TECHNICAL MEMORANDUM (NASA) 72

DATA REDUCTION SOFTWARE FOR LORAN-C FLIGHT TEST EVALUATION

This paper describes a set of programs written for use on Ohio University's 370 computer for reducing and analyzing flight test data.

by

Joseph P. Fischer

Avionics Engineering Center
Department of Electrical Engineering
Ohio University
Athens, Ohio 45701

December 1979

Supported by

National Aeronautics and Space Administration
Langley Research Center
Langley Field, Virginia
Grant NGR 36-009-017

TABLE OF CONTENTS

		PAGE
I	INTRODUCTION	1
II	THE TAPE READER PROGRAM	1
III	SUPERVISOR PROGRAM	6
IV	SUBPROGRAM DBREAD	6
V	SUBPROGRAM TDPOS	10
VI	RANGE/AZIMUTH SUBPROGRAM	10
VII	PLOTTING	11
VIII	SUMMARY	11
IX	ACKNOWLEDGEMENTS	11
X	REFERENCES	13
XI	BIBLIOGRAPHY	13
APPENDIX I	LISTING FOR THE TAPE READER PROGRAM	14
APPENDIX II	LISTING FOR THE SUPERVISOR PROGRAM	23
APPENDIX III	LISTING FOR SUBPROGRAM DBREAD	26
APPENDIX IV	LISTING FOR SUBPROGRAM TDPOS	28
APPENDIX V	LISTING FOR BLOCK DATA CONSTANTS REQUIRED BY SUBPROGRAM TDPOS	35
APPENDIX VI	LISTING FOR SUBPROGRAM RNGAZ	37
APPENDIX VII	LISTING OF SAMPLE PLOTTING PROGRAM	39

I. INTRODUCTION

This paper describes a set of programs designed to be run on the IBM 370/158 computer at Ohio University. These programs are used to read the recorded time differences from the tape produced by the Loran data collection system[1], convert them to latitude/longitude and produce various plotting input files. The programs have been written so they may be tailored easily to meet the demands of a particular data reduction job. The tape reader program is written in 370 assembler language and the remaining programs are written in standard IBM FORTRAN-IV language. The tape reader program is dependent upon the recording format used by the data collection system and on the I/O macros used at the computing facility. The other programs are generally device-independent, although the plotting routines will be dependent upon the plotting method used.

Figure 1 shows an overall view of the flow of data from the receiver, through the Loran interface unit, to the microprocessor, and the main computing facility. The basic function of the data reduction programs is to convert the recorded data to a more readily usable form; convert the time difference (TD) numbers to latitude/longitude (lat/long), to format a printed listing of the TDs, lat/long, reference times, and other information derived from the data, and to produce data files which may be used for subsequent plotting.

II. THE TAPE READER PROGRAM

Figure 2 shows a functional flow diagram of the tape reader program. Its main function is to find and separate valid data from the tape input. The present format of the recorded tapes is a continuous string of numbers containing the GRI count, and the two TDs. A typical tape record is shown in Figure 3. Although the record format depicted in Figure 3 is eight words of ten bytes each, this is not always the case. Because of the recording operation of the microcomputer, the word positions may be offset to the left or right of the start-record position by a variable number of bytes. It is also possible that an incomplete word was recorded, i.e., one of the three fields (GRI or the TDs) may have been omitted for some reason (usually an interrupt timing problem). Completely invalid data may be included because of initialization problems. The recorded data is in a packed BCD format, as opposed to ASCII or EBCDIC which is more commonly used in the computer. The tape reader program, then, must correctly identify complete data fields, rejecting those which are incomplete or otherwise invalid. The program then converts the data from the packed decimal form to EBCDIC and stores it on a disk file and/or a tape volume. A running count is kept of errors encountered while reading the input tape.

For the discussion which follows, reference is made to the complete program listing in Appendix I. After reading a record from the tape input, a check is made to determine if any errors have been returned from the operating system. An end-of-file (EOF) indication is handled separately and causes the program to close all files and terminate execution (normal exit). If a permanent I/O error is encountered, the record is skipped and the next record is read. A count is maintained of all records skipped in this manner. There are several reasons for permanent I/O errors such as, improper inter-record gaps, or improper recording of the

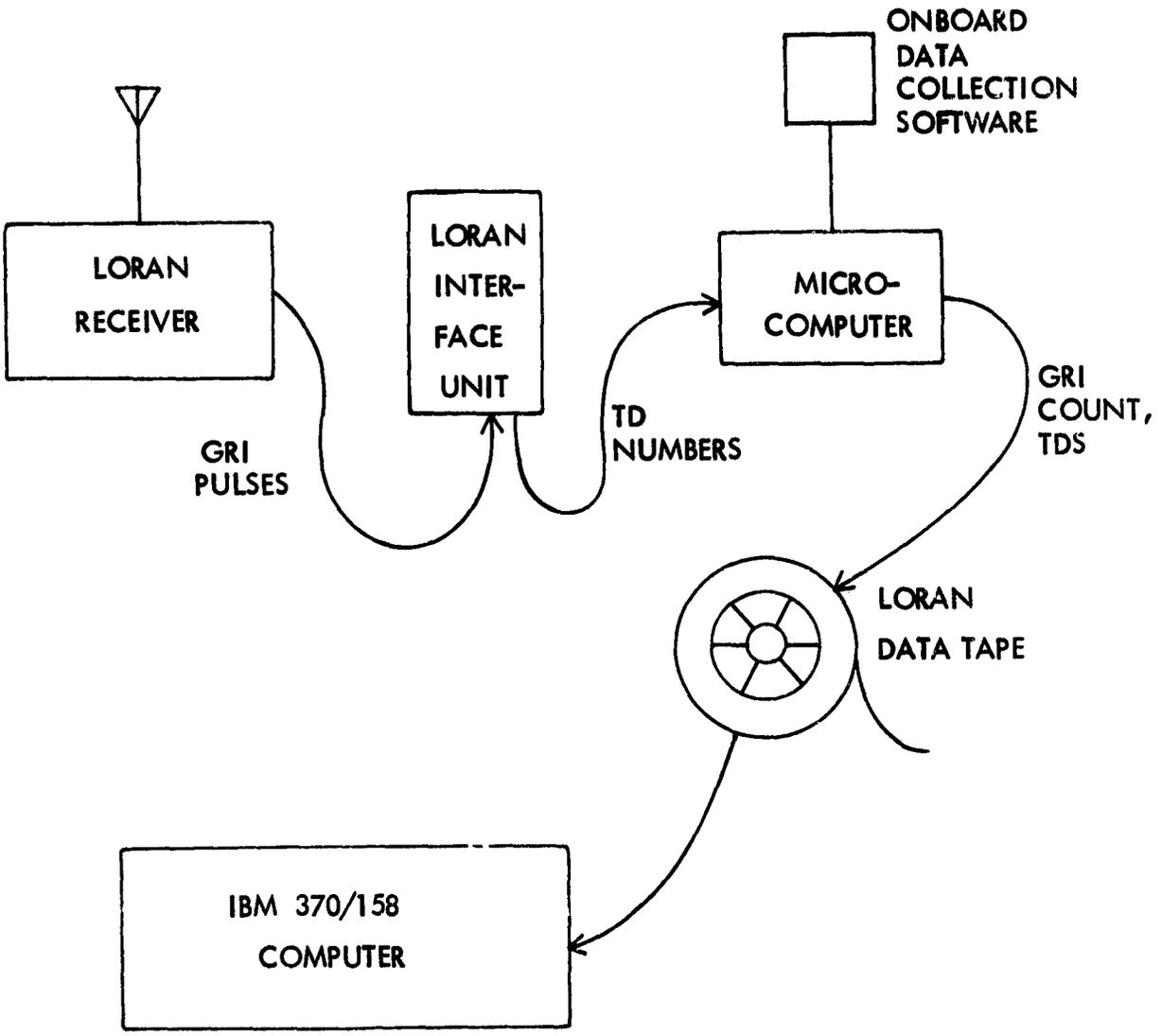

Figure 1. Flow of Data Through Loran Data Collection System.

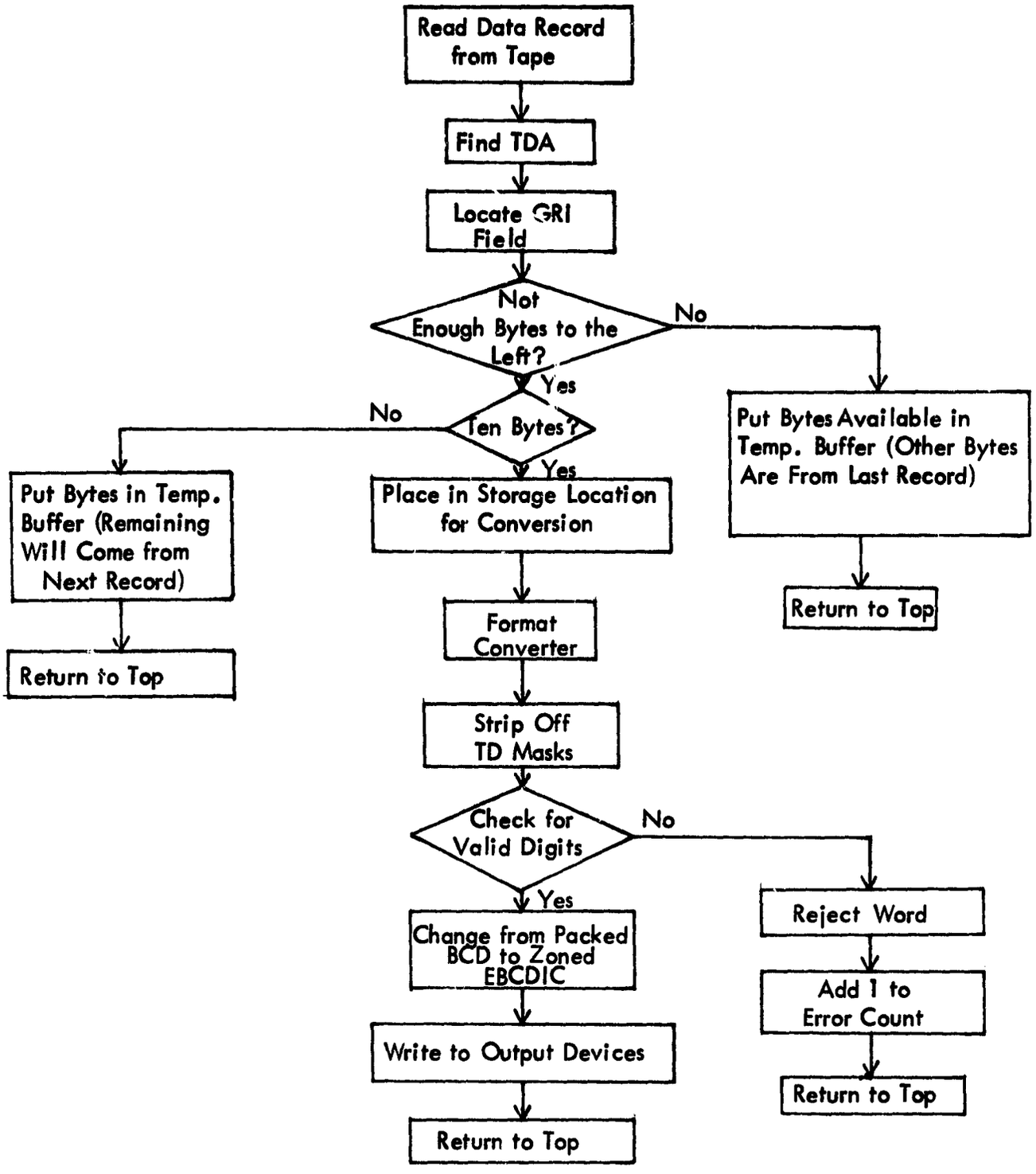

Figure 2. Flow Diagram for Tape Reader Program.

```
00000001A42596B5677700000002A42596B5677600000003A42597B5677800000004A42596B56776
00000005A42598B5677700000006A42597B5677600000007A42596B5677600000000A42596B56777
```

GRI | TDA | TDB

Figure 3. Example of GRI-TD Record as Found on Loran Data Tape.

record. The detection and handling of such errors are functions of the tape unit and tape channel used at the computing facility. Any other errors encountered cause an error message to be generated, all files closed, and execution to be terminated.

