U.S. DEPARTMENT OF COMMERCE
ENVIRONMENTAL SCIENCE SERVICES ADMINISTRATION
WEATHER BUREAU

Weather Bureau Technical Memorandum TDL-6

A FORTRAN PROGRAM FOR THE CALCULATION OF HOURLY VALUES OF ASTRONOMICAL TIDE AND TIME AND HEIGHT OF HIGH AND LOW WATER

N. A. Pore
Techniques Development Laboratory, SDO, Weather Bureau

and

R. A. Cummings
Oceanography Division, Coast and Geodetic Survey

SYSTEMS DEVELOPMENT OFFICE
TECHNIQUES DEVELOPMENT LABORATORY
TECHNICAL MEMORANDUM NO. 6

SILVER SPRING, MD.
January 1967

Reprinted December 1975

NOAA--S/T 76-2035

CONTENTS

A FORTRAN PROGRAM FOR THE CALCULATION OF HOURLY VALUES OF ASTRONOMICAL TIDE AND TIME AND HEIGHT OF HIGH AND LOW WATER

N. A. PORE
Techniques Development Laboratory, SDO, Weather Bureau

and

R. A. CUMMINGS
Oceanography Division, Coast and Geodetic Survey

ABSTRACT

A listing of the FORTRAN statements for a program for computing hourly values and times and heights of high and low astronomical tide is presented. Also shown are a flow chart of the logic of the program, descriptions and samples of the program control cards and data cards, and a sample of each type of output.

1. INTRODUCTION

The change from analog to digital computation of tides began in 1956 when the first program for the prediction of hourly tide heights by the harmonic method was prepared for the IBM 701 in machine language. This method was subsequently programed in machine language for the IBM 704 and the IBM 7094. The 7094 program had a subroutine which computed the times and heights of high and low water. This method was described by Harris, Pore, and Cummings [1]. The most recent version of this program is written in FORTRAN IV for the IBM 7094 and consequently is easily adapted to any computer with adequate memory using the FORTRAN system. It requires 1.7 minutes on the IBM 7094 to compute one year of hourly heights and times and heights of high and low water.

The purpose of this report is to document the actual FORTRAN program for prospective users. Full details of the FORTRAN program are presented along with instructions for the preparation of the program control cards. Little description of the actual harmonic tide prediction method is presented because this is available elsewhere [1, 2].

2. MATHEMATICAL PROCEDURE

The height of the tide at any time, as given by Schureman [2], may be written:

$$h = H_o + \sum_{h=1}^{N} f_n H_n \cos [a_n t + (V_o + u)_n - K'_n], \tag{1}$$

where

h	=	height of tide at any time t.
H_o	=	mean height of water level above datum used for prediction.
H_n	=	mean amplitude of any constituent A_n.
f_n	=	factor for reducing mean amplitude to year of prediction.
a_n	=	hourly speed of constituent A_n.
t	=	time, in hours, reckoned from beginning of year of prediction.
$(V_o + u)_n$	=	Greenwich equilibrium argument of constituent A_n when t=0.
K'_n	=	modified epoch of constituent A_n.
N	=	number of constituents used for the particular station.

The cosines of the argument $[a_n t + (V_o + u)_n - K'_n]$ are supplied for the calculation by a relatively efficient "table look-up" procedure.

The logic of the procedure is shown in the flow chart of figure 1.

3. PROGRAM DESCRIPTION

The program is written in FORTRAN IV language and has been compiled and tested on the IBM 7094. It should be adaptable to any computer with adequate memory which utilizes the FORTRAN system.

A listing of the FORTRAN source statements is presented in Appendix A. The steps in the program described below are keyed to the source statements by the numbers and scale added to the right edge of the listing in Appendix A.

1. Initialize MS, MY, and MD, the control words which determine if more than one problem is to be done. Specify that constants for the station, year, and date will be read in. After the first set of calculations these variables may be set to zero if station, year, or date are not being changed.

2. Read-in the table of constituent speeds A(J). The program is compiled to accept 37 constituents. This number can be increased by a change of the DIMENSION statement and a change of the indexing of the appropriate statements.

3. Convert the constituent speeds to units such that 1024 units are equal to $\pi/2$.

4. Read station name from punched card (72H). The amplitude of each constituent (AMP(J)), and the modified epoch of each constituent (EPOCH(J)) are read in for the station. These require 6 punched cards. The first word on each card is the assigned station number. The second word is the card number, 1 through 6. These station cards must be in correct numerical order

$$h_t = H_o + \sum_{n=1}^{N} f_n H_n \cos\left[a_n t + (V_o + u)_n - K'_n\right]$$

