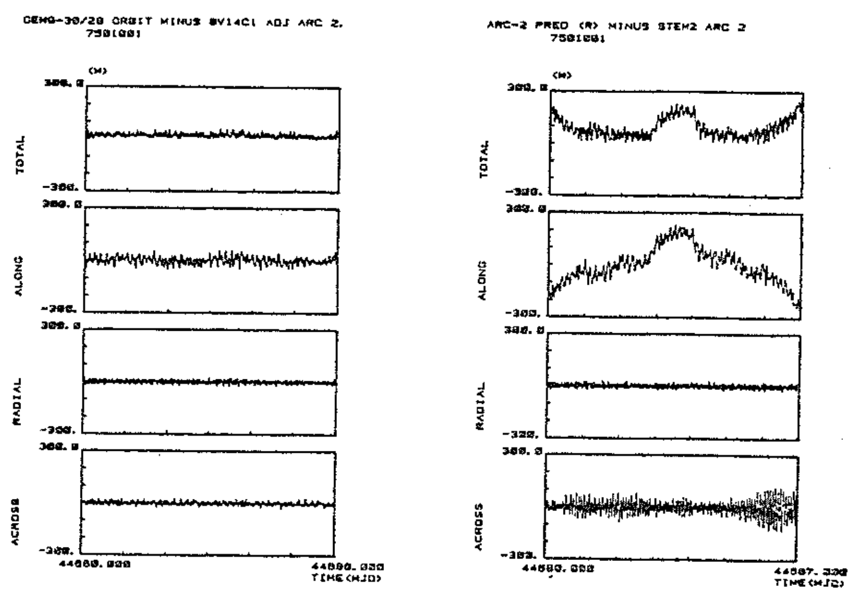


Figure 7. *Another STARLETTE arc. In the left hand figure only GEM9 was incorporated in the reference orbit. The right hand figure illustrates the result of an approximation to the reference orbit described in table 1. In this case a significant along-track error occurs (compare figure 6).*

strongly simplified tailored force field from which the equations of motion of

	STARLETTE	LAGEOS
Earth gravity	GEM9 up to (30,28)	GEM10B up to (13,13)
Lunar gravity	mass ratio 1.229997E-2	mass ratio 1.229997E-2
Solar gravity	mass ratio 3.329456E+5	mass ratio 3.329456E+5
Earth tides	K2= 0.290 K3= 0.0 Phase angle= 2.5 degr.	K2= 0.350 K3= 0.0 Phase angle= 0.0 degr.
Air drag	C_d= 3.5 Jacchia 1965 static density model	-
Solar radiation	4.500E-6 N/M^2 C_r= 1.5 Area= 4.522 E-2 M^2 Mass= 4.700E+1 KG	4.500E-6 N/M^2 C_r= 1.158 Area= 2.827E-1 M^2 Mass= 4.110E+2 KG
Auxiliary	-	Along-track acceleration -3.860 pm/S^2

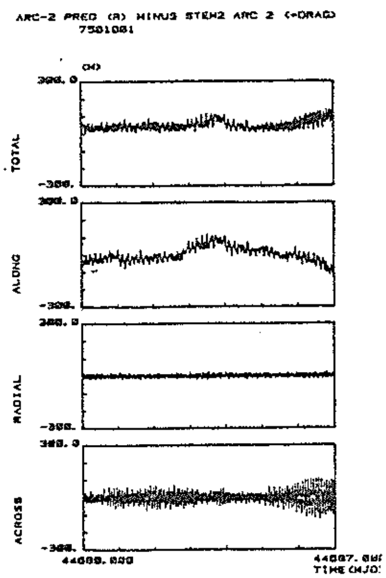
Table 1. Description of the force models defining the reference orbits of STARLETTE and LAGEOS used in the analysis.

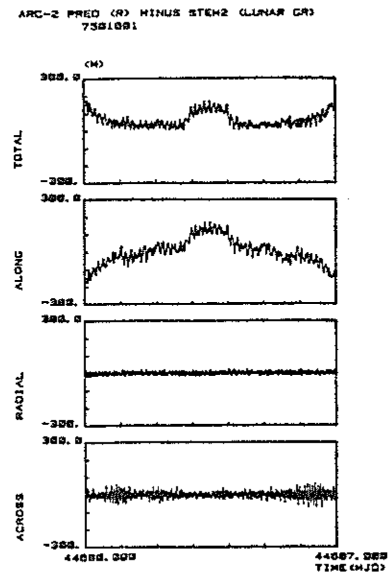
the satellite are to be derived. Maximum errors of 50 to 75 m can be expected. The mathematics involved could hardly impose serious problems in developing software even for very small computer systems.

In case of LAGEOS it might be sufficient to have only one single state vector to predict the orbit during an entire observation campaign period of several months time.

6 Acknowledgment

Discussions with B.A.C. Ambrosius and B.H.W. van Gelder substantially incited these investigations.





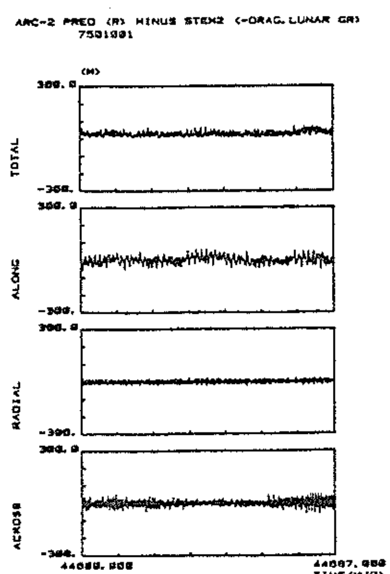


Figure 8.

The solution to the problem of figure 7. The effects of air drag (top left) and lunar gravity (top right) were individually accounted for. If both the effects of air drag and lunar gravity are included in the taylored model, the result of the approximation for this arc is quite acceptable (bottom).

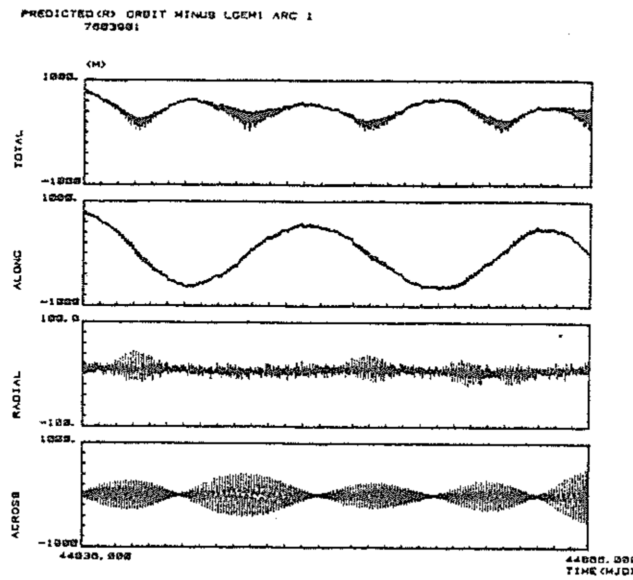


Figure 9. *The approximation technique applied to a 30-day arc of LAGEOS. The taylored model comprises earth gravity coefficients up to degree and order 4 exclusively (top) and together with models for solar and lunar gravity (bottom).*