After a record has been correctly read, the program scans the record for the first occurrence of the TDA mask. Currently, this mask is a hexadecimal "A" in the upper four bits of the most-significant-digit (MSB) of the TDA field. When the TDA mask is found, the digit pointer backspaces four bytes to point to the first byte of the GRI count. Then a check is made for the TDB mask which is the hexadecimal character "B" in the upper four bits of the MSB for the TDB field. After this is verified, a check is made to ensure that there are ten digits in the word. If any of these checks fails, the word is skipped and the next word is tried. A separate count of these errors is kept.

The tape reader program must take into account the fact that part of a word may be on the previous line or on the succeeding line (tape record). If, while checking the word lengths, it is found that some of the first several bytes are missing, the program assumes that these bytes were on the last record. In this case, it assumes that these bytes are already in a temporary buffer which was built during processing of the previous record. Then the remaining bytes on the present record are used to fill out this temporary buffer which is then sent to the format conversion routine. If, while checking the word length, it is found that some of the bytes at the end of the word are missing, the bytes that are present are stored in a temporary buffer and another record is obtained, where it is assumed that the remaining bytes will be found.

When it has been ascertained that a proper ten-byte word exists, a subprogram is called which changes the packed decimal form to a zoned (EBCDIC) format. A check is made to be sure that ten bytes have actually been passed to the conversion routine; also the ten bytes are checked for valid characters. If these checks fail, the word is purged and the invalid word count is increased by one. Before the format conversion takes place, the two TD masks are stripped off (replaced by a four-bit zero). The format conversion is accomplished by taking each four-bit digit (20 in total) and prefixing it with the four-bit hexadecimal "F". Thus, the ten-byte word is converted to a 20-byte EBCDIC word.

The reformatted word is placed in an output buffer. The possible outputs are: a disk file with a dataset name specified when the tape reader program was called, and/or a tape volume. After the output record (s) are written, another pass is made to process another TD data word.

An additional feature built into the data recording system allows event-marks to be identified uniquely by writing an incrementing number into the most significant byte of the GRI count. For no events entered, this MSB digit is set to zero. The tape reader program tests the MSB of the GRI number; if it is non-zero, the entire TD word is converted to zoned format and stored on a separate disk file.

III. SUPERVISOR PROGRAM

The supervisor program is used during the TD-to-position conversion operation. Figure 4 shows a functional diagram of this program. A complete listing is given in Appendix II. The particular version of the supervisor to be discussed here was designed primarily for the CMS interactive system. To be used in other systems, such as OS/VS, some of the read and write statements and possibly some format statements need be changed. In order to facilitate use by other agencies, I/O operations are confined to this supervisor program; except for subroutine DBREAD, which does its own file-read operations.

The supervisor requests the user to enter parameters for the data reduction job. These parameters, in order of entry, are: the dead-reckoned latitude, dead-reckoned longitude, the number of input data points to be skipped between calculations, the basic Loran chain rate, the reference longitude. The dead-reckoned latitude and longitude are used to initialize the TD-to-position subprogram. The position may be approximate; within one to two degrees of the actual position. Since the conversion program requires a dead-reckoned position each time it is called, the last position calculated is used for the new dead-reckoned position. Because the input file containing the time differences can be quite large, it would take an excessive amount of time to convert every point. If the data is collected at the maximum rate, then there will be approximately ten points every second of real time. Under normal circumstances, one-tenth second represents a very small change in position; thus, the third entry to the supervisor program allows a number of input data points to skipped in between calculations, saving time. The maximum value for this entry is to do one calculation for every 9999th data point. The entry for basic chain rate is used for calculating the time between each set of data points. This is found by multiplying the GRI count read in from the data by the basic chain rate and dividing by 3600, giving the time in hours. This entry is made in decimal format; e.g. if the chain rate is 89700, enter 0.0897. This is the chain rate in fractional seconds. The final two entries are a reference position passed to the range/azimuth routine. Thus the range and azimuth of each calculated point may be found relative to the reference.

After the parameters are read in, the appropriate subprograms are called to read in a data point, convert it to lat/long, and convert it to range/azimuth. The first point is used as a time reference and the GRI count of each succeeding point is used to calculate the time elapsed since the first point. All of the information obtained by the supervisor program is formatted into a listing which is generally printed. Figure 5 shows a sample listing. A separate dataset file is also employed to write the range/azimuth or lat/long, which is then used as input to the plotting programs.

IV. SUBPROGRAM DBREAD

Subprogram DBREAD is used to obtain a GRI count and the two time differences from the input file. The operation of DBREAD is outlined in Figure 6. The complete listing is shown in Appendix III. The present version of this program reads from an 80-byte input file, each record containing four GRI and TD words. Several checks of the numbers read in are performed to minimize problems in other subprograms. One of these checks is a blunder point

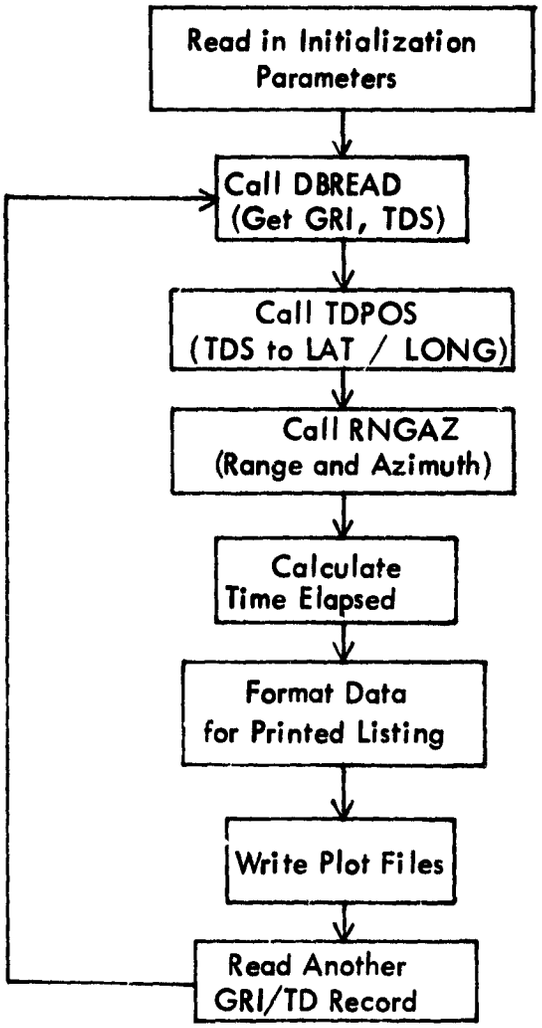

Figure 4. Flow Diagram for Supervisor Program.

PARAMETERS FOR LORAN DATA REDUCTION PROGRAM.

DEAD RECKONED LATITUDE: 33.3222
DEAD RECKONED LONGITUDE: 82.0875

DATA TAKEN EVERY 1 POINTS.

CHAIN RATE: 0.0996

REFERENCE LATITUDE: 39 13 22.91
REFERENCE LONGITUDE: 82 5 58.42

COUNT	TDA	TDB	LATITUDE DD MM SS	LONGITUDE DD MM SS	RANGE N.MILES	AZIMUTH DEGREES	TIME MM MM SS
1	42594.0	56794.0	39 19 11.99	82 3 54.06	1.6180	96.4288	0 0 0.1
2	42594.0	56793.0	39 19 12.69	82 4 1.53	1.5207	96.3917	0 0 0.2
3	42595.0	56795.0	39 19 17.99	82 3 [illegible]8.12	1.6738	92.7821	0 0 0.3
4	42594.0	56793.0	39 19 12.69	82 4 1.53	1.5207	96.3917	0 0 0.4
5	42596.0	56794.0	39 19 25.32	82 3 [illegible]8.17	1.5423	98.4231	0 0 0.5
6	42595.0	56793.0	39 19 19.39	82 4 4.06	1.4796	92.1917	0 0 0.6
7	42594.0	56793.0	39 19 12.69	82 4 1.53	1.5206	96.3574	0 0 0.7
8	42594.0	56794.0	39 19 11.99	82 3 54.06	1.6180	96.4288	0 0 0.8

Figure 5. Example of Printer Listing Produced by Loran Data Reduction Program.

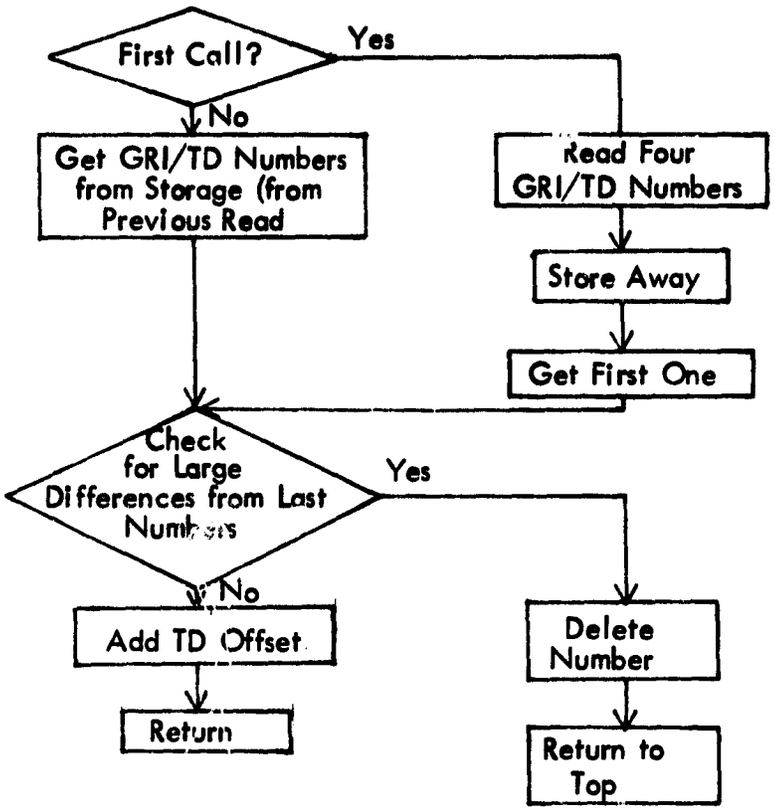

Figure 6. Flow Diagram for Subprogram DBREAD.

trap to delete a number if it differs significantly from the last number read in. Each TD count is held and compared to the new ones read in. If the difference is greater than five microseconds, the new number is deleted and the next number is tested. If the test fails ten consecutive times, the new number replaces the old number used for the comparisons. The purpose of this is to remove one TD reading which is off from the rest for some reason, but to be able to recover if there is a gap in the data because the data collection system was stopped briefly.

Other data checks include a provision to defeat the TD comparator on the first call to DBREAD. An offset may be added to the time differences if desired for analysis of tracking errors. When an end-of-file condition is detected on the input, the GRI count and the time differences are set to zero.

Care should be exercised if this program is modified since this program depends on values from previous calls to determine if any error conditions exist. For this reason, some variables in the program are initialized in a DATA statement and some are initialized in a arithmetic statement.

V. SUBPROGRAM TDPOS

Subprogram TDPOS is a subroutine which converts two time difference numbers to latitude and longitude. This program is a modification of an interactive program[2] already developed for TD conversion[3]. A listing of this program is given in Appendix IV. This program requires that the two TD numbers and the dead-reckoned position be included in the argument list each time it is called. The dead-reckoned position has been discussed briefly in the last section. The further off the dead-reckoned position is from the actual position, the more iterations are necessary to obtain a solution. If the dead-reckoned position is too far off, the calculations may diverge and result in no solution. The number of iterations is kept, and an error condition is noted if the number of iterations becomes too great. Any error conditions are returned in the IERROR argument. IERROR is positive for no error and negative if an error occurs. The computed position is returned in the argument POS.