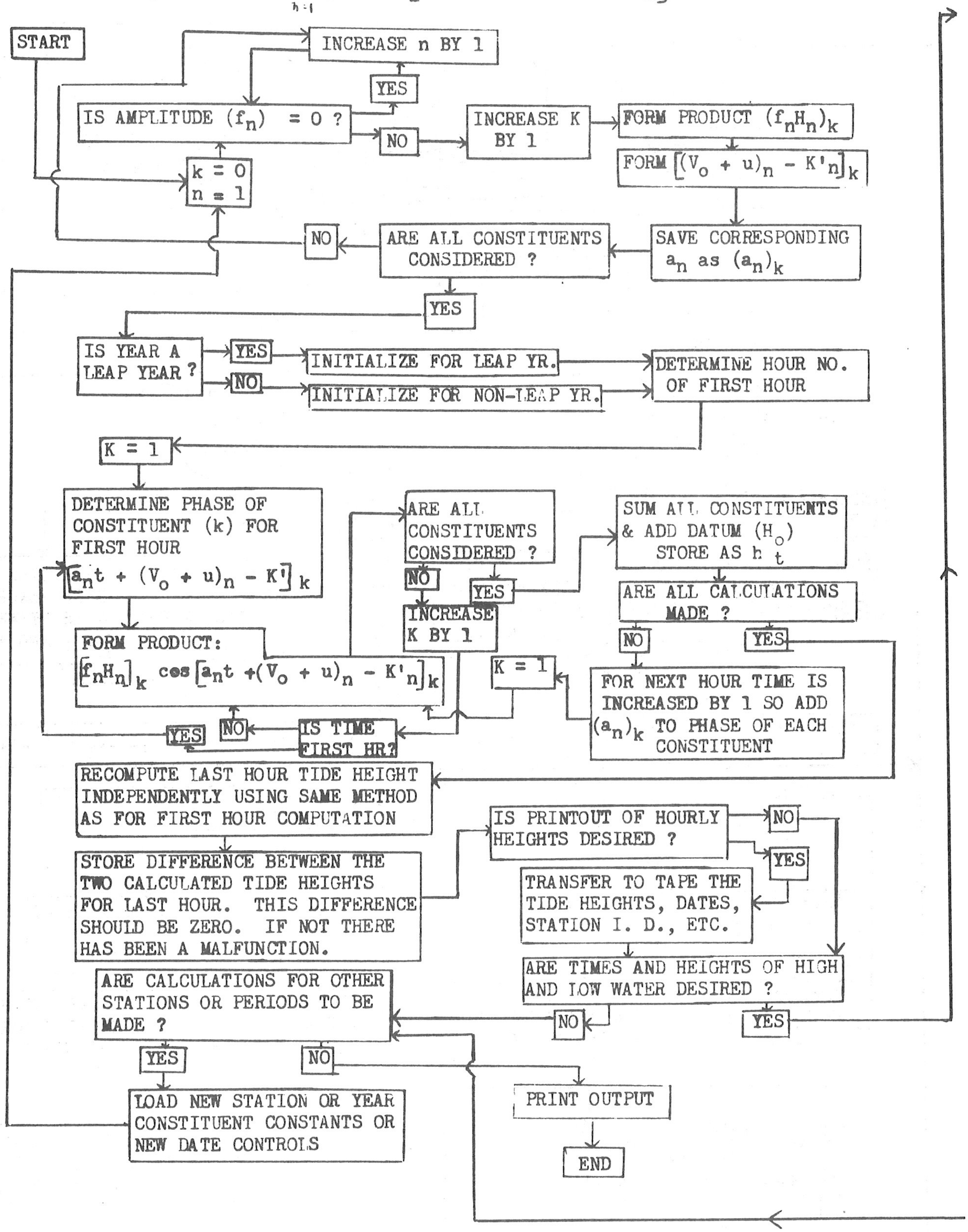
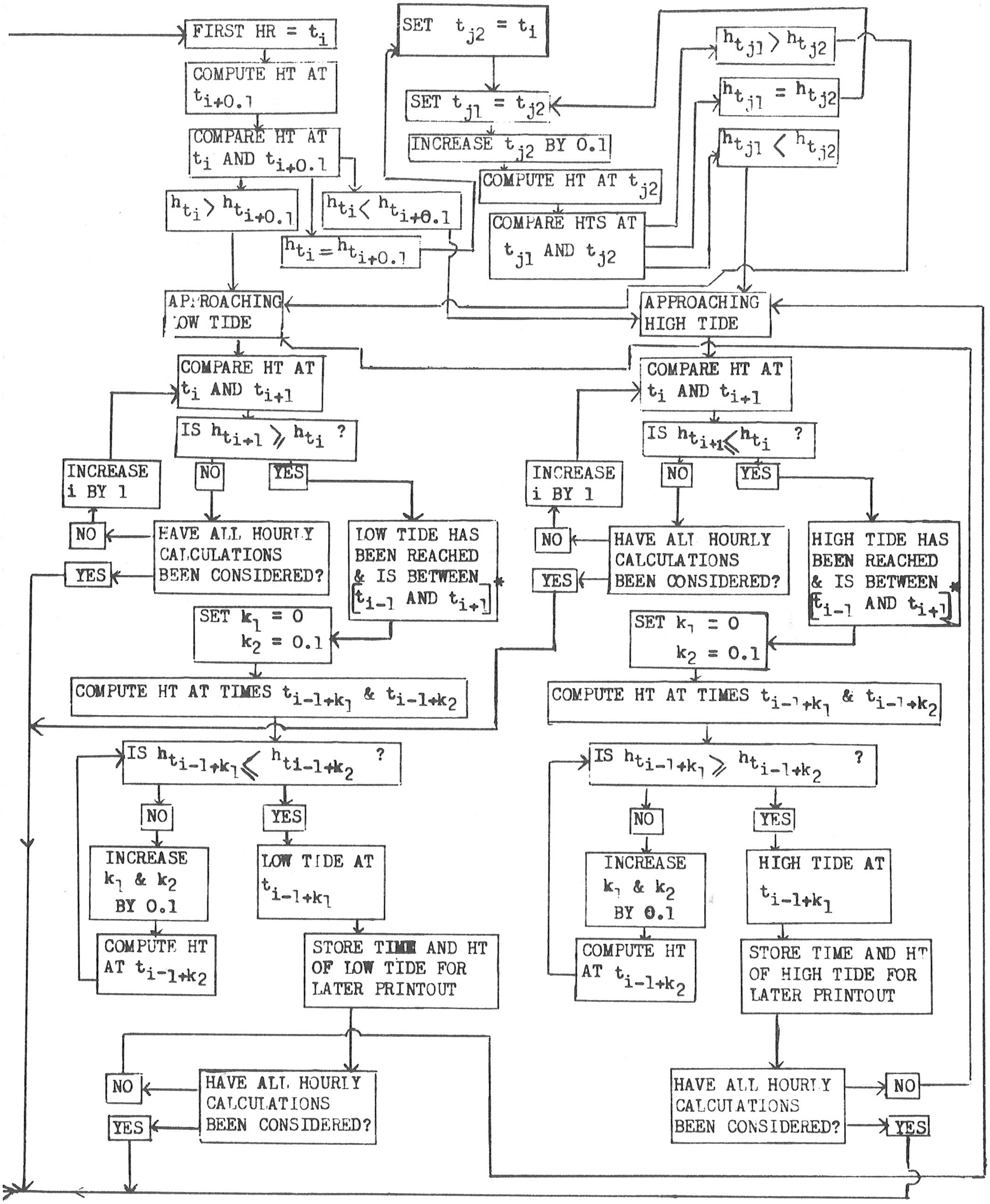


Figure 1. - Flow chart showing the logic of the procedure for the astronomical tide prediction program.



* If t_i is first hour, extreme tide is between t_i and t_{i+1} rather than between t_{i-1} and t_{i+1}. The search for high or low tide is made between these two hours.

so that the amplitudes and epochs are read in for the appropriate constituents. The station numbers are compared for consistency and the order of the six cards is checked.

5. Read-in the datum plane (DATUM) and control word (IND). The tide heights are calculated with reference to mean sea level datum plane. The amount in feet specified by DATUM is added to the calculations before they are printed. The control word IND determines if hourly calculations, high and low water calculations, or both are to be printed. IND is set to 1 for highs and lows only, 2 for both hourly values and highs and lows, and 3 for hourly values only.

6. Determine if year constants are to be read in.

7. Read-in the node factor (XODE (J)) and the equilibrium argument (VPU(J)) for each constituent for the particular year. This requires 5 cards. The first word is the year, the second specifies whether the year of calculation is a leap year or a non-leap year (1 for leap year and 0 for non-leap year). The third word is the card number, 1 through 5. The cards are checked for year consistency and for card order.

8. Determine if a date card is to be read in.

9. Read-in date control card consisting of 12 time periods. MO(J) is month, NBDAY(J) is the beginning day, and NEDAY(J) is the ending day. If calculations are for less than 12 time periods, the unused portion of the card is left blank. The unused words MO(J), NBDAY(J), NEDAY(J) are therefore read in as zero.

10. Examine amplitudes of the constituents (AMP(J)) for zero and form a set of tables consisting of AMPA(K) which is $f_n H_n$ in equation 1, EPOCH(K) which is $(V_o + u)_n - K'_n$ in equation 1, and SPD(K) which is a_n in equation 1. The resulting set of constants is for the constituents of non-zero amplitude. Also constituent speeds in tenths of hours are computed and stored as SP(K).

