OAA Eastern Region Computer Programs
nd Problems NWS ERCP - 45

SLOS: Displaying Time Histories of Storm Surge Data from SLOSH Output

Charles D. Little
National Weather Service Forecast Office
Columbia, SC

Scientific Services Division
Eastern Region Headquarters
April 1989

U.S. DEPARTMENT OF COMMERCE / National Oceanic and Atmospheric Administration / National Weather Service

NOAA TECHNICAL MEMORANDUM

National Weather Service, Eastern Region Computer Programs and Problems

The Eastern Region Computer Programs and Problems (ERCP) series is a subset of the Eastern Region Technical Memorandum series. It will serve as the vehicle for the transfer of information about fully documented AFOS application programs. The format ERCP - No. 1 will serve as the model for future issuances in this series.

1 An AFOS version of the Flash Flood Checklist. Cynthia M. Scott, March 1981. (PB81 211252).

2 An AFOS Applications Program to Compute Three-Hourly Stream Stages. Alan P. Blackburn, September 1981. (PB82 156886).

3 PUPPY (AFOS Hydrologic Data Reporting Program). Daniel P. Provost, December 1981. (PB82 199720).

4 Special Search Computer Program. Alan P. Blackburn, April 1982. (PB83 175455).

5 Conversion of ALEMBIC$ Workbins. Alan P. Blackburn, October 1982. (PB83 138313).

6 Real-Time Quality Control of SAOs. John A. Billet, January 1983. (PB83 166082).

7 Automated Hourly Weather Collective from HRR Data Input. Lawrence Cedrone, January 1983 (PB83 167122).

8 Decoders for FRH, FTJ and FD Products. Cynthia M. Scott, February 1983. (PB83 176057).

9 Stability Analysis Program. Hugh M. Stone, March 1983. (PB83 197947).

10 Help for AFOS Message Comp. Alan P. Blackburn, May 1983. (PB83 213561).

11 Stability and Other Parameters from the First Transmission RAOB Data. Charles D. Little, May 1983. (PB83 220475).

12 TERR, PERR, and BIGC: Three Programs to Compute Verification Statistics. Matthew R. Peroutka, August 1983. (PB84 127521).

13 Decoder for Manually Digitized Radar Observations. Matthew R. Peroutka, June 1983. (PB84 127539).

14 Slick and Quick Data Entry for AFOS Era Verification (AEV) Program. Alan P. Blackburn, December 1983. (PB84 138726).

15 MDR--Processing Manually Digitized Radar Observations. Matthew R. Peroutka, November 1983. (PB84 161462) (Revised June 1985, PB85-220580/AS)

16 RAMP: Stability Analysis Program. Hugh M. Stone, February 1984.(PB84 161447)

17 ZONES. Gerald G. Rigdon, March 1984. (PB84 174325)

18 Automated Analysis of Upper Air Soundings to Specify Precipitation Type. Joseph R. Bocchieri and Gerald G. Rigdon, March 1984. (PB84 174333)