The chain constants required by TDPOS are provided in a separate BLOCK DATA file. An example is given in Appendix V. Each such file contains the necessary chain data for the master and two selected secondary stations along with the semi-major and semi-minor radii of the earth.

VI. RANGE/AZIMUTH SUBPROGRAM

Subprogram RNGAZ is used to obtain the geodesic arc length and the bearing angle from a given position to a reference position. This program is based on an arc-length computational method used to calculate predicted Loran-C time differences[4]. A complete listing may be found in Appendix VI. The arc length is computed on a reference ellipsoid. This procedure incorporates similar corrections used for Loran-C; i.e., basic shape of the ellipsoid and propagation constants.

Double precision arithmetic is used in this subprogram to maintain accuracy at short arc-lengths. It was found in testing a similar program written in single-precision, that when the arc length was less than several miles, that some of the intermediate results suffered severely from round-off errors. As a result, underflow and overflow problems occurred when the program was run. With the double-precision method, it has been found that no such problems occur and the results are sufficiently accurate at arc-lengths down to one-half mile.

The results returned by this subprogram are in nautical miles for the arc-length and degrees for the bearing. If desired, other units may be easily obtained. Input to this program is the latitude and longitude in decimal degrees.

VII. PLOTTING

Plotting the results of a test flight provides a quick means of analyzing visually the data obtained rather than relying on the printed listing. The listing may be used for accurate point-by-point evaluation, if desired. This section will briefly discuss how the data reduction programs may be used for plotting the data. The plotting routines used in this case are contained in the standard Calcomp plotting package available to FORTRAN users. An example of a FORTRAN-IV program using thr range/angle information as input is shown in Appendix VII. A typical plot is shown in Figure 7. The plotting package is quite flexible and allows the user to write a plotting program to satisfy the requirements of the job at hand.

The data reduction programs can produce one or more compact data files which may be used as input for a plotter program. These files usually will only contain the computed latitude and longitude or the range and bearing. Thus these files may be considerably smaller in size than files used to produce them. This makes them easier to store and allow a number of plots on different scales to be produced using the same input file.

VIII. SUMMARY

The data reduction programs consist of several separate programs used for reading Loran data and producing more usable information. The tape reader program may be used separately to obtain formatted time difference numbers. The remaining programs convert the time differences to latitude/longitude and produce plotter input files. These programs have been written so that they may be modified easily to meet the demands of a particular data reduction job.

IX. ACKNOWLEDGEMENTS

These programs were written as an aid to flight test work using Loran-C navigation methods in general aviation. This work is being supported by NASA Grant NGR 36-009-017. Subprogram DBREAD is a modification of a similar program written by Dr. R.W. Lilley.

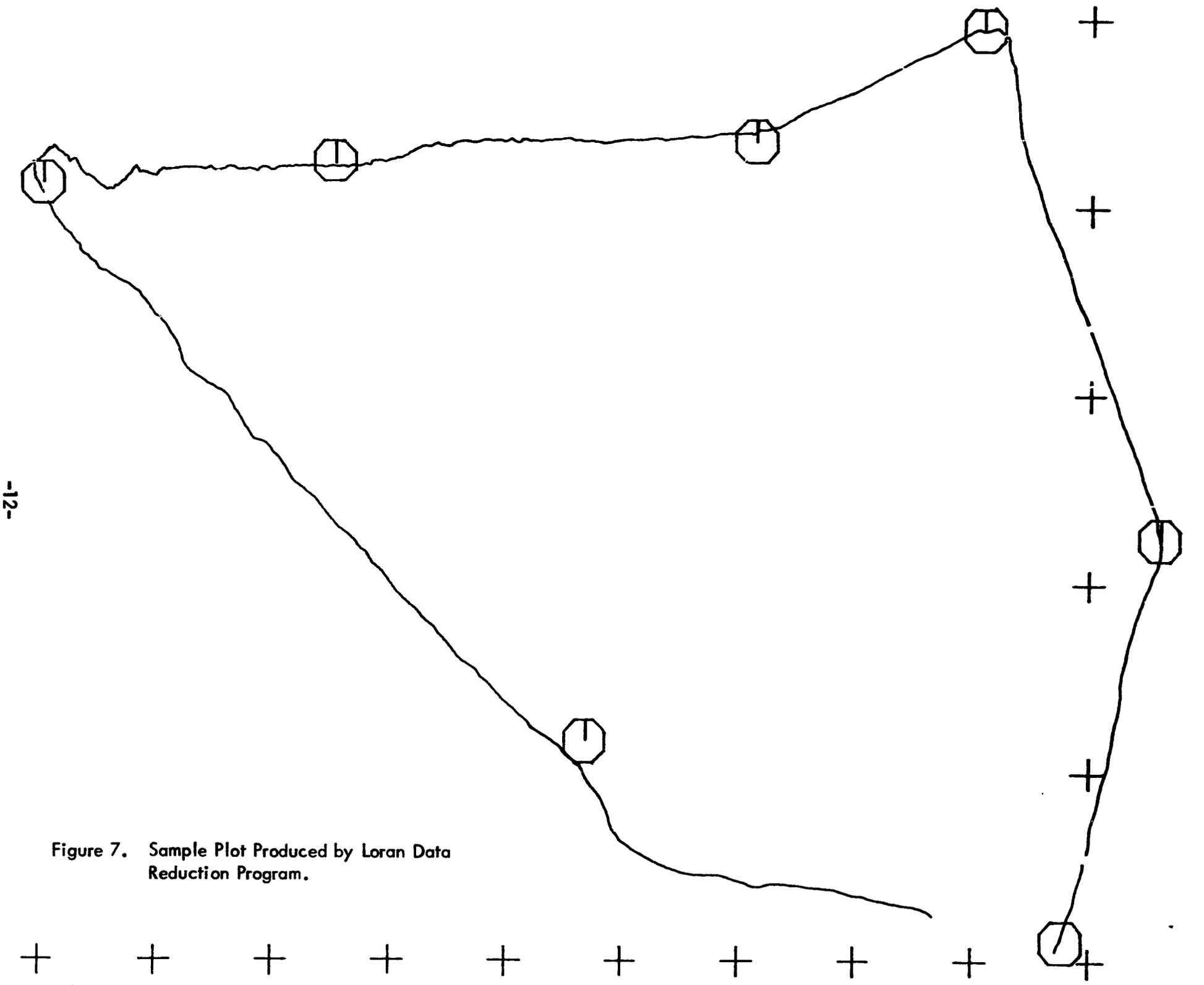

Figure 7. Sample Plot Produced by Loran Data Reduction Program.

X. REFERENCES

[1] Nickum, James D., "Loran-C Flight Test Software," NASA Technical Memorandum No. 61, Avionics Engineering Center, Department of Electrical Engineering, Ohio University, Athens, Ohio, August 1978.

[2] Piecuch, Lynn M. and R. W. Lilley, "Interactive Loran-C to Geographic and Geographic-to-Loran-C Computation," NASA TM 52, Avionics Engineering Center, Department of Electrical Engineering, Ohio University, Athens, Ohio, August 1977.

[3] "Loran-to-Geographic Conversion and Geographic-to-Loran Conversion," Informal Report N-3-64, Naval Oceanographic Office, Washington, D.C., June 1964.

[4] Kayton, Myron and Walter R. Fried, editors, "Avionics Navigation Systems," New York, John Wiley and Sons, inc., 1969, pp. 26-7.

XI. BIBLIOGRAPHY

The following publications may prove useful in explaining the computer languages used in this report and how to properly interface these programs to the operating system.

IBM System/360 and System/370 FORTRAN IV Language, GC28-6515-10, IBM Corporation, May 1974.

OS/VS-DOS/VS-VM/370 Assembler Language, GC33-4010-4, IBM Corporation, January 1975.

IBM System/370 Principles of Operation, GA22-7000-5, IBM Corporation, August 1976.

IBM Virtual Machine Facility/370: CMS Command and Macro Reference, GC20-1818-1, IBM Corporation, August 1977.

IBM Virtual Machine Facility/370: CMS User's Guide, GC20-1819-0, IBM Corporation, October 1976.