11. Compute NOCON, the number of non-zero constituents, which is the number of constituents to be used in the calculations.

12. Put the month, beginning, and ending dates in MO(13), NB(13), and NEDAY(13).

13. Check for blank (zero) month number which indicates calculations for the last time period have been completed.

14. Determine number of days (NODAYS) and number of hours (NOHRS) for which calculations are to be made.

15. Check whether year is leap year or non-leap year.

16. Determine the first hour number of the year for which calculations are to be made (FIRST). The first hour number of each month is stored in table TABHR(K). There are 12 values for non-leap years and 12 for leap years. First is saved as NFIRST.

17. Determine phase angle of each constituent for the first hour of calculation. These arguments are then reduced to values of less than 2π.

18. Determine for all hours after the first the phases of the constituents by adding the speed of each constituent SPD(J) to the previous phase (ARG(J)). These also are reduced to less than 2π.

19. Determine the quadrant of the phase of the constituent. This is necessary as program utilizes a cosine table of 1025 values, from 0 to $\pi/2$.

20. From NP determine which value of the cosine table to use for the particular constituent; NP is determined by rounding.

21. Accumulate the value of the tide for a particular hour STORX(K) for those constituents which are in an increasing phase.

22. Same as 20.

23. Same as 21 except for constituents which are in a decreasing phase.

24. Find time of last hour for which calculations are made. This is used for calculating the tide height for the last hour the second time by the method used in the first hour in step 17. This independent calculation of the last hour is compared to the value for the last hour obtained by the short cut method of step 18.

25. Transfer for calculation of the last hour tide height.

26. Find CKSUM, the difference between the two calculations of the last hour tide. Should be very near zero.

27. Add datum plane to the tide calculations.

28. Write a heading line consisting of year, month, datum, number of constituents, and CKSUM.

29. Write out the hourly calculations of tide.

30. Check to determine if hourly values only are desired.

31. Check constituent constants cards for errors. (Format of error messages is shown in listing in Appendix.)

32. Initialize ITEMS to zero. ITEMS builds up to twice the number of extreme tides.

33. Set number of one-hour time intervals.

34. Begin search for tide extremes.

35. Test to determine if first hour is being considered.

36. Initialize for first hour.

37. States that an extreme tide will occur between TIME - 1 and TIME + 1 hours.

38. Compute tide heights at tenth of hour intervals. Similar to computation of hourly tide heights.

39. Transfer. (Depends on whether the program is searching for a high tide or a low tide.)

40. Initialize for low tide search.

41. Search for low tide.

42. Check to determine if low tide has been found.

43. Initialize for search of high tide.

44. Search for high tide.

45. Check to determine if high tide has been found.

46. Determine if first tide extreme is a high or low tide.

47. Store time and height of extreme tide.

48. Transfer appropriately to search for high or low tide.

49. Change hours and tenths of hours to hours and minutes.

50. Set sign of all zero heights to plus.

51. Set initial day number.

52. Write four heading lines.

53. Write times and heights of extreme tides.

54. Read MS, MY, MD to determine if more problems are to be done. If MS, MY, and MD are all zero, all calculations are finished. If any of the three are not zero, more calculations are to follow. If MS is non-zero, new station cards are to be loaded. If MY is non-zero, new year cards are to be loaded. If MD is non-zero, a new date control card is to be loaded.

4. CONTROL CARDS

The data and control cards are arranged in the following order:

Cosines	(86 cards)
Constituent Speeds	(6 cards)
Title card	(1 card)
Station constants	(6 cards)
Datum plane, Output control word	(1 card)
Year constants	(5 cards)

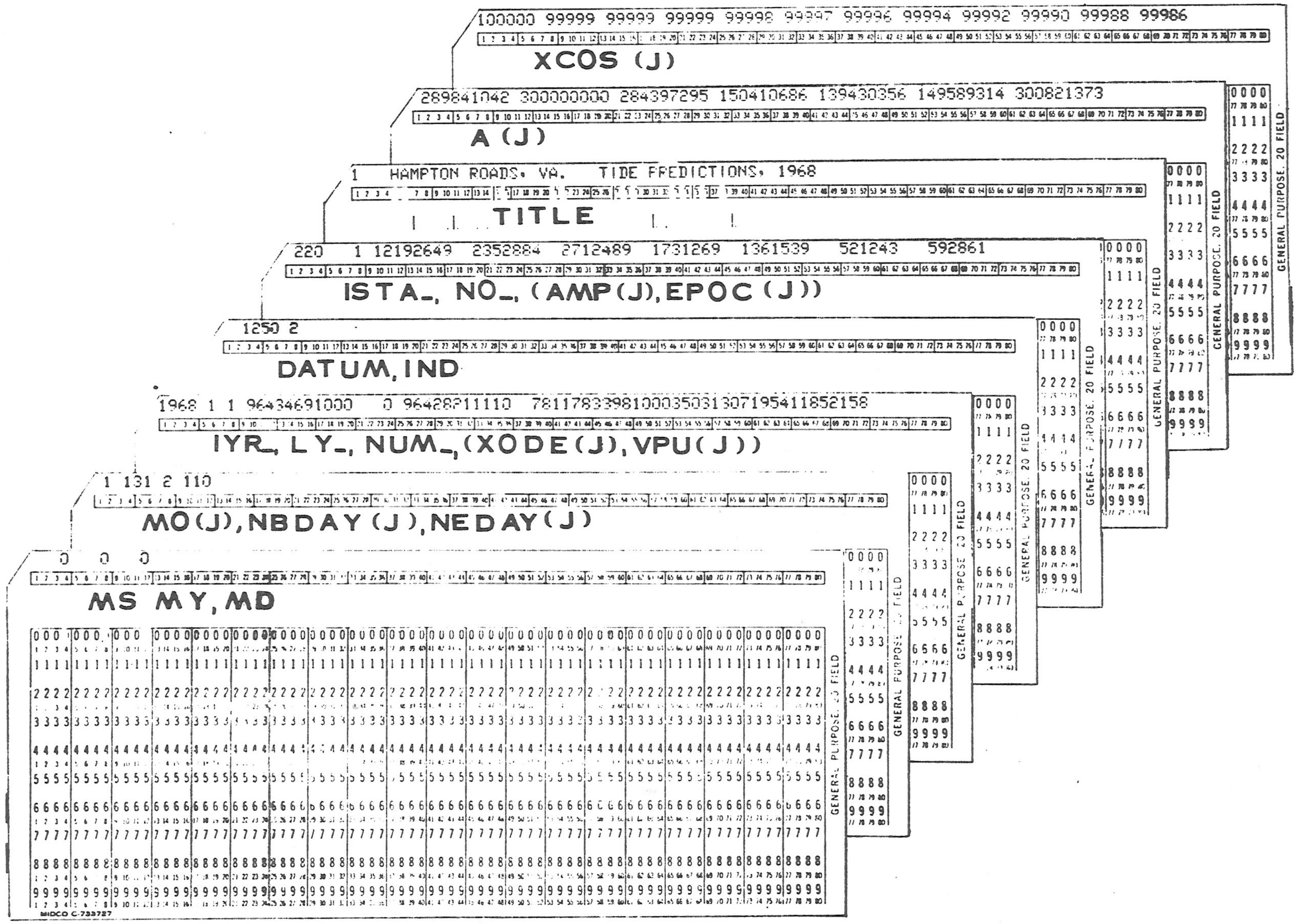


Figure 2. - Samples of the input data and program control cards.