19 Verification of Asynchronous Transmissions. Lawrence Cedrone, March 1984. (PB84 189885)

20 AFOS Hurricane Plotter. Charles Little, May 1984. (PB84 199629)

21 WARN - A Warning Formatter. Gerald G. Rigdon, June 1984. (PB84 204551)

22 Plotting TDL Coastal Wind Forecasts. Paula Severe, June 1984 (Revised) (PB84 220789)

23 Severe Weather Statistics STADTS Decoder (SWX) and Plotter (SWY). Hugh M. Stone, June 1984. (PB84 213693)

24 WXR. Harold Opitz, August 1984. (PB84 23722) (Revised August 1985, PB84 100815/AS)

25 FTASUM: Aviation Forecast Summaries. Matthew Peroutka, August 1984. (PB85 112977)

26 SAOSUM: A Short Summary of Observations. Matthew Peroutka, October 1984. (PB85 120384)

27 TRAJ - Single Station Trajectory Plot. Tom Niziol, December 1984. (PB85 135002)

28 VIDTEX. Gerald G. Rigdon, February 1985. (PB85 175669/AS)

29 Isentropic Plotter. Charles D. Little, February 1985. (PB85 175651/AS)

30 CERR: An Aviation Verification Program. M. Peroutka, April 1985. (PB85 204824/AS)

31 Correlation and Regression Equation - REGRS. Hugh M. Stone, May 1985. (PB85 213353/AS)

32 Scatter Diagram and Histogram Program - SCATR. Hugh M. Stone, May 1985. (PB85 213346/AS)

33 TIMCHEK. Gerald G. Rigdon, June 1985. (PB85-221257/AS)

34 A MOS Temperature - PoP Forecast Plot. William C. Randel, October 1985. (PB86 120029/AS)

35 ROTODRAW. Thomas Niziol, November 1985 (PB86 131828/AS)

36 LAMEB: Data Processing for the Great Lakes. William C. Randel and Matthew R. Peroutka, March 1986. (PB86 176658/AS)

37 Convective Parameters & Hodograph Program - Convect. Hugh M. Stone, April 1986. (PB86-197225/AS)

38 DWXR - SMEF Product Compression Program. Harold H. Opitz, September 1986.

39 CRASHQ: Listing Products Being Transmitted At the Time of a Crash. William C. Randel, January 1987 (PB87-151890/AS)

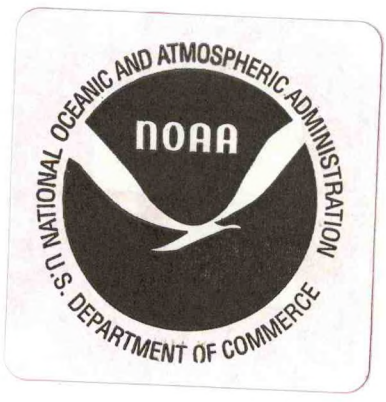
40 AVGPLOT and AVGCLIM. Alan Blackburn, March 1987 (PB87-180626/AS)

41 Severe Weather Potential (SPOT) Plotfile Generator. Ken LaPenta, July 1987. (PB87 217717/AS)

42 COARS Family of Programs. Lawrence Cedrone, November 1987 (PB88-131602)

ACKNOWLEDGEMENTS

Special thanks to Dick Shenot and John Townsend (WSO CHS) for their help in transferring the numerous data files.



Displaying Time Histories of Storm Surge Data from SLOSH Output

Charles D. Little
WSFO Columbia, SC

I. Introduction

The "Sea, Lake and Overland Surge from Hurricanes" (SLOSH) program is operated and maintained by the National Hurricane Center (Purvis, 1984). This program simulates the effects of hurricanes as they approach a coastal basin, producing surge height data that is useful in disaster planning and evacuation (Townsend, 1984). The computer model is specifically adapted to each coastal basin. After the basin has been modeled, the SLOSH program is run for an average of 250 simulations (with different hurricane tracks and strengths). The output from these simulations consists of pages and pages of numerical data. It is this data that is used by disaster planners and meteorologists. Analyzing and displaying storm surge time histories is labor intensive and time consuming under normal conditions, let alone under the threat of an actual hurricane.

An AFOS applications program, SLOS, has been developed to provide a means of rapidly displaying the time histories of storm surge data generated from the SLOSH program. The SLOS program plots the mean tide and the mean tide plus the storm surge expected from an approaching hurricane for the 12 hours before and after the forecast landfall. The beginning and ending times of both 40 mph winds and hurricane force winds are also plotted on the graph. Under hurricane threat conditions, the user can easily update the graphs as conditions change (e. g., time and location of landfall; storm strength, speed or direction of movement).

For planning purposes, any combination of storm track and intensity can be selected, along with the time of arrival and time of high tide, to arrive at an almost infinite combination of possible storm conditions. Worst case scenarios may be developed, and timing of evacuation and location of shelters may then be refined.

The original program to display the SLOSH time histories was developed by John Townsend, WSO Charleston, SC. The program ran on a Radio Shack 64K TRS-80 Color Computer. This program is coded in FORTRAN designed to run in the background partition of AFOS. The original datafiles were modified to aid in reading and data conversion.

SLOS requires the SLOSH storm surge data for the desired basin(s) in RDOS file form. Anyone interested in implementing this program should contact Eastern Region SSD; we will attempt to obtain the necessary data printouts for you. This data must be entered manually to create datafiles for various storm strengths, speeds and tracks, a tedious but unavoidable job.

II. Methodology and Software Structure

The user specifies the time of landfall when starting SLOS (see section V, Part B for command line). The menu will then appear on GDM 2 (default; can be overridden in command line). Figure 1 shows a sample blank menu.