APPENDIX I Listing for the Tape Reader program.

```
         TITLE 'TAPUNF - UNFORMAT PROGRAM FOR LORAN-C DATA RECORDED ON *TAP00010
               TAPE.'                                                  TAP00020
         PRINT ON,GEN,NODATA                                           TAP00030
         SPACE 2                                                       TAP00040
*****************************************************************    TAP00050
*                                                               *      TAP00060
*                                                               *      TAP00070
*                                                               *      TAP00080
*        THIS PROGRAM IS DESIGNED TO READ IN LORAN DATA RECORDED *      TAP00090
*        ON MAGNETIC TAPE.  IT REMOVES THE TD MASKS AND CHANGES *      TAP00 00
*        THE DATA FROM PACKED TO ZONED (EBCDIC).  THE ZONED     *      TAP0 110
*        DATA MAY THEN BE STORED ON DISK, USING THE FILEID      *      TAP00120
*        SPECIFIED, AND/OR MAY BE STORED ON ANOTHER TAPE.       *      TAP00130
*        LORAN EVENT MARKS ARE REMOVED AND STORED ON DISK       *      TAP00140
*        WITH THE FILEID '(FN) EVENTS C1.'                      *      TAP00150
*                                                               *      TAP00160
*                                                               *      TAP00170
*                                                               *      TAP00180
*****************************************************************    TAP00190
         SPACE 3                                                       TAP00200
         LCLA  &RECLNG          DESIRED TAPE LENGTH                    TAP00210
&RECLNG  SETA  250                                                     TAP00220
TAPUNF   START X'20000'                                                TAP00230
         USING TAPUNF,12                                               TAP00240
         LR    12,15                                                   TAP00250
         LA    1,8(,1)          POINT PAST NAME                        TAP00260
         CLI   0(1),X'FF'       NOTHING SPECIFIED?                     TAP00270
         BE    CMDERROR         IF SO, THEN ERROR                      TAP00280
         CLI   0(1),C'('        OPTIONS DELIMITER?                     TAP00290
         BNE   *+8              SEE IF NODISK SPECIFIED,               TAP00300
         B     NODISKTS         SEE IF IT'S NODISK                     TAP00310
*                               IF NOT, THEN ERROR                     TAP00320
         CLI   8(1),X'FF'       SEE IF THERE IS A FILETYPE             TAP00330
         BE    CMDERROR         IF NOT, THEN ERROR                     TAP00340
         CLI   8(1),C'('        OPTIONS?                               TAP00350
         BNE   *+12             YES, CHECK FOR NODISK                  TAP00360
         LA    1,8(,1)          POINT TO OPTION (?)                    TAP00370
         B     NODISKTS         CHECK FOR "NODISK"                     TAP00380
         CLI   16(1),X'FF'      CHECK FOR FILEMODE LETTER              TAP00390
         BE    PUTINA           IF NONE, THEN SUBSTITUTE "A"           TAP00400
         CLI   16(1),C'('       OPTIONS?                               TAP00410
         BE    PUTINA           MOVE IN DISK "A"                       TAP00420
         MVC   DISKNAME+8(18),0(1) COMPLETE ID, MOVE TO PSCB           TAP00430
         MVC   EVNTCTL+8(8),0(1)   FN FOR EVENTS                       TAP00440
         MVI   DISKNAME+25,C' '    STICK IN ADDITIONAL SPACE           TAP00450
         LA    1,24(,1)         LOOK AT BEGINNING OF OPTIONS           TAP00460
         B     OPTCHECK         NOW CHECK FOR MODIFIERS                TAP00470
         SPACE                                                         TAP00480
PUTINA   MVC   DISKNAME+8(16),0(1) MOVE IN PARTIAL ID                  TAP00490
         MVC   EVNTCTL+8(8),0(1)   FN FOR EVENTS                       TAP00500
         MVI   DISKNAME+24,C'A'    MOVE IN "A"                         TAP00510
         MVI   DISKNAME+25,C' '    MOVE IN SPACE                       TAP00520
         LA    1,16(,1)         LOOK AT OPTIONS                        TAP00530
         SPACE                                                         TAP00540
OPTCHECK CLI   0(1),C'('        SEE IF THERE IS OPTION DELIM.          TAP00550
```

```
         BNE   BEGINEX         NO, BEGIN MAIN EXECUTION                    TAP00560
         LA    1,8(,1)         MOVE PAST OPEN PAREN.                       TAP00570
         SPACE 2                                                           TAP00580
*****************************************************************        TAP00590
*                                                               *        TAP00600
*        THE OPTIONS ALLOWED BY THIS PROGRAM ARE:  "IN" TO      *        TAP00610
*        CHANGE THE TAPE IN VADDR FROM THE DEFAULT 181 TO ANOTHER;*       TAP00620
*        "OUT" TO CHANGE THE TAPE OUT VADDR FROM THE DEFAULT    *        TAP00630
*        182 TO ANOTHER; "NODISK" TO SUPPRESS WRITING THE CONVERTED*      TAP00640
*        DATA ONTO A DISK BUFFER, THIS OPTION MAY ONLY BE USED IF *       TAP00650
*        "TAPE" IS SPECIFIED, EVENTS MARKS ARE STILL WRITTEN;   *        TAP00660
*        "TAPE" TO DUMP THE CONVERTED DATA ONTO TAPE AT ADDRESS *        TAP00670
*        182 OR THE MODIFIED ADDRESS.                           *        TAP00680
*                                                               *        TAP00690
*****************************************************************        TAP00700
         SPACE 2                                                           TAP00710
OPTLOOP  CLI   0(1),X'FF'      NOTHING THERE?                              TAP00720
         BE    BEGINEX         IF NOT, NO OPTIONS; RUN MAIN PROG.          TAP00730
         CLI   0(1),C')'       CLOSING PAREN.?                             TAP00740
         BE    BEGINEX         IF YES, START EXECUTION                     TAP00750
         CLC   0(8,1),OPTIN    SEE IF "IN" WAS SPECIFIED                   TAP00760
         BE    MVTPINAD        YES, MODIFY TAPE-IN VADDR                   TAP00770
         CLC   0(8,1),OPTOUT   SEE IF "OUT" SPECIFIED                      TAP00780
         BE    MVTPOUTD        YES, GO MODIFY TAPE-OUT VADDR               TAP00790
         CLC   0(8,1),OPTNDSK  SEE IF "NODISK" SPECIFIED                   TAP00800
         BE    NODSKPRC        SET FLAG SO WRITING TO DISK                 TAP00810
         CLC   0(8,1),OPTTAPE  SEE IF SECOND TAPE TO USE                   TAP00820
         BE    SETTPPL         IF YES, SET FLAG TO INDICATE SUCH           TAP00830
         SPACE                                                             TAP00840
*****************************************************************        TAP00850
*                                                               *        TAP00860
*              ONLY FOUR OPTIONS ALLOWED, IF THE OPTIONS        *        TAP00870
*              SPECIFIED FAILED THE ABOVE TESTS, THEN           *        TAP00880
*              WRITE AN ERROR MESSAGE AND LEAVE.                *        TAP00890
*                                                               *        TAP00900
*****************************************************************        TAP00910
         SPACE                                                             TAP00920
         LR    2,1             USE 2 SINCE GPR1 IS BOMBED                  TAP00930
         LINEDIT TEXT='DMSTPU001E INVALID OPTION ''........''',           *TAP00940
               SUB=(CHARA,(2)),RENT=NO,DOT=NO,DISP=ERRMSG                  TAP00950
         LA    15,20           ERROR LEVEL 20                              TAP00960
         BR    14              RETURN TO CMS                               TAP00970
         SPACE 2                                                           TAP00980
MVTPINAD CLI   8(1),C'0'       SEE IF VADDR CHANGE IS IN                   TAP00990
*                               NUMERIC OR CHARACTER FORMAT                TAP01000
         BL    MVTCH           IF CHARACTER, BRANCH                        TAP01010
*                                  MUST BE NUMERIC                        TAP01020
         MVC   DMS0087A-4+1(3),8(1)    MOVE IN VADDR                       TAP01030
         MVI   DMS0087A-4,C'0'         INNSERT ZERO                        TAP01040
         MVC   DMS0115A-16+1(3),8(1)   CHANGE VADDR                        TAP01050
         MVI   DMS0115A-16,C'0'        INSERT ZERO                         TAP01060
         LA    1,16(,1)        POINT TO NEXT OPTION                        TAP01070
         B     OPTLOOP         CHECK FOR MORE                              TAP01080
         SPACE                                                             TAP01090
MVTCH    MVC   DMS0087A-4(4),8(1)  CHANGE VADDR FOR TAPE IN                TAP01100
```

```
         MVC   DMS0115A-16(4),8(1) CHANGE VADDR FOR TAPE COMMAND          TAP01110
         LA    1,16(,1)          POINT TO NEXT OPTION (IF ANY)          TAP01120
         B     OPTLOOP           CHECK FOR MORE                         TAP01130
         SPACE 2                                                        TAP01140
MVTPOUTD CLI   8(1),C'0'         SEE IF NUMERIC OR CHAR.                TAP01150
         BL    MVTCHOUT          IF CHAR., BRANCH DOWN                  TAP01160
         MVC   DMS0202A-4+1(3),8(1)     CHANGE VADDR                    TAP01170
         MVI   DMS0202A-4,C'0'     INSERT ZERO                          TAP01180
         LA    1,16(,1)          POINT TO NEXT OPTION                   TAP01190
         B     OPTLOOP           CHECK FOR MORE                         TAP01200
         SPACE                                                          TAP01210
MVTCHOUT MVC   DMS0202A-4(4),8(1)  CHANGE VADDR FOR TAPE OUT            TAP01220
         LA    1,16(,1)          POINT TO NEXT OPTION (IF ANY)          TAP01230
         B     OPTLOOP           CHECK FOR MORE                         TAP01240
         SPACE 2                                                        TAP01250
NODSKPRC OI    FLAGS,1           SET FLAG SO NO DISK WRITING            TAP01260
         LA    1,8(,1)           POINT TO NEXT OPTION (IF ANY)          TAP01270
         B     OPTLOOP                                                  TAP01280
         SPACE 2                                                        TAP01290
SETTPFL  OI    FLAGS,4           SET TAPE FLAG                          TAP01300
         LA    1,8(,1)           POINT TO NEXT OPTION                   TAP01310
         B     OPTLOOP                                                  TAP01320
         SPACE 2                                                        TAP01330
NODISKTS OI    FLAGS,2           SET NO FILEID BIT                      TAP01340
         B     OPTCHECK                                                 TAP01350
         SPACE 2                                                        TAP01360
CMDERROR LINEDIT TEXT='DMSTPU002E INCOMPLETE FILEID SPECIFIED.',       *TAP01370
               DOT=NO,RENT=NO,DISP=ERRMSG                               TAP01380
         LA    15,24             ERROR LEVEL 24                         TAP01390
         BR    14                RETURN TO CMS                          TAP01400
         SPACE 2                                                        TAP01410
BEGINEX  TM    FLAGS,2           CHECK THE NO-ID BIT                    TAP01420
         BNO   RDYRUN            IF SET THEN NODISK BIT MUST BE SET     TAP01430
         TM    FLAGS,1           SEE IF IT IS                           TAP01440
         BO    RDYRUN            IF SET, OK                             TAP01450
         SPACE                                                          TAP01460
         LINEDIT TEXT='DMSTPU005E NO FILEID SPECIFIED, USE NODISK.',    *TAP01470
               DISP=ERRMSG,DOT=NO,RENT=NO                               TAP01480
         LA    15,4              ERROR LEVEL 4                          TAP01490
         BR    14                RETURN TO CMS                          TAP01500
         SPACE 2                                                        TAP01510
*****************************************************************      TAP01520
*                                                               *      TAP01530
*.       A TAPE RECORD IS READ IN AND ANY ERRORS ARE CHECKED.  IF *     TAP01540
*        AN "EOT" IS DETECTED, THE PROGRAM WRITES THE INFORMATIONAL *   TAP01550
*        MESSAGES AND LEAVES, IF A PERMANENT I/O ERROR IS DETECTED, *   TAP01560
*        THE BAD RECORD IS SKIPPED OVER AND THE NEXT RECORD IS TRYED. *  TAP01570
*        AFTER THE RECORD IS READ IN, THE TDA MASK IS SEARCHED FOR, *   TAP01580
*        AND WHEN FOUND, BACKSPACED TO THE FIRST BYTE OF THE GRI.  *    TAP01590
*        CHECKS ARE MADE TO BE SURE TEN BYTES ARE THERE, IF NOT, THE *  TAP01600
*        MISSING BYTES ARE OBTAINED FROM THE PREVIOUS RECORD, OR ADDED* TAP01610
*        TO THE NEXT RECORD.  THEN THE TEN BYTES OF DATA ARE CONVERTED* TAP01620
*        TO EBCDIC AND STORED.                                  *       TAP01630
*                                                               *      TAP01640
*****************************************************************      TAP01650
```

```
         SPACE 2                                                        TAP01660
RDYRUN   SLR   7,7             CLEAR GPR7                              TAP01670
         LA    6,DATACNVT      SET TO START OF CONVERSION AREA         TAP01680
         LA    8,DSKBF         INITIALIZE GPR8                         TAP01690
         LA    9,TPOUTBF       INITIALIZE GPR9                         TAP01700
         LA    5,TPINBF        SET ADDRESS OF START OF TAPE BUFFER     TAP01710
TPREAD   RDTAPE TPINBF,&RECLNG     READ A RECORD FROM TAPE              TAP01720
         LTR   15,15           CHECK THE RETURN CODE                   TAP01730
         BZ    TPCONT          IF ZERO, START BYTE MOVE OPERATION      TAP01740
         STH   15,R15RR        STORE RETURN CODE FOR TESTING           TAP01750
         CLI   R15RR+1,2       SEE IF EOT DETECTED                     TAP01760
         BE    EOT             YES, WRAP UP OPERATION                  TAP01770
         CLI   R15RR+1,3       SEE IF I/O ERROR                        TAP01780
         BE    TRYAGAIN        IF NOT, CAN'T CONTINUE                  TAP01790
         LR    2,15            SAVE RETURN CODE                        TAP01800
         LINEDIT TEXT='DMSTUP010E ERROR ON READING TAPE.',             *TAP01810
               DISP=ERRMSG,DOT=NO,RENT=NO                              TAP01820
         LR    15,2            GET RETURN CODE BACK...                 TAP01830
         BR    14              ...AND RETURN TO CMS                    TAP01840
         SPACE 2                                                       TAP01850
TRYAGAIN TAPECTL FSR           SKIP OVER BAD RECORD                    TAP01860
         AP    TPERRNBR(2),PONE    ADD ONE TO ERROR COUNT             TAP01870
         B     TPREAD          TRY TO READ ANOTHER RECORD              TAP01880
         SPACE 2                                                       TAP01890
TPCONT   LTR   7,7             TEST TO SEE IF FILL ON LEFT             TAP01900
         BZ    NOFILL          IF ZERO, NO FILL REQUIRED               TAP01910
FILL     MVC   0(1,6),0(5)     MOVE A BYTE TO CONVERT AREA             TAP01920
         LA    6,1(,6)         NEXT LOCATION IN CONVERT                TAP01930
         LA    5,1(,5)         NEXT LOCATION TO BE MOVED               TAP01940
         BCTR  7,0             ONE LESS TO DO                          TAP01950
         LTR   7,7             SEE IF IT IS ZERO                       TAP01960
         BZ    CONVERB         YES, COMPLETE FIELD, SO CONVERT         TAP01970
         B     FILL            OTHERWISE, DO IT AGAIN                  TAP01980
         SPACE                                                         TAP01990
NOFILL   LA    3,10(,5)        LOAD TEN LOCATIONS DOWN...              TAP02000
         CL    3,TPBFEND       ...AND SEE IF OUT-OF-BOUNDS             TAP02010
         BNH   NOEND           IF SMALLER, THEN OK                     TAP02020
         LA    7,10            TEN BYTE COUNT                          TAP02030
DROPRT   MVC   0(1,6),0(5)     MOVE IN A BYTE                          TAP02040
         LA    5,1(,5)         INCREMENT 5                             TAP02050
         LA    6,1(,6)         INCREMENT 6                             TAP02060
         BCTR  7,0             DECREMENT 7                             TAP02070
         CL    5,TPBFEND       SEE IF AT EDGE YET                      TAP02080
         BNP   DROPRT          NO, CONTINUE MOVING                     TAP02090
         LA    5,TPINBF        BACK TO BEGINNING                       TAP02100
         B     TPREAD          READ ANOTHER RECORD                     TAP02110
         SPACE                                                         TAP02120
NOEND    SLR   3,3             CLEAR 3 AGAIN                           TAP02130
         IC    3,0(,5)         LOAD IN A BYTE                          TAP02140
         SRL   3,4             GET RID OF LOWER FOUR BITS              TAP02150
         CH    3,AMASK         SEE IF TDA MASK IS THERE                TAP02160
         BE    FA              YES, FOUND IT                           TAP02170
         LA    5,1(,5)         NO, TRY NEXT BYTE                       TAP02180
         CL    5,TPBFEND       AT END OF RECORD?                       TAP02190
         BP    TPREAD-4        YES, GET ANOTHER RECORD                 TAP02200
```

```
         B     NOEND+2          GO THROUGH LOOP AGAIN                    TAP02210
         SPACE                                                           TAP02220
FA       SL    5,P4             POINT TO GRI FIELD                       TAP02230
         MVC   0(10,6),0(5)     MOVE THE FIELD TO CONVERT AREA           TAP02240
         LA    6,DATACNVT+10    DO THIS FOR TEST LATER ON                TAP02250
         LA    5,10(,5)         NEXT FIELD TO BE MOVED                   TAP02260
         SPACE 2                                                         TAP02270
*****************************************************************      TAP02280
*                                                               *      TAP02290
*        CONVERSION IS DONE ONE BYTE AT A TIME.  THE BYTES      *      TAP02300
*        ARE CHECKED TO BE SURE THEY ARE IN THE RANGE OF        *      TAP02310
*        ZERO THROUGH NINE, INCLUSIVE.  IF THEY ARE NOT, THE    *      TAP02320
*        ENTIRE DATA RECORD IS REJECTED.                        *      TAP02330
*                                                               *      TAP02340
*****************************************************************      TAP02350
         SPACE 2                                                         TAP02360
CONVRBB  CL    6,DATACNVE       AT END OF CONVERT BUFFER                 TAP02370
         BNE   BADRECRD         IF NOT, BAD RECORD DETECTED              TAP02380
         LA    6,DATACNVT       POINT TO BEGINNING                       TAP02390
         IC    3,4(,6)          TEST FOR TDA MASK                        TAP02400
         SRL   3,4              STRIP OFF LOWER BITS                     TAP02410
         CH    3,AMASK          IS "A"?                                  TAP02420
         BNE   BADRECRD         NO, ERROR                                TAP02430
         IC    3,7(,6)          TEST FOR TDB MASK                        TAP02440
         SRL   3,4              STRIP OFF LOWER BITS                     TAP02450
         CH    3,BMASK          IS "B"?                                  TAP02460
         BNE   BADRECRD         NO, ERROR                                TAP02470
         NI    4(6),X'0F'       STRIP OFF TDA MASK                       TAP02480
         NI    7(6),X'0F'       STRIP OFF TDB MASK                       TAP02490
         SPACE                                                           TAP02500
         LA    10,CONVERT       START AT BEGINNING                       TAP02510
CVZONE   IC    3,0(,6)          GET A BYTE                               TAP02520
         SRL   3,4              SHIFT OFF LOWER FOUR BITS                TAP02530
         CH    3,H0             SEE IF NUMBER LESS THAN "0"              TAP02540
         BM    BADRECRD         IF YES, REJECT RECORD                    TAP02550
         CH    3,H9             SEE IF NUMBER GREATER THAN "9"           TAP02560
         BH    BADRECRD         IF YES, REJECT RECORD                    TAP02570
         O     3,ZONEMASK       ADD THE ZONE MASK                        TAP02580
         STC   3,0(,10)         AND PLACE IN CONVERT BUFFER              TAP02590
         LA    10,1(,10)        NEXT BYTE IN BUFFER                      TAP02600
         IC    3,0(,6)          GET THE SAME CHARACTER                   TAP02610
         N     3,STRIPU4        STRIP OFF THE UPPER FOUR BITS            TAP02620
         CH    3,H0             CHECK IF VALID                           TAP02630
         BM    BADRECRD         REJECT RECORD IF NOT                     TAP02640
         CH    3,H9             CHECK IF VALID                           TAP02650
         BH    BADRECRD         REJECT IF NOT                            TAP02660
         O     3,ZONEMASK       ADD THE ZONE MASK                        TAP02670
         STC   3,0(,10)         STORE IN BUFFER                          TAP02680
         LA    10,1(,10)        NEXT BYTE                                TAP02690
         LA    6,1(,6)          NEXT BYTE TO BE DONE                     TAP02700
         CL    6,DATACNVE       SEE IF DONE                              TAP02710
         BNP   CVZONE           NO, CONTINUE THIS LOOP                   TAP02720
         LA    6,DATACNVT       LOAD GPR6                                TAP02730
         SPACE                                                           TAP02740
         TM    DATACNVT,X'FF'       CHECK IF FIRST NUMBER IS ZERO        TAP02750
```

```
         BZ    SEEDSK          IF ZERO, NO EVENT ENTERED                    TAP02760
         FSWRITE FSCB=EVNTCTL  PUT EVENT ON DISK                            TAP02770
         LTR   15,15           CHECK THE RETURN CODE                        TAP02780
         BZ    NOFILL          IF ZERO, DO NEXT RECORD                      TAP02790
         SPACE                                                              TAP02800
         LR    2,15            SAVE RETURN CODE                             TAP02810
         LINEDIT TEXT='DMSTPU021E ERROR ON WRITING TO ''LORAN EVENTS C1*TAP02820
               ''',DISP=ERRMSG,DOT=NO,RENT=NO                               TAP02830
         LR    15,2            GET RETURN CODE BACK                         TAP02840
         BR    14              AND RETURN TO CMS                            TAP02850
         SPACE 2                                                            TAP02860
*****************************************************************************TAP02870
*                                                                           * TAP02880
*        THE WRITING TO DISK AND/OR TAPE IS DONE HERE.  THE                 * TAP02890
*        FORMATS USED ARE FOR DISK:  RECFM = F, LRECL = 80,                 * TAP02900
*        BLKSIZE = 80.  FOR TAPE:  RECFM = U, BLKSIZE = 800.                * TAP02910
*                                                                           * TAP02920
*****************************************************************************TAP02930
         SPACE 2                                                            TAP02940
SEEDSK   TM    FLAGS,1         SEE IF WRITING TO DISK                       TAP02950
         BO    SEETP           IF NOT, CHECK FOR TAPE                       TAP02960
         MVC   0(20,8),CONVERT     MOVE THE NUMBER TO DISK BUFFER           TAP02970
         LA    8,20(,8)        NEXT FIELD AREA                              TAP02980
         CL    8,DSKBFEND      CHECK IF BUFFER IS FULL                      TAP02990
         BL    SEETP           IF NOT, GO ON                                TAP03000
         LA    8,DSKBF         RE-INITIALIZE GPR8                           TAP03010
         FSWRITE FSCB=DISKNAME WRITE TO DISK                                TAP03020
         LTR   15,15           CHECK RETURN CODE                            TAP03030
         BZ    SEETP           IF ZERO, GO ON                               TAP03040
         SPACE                                                              TAP03050
         LR    2,15            SAVE RETURN CODE                             TAP03060
         LINEDIT TEXT='DMSTPU020E ERROR ON WRITING TO ''..... .....*TAP03070
               .''',SUB=(CHARA,DISKNAME+8,CHARA,DISKNAME+16,CHARA,DISK*TAP03080
               NAME+24),DISP=ERRMSG,DOT=NO,RENT=NO                          TAP03090
         LR    15,2            GET RETURN CODE BACK                         TAP03100
         BR    14              RETURN TO CMS                                TAP03110
         SPACE 2                                                            TAP03120
SEETP    TM    FLAGS,4         CHECK IF TAPE BEING USED                     TAP03130
         BZ    NOFILL                                                       TAP03140
         MVC   0(20,9),CONVERT     MOVE DIGITS TO TAPE BUFFER               TAP03150
         LA    9,20(,9)        POINT TO NEXT FIELD                          TAP03160
         CL    9,TPOUTEND      SEE IF BUFFER FULL                           TAP03170
         BNH   NOFILL          IF NOT, PROCESS ANOTHER FIELD                TAP03180
         LA    9,TPOUTBF       POINT BACK TO BEGINNING                      TAP03190
         WRTAPE (9),800,182    DUMP BUFFER TO TAPE                          TAP03200
         LTR   15,15           CHECK RETURN CODE FROM DUMP                  TAP03210
         BZ    NOFILL          IF ZERO, PROCESS ANOTHER RECORD              TAP03220
         LR    2,15            SAVE RETURN CODE                             TAP03230
         SPACE                                                              TAP03240
         LINEDIT TEXT='DMSTPU022E ERROR ON WRITING TO TAPE.',          *TAP03250
               DISP=ERRMSG,DOT=NO,RENT=NO                                   TAP03260
         SPACE                                                              TAP03270
         LR    15,2            GET RETURN CODE BACK                         TAP03280
         BR    14              GO TO  CMS                                   TAP03290
         SPACE 3                                                            TAP03300
```

```
BADRECRD LA    6,DATACNVT      POINT TO BEGINNING OF CONVERT                TAP03310
         AP    RECERROR(2),PONE   ADD ONE TO ERROR COUNT                    TAP03320
         B     NOFILL          TRY NEXT FIELD                               TAP03330
         SPACE 2                                                            TAP03340
*********************************************************************      TAP03350
*                                                                   *      TAP03360
*        THE PROGRAM COMES HERE AFTER READING A FILE MARK           *      TAP03370
*        ON THE TAPE.  THE COUNTS FOR TAPE RECORDS SKIPPED          *      TAP03380
*        AND DATA RECORDS REJECTED ARE UNPACKED AND DISPLAYED       *      TAP03390
*        ON THE TERMINAL.  THEN, CONTROL IS RETURNED TO CMS.        *      TAP03400
*                                                                   *      TAP03410
*********************************************************************      TAP03420
         SPACE 2                                                            TAP03430
EOT      UNPK  MSG1+23(3),TPERRNBR(2)   CHANGE TO EBCDIC                    TAP03440
         OI    MSG1+25,X'F0'  REMOVE SIGN                                   TAP03450
         UNPK  MSG2+23(3),RECERROR(2)   CHANGE TO EBCDIC                    TAP03460
         OI    MSG2+25,X'F0'  REMOVE SIGN                                   TAP03470
         WRTERM ' '           CARRIAGE RETURN                              TAP03480
         WAITT                                                              TAP03490
         WRTERM MSG1,26                                                     TAP03500
         WAITT                                                              TAP03510
         WRTERM MSG2,26                                                     TAP03520
         WAITT                                                              TAP03530
         SLR   15,15           ZERO RETURN CODE                             TAP03540
         BR    14              BACK TO CMS                                  TAP03550
         EJECT                                                              TAP03560
         DS    0D                                                           TAP03570
OPTIN    DC    CL8'IN'                                                      TAP03580
OPTOUT   DC    CL8'OUT'                                                     TAP03590
OPTNDSK  DC    CL8'NODISK'                                                  TAP03600
OPTTAPE  DC    CL8'TAPE'                                                    TAP03610
F4       DC    F'4'                                                         TAP03620
ZONEMASK DC    H'0',X'00F0'                                                 TAP03630
STRIP04  DC    H'0',X'000F'                                                 TAP03640
DATACNVE DC    AL4(DATACNVT+10)                                             TAP03650
TPBFEND  DC    AL4(TPINBF+&RECLNG)                                          TAP03660
DSKBFEND DC    AL4(DSKBF+80)                                                TAP03670
TPOUTEND DC    AL4(TPOUTBF+800)                                             TAP03680
DISKNAME FSCB  '      ',RECFM=F,BUFFER=DSKBF,BSIZE=80                       TAP03690
EVNTCTL  FSCB  'LORAN EVENTS C1',RECFM=F,BUFFER=CONVERT,BSIZE=20            TAP03700
E15RR    DS    H                                                            TAP03710
H0       DC    H'0'                                                         TAP03720
H9       DC    H'9'                                                         TAP03730
AMASK    DC    X'000A'                                                      TAP03740
BMASK    DC    X'000B'                                                      TAP03750
FLAGS    DC    X'00'                                                        TAP03760
CONVERT  DS    21X                                                          TAP03770
TPINBF   DS    &RECLNG.X                                                    TAP03780
TPOUTBF  DS    800X                                                         TAP03790
DSKBF    DS    80X                                                          TAP03800
DATACNVT DC    10C' '                                                       TAP03810
         DS    10X                                                          TAP03820
TPERRNBR DC    PL2'0'                                                       TAP03830
RECERROR DC    PL2'0'                                                       TAP03840
PONE     DC    P'1'                                                         TAP03850
```

```
FILE: TAPUNP    ASSEMBLE C              OHIO UNIVERSITY AVIONICS ENGINEERING CENTER

MSG1      DC    C'TAPE RECORDS REJECTED:    '                          TAP03860
MSG2      DC    C'DATA RECORDS REJECTED:    '                          TAP03870
          END   TAPUNP                                                 TAP03880
```