Date control (1 card)
Termination control (1 card)

A sample of each type of card is shown in figure 2. The formats and card descriptions of the data and control words used in the program follow:

Cosine cards: XCOS(J)

(12F6.5) 1025 cosine values from 0 to $\pi/2$ at intervals of 90/1024 degrees. A listing of the cosines is presented as Appendix B.

Constituent Speeds cards: (A(J))

(7F10.7) 37 values of constituent speeds in degrees per hour.

Title card:

(72H) Any 72 Hollerith characters desired as the title line of output.

Station cards: ISTA_, NO., (AMP(J), EPOCH(J))

(2I4, F5.3, F4.1, F5.3, F4.1, F5.3, F4.1, F5.3, F4.1, F5.3, F4.1, F5.3, F4.1, F5.3, F4.1). The first two words on each of the six station cards are the station number and the card number which varies from 1 through 6. The remainder of each card consists of seven pairs of amplitudes (AMP(J)) in feet and phase angles (EPOCH(J)) in degrees for seven constituents.

Datum plane and output control card: DATUM, IND

(F6.3, I2) The datum plane (DATUM) is expressed in feet. The output control word (IND) indicates which calculations and output are desired. IND is punched 1 for high and low tides only, 2 for hourly values and high and low tides, and 3 for hourly values only.

Year cards: IYR_, LY_, NUM_, (XODE(J), VPU(J))

(I4, 2I2, F4.3, F4.1, F4.3, F4.1, F4.3, F4.1, F4.3, F4.1, F4.3, F4.1, F4.3, F4.1, F4.3, F4.1, F4.3, F4.1). The first three words on each of the five year cards are the year (IYR_), a word (LY_) which specifies if the year is a leap year, and the card number (NUM_). LY_ is punched 0 for a non-leap year and 1 for a leap year. The remainder of the card consists of eight pairs of amplitude modification factors (XODE(J)) and Greenwich equilibrium arguments (VPU (J)) in degrees.

Date control card: MO(J), NBDAY(J), NEDAY(J)

(36I2) Each time period for which calculations are to be made is specified by three words, the month number (MO(J)), the beginning date (NBDAY(J)), and the last date of the period (NEDAY(J)). Twelve time periods is the maximum number which can be specified on one date control card.

HAMPTON ROADS, VA.   TIDE PREDICTIONS, 1968
YEAR 1968   MONTH 3   DATUM 1.25   NO. OF CONSTITUENTS 15   CHECKSUM -0.0000000

1	2.0	1.4	0.7	0.1	-0.2	-0.2	0.1	0.7	1.4	2.0	2.3	2.4
1	2.1	1.5	0.8	0.2	-0.2	-0.3	0.0	0.5	1.2	1.9	2.3	2.5
2	2.3	1.8	1.2	0.5	0.1	-0.1	-0.0	0.4	1.0	1.6	2.1	2.3
2	2.2	1.8	1.2	0.6	0.1	-0.2	-0.1	0.3	0.8	1.5	2.0	2.4
3	2.4	2.1	1.6	1.0	0.4	0.1	-0.0	0.2	0.6	1.2	1.7	2.0
3	2.1	1.9	1.5	0.9	0.4	0.0	-0.1	0.1	0.5	1.1	1.7	2.1
4	2.3	2.2	1.9	1.4	0.8	0.4	0.2	0.2	0.4	0.9	1.4	1.8
4	2.0	1.9	1.6	1.2	0.7	0.3	0.0	0.0	0.3	0.8	1.3	1.8
5	2.1	2.2	2.1	1.7	1.2	0.8	0.4	0.3	0.3	0.6	1.0	1.4
5	1.7	1.8	1.7	1.4	1.0	0.6	0.3	0.1	0.2	0.5	1.0	1.5
6	1.9	2.1	2.2	2.0	1.6	1.2	0.7	0.5	0.4	0.5	0.7	1.1
6	1.4	1.7	1.7	1.6	1.3	0.9	0.6	0.3	0.2	0.3	0.6	1.1
7	1.5	1.9	2.1	2.1	1.9	1.5	1.1	0.8	0.5	0.4	0.5	0.8
7	1.1	1.4	1.6	1.6	1.5	1.2	0.9	C.6	0.3	0.3	0.4	0.7
8	1.1	1.5	1.9	2.1	2.1	1.9	1.6	1.2	0.8	0.5	0.4	0.5
8	0.7	1.1	1.4	1.6	1.6	1.5	1.3	0.9	0.6	0.3	0.3	0.4
9	0.7	1.1	1.6	1.9	2.1	2.2	2.0	1.6	1.2	0.7	0.5	0.3
9	0.4	0.7	1.0	1.4	1.6	1.7	1.6	1.3	1.0	0.6	0.3	0.2
10	0.3	0.6	1.1	1.6	2.0	2.3	2.3	2.C	1.6	1.1	0.7	0.3
10	0.2	0.3	0.7	1.1	1.5	1.8	1.9	1.8	1.4	1.0	0.5	0.2

HAMPTON ROADS, VA.   TIDE PREDICTIONS, 1968
YEAR 1968   MONTH 3   DATUM 1.25   NO. OF CONSTITUENTS 15   CHECKSUM -0.0000000

TIMES AND HEIGHTS OF HIGH AND LOW WATERS

DAY	TIME	HT.	TIME	HT.	TIME	HT.	TIME	HT.	TIME	HT.	TIME	HT.
1	430	-0.3	1036	2.4	1642	-0.3	2300	2.5				
2	506	-0.2	1112	2.3	1718	-0.2	2336	2.4				
3	548	0.0	1142	2.1	1748	-0.1						
4	12	2.3	624	0.1	1218	2.0	1830	0.0				
5	54	2.2	706	C.3	1300	1.8	1906	0.1				
6	136	2.2	800	C.4	1348	1.7	1954	0.2				
7	230	2.1	854	C.4	1448	1.6	2048	0.3				
8	330	2.1	1000	0.4	1548	1.6	2154	0.3				
9	430	2.2	1100	C.3	1654	1.7	2300	0.2				
10	530	2.3	1200	C.2	1754	1.9						

Figure 3. - Samples of the printed output of the program. These printouts show (above) the calculated hourly tide and (below) the times and heights of high and low waters for Hampton Roads, Va. for March 1 - 10, 1968.

Termination control card: MS, MY, MD

(3I4) These three words determine if more problems are to be done. If all three are zero, no other problems will be done. If MS is punched 1, new station constants will be read. If MY is punched 1, new year constants will be read. If MD is punched 1, a new date control card will be read.