The menu is used to select the desired datafile. Each datafile contains the information needed to create a graph for all points listed under the LOCATIONS column. The menu is also used to enter and change the time of high tide. Status or error messages will be displayed just above the LOCATIONS column. These messages will inform the user of needed entries or minor errors (see section V, Part B for a complete error listing).

Columns 1 through 5 are used to construct the name of the desired datafile. Each datafile name consists of:

1. CATEGORY - Hurricane class 1 through 5
2. DIRECTION - Direction the storm is moving
3. SPEED - Speed, in miles per hour, the storm is moving
4. TRACK - Left, right or at the reference point
5. DISTANCE - Distance, in miles, from the reference point of landfall if the TRACK is left or right.

For example, a datafile with the name 5NW12L20 contains data for a category 5 hurricane moving toward the northwest at 12 mph. Landfall is expected to be 20 miles to the left of the reference point.

To create the name of a datafile, use the track ball to position the cursor in any box (columns 1-5) and press the ENTER CURSOR button. The columns may be entered in any order. A ** will appear in the chosen box.

When columns 1 through 5 have been completed, move the cursor to the LOCATION columns and choose the location to be plotted. If this is the first graph, you will then be prompted by a status message to enter the time of high tide. Enter the appropriate time and the graph will appear on GDM 1. Figure 2 shows a completed menu and Figure 3, the resulting graph.

Once a graph has been plotted, you have four options:

1. ANOTHER LOCATION - You may choose another location using the current datafile. Just move the cursor to the desired location and press the ENTER CURSOR button twice. A new graph should appear in a few seconds on GDM 1.

2. TRACK - Allows the user to choose a different datafile (different track, speed or strength of storm). Move the cursor to the box at the left of TRACK and press the ENTER CURSOR button. Enter new data in columns 1 through 5 and chose a location.

3. STOP - Allows the user to end the program. Move the cursor to the box at the left of STOP and press the ENTER CURSOR button.

4. CHANGE THE TIME OF HIGH TIDE - Allows the user to change the time of high tide at any time. To do this, move the cursor to the box next to the desired time and press the ENTER CURSOR button. The new time of high tide should appear on the menu.

Once a selection has been made, you may not change it. If a mistake is made in columns 1 through 5, continue to enter the data. When all the columns have been completed, one of two things will occur:

1. The program will plot a graph of the wrong datafile. At this time, you may choose the TRACK option and enter the correct filename.

2. The program will not be able to find the datafile and the status error NO SUCH FILENAME -- TRY ANOTHER will appear. Enter the correct filename.

The file SLOS.DT contains the entire menu, most of the graph labels and many of the error messages. Some of the entries will have to be edited to meet local requirements. See Figure 4 and Appendix A. The format for the datafiles (which must be created locally) is described in Appendix B. A sample datafile is shown in Figure 5.

The main program SLOS.FR is the driver program for most of the reading, decoding, plotting and error routines. In addition to this, SLOS.FR opens channels to the GDMs and to the desired datafile, displays several of the status messages and reads the menu options.

Figure 6 illustrates the structure of the program.

SLOS subroutines:

SLOS6 This subroutine reads the AFOS command line for:

THE TIME OF LANDFALL - to the nearest whole hour, based on a 24-hour clock. The time must be entered as four characters. Any time zone may be used (EDT, EST, Z or any other) because the data plotting is based on hours before and after landfall.

CONSOLE - The default console is 0.

SCREEN - The default screen for the menu is 2. The graph will always appear on screen 1.

SLOS1 This subroutine reads the file SLOS.DT (Figure 4) for the following information:

1. Graph labels
2. Menu
3. Errors
4. Information that will be used for data retrieval, display and conversion.

This subroutine writes the menu to the GDM (Egger, 1983). SLOS1 makes calls to SLOS0, SLOS8 and SLOSI to aid in decoding and formatting data read from SLOS.DT. For more information on the contents of SLOS.DT, see Appendix A.

SLOSX SLOSX will display errors that cause the program to halt. If SLOSX is called, the subroutine will clear the menu from the screen, then display the appropriate error message. This routine also closes open files and channels and halts the program. For more information on errors and corrective actions, see section V, Part B.

SLOS2 Reads the cursor location when the ENTER CURSOR button is pressed. This subroutine checks to see if the cursor is in a known column. If the cursor is in a valid location then SLOS3 and SLOS4 are called; if not, then the appropriate error message is displayed on the menu.