APPENDIX II Listing for the Supervisor program.

```
      DIMENSION TDS(2),POS(2),IPOS(2),FPOS(2),APOS(2),MPOS(2),SPOS(2),DRLRN00010
     :POS(2)                                                          LRN00020
      REAL*8 RLAT,RLONG,PI                                            LRN00030
      COMMON/RNGPOS/RLAT,RLONG                                        LRN00040
      DATA PI/3.14159265358979/                                       LRN00050
      TIME=0.0                                                        LRN00060
      GRI$=0.0                                                        LRN00070
      READ 3,IDEG,MIN,SEC,IDEG1,MIN1,SEC1                            LRN00080
      DRPOS(1)=IDEG+(MIN+SEC/60.)/60.                                 LRN00090
      DRPOS(2)=IDEG1+(MIN1+SEC1/60.)/60.                              LRN00100
      READ 2,LOOPS                                                    LRN00110
      READ 1,CRR                                                      LRN00120
      READ 3,IDEG,MIN,SEC,IDEG1,MIN1,SEC1                            LRN00130
      RLAT=(IDEG+(MIN+SEC/60.)/60.)*PI/180.                           LRN00140
      RLONG=(IDEG1+(MIN1+SEC1/60.)/60.)*PI/180.                       LRN00150
      PRINT 9,DRPOS,LOOPS,CRR,IDEG,MIN,SEC,IDEG1,MIN1,SEC1           LRN00160
      PRINT 8                                                         LRN00170
      CALL DRREAD(GRI,TDS)                                            LRN00180
      IF(GRI.EQ.0.)GOTO 100                                           LRN00190
      ICOUNT=1                                                        LRN00200
      GOTO 14                                                         LRN00210
   10 DO 13 I=1,LOOPS                                                 LRN00220
      CALL DRREAD(GRI,TDS)                                            LRN00230
      IF(GRI.EQ.0.0)GOTO 100                                          LRN00240
   13 CONTINUE                                                        LRN00250
      ICOUNT=ICOUNT+I                                                 LRN00260
   14 GRII=GRI-GRI$                                                   LRN00270
      GRI$=GRI                                                        LRN00280
      TIME=TIME+GRII*CRR/3600.0                                       LRN00290
      CALL TDPOS(TDS,POS,DRPOS,IERROR)                                LRN00300
      IF(IERROR)111,110,110                                           LRN00310
  110 DO 21 I=1,2                                                     LRN00320
      DRPOS(I)=POS(I)                                                 LRN00330
      IPOS(I)=POS(I)                                                  LRN00340
      FPOS(I)=POS(I)-IPOS(I)                                          LRN00350
      APOS(I)=FPOS(I)*60.0                                            LRN00360
      MPOS(I)=APOS(I)                                                 LRN00370
   21 SPOS(I)=(APOS(I)-MPOS(I))*60.0                                  LRN00380
      IHR=TIME                                                        LRN00390
      XMIN=(TIME-IHR)*60.0                                            LRN00400
      MIN=XMIN                                                        LRN00410
      SCS=(XMIN-MIN)*60.0                                             LRN00420
      CALL RNGAZ(POS,RHO,AZIM)                                        LRN00430
      PRINT 5,ICOUNT,TDS,(IPOS(I),MPOS(I),SPOS(I),I=1,2),RHO,AZIM,IHR,MILRN00440
     >N,SCS                                                           LRN00450
      WRITE(11,6)RHO,AZIM                                             LRN00460
      GOTO 10                                                         LRN00470
  111 PRINT 7,TDS                                                     LRN00480
      GOTO 10                                                         LRN00490
  100 STOP                                                            LRN00500
    1 FORMAT(F10.0)                                                   LRN00510
    2 FORMAT(I4)                                                      LRN00520
    3 FORMAT(I4,1X,I2,1X,F5.2)                                        LRN00530
    5 FORMAT(1X,I5,8X,2(F7.1,8X),2(I4,1X,I2,1X,F5.2,8X),2(F8.4,8X),2(I2,LRN00540
     >1X),F4.1)                                                       LRN00550
```

```
    6 FORMAT(2(F8.4))                                                    LRN00560
    7 FORMAT('0 TDA = ',F7.1,8X,'TDB = ',F7.1,T70,15('*'),'  SOLUTION CALRN00570
     :NNOT BE OBTAINED.  ',15('*'))                                      LRN00580
    8 FORMAT(1H1,' COUNT',T15,'TDA',T30,'TDB',T47,'LATITUDE',T68,'LONGITLRN00590
     >UDE',T87,'RANGE',T103,'AZIMUTH',T120,'TIME'/1X,T47,'DD MM SS',T68,LRN00600
     >'DD MM SS',T87,'N.MILES',T103,'DEGREES',T119,'HH MM SS'/)          LRN00610
    9 FORMAT('1   PARAMETERS FOR LORAN DATA REDUCTION PROGRAM.'///'0DEAD LRN00620
     >RECKONED LATITUDE: ',F9.4/' DEAD RECKONED LONGITUDE: ',F8.4///' DATLRN00630
     >A TAKEN EVERY ',I4,' POINTS.'///' CHAIN RATE: ',F6.4///' REFERENCE LLRN00640
     >ATITUDE: ',I5,1X,I2,1X,F5.2/' REFERENCE LONGITUDE: ',I4,1X,I2,1X,FLRN00650
     >5.2)                                                               LRN00660
      END                                                                LRN00670
```