5. SAMPLES OF THE OUTPUT

A sample of each type of printed output is shown in figure 3. The top half of the figure is a set of hourly tide heights and the lower portion is a set of times and heights of high and low waters. The datum plane is 1.25 feet (which is mean low water at Hampton Roads) and 15 non-zero constituents were included in the calculations.

Each day of hourly tide heights requires two lines of printing. The first number of each line is the date of the month and the other twelve numbers are the hourly tide heights expressed in feet. The first line of each day contains heights from 0000 Local Standard Time (LST) to 1100 LST. The second line covers the period 1200 LST through 2300 LST.

The section of the output containing the highs and lows shows the time (LST) of high or low tide in hours and minutes with no separation or punctuation between hours and minutes. Heights are printed in feet. For example, figure 3 shows the first low tide on March 1 to be -0.3 ft. MLW at 0430 LST.

REFERENCES

1. D. L. Harris, N. A. Pore, and R. A. Cummings, "Tide and Tidal Current Prediction by High Speed Digital Computer," International Hydrographic Review, vol. XLII, No. 1, Jan. 1965, pp. 95-103.

2. P. Schureman, Manual of Harmonic Analysis and Prediction of Tides, Special Publication No. 98, U. S. Coast and Geodetic Survey, Washington, D. C., 1958, 317 pp.