SLOS3 Decodes the row position of the cursor.

SLOS4 Decodes the row and column position to determine the correct input value.

SLOS5 Creates the filename of the datafile. This subroutine also completes the array used to generate a link to the proper directory (DIRECTORY:FILENAME).

SLOS9 Computes the time of high tide based on the menu entry.

SLOSG This subroutine reads the selected datafile (Figure 5) with a block read. Each datafile has a 7 block or 7168 byte limit. The data is put into an array in unpacked format. For more information on the datafiles, see Appendix B.

SLOS7 This subroutine converts the data for each location to binary.

SLOSA — This subroutine is the driver routine for creating and displaying the graph. SLOSA starts by drawing and labeling the time lines. It also calls the other display routines.

SLOSB — Computes the mean tide and the storm surge (mean tide plus the forecast surge).

SLOSC — Plots the height lines on overlay 3, the mean tide on overlay 1 and the storm tide on overlay 2.

SLOSD — Decodes and plots the time, beginning/ending of hurricane force/40 mph winds. The large bars are the start/end of hurricane force winds. The start/end of 40 mph winds are represented by the shorter bars.

SLOSE — Converts maximum and minimum storm surge values to ASCII.

SLOSF — Converts height data to ASCII.

CFLTCVT — CFLTCVT is a modified version of FLTCVT (Peroutka, 1981) that allows the main program to continue even if an error is detected in data conversion. If an error is detected, a value of 999999 is returned.

CURSR, RDCUR — (Fors, et al, 1981) These two subroutines read the current location of the cursor and return the X and Y values in pixels.

III. CAUTIONS AND RESTRICTIONS

1. The size of each datafile is limited to 7 blocks (7168 bytes).
2. The names of the locations in the datafile may not exceed 24 characters.
3. If the datafiles are on DP0 or DP0F you must be sure the disk drive is READY before running the program.

IV. REFERENCES

Egger, Thomas J., 1983: Assembly Language Graphics Library with FORTRAN Interfacing. NOAA Central Region Computer Programs and Problems NWS CRCP - No. 9, NWS, Kansas City, MO.

Fors, Jim, Don Laurine and Sandy MacDonald, 1981: AFOS Interactive Graphics, NOAA Western Region Computer Programs and Problems NWS WRCP - No. 28, NWS, Salt Lake City, UT.

Peroutka, M., 1981: Accessing the AFOS Database, NOAA Western Region Computer Programs and Problems NWS WRCP - No. 23, NWS, Salt Lake City, UT.

Purvis, John C., Mark Perry and Michael T. Holland, 1984: Maximum Envelope of Water and Time History for Hurricanes Affecting the South Carolina Coast. South Carolina Water Resources Commission, Columbia, SC, 667p.

Townsend, John F., 1984: A Computer Calculation and Display System for Slosh Hurricane Surge Model Data, NOAA Technical Memorandum NWS ER-67, NWS, Garden City, NY.

V.

ERCP #45
April 1989

DISPLAYING TIME HISTORIES OF STORM SURGE DATA FROM SLOSH OUTPUT

PART A: INFORMATION AND INSTALLATION

PROGRAM NAME: SLOS.SV

AAL ID:
REVISION NO.: 1.00

PURPOSE: Allows the user to rapidly display time histories of storm surge data (12 hours before and after landfall) from NWS SLOSH model output. Under hurricane threat conditions, the user can quickly update the display as the time of landfall and/or the strength of the storm change.

The original version of this program was written for a TRS-80 Color Computer by John Townsend of WSO CHS.

PROGRAM INFORMATION:

Development Programmer:
Chuck Little
Location: WSFO CAE
Phone: (FTS) 677-5501
Language: DG FORTRAN IV/5.57

Maintenance Programmer:
Chuck Little
Location: WSFO CAE
Phone: (FTS) 677-5501
Type: Standard

Save File Creation Dates:
Original Release/Version 1.00 - 05/12/88

Running Time: About 4 seconds to generate one graph once the menu selections have been made.

Disk Space:

Program	50 RDOS blocks
Data	Variable (depending on number of locations) (Each datafile is limited to 7 blocks/7168 bytes.)