APPENDIX III Listing for subprogram DBREAD.

```
      SUBROUTINE DBREAD(GRI,TDS)                                        DBR00010
      DIMENSION TDS(2),IBUF(4,3),TDOLD(2),TDOF(2)                       DBR00020
      DATA IPASS/4/,TDOLD/0.,0./,ICALL/0/,TDOF/-2.,17./                 DBR00030
      NERR=0                                                            DBR00040
    7 IF(IPASS.NE.4)GOTO 2                                              DBR00050
      READ(10,40,END=5)((IBUF(I,J),J=1,3),I=1,4)                        DBR00060
      IPASS=0                                                           DBR00070
    2 IPASS=IPASS+1                                                     DBR00080
      GRI=IBUF(IPASS,1)                                                 DBR00090
      IF(GRI.EQ.0.)GRI=1.                                               DBR00100
      DO 11 I=1,2                                                       DBR00110
   11 TDS(I)=IBUF(IPASS,I+1)+TDOF(I)                                    DBR00120
      IF(ICALL)9,10,9                                                   DBR00130
   10 ICALL=1                                                           DBR00140
      DO 12 I=1,2                                                       DBR00150
   12 TDOLD(I)=TDS(I)                                                   DBR00160
    9 IF(ABS(TDOLD(1)-TDS(1)).GT.5..OR.ABS(TDOLD(2)-TDS(2)).GT.5.)GOTO13DBR00170
      DO 8 I=1,2                                                        DBR00180
    8 TDOLD(I)=TDS(I)                                                   DBR00190
      RETURN                                                            DBR00200
    5 GRI=0.0                                                           DBR00210
      DO 6 I=1,2                                                        DBR00220
    6 TDS(I)=0.0                                                        DBR00230
      RETURN                                                            DBR00240
   13 NERR=NERR+1                                                       DBR00250
      IF(NERR.LT.10)GOTO7                                               DBR00260
      NERR=0                                                            DBR00270
      ICALL=0                                                           DBR00280
      GOTO 7                                                            DBR00290
   40 FORMAT(4(I8,I6,I6))                                               DBR00300
      END                                                               DBR00310
```