APPENDIX A

Listing of FORTRAN IV source statements for Astronomical Tide Prediction Program as compiled on the IBM 7094.

```
      DIMENSION A(40),AMP(40),EPOC(40),XODE(40),VPU(40),MO(13),SP(40),
     1 NBDAY(13),NEDAY(13),XCOS(1026),SPD(40),ARG(40),TABHR(25),
     2 ANG(40),KDAY(32),STORX(769),EXTIM(400),JXTIM(200),XHT(200),
     3 EPOCH(40),AMPA(40),XXHT(12),JJXTI(12)
      DATA (TABHR(I), I=1,24)/  -24.,  720., 1392., 2136., 2856., 3600.,
     1     4320., 5064., 5808., 6552., 7272., 7992.,  -24.,  720., 1416.,
     2     2160., 2880., 3624., 4344., 5088., 5832., 6552., 7296., 8016./
      READ (5,535) (XCOS(J),  J = 1,1025)
      MS = 1
      MY = 1                                                            1
      MD = 1
      CON = 1024. / 90.
      READ (5,530) (A(J),  J = 1,37)                                    2
      DO 90 J = 1,37
   90 A(J) = A(J) * CON                                                 3
  100 IF (MS.EQ.0) GO TO 120
  110 READ (5,550)
      READ (5,531) ISTA1,NO1,(AMP(J),EPOC(J),J=1,7),ISTA2,NO2,(AMP(J),
     1 EPOC(J),J=8,14),ISTA3,NO3,(AMP(J),EPOC(J),J=15,21),ISTA4,NO4,
     2 (AMP(J),EPOC(J),J=22,28),ISTA5,NO5,(AMP(J),EPOC(J),J=29,35),
     3 ISTA6,NO6,(AMP(J),EPOC(J),J=36,40)
       IF (ISTA1 .NE. ISTA2)  GO TO 451
       IF (ISTA2 .NE. ISTA3)  GO TO 451
       IF (ISTA3 .NE. ISTA4)  GO TO 451                                 4
       IF (ISTA4 .NE. ISTA5)  GO TO 451
       IF (ISTA5 .NE. ISTA6)  GO TO 451
       IF (NO1  .NE.  1) GO TO 450
       IF (NO2  .NE.  2) GO TO 450
       IF (NO3  .NE.  3) GO TO 450
       IF (NO4  .NE.  4) GO TO 450
       IF (NO5  .NE.  5) GO TO 450
       IF (NO6  .NE.  6) GO TO 450
  119 READ (5,532) DATUM,IND                                            5
  120 IF (MY.EQ.0) GO TO 131                                            6
  130 READ (5,533) IYR1,LY1,NUM1,(XODE(J),VPU(J), J = 1,8), IYR2, LY2,
     1 NUM2, (XODE(J),VPU(J), J=9,16), IYR3,LY3,NUM3, (XODE(J),
     2 VPU(J),J = 17,24),IYR4,LY4,NUM4, (XODE(J),VPU(J), J = 25,32),
     3 IYR5,LY5,NUM5, (XODE(J),VPU(J), J = 33,40)
       IF(IYR1 .NE. IYR2) GO TO 452                                     7
       IF(IYR2 .NE. IYR3) GO TO 452
       IF(IYR3 .NE. IYR4) GO TO 452
       IF(IYR4 .NE. IYR5) GO TO 452
       IF(NUM1 .NE. 1)   GO TO 453
       IF(NUM2 .NE. 2)   GO TO 453
       IF(NUM3 .NE. 3)   GO TO 453
       IF(NUM4 .NE. 4)   GO TO 453
       IF(NUM5 .NE. 5)   GO TO 453
  131  IF (MD.EQ.0) GO TO 160                                           8
  140  READ (5,534) (MO(J), NBDAY(J), NEDAY(J), J = 1,12)               9
C      SET UP TABLES FOR NON-ZERO CONSTITUENTS
  160  K = 0
       DO 180  J = 1,40
  161  IF (AMP(J).EQ.0.0) GO TO 180
  170  K = K + 1
       AMPA(K) = AMP(J) * XODE(J)
       TEMX = VPU(J) - EPOC(J)                                          10
       IF (TEMX .GE. 0.) GO TO 171
       TEMX = TEMX + 360.
  171  EPOCH(K) = TEMX * CON
       SPD(K) = A(J)
       SP(K) = SPD(K) / 10.
  180  CONTINUE                                                         11
       NOCON = K
C      OPERATING TABLES NOW STORED AS AMPA(K),EPOCH(K),SPD(K)
       DO 2000 JP = 1,12
       MO(13) = MO(JP)                                                  12
       NBDAY(13) = NBDAY(JP)
       NEDAY(13) = NEDAY(JP)
       NNEDA = NEDAY(13) + 1
  181  IF (MO(13).EQ.0) GO TO 2005                                      13
  190  NODAYS = NEDAY(13) - NBDAY(13) + 2
       NOHRS = NODAYS * 24                                              14
       HRS = NOHRS
C      DETERMINE FIRST HOUR OF TIME PERIOD                              15
  191  IF (LY1.GT.0) GO TO 210
  200  K = MO(13)
       GO TO 215
  210  K  = MO(13) + 12                                                 16
```

```
  215 BDAY = NBDAY(13)
      FIRST = TABHR(K) + BDAY * 24.
      NFIRST = FIRST
      DO 220 J = 1,769
  220 STORX(J) = 0.
C     TIDE = DATUM + AMPA(K) * COS(A(K) * T + EPOCH(K))
      KOUNT = 0
      KT = 0
      DO 380  K = 1,NOHRS
  221 IF (KOUNT.GT.0) GO TO 260
  230 KOUNT = 1
  231 DO 250 J = 1,NOCON
      ARGU = SPD(J) * FIRST + EPOCH(J)                                  17
  250 ARG(J) = AMOD(ARGU,4096.)
      GO TO 290
  260 DO 280  J = 1,NOCON
      ARG(J) = ARG(J) + SPD(J)
  270 IF (ARG(J).LT.4096.) GO TO 280                                    18
      ARG(J) = ARG(J) - 4096.
      GO TO 270
  280 CONTINUE
  290 DO 374 J = 1,NOCON
      IF (ARG(J) - 1024.)  320,320,300
  300 IF (ARG(J) - 2048.)  350,350,310                                  19
  310 IF (ARG(J) - 3072.)  360,360,330
  320 ANG(J) = ARG(J)
      GO TO 340
  330 ANG(J) = 4096. - ARG(J)
  340 NP = ANG(J) + 1.5                                                 20
      STORX(K) = STORX(K) + AMPA(J) * XCOS(NP)                          21
      GO TO 374
  350 ANG(J) = 2048. - ARG(J)
      GO TO 370
  360 ANG(J) = ARG(J) - 2048.
  370 NP = ANG(J) + 1.5                                                 22
      STORX(K) = STORX(K) - AMPA(J) * XCOS(NP)                          23
  374 CONTINUE
  375 IF (K.NE.NOHRS) GO TO 380
  376 IF (KT.EQ.1) GO TO 378
      FIRST = FIRST + HRS - 1.                                          24
      KT = 1
      CHECK = STORX(K)
      STORX(K) = 0.
      GO TO 231                                                         25
  378 CKSUM = CHECK - STORX(K)                                          26
  380 CONTINUE
      DO 400  K = 1,NOHRS
  400 STORX(K) = STORX(K) + DATUM                                       27
      GO TO (419,401,401),IND
  401 KDAY(1) = NBDAY(13)
      NODAYS = NODAYS - 1
      DO 410 I = 2,NODAYS
  410 KDAY(I) = KDAY(I-1) + 1
      WRITE  (6,550)
      WRITE (6,560) IYR1,MO(13),DATUM,NOCON,CKSUM                       28
      WRITE (6,537) (KDAY(I), STORX(24*I-23), STORX(24*I-22), STORX(24*I
     1 -21),STORX(24*I-20), STORX(24*I-19), STORX(24*I-18), STORX(24*I-
     2 17), STORX(24*I-16), STORX(24*I-15), STORX(24*I-14), STORX(24*I-
     3 13), STORX(24*I-12), KDAY(I), STORX(24*I-11), STORX(24*I-10),    29
     4 STORX(24*I-9), STORX(24*I-8), STORX(24*I-7), STORX(24*I-6),
     5 STORX(24*I-5), STORX(24*I-4), STORX(24*I-3), STORX(24*I-2),
     6 STORX(24*I-1), STORX(24*I), I=1,NODAYS)
  419 IF (IND.EQ.3) GO TO 2000                                          30
      GO TO 1000
  450 WRITE (6,501)
      STOP
  451 WRITE  (6,502)
      STOP
  452 WRITE  (6,503)
      STOP
  453 WRITE  (6,504)
      STOP
  501 FORMAT (27H STATION CARDS OUT OF ORDER)
  502 FORMAT (31H STATION NUMBERS NOT CONSISTENT)
  503 FORMAT (28H YEAR NUMBERS NOT CONSISTENT)                          31
  504 FORMAT (24H YEAR CARDS OUT OF ORDER)
  530 FORMAT (7F10.7)
  531 FORMAT (2I4,F5.3,F4.1,F5.3,F4.1,F5.3,F4.1,F5.3,F4.1,F5.3,F4.1,
```

```
    1 F5.3,F4.1,F5.3,F4.1)
  532 FORMAT (F6.3,I2)
  533 FORMAT (I4,2I2,F4.3,F4.1,F4.3,F4.1,F4.3,F4.1,F4.3,F4.1,
    1 F4.3,F4.1,F4.3,F4.1,F4.3,F4.1,F4.3,F4.1)
  534. FORMAT (36I2)
  535 FORMAT (12F6.5)
  537 FORMAT (I9,12F9.1)
  538 FORMAT (3I4)
  550 FORMAT (72H
    1                                                                   )
  560 FORMAT   (6H YEAR ,I4,7H  MONTH,I3,9H  DATUM   ,F7.2,  24H   NO. OF
    1 CONSTITUENTS   ,I4,13H      CHECKSUM,F12.7//)
  575   FORMAT (//41H TIMES AND HEIGHTS OF HIGH AND LOW WATERS//)
  580   FORMAT (112H         DAY      TIME      HT.      TIME      HT.      TIME
    1        HT.       TIME       HT.       TIME       HT.        TIME       HT.)
  581   FORMAT (28H CONSECUTIVE TIDES SAME TIME)
  585   FORMAT (1H0,I9,6(5XI4,F8.1)/10X,6(5XI4,F8.1))
C       COMPUTE TIMES AND HEIGHTS OF HIGH AND LOW WATERS
 1000   ITEMS = 0                                                          32
 1010   K = 1
        NOHRS = NOHRS - 1                                                  33
 1035   DO 1500 I = 1,NOHRS                                                34
        IF (I.EQ.1) GO TO 1039                                             35
 1038   GO TO (1270,1287),NST
 1039   NWHOA = 7
        TIME = NFIRST * 10                                                 36
        NARC = 1
        GO TO 1060
 1040   IF(STORX(I) - STORX(I+1)) 1285,1045,1265
 1045   NWHOA = 1
      NARC = 1
        TIME = NFIRST * 10
        GO TO 1060
 1050   TIME = (NFIRST + I - 2) * 10                                       37
        NARC = 1
 1060   STOXR = DATUM
        GO TO (1075,1100),NARC
 1075   DO 1090 J = 1,NOCON
        ARGU = SP(J) * TIME + EPOCH(J)
 1090   ARG(J) = AMOD(ARGU,4096.)
        GO TO 1120
 1100   DO 1110 J = 1,NOCON
        ARG(J) = ARG(J) + SP(J)
 1105   IF (ARG(J).LT.4096.) GO TO 1110
        ARG(J) = ARG(J) - 4096.
        GO TO 1105
 1110   CONTINUE
 1120   DO 1220 J = 1,NOCON
        IF (ARG(J) - 1024.) 1150,1150,1130
 1130   IF (ARG(J) - 2048.) 1180,1180,1140                                 38
 1140   IF (ARG(J) - 3072.) 1190,1190,1160
 1150   ANG(J) = ARG(J)
        GO TO 1170
 1160   ANG(J) = 4096. - ARG(J)
 1170   NP = ANG(J) + 1.5
        STOXR = STOXR + AMPA(J) * XCOS(NP)
        GO TO 1220
 1180   ANG(J) = 2048. - ARG(J)
        GO TO 1200
 1190   ANG(J) = ARG(J) - 2048.
 1200   NP = ANG(J) + 1.5
 1210   STOXR = STOXR - AMPA(J) * XCOS(NP)
 1220   CONTINUE
        GO TO (1250,1260,1275,1280,1290,1295,1400,1410,1412), NWHOA        39
 1250   POINT1 = STOXR
        NWHOA = 2
        NARC = 2
        GO TO 1060
 1260   IF (POINT1 - STOXR) 1285,1060,1265
 1265   NWHOA = 3
      NST = 1                                                              40
 1270   IF (STORX(I) - STORX(I+1)) 1050,1050,1500                          41
 1275   POINT1 = STOXR
        JHOL = 1
        NWHOA = 4
        NARC = 2
        TIME = TIME + 1.
        GO TO 1060
```

```
1280  IF (POINT1 - STOXR) 1425,1425,1300                              42
1285  NWHOA = 5
      NST = 2                                                        43
1287  IF (STORX(I) - STORX(I+1)) 1500,1050,1050                      44
1290  POINT1 = STOXR
      JHOL = 2
      NWHOA = 6
      NARC = 2
      TIME = TIME + 1.
      GO TO 1060
1295  IF (POINT1 - STOXR) 1300,1425,1425                             45
1300  TIME = TIME + 1.
      POINT1 = STOXR
      GO TO 1060
1400  POINT1 = STOXR
      NWHOA = 8
      NARC = 2
      TIME = TIME + 1.
      GO TO 1060
1410  IF (POINT1-STOXR) 1415,1411,1420
1411  NWHOA = 9
      POINT2 = STOXR
      GO TO 1060                                                     46
1412  IF (POINT2-STOXR) 1413,1040,1414
1413  JHOL = 1
      GO TO 1425
1414  JHOL = 2
      GO TO 1425
1415  IF (STOXR - STORX(I+1)) 1285,1290,1290
1420  IF (STOXR - STORX(I+1)) 1275,1275,1265
1425  EXTIM(K) = TIME - 1.
      EXTIM(K+1) = POINT1                                            47
      IF (K.EQ.1) GO TO 1430
      IF (EXTIM(K).GT.EXTIM(K-2)) GO TO 1430
      GO TO 1300
1430  K = K + 2
      ITEMS = ITEMS + 2
      GO TO (1285,1265),JHOL                                         48
1500  CONTINUE
      KAY = ITEMS / 2
      J = 0
      DO 1650 K = 1,ITEMS,2
      J = J + 1                                                      49
      JHR = EXTIM(K)
      JHR = MOD(JHR,240)
      JTEN = MOD(JHR,10)
      JXTIM(J) = ((JHR - JTEN) * 10) + JTEN * 6
      XHT(J) = EXTIM(K+1)
      IF (XHT(J)) 1600,1650,1650
1600  IF (XHT(J).LE.-0.05) GO TO 1650                                50
      XHT(J) = -XHT(J)
1650  CONTINUE
1670  NDAY = NBDAY(13)                                               51
      NCOUNT = 0
      NNJ = 1
1674  WRITE (6,550)
      WRITE (6,560) IYR1,MO(13),DATUM,NOCON,CKSUM                    52
      WRITE (6,575)
      WRITE (6,580)
1704  DO 1750 I = 1,KAY
      IF (JXTIM(I) - JXTIM(I+1)) 1705,1710,1715
1705  NCOUNT = NCOUNT + 1
      GO TO 1750
1710  WRITE (6,581)
      STOP
1715  NLAST = NNJ + NCOUNT                                           53
1717  WRITE (6,585) NDAY,(JXTIM(J),XHT(J), J = NNJ,NLAST)
1740  NNJ = NLAST + 1
      NCOUNT = 0
      NDAY = NDAY + 1
1745 IF (NDAY .EQ. NNEDA) GO TO 2000
1750  CONTINUE
2000  CONTINUE
2005  READ (5,538) MS,MY,MD                                          54
2010  IF (MS+MY+MD) 2020,2020,100
2020  STOP
     END
```