APPENDIX IV Listing for subprogram TDPOS.

```
      SUBROUTINE TDPOS(TH,POS,DRPOS,IERROR)                              TDP00010
      DIMENSION TQSV(2),QSV(2),ANG(8),AD(8),DM(8),CS(8),POS(2),ZIWD(2),FTDP00020
     :WM(2),DEL(2),IDR(2),DRD(2),DRM(2),TH(2),A(11),B(11),C(11),D(11),E(TDP00030
     :11),CC(11),TM(2),BLEM(2),BEDEL(2),RADR(2),BETA(2),OMG(2),IWD(2),IDTDP00040
     :RD(2),DRPOS(2)                                                    TDP00050
C -                                                                     TDP00060
      DATA A1/24.0305/,A2/-0.40758/,A3/3.46776E-3/,B1/0.510483/,B2/-0.01TDP00070
     :1402/,B3/0.001760/,RD/1.745329E-2/,RM/2.908882E-4/,RS/4.848137E-6/TDP00080
     :,PI/3.141592/,A4/2.996912E2/                                      TDP00090
C -                                                                     TDP00100
      COMMON/CHAIND/DEL,A5,A6,AD,DM,CS                                  TDP00110
C -                                                                     TDP00120
C - BEGIN TIME DIFFERENCE TO POSITION CONVERSION.                       TDP00130
C -                                                                     TDP00140
      DO 1 I=1,2                                                        TDP00150
      IDRD(I)=DRPOS(I)                                                  TDP00160
      DRD(I)=IDRD(I)                                                    TDP00170
    1 DRM(I)=(DRPOS(I)-DRD(I))*60.0                                     TDP00180
      IERROR=1                                                          TDP00190
      A10=(A5*A5-A6*A6)/(A5*A5)                                         TDP00200
      A14=1.0-A6/A5                                                     TDP00210
      A50=(1.0+A14+A14*A14)                                             TDP00220
      A51=(A50-1.0)                                                     TDP00230
      A52=(A14*A14)/2.0                                                 TDP00240
      A53=-A51/2.0                                                      TDP00250
      A54=(A14*A14)/16.0                                                TDP00260
      A55=(A14*A14)/8.0                                                 TDP00270
      A56=A14*A14                                                       TDP00280
      A57=A56*1.25                                                      TDP00290
      A58=A56/4.0                                                       TDP00300
      DO 128 K=1,8                                                      TDP00310
      IF(AD(K))124,126,126                                              TDP00320
  124 ANG(K)=RD*AD(K)-RM*DM(K)-RS*CS(K)                                 TDP00330
      GO TO 128                                                         TDP00340
  126 ANG(K)=RD*AD(K)+RM*DM(K)+RS*CS(K)                                 TDP00350
  128 CONTINUE                                                          TDP00360
      A12=(ANG(1)-ANG(5)+ANG(2)-ANG(6))                                 TDP00370
      A12=ABS(A12)                                                      TDP00380
      IF(A12-0.00001)7,7,8                                              TDP00390
    7 A11=-1.0                                                          TDP00400
      GO TO 9                                                           TDP00410
    8 A11=1.0                                                           TDP00420
C -                                                                     TDP00430
C - APPROXIMATE POSITIONS AND STATION COORDINATES.                      TDP00440
C -                                                                     TDP00450
    9 E(1)=ANG(1)                                                       TDP00460
      E(2)=ANG(2)                                                       TDP00470
      CC(1)=ANG(3)                                                      TDP00480
      CC(2)=ANG(4)                                                      TDP00490
      E(3)=SIN(E(1))                                                    TDP00500
      E(4)=COS(E(1))                                                    TDP00510
      E(5)=E(3)/E(4)                                                    TDP00520
      E(8)=(E(5))*(1.0-A14)                                             TDP00530
      A62=ATAN(E(8))                                                    TDP00540
      E(6)=SIN(A62)                                                     TDP00550
```

```
      E(7)=COS(A62)                                                    TDP00560
      CC(3)=SIN(CC(1))                                                 TDP00570
      CC(4)=COS(CC(1))                                                 TDP00580
      CC(5)=CC(3)/CC(4)                                                TDP00590
      CC(8)=(CC(5))*(1.0-A14)                                          TDP00600
      A62=ATAN(CC(8))                                                  TDP00610
      CC(6)=SIN(A62)                                                   TDP00620
      CC(7)=COS(A62)                                                   TDP00630
      I=1                                                              TDP00640
      GO TO 500                                                        TDP00650
   15 E(9)=A35                                                         TDP00660
      E(10)=A46                                                        TDP00670
      E(11)=A47                                                        TDP00680
      DO 17 J=1,11                                                     TDP00690
      A(J)=E(J)                                                        TDP00700
   17 B(J)=CC(J)                                                       TDP00710
      E(1)=ANG(5)                                                      TDP00720
      E(2)=ANG(6)                                                      TDP00730
      CC(1)=ANG(7)                                                     TDP00740
      CC(2)=ANG(8)                                                     TDP00750
      E(3)=SIN(E(1))                                                   TDP00760
      E(4)=COS(E(1))                                                   TDP00770
      E(5)=E(3)/E(4)                                                   TDP00780
      E(8)=(E(5))*(1.0-A14)                                            TDP00790
      A62=ATAN(E(8))                                                   TDP00800
      E(6)=SIN(A62)                                                    TDP00810
      E(7)=COS(A62)                                                    TDP00820
      CC(3)=SIN(CC(1))                                                 TDP00830
      CC(4)=COS(CC(1))                                                 TDP00840
      CC(5)=CC(3)/CC(4)                                                TDP00850
      CC(8)=(CC(5))*(1.0-A14)                                          TDP00860
      A62=ATAN(CC(8))                                                  TDP00870
      CC(6)=SIN(A62)                                                   TDP00880
      CC(7)=COS(A62)                                                   TDP00890
      I=2                                                              TDP00900
      GO TO 500                                                        TDP00910
   19 E(9)=A35                                                         TDP00920
      E(10)=A46                                                        TDP00930
      E(11)=A47                                                        TDP00940
      DO 21 J=1,11                                                     TDP00950
      C(J)=E(J)                                                        TDP00960
   21 D(J)=CC(J)                                                       TDP00970
      TM(1)=A(10)+A(11)                                                TDP00980
      TM(2)=C(10)+C(11)                                                TDP00990
      DO 45 M=1,2                                                      TDP01000
      BETA(M)=TM(M)                                                    TDP01010
      BEDEL(M)=BETA(M)+DEL(M)                                          TDP01020
   45 BLEM(M)=BETA(M)+BEDEL(M)                                         TDP01030
      IQSV(1)=99999                                                    TDP01040
      IQSV(2)=99999                                                    TDP01050
      ITER=0                                                           TDP01060
   82 SDR=DRD(1)+DRM(1)+DRD(2)+DRM(2)                                  TDP01070
      IF(SDR) 83,84,83                                                 TDP01080
   83 DO 30 K=1,2                                                      TDP01090
      IF(DRD(K))32,34,34                                               TDP01100
```

```
   3  RADR(K)=RD*DRD(K)-RM*DRM(K)                                    TDP01110
      GO TO 30                                                       TDP01120
   34 RADR(K)=RD*DRD(K)+RM*DRM(K)                                    TDP01130
   30 CONTINUE                                                       TDP01140
      E(1)=RADR(1)                                                   TDP01150
      E(2)=RADR(2)                                                   TDP01160
      A28=-1.0                                                       TDP01170
   84 E(3)=SIN(E(1))                                                 TDP01180
      E(4)=COS(E(1))                                                 TDP01190
      E(5)=E(3)/E(4)                                                 TDP01200
      E(8)=(E(5))*(1.0-A14)                                          TDP01210
      A62=ATAN(E(8))                                                 TDP01220
      E(6)=SIN(A62)                                                  TDP01230
      E(7)=COS(A62)                                                  TDP01240
      DO 86 J=1,8                                                    TDP01250
   86 CC(J)=D(J)                                                     TDP01260
      I=3                                                            TDP01270
      GO TO 500                                                      TDP01280
   90 C1=A35                                                         TDP01290
      C2=A44                                                         TDP01300
      C3=A45                                                         TDP01310
      C101=A47                                                       TDP01320
      DO 92 J=1,8                                                    TDP01330
   92 CC(J)=C(J)                                                     TDP01340
      I=4                                                            TDP01350
      GO TO 500                                                      TDP01360
   95 C4=A35                                                         TDP01370
      C5=A44                                                         TDP01380
      C6=A45                                                         TDP01390
      C104=A47                                                       TDP01400
      DO 50 J=1,8                                                    TDP01410
   50 CC(J)=B(J)                                                     TDP01420
      I=5                                                            TDP01430
      GO TO 500                                                      TDP01440
   55 C7=A35                                                         TDP01450
      C8=A44                                                         TDP01460
      C9=A45                                                         TDP01470
      C107=A47                                                       TDP01480
      IF(A11) 52,99,53                                               TDP01490
   53 C10=C7                                                         TDP01500
      C11=C8                                                         TDP01510
      C12=C9                                                         TDP01520
      C110=C107                                                      TDP01530
      DO 63 J=1,8                                                    TDP01540
   63 CC(J)=A(J)                                                     TDP01550
      I=6                                                            TDP01560
      GO TO 500                                                      TDP01570
   65 C7=A35                                                         TDP01580
      C8=A44                                                         TDP01590
      C9=A45                                                         TDP01600
      C107=A47                                                       TDP01610
      C13=TH(2)-C(10)-C(11)-C101+C104-DEL(2)                         TDP01620
      C17=C13*A4                                                     TDP01630
      C18=TH(1)-A(10)-A(11)-C110+C107-DEL(1)                         TDP01640
      C22=C18*A4                                                     TDP01650
```

```
      C23=C1-C17                                                        TDP01660
      C24=C4                                                            TDP01670
      C25=C7+C22                                                        TDP01680
      C26=C10                                                           TDP01690
      C27=((C2-C5)*(C25-C26)+(C23-C24)*(C11-C8))                        TDP01700
      C29=((C2-C5)*(C12-C9)+(C3-C6)*(C8-C11))                           TDP01710
      C30=C27/C29                                                       TDP01720
      C28=(C23-C24+C30*(C3-C6))/(C5-C2)                                 TDP01730
      GO TO 130                                                         TDP01740
   52 C13=TH(2)-C(10)-C(11)-C101+C104-DEL(2)                            TDP01750
      C17=C13*A4                                                        TDP01760
      C18=TH(1)-A(10)-A(11)-C107+C104-DEL(1)                            TDP01770
      C22=C18*A4                                                        TDP01780
      C23=C1-C17                                                        TDP01790
      C24=C4                                                            TDP01800
      C25=C7-C22                                                        TDP01810
      C27=(C2*(C25-C24)+C23*(C5-C8)+C8*C24-C5*C25)                      TDP01820
      C29=(C2*(C6-C9)+C3*(C8-C5)+C5*C9-C8*C6)                           TDP01830
      C30=C27/C29                                                       TDP01840
      C28=(C23-C24+C30*(C3-C6))/(C5-C2)                                 TDP01850
  130 C31=(A5*A4*(1.0-A10))/(1.0-A10*E(3)*E(3))**1.5                    TDP01860
      C32=(A5*A4)/(1.0-A10*E(3)*E(3))**0.5                              TDP01870
      C33=(C30/C31)                                                     TDP01880
      C34=(-C28/(C32*E(4)))                                             TDP01890
      E(1)=E(1)+C33                                                     TDP01900
      E(2)=E(2)+C34                                                     TDP01910
      IF(A28) 132,99,134                                                TDP01920
  132 A28=1.0                                                           TDP01930
      GO TO 84                                                          TDP01940
C -                                                                     TDP01950
C - CONVERSION DONE, RETURN TO DISTANCE-BEARING ROUTINE.                TDP01960
C -                                                                     TDP01970
  900 IF(IQSV(1).NE.IWD(1))GO TO 7713                                   TDP01980
      IF(IQSV(2).NE.IWD(2))GO TO 7713                                   TDP01990
      IF(ABS(QSV(1)-FWM(1)).GT.0.1)GO TO 7713                           TDP02000
      IF(ABS(QSV(2)-FWM(2)).GT.0.1)GO TO 7713                           TDP02010
      IDR(1)=IDR(1)*10                                                  TDP02020
      DO 839 I=1,2                                                      TDP02030
      ZIWD(I)=IWD(I)                                                    TDP02040
  839 POS(I)=ZIWD(I)+FWM(I)/60.0                                        TDP02050
      RETURN                                                            TDP02060
C -                                                                     TDP02070
C - CONTINUE ITERATIONS.                                                TDP02080
C -                                                                     TDP02090
 7713 DO 7712 M=1,2                                                     TDP02100
      DRD(M)=0.0                                                        TDP02110
      DRM(M)=0.0                                                        TDP02120
      QSV(M)=FWM(M)                                                     TDP02130
 7712 IQSV(M)=IWD(M)                                                    TDP02140
      ITER=ITER+1                                                       TDP02150
      IF(ITER.LT.100)GO TO 82                                           TDP02160
      IERROR=-1                                                         TDP02170
      RETURN                                                            TDP02180
  134 OMG(1)=E(1)                                                       TDP02190
      OMG(2)=E(2)                                                       TDP02200
```

```
      DO 4840 M=1,2                                                    TDP02210
      W=OMG(M)/RD                                                      TDP02220
      IWD(M)=W                                                         TDP02230
      FWD=IWD(M)                                                       TDP02240
      DWD=W-FWD                                                        TDP02250
      EWM=DWD*60.0                                                     TDP02260
      FWM(M)=ABS(EWM)                                                  TDP02270
      IF(FWM(M)-59.9995)4840,4810,4810                                 TDP02280
 4810 FWM(M)=0.0                                                       TDP02290
      IF(IWD(M))4820,4830,4830                                         TDP02300
 4820 IWD(M)=IWD(M)-1                                                  TDP02310
      GO TO 4840                                                       TDP02320
 4830 IWD(M)=IWD(M)+1                                                  TDP02330
 4840 CONTINUE                                                         TDP02340
      GO TO 900                                                        TDP02350
C -                                                                    TDP02360
C - CALCULATION OF INVERSE VARIABLES.                                  TDP02370
C -                                                                    TDP02380
  500 A59=-CC(2)                                                       TDP02390
      A60=-E(2)                                                        TDP02400
      C35=A59-A60                                                      TDP02410
      C36=ABS(C35)                                                     TDP02420
      IF(C36-PI)501,502,502                                            TDP02430
  502 A16=2.0*PI-C36                                                   TDP02440
      GO TO 505                                                        TDP02450
  501 A16=C36                                                          TDP02460
  505 IF(A16)506,507,506                                               TDP02470
  507 A16=0.00000005                                                   TDP02480
  506 A17=SIN(A16)                                                     TDP02490
      A18=COS(A16)                                                     TDP02500
      A19=E(6)*CC(6)                                                   TDP02510
      A20=E(7)*CC(7)                                                   TDP02520
      A21=A19+A20*A18                                                  TDP02530
      A22=((A17*CC(7))**2+(CC(6)*E(7)-E(6)*CC(7)*A18)**2)**0.5         TDP02540
      A23=(A20*A17)/A22                                                TDP02550
      A24=1.0-A23*A23                                                  TDP02560
      A25=ARSIN(A22)                                                   TDP02570
      A26=A25*A25                                                      TDP02580
      A27=1.0/A22                                                      TDP02590
      A28=A21/A22                                                      TDP02600
      A29=A24*A24                                                      TDP02610
      A30=(A50*A25)+A19*(A51*A22-A52*A26*A27)                          TDP02620
      A31=A24*(A53*A25+A53*A22*A21+A52*A26*A28)                        TDP02630
      A32=A19*A19*(-A52*A21*A22)                                       TDP02640
      A33=A29*(A54*A25+A54*A21*A22-A52*A26*A28-A55*A22*(A21**3))       TDP02650
      A34=A19*A24*(A52*A26*A27+A52*A22*A21*A21)                        TDP02660
      A35=(A30+A31+A32+A33+A34)*A6*A4                                  TDP02670
      A36=(A51*A25)+A19*(-A52*A22-A14*A14*A26*A27)                     TDP02680
      A37=A24*(-A57*A25+A58*A22*A21+A14*A14*A26*A28)                   TDP02690
      A38=(A36+A37)*A23+A16                                            TDP02700
      A39=SIN(A38)                                                     TDP02710
      A40=COS(A38)                                                     TDP02720
      A41=(CC(6)*E(7)-A40*E(6)*CC(7))/(A39*CC(7))                      TDP02730
      IF(A41)510,509,510                                               TDP02740
  509 A41=0.00000005                                                   TDP02750
```

```
  510 A42=1.0/A41                                                    TDP02760
      A43=ATAN(A42)                                                  TDP02770
      IF(C35) 515,514,514                                            TDP02780
  514 IF(C35-PI) 511,512,512                                         TDP02790
  511 IF(A41) 520,521,521                                            TDP02800
  520 A43=PI+A43                                                     TDP02810
      GO TO 521                                                      TDP02820
  512 IF(A41) 517,518,518                                            TDP02830
  515 IF(C35+PI) 511,511,516                                         TDP02840
  516 IF(A41) 517,518,518                                            TDP02850
  517 A43=PI-A43                                                     TDP02860
      GO TO 521                                                      TDP02870
  518 A43=2.0*PI-A43                                                 TDP02880
  521 A43=A43+PI                                                     TDP02890
      A43=A43-2.0*PI                                                 TDP02900
      IF(A43) 522,523,523                                            TDP02910
  522 A43=A43+2.0*PI                                                 TDP02920
  523 A44=SIN(A43)                                                   TDP02930
      A45=COS(A43)                                                   TDP02940
      A46=A35/1609.344                                               TDP02950
      IF(A46-100.0) 525,526,526                                      TDP02960
  525 A47=B1/A46+B2+B3*A46                                           TDP02970
      GO TO 527                                                      TDP02980
  526 A47=A1/A46+A2+A3*A46                                           TDP02990
  527 A46=A35/A4                                                     TDP03000
      GO TO(15,19,90,95,55,65),I                                     TDP03010
   99 RETURN                                                         TDP03020
      END                                                            TDP03030
```