APPENDIX B

Cosine table for use in the Astronomical Tide Program. The format is (12F6.5). The 1025 cosines are in order from 0° to 90° at intervals of 90/1024 degrees. The decimal points need not be punched; they are shown here for clarity.

COSINES

100000	.99999	.99999	.99999	.99998	.99997	.99996	.99994	.99992	.99990	.99988	.99986
.99983	.99980	.99977	.99974	.99970	.99966	.99962	.99958	.99953	.99948	.99943	.99938
.99932	.99927	.99920	.99914	.99908	.99901	.99894	.99887	.99879	.99872	.99864	.99856
.99848	.99839	.99830	.99821	.99812	.99802	.99793	.99782	.99772	.99762	.99751	.99740
.99729	.99717	.99706	.99694	.99682	.99670	.99657	.99644	.99631	.99618	.99604	.99591
.99577	.99562	.99548	.99534	.99518	.99503	.99488	.99472	.99457	.99441	.99424	.99408
.99391	.99373	.99357	.99339	.99321	.99303	.99284	.99266	.99248	.99228	.99210	.99191
.99170	.99151	.99131	.99110	.99090	.99070	.99048	.99027	.99006	.98984	.98962	.98940
.98917	.98895	.98873	.98849	.98826	.98803	.98778	.98754	.98731	.98705	.98681	.98657
.98630	.98605	.98580	.98553	.98528	.98502	.98474	.98448	.98422	.98393	.98366	.98339
.98310	.98283	.98255	.98225	.98197	.98168	.98138	.98109	.98079	.98048	.98019	.97987
.97957	.97926	.97894	.97863	.97832	.97799	.97768	.97736	.97702	.97670	.97638	.97603
.97570	.97537	.97502	.97468	.97435	.97398	.97364	.97330	.97293	.97258	.97223	.97186
.97150	.97115	.97076	.97040	.97004	.96965	.96928	.96891	.96852	.96814	.96776	.96736
.96698	.96660	.96618	.96580	.96541	.96499	.96459	.96420	.96377	.96337	.96297	.96253
.96213	.96169	.96128	.96086	.96042	.96000	.95958	.95913	.95870	.95827	.95781	.95738
.95695	.95648	.95604	.95560	.95513	.95468	.95424	.95376	.95330	.95285	.95236	.95191
.95144	.95095	.95049	.95002	.94952	.94905	.94857	.94807	.94759	.94711	.94659	.94611
.94562	.94510	.94461	.94411	.94359	.94309	.94259	.94205	.94155	.94104	.94050	.93999
.93948	.93893	.93841	.93789	.93733	.93681	.93625	.93572	.93519	.93462	.93409	.93355
.93298	.93244	.93189	.93131	.93076	.93021	.92963	.92907	.92852	.92792	.92736	.92680
.92620	.92563	.92505	.92445	.92388	.92330	.92269	.92211	.92153	.92091	.92032	.91973
.91910	.91851	.91792	.91728	.91668	.91608	.91544	.91483	.91423	.91358	.91297	.91235
.91170	.91108	.91046	.90980	.90917	.90855	.90788	.90725	.90657	.90594	.90530	.90462
.90398	.90334	.90265	.90201	.90136	.90066	.90001	.89936	.89866	.89800	.89734	.89663
.89596	.89530	.89458	.89391	.89324	.89252	.89184	.89116	.89043	.88975	.88906	.88833
.68764	.88695	.88621	.88551	.88481	.88407	.88337	.88266	.88191	.88120	.88049	.87973
.87902	.87830	.87753	.87681	.87609	.87532	.87457	.87387	.87309	.67235	.87162	.87083
.87010	.86931	.86856	.86782	.86702	.86627	.86552	.86472	.86397	.86321	.86240	.86164
.86088	.86006	.85930	.85853	.85771	.85694	.85616	.85533	.85456	.85378	.85294	.85216
.85137	.85053	.84974	.84895	.84810	.84731	.84651	.84566	.84486	.84405	.84319	.84239
.84158	.84071	.83990	.83908	.83821	.83739	.83657	.83569	.83487	.83404	.83316	.83233
.83150	.83061	.82977	.82888	.82804	.82719	.82629	.82545	.82460	.82369	.82284	.82199
.82108	.82022	.81936	.81844	.81758	.81672	.81579	.81492	.81405	.81312	.81225	.81137
.81044	.80956	.80868	.80773	.80685	.80596	.80501	.80412	.80323	.80227	.80138	.80048
.79952	.79862	.79771	.79675	.79584	.79493	.79396	.79305	.79213	.79116	.79024	.78932
.78833	.78741	.78648	.78550	.78457	.78364	.78264	.78171	.78071	.77977	.77883	.77782
.77688	.77594	.77492	.77398	.77303	.77201	.77106	.77010	.76908	.76812	.76716	.76613
.76516	.76420	.76316	.76219	.76122	.76018	.75921	.75823	.75719	.75621	.75522	.75417
.75319	.75220	.75115	.75015	.74916	.74810	.74711	.74611	.74504	.74404	.74304	.74196
.74096	.73995	.73887	.73786	.73685	.73577	.73475	.73373	.73264	.73162	.73053	.72950
.72848	.72738	.72635	.72532	.72422	.72318	.72214	.72104	.72000	.71895	.71784	.71680
.71575	.71463	.71358	.71253	.71141	.71035	.70929	.70817	.70711	.70604	.70491	.70385
.70278	.70164	.70057	.69950	.69835	.69728	.69620	.69505	.69398	.69289	.69174	.69066
.68957	.68841	.68732	.68623	.68507	.68397	.68288	.68171	.68061	.67951	.67834	.67723
.67613	.67495	.67384	.67266	.67155	.67044	.66925	.66813	.66702	.66582	.66470	.66358
.66238	.66126	.66013	.65893	.65780	.65667	.65546	.65433	.65319	.65198	.65084	.64971
.64849	.64735	.64620	.64498	.64383	.64268	.64146	.64031	.63915	.63792	.63677	.63561
.63437	.63321	.63205	.63081	.62965	.62848	.62723	.62606	.62489	.62364	.62247	.62130
.62004	.61886	.61763	.61643	.61524	.61398	.61280	.61161	.61035	.60916	.60797	.60669
.60550	.60431	.60303	.60183	.60064	.59936	.59815	.59695	.59567	.59446	.59326	.59197
.59076	.58955	.58825	.58704	.58582	.58453	.58331	.58209	.58079	.57957	.57834	.57704
.57581	.57458	.57327	.57204	.57081	.56950	.56827	.56703	.56571	.56447	.56324	.56191
.56067	.55943	.55810	.55686	.55561	.55428	.55303	.55178	.55044	.54919	.54785	.54660
.54534	.54400	.54274	.54148	.54013	.53887	.53761	.53626	.53499	.53372	.53237	.53110
.52983	.52847	.52720	.52592	.52456	.52328	.52200	.52064	.51936	.51807	.51670	.51542
.51413	.51276	.51147	.51018	.50881	.50751	.50622	.50484	.50354	.50225	.50086	.49956
.49826	.49688	.49557	.49427	.49288	.49157	.49027	.48887	.48756	.48625	.48485	.48354
.48214	.48083	.47951	.47811	.47679	.47547	.47406	.47274	.47142	.47000	.46868	.46735
.46594	.46461	.46328	.46186	.46053	.45920	.45778	.45645	.45511	.45369	.45235	.45101
.44958	.44824	.44690	.44547	.44412	.44278	.44134	.44000	.43865	.43721	.43586	.43451
.43307	.43172	.43036	.42892	.42756	.42621	.42476	.42340	.42204	.42059	.41923	.41787
.41641	.41505	.41369	.41223	.41086	.40940	.40803	.40666	.40520	.40383	.40245	.40099
.39961	.39824	.39677	.39539	.39402	.39254	.39116	.38978	.38831	.38693	.38554	.38407
.38268	.38129	.37982	.37843	.37704	.37556	.37417	.37277	.37129	.36990	.36850	.36701
.36562	.36422	.36273	.36133	.35993	.35844	.35704	.35564	.35414	.35274	.35133	.34984
.34843	.34702	.34552	.34412	.34271	.34120	.33979	.33829	.33688	.33546	.33395	.33254
.33113	.32962	.32820	.32678	.32527	.32385	.32243	.32092	.31949	.31807	.31656	.31513
.31371	.31219	.31076	.30934	.30782	.30639	.30496	.30344	.30201	.30058	.29905	.29762
.29619	.29466	.29322	.29179	.29026	.28882	.28729	.28585	.28442	.28298	.28144	.28000
.27856	.27702	.27558	.27414	.27260	.27116	.26971	.26817	.26673	.26528	.26374	.26229
.26075	.25930	.25785	.25630	.25485	.25340	.25185	.25040	.24895	.24740	.24595	.24449
.24294	.24149	.24003	.23848	.23702	.23556	.23401	.23255	.23109	.22953	.22807	.22661
.22505	.22359	.22213	.22057	.21910	.21764	.21608	.21461	.21315	.21158	.21012	.20865
.20709	.20562	.20415	.20258	.20112	.19965	.19808	.19661	.19514	.19357	.19209	.19062
.18905	.18758	.18601	.18453	.18306	.18149	.18001	.17853	.17696	.17548	.17401	.17243
.17095	.16947	.16790	.16642	.16494	.16336	.16188	.16040	.15882	.15734	.15586	.15428
.15280	.15131	.14973	.14825	.14676	.14518	.14370	.14221	.14053	.13914	.13766	.13607
.13459	.13310	.13151	.13003	.12854	.12695	.12546	.12398	.12239	.12090	.11941	.11782
.11633	.11484	.11325	.11176	.11027	.10868	.10719	.10560	.10411	.10262	.10102	.09953
.09804	.09645	.09495	.09346	.09187	.09037	.08888	.08729	.08579	.08430	.08270	.08121
.07971	.07812	.07662	.07513	.07353	.07203	.07054	.06894	.06745	.06595	.06435	.06285
.06136	.05976	.05826	.05677	.05517	.05367	.05217	.05057	.04908	.04758	.04598	.04448
.04298	.04138	.03989	.03839	.03679	.03529	.03379	.03219	.03069	.02909	.02759	.02609
.02449	.02299	.02149	.01990	.01840	.01690	.01530	.01380	.01230	.01070	.00920	.00770
.00610	.00460	.00310	.00150	.00000							