APPENDIX V Lisitng for Block Data constants required by subprogram TDPOS.

```
      BLOCK DATA                                                        99600010
C -                                                                     99600020
C - THIS IS THE CHAIN DATA FOR THE 9960 (US NORTHEAST) STATIONS         99600030
C - W (CARIBOU) AND X (NANTUCKET).                                      99600040
C -                                                                     99600050
C - MODIFIED CONSTANTS TO NORTH AMERICAN DATUM 1927 (NAD-27).           99600060
C -                                                                     99600070
      COMMON/CHAIND/DEL(2),A5,A6,AD(8),DM(8),CS(8)                      99600080
      DATA DEL/11.0E3,25.0E3/                                           99600090
      DATA A5/2.1275406E4/,A6/2.1203281E4/                              99600100
      DATA AD/42.0,76.0,46.0,67.0,                                      99600110
     >        42.0,76.0,41.0,69.0/                                      99600120
      DATA DM/42.0,49.0,48.0,55.0,                                      99600130
     >        42.0,49.0,15.0,58.0/                                      99600140
      DATA CS/50.47,34.44,27.86,39.16,                                  99600150
     >        50.47,34.44,11.98,40.51/                                  99600160
      END                                                               99600170
```

APPENDIX VI Listing for subprogram RUGAZ.

```
      SUBROUTINE RNGAZ(POS,RHO,AZIM)                                    RNG00010
      IMPLICIT REAL*8(A-H,O-Z)                                          RNG00020
      REAL*4 POS(2),RHO,AZIM                                            RNG00030
      COMMON/RNGPOS/PHI,XLNG1                                           RNG00040
      DATA A/6.3782064D6/,F/3.390075304D-3/,PI/3.141592653589793/       RNG00050
      PHII=POS(1)*PI/180.                                               RNG00060
      XLNG2=POS(2)*PI/180.                                              RNG00070
      DXLNG=XLNG1-XLNG2                                                 RNG00080
      BETA=DATAN((1.-F)*DTAN(PHI))                                      RNG00090
      BETAI=DATAN((1.-F)*DTAN(PHII))                                    RNG00100
      C1=DCOS(BETAI)*DSIN(DXLNG)                                        RNG00110
      C2=DCOS(BETA)*DSIN(BETAI)-DSIN(BETA)*DCOS(BETAI)*DCOS(DXLNG)      RNG00120
      C3=DSIN(BETA)*DSIN(BETAI)+DCOS(BETA)*DCOS(BETAI)*DCOS(DXLNG)      RNG00130
      PSI=DATAN(C1/C2)                                                  RNG00140
      THETA=DATAN((C2*DCOS(PSI)+C1*DSIN(PSI))/C3)                       RNG00150
      XM=(DSIN(BETA)+DSIN(BETAI))**2                                    RNG00160
      XN=((DSIN(BETA)-DSIN(BETAI))/DSIN(THETA))**2                      RNG00170
      XU=(1.-DCOS(THETA))/DSIN(THETA)*(THETA-DSIN(THETA))/DSIN(THETA)   RNG00180
      XV=(1.+DCOS(THETA))*(THETA+DSIN(THETA))                           RNG00190
      RHO=DABS(A*THETA-A*F*(XM*XU+XN*XV)/4.)/1852.0                     RNG00200
      IF(C1.GE.0..AND.C2.GE.0.)PSI=PSI*180./PI                          RNG00210
      IF(C1.GE.0..AND.C2.LE.0.)PSI=(PI+PSI)*180./PI                     RNG00220
      IF(C1.LE.0..AND.C2.LE.0.)PSI=(PI+PSI)*180./PI                     RNG00230
      IF(C1.LE.0..AND.C2.GE.0.)PSI=(2.*PI+PSI)*180./PI                  RNG00240
      AZIM=PSI                                                          RNG00250
      RETURN                                                            RNG00260
      END                                                               RNG00270
```

APPENDIX VII Lisitng of sample plotting program.

```
      DATA PI/3.14159265/,XORG/9.5/,YORG/0.25/,XSCL/5./,YSCL/3.125/,IYES RHO00010
     >/3HYES/,IGO/-1/                                                    RHO00020
      PRINT 60                                                           RHO00030
      READ 61,IANS                                                       RHO00040
      IF(IANS.EQ.IYES) IGO=1                                             RHO00050
      CALL PLOTS(BUF,4,11)                                               RHO00060
      CALL SYMBOL(XORG,YORG,0.25,3,0.,-1)                                RHO00070
      DO 20 I=1,9                                                        RHO00080
   20 CALL SYMBOL(XORG-I,YORG,0.25,3,0.,-1)                              RHO00090
      DO 21 I=1,5                                                        RHO00100
   21 CALL SYMBOL(XORG,YORG+1.6*I,0.25,3,0.,-1)                          RHO00110
C                                                                        RHO00120
      CALL SYMBOL(XORG-0.9994/XSCL,YORG+0.8092/YSCL,0.35,1,0.,-1)        RHO00130
      CALL SYMBOL(XORG+3.1184/XSCL,YORG+11.3414/YSCL,0.35,1,0.,-1)       RHO00140
      CALL SYMBOL(XORG-4.5399/XSCL,YORG+24.7055/YSCL,0.35,1,0.,-1)       RHO00150
      CALL SYMBOL(XORG-14.3832/XSCL,YORG+21.7253/YSCL,0.35,1,0.,-1)      RHO00160
      CALL SYMBOL(XORG-32.7287/XSCL,YORG+21.1557/YSCL,0.35,1,0.,-1)      RHO00170
      CALL SYMBOL(XORG-45.2794/XSCL,YORG+20.5569/YSCL,0.35,1,0.,-1)      RHO00180
      CALL SYMBOL(XORG-21.6659/XSCL,YORG+6.0122/YSCL,0.35,1,0.,-1)       RHO00190
C                                                                        RHO00200
   11 READ(10,1,END=100)RNG,AZM                                          RHO00210
      RNG$=RNG                                                           RHO00220
      AZM$=AZM                                                           RHO00230
      RHOX=RNG*SIN(AZM*PI/180.)/XSCL                                     RHO00240
      RHOY=RNG*COS(AZM*PI/180.)/YSCL                                     RHO00250
      CALL PLOT(XORG+RHOX,YORG+RHOY,3)                                   RHO00260
C                                                                        RHO00270
   10 READ(10,1,END=100)RNG,AZM                                          RHO00280
      IF(ABS(RNG-RNG$).GT.5.)GOTO 11                                     RHO00290
      IF(ABS(AZM-AZM$).GT.10.)GOTO 11                                    RHO00300
      AZM$=AZM                                                           RHO00310
      RNG$=RNG                                                           RHO00320
      RHOX=RNG*SIN(AZM*PI/180.)/XSCL                                     RHO00330
      RHOY=RNG*COS(AZM*PI/180.)/YSCL                                     RHO00340
      CALL PLOT(XORG+RHOX,YORG+RHOY,2)                                   RHO00350
      GOTO 10                                                            RHO00360
C                                                                        RHO00370
  100 IF(IGO)101,101,52                                                  RHO00380
   52 CALL PLOT(0.,0.,-3)                                                RHO00390
C                                                                        RHO00400
   51 READ(12,1,END=101)RNG,AZM                                          RHO00410
      RNG$=RNG                                                           RHO00420
      AZM$=AZM                                                           RHO00430
      RHOX=RNG*SIN(AZM*PI/180.)/XSCL                                     RHO00440
      RHOY=RNG*COS(AZM*PI/180.)/YSCL                                     RHO00450
      CALL PLOT(XORG+RHOX,YORG+RHOY,3)                                   RHO00460
C                                                                        RHO00470
   50 READ(12,1,END=101)RNG,AZM                                          RHO00480
      IF(ABS(RNG-RNG$).GT.5.)GOTO 51                                     RHO00490
      IF(ABS(AZM-AZM$).GT.10.)GOTO 51                                    RHO00500
      AZM$=AZM                                                           RHO00510
      RNG$=RNG                                                           RHO00520
      RHOX=RNG*SIN(AZM*PI/180.)/XSCL                                     RHO00530
      RHOY=RNG*COS(AZM*PI/180.)/YSCL                                     RHO00540
      CALL PLOT(XORG+RHOX,YORG+RHOY,2)                                   RHO00550
```

```
      GOTO 50                                                          RHO00560
  101 CALL PLOT(0.,0.,999)                                             RHO00570
      STOP                                                             RHO00580
    1 FORMAT(2(F8.4))                                                  RHO00590
   60 FORMAT(1X,'TWO PLOTS? YES/NO')                                   RHO00600
   61 FORMAT(A3)                                                       RHO00610
      END                                                              RHO00620
```