PROGRAM REQUIREMENTS

Program Files:

Name	Disk Location	Comments
SLOS.SV	APPL1	

Data Files:

Name	Disk Location	R/W	Comments
SLOS.DT	APPL1	R	Linked from SYSZ

datafiles	variable	R	Contain SLOSH data to be plotted. (location must be entered in file SLOS.DT)

AFOS Products:

ID	Action	Comments
none		

LOAD LINE

```
RLDR/P SLOS SLOS<0 1 2 3 4 5 6 7 8 9 A B C D E F G I X> CFLTCVT CURSR
RDCUR SLOSREV <EGR2 BG UTIL FORT>.LB SLOS.LM/L
```

PROGRAM INSTALLATION

1. Move SLOS.SV and SLOS.DT to APPL1 and create links to them in SYSZ.

2. Create datafiles from SLOSH printouts using M:F/ or an RDOS text editor. See Appendix B of ERCP #45 for format. You may want to store these on a Phoenix disk or floppies, since they will not be frequently used.

3. Edit SLOS.DT (provided) to reflect local setup using E:F/APPL1:SLOS.DT. See Appendix A of ERCP #45 for instructions. Make sure to correctly indicate the location of your datafiles!

DISPLAYING TIME HISTORIES OF STORM SURGE DATA FROM SLOSH OUTPUT

PART B: EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: SLOS.SV

AAL ID:
REVISION NO.: 1.00

PROGRAM EXECUTION:

1. If the SLOSH datafiles have been stored on DP0 or DP0F, the Phoenix drive must be on-line (READY) to access them.

2. Set the zoom to 1:1 on the GDM selected for the display.

3. At an ADM, enter the following command line:

RUN:SLOS xxxx/L n/C m/S

where

xxxx - the time of landfall to the nearest whole hour based on a 24-hour clock. (Z or military - it doesn't matter since all data is based on hours before and after landfall. Likewise, time zone is inconsequential, except that the time must agree with the time zone in SLOS.DT.) The time must be entered as four characters.

n - the console number of the GDM where the menu will be displayed (default=0)

m - the screen number of the GDM where the menu will be displayed (default=2). The output graph always appears on screen 1.

4. The menu should appear on the selected GDM (sample on attached page). The menu is used to generate the name of the desired datafile (each datafile contains the information needed to create a graph for all points listed under the menu's LOCATIONS column) and to enter and change the time of high tide. Status and error messages are displayed just above the LOCATIONS column, informing the user of minor errors and prompting for necessary entries.

Columns 1-5 are used to construct the datafile name. Each datafile name consists of:

1. CATEGORY - hurricane class 1 through 5
2. DIRECTION - direction of storm movement
3. SPEED - speed (mph) of storm movement

4. TRACK - left, right or at the reference point
5. DISTANCE - distance (miles) from landfall to reference point if TRACK is left or right.

(For example, a datafile with the name 5NW12L20 would contain data for a category 5 hurricane moving towards the northwest at 12 MPH. Landfall is expected to be 20 miles to the left of the reference point.) To create the name of the datafile, use the GDM trackball to position the cursor in any box (columns 1-5) and press the ENTER CURSOR button. The columns may be entered in any order. A "***" will appear in the chosen box.

When columns 1 through 5 have been completed, move the cursor to the LOCATION columns and choose the location to be plotted.

If you are plotting the first graph, you will be prompted by a status message to enter the time of high tide. Once this is entered, the storm surge display should appear on GDM 1.

After the first graph has been plotted, you can either stop or create more displays for different locations and/or tracks and/or times of high tide:

ANOTHER LOCATION - You may choose another location using the current datafile by moving the cursor to the desired location and pressing ENTER CURSOR twice. A new graph should appear in seconds on GDM 1.

DIFFERENT TRACK - You can select another hurricane track by changing the name of the datafile. Move the cursor to the box at the left of TRACK and press ENTER CURSOR. Enter new data in columns 1 through 5 and the LOCATION column.

DIFFERENT TIME OF HIGH TIDE - You can change the time of high tide any time by moving the cursor to the box next to the desired time and pressing ENTER CURSOR. The new time of high tide should appear on the menu.

STOP - Exit the program by moving the cursor to the box at the left of STOP and press ENTER CURSOR.

Once any selection has been made (by hitting ENTER CURSOR) you cannot change it. If a mistake is made in columns 1 through 5 (datafile name) continue to enter the data. When all of the columns have been completed one of two things will occur:

1. The program will plot a graph of the wrong datafile. At this time you can choose the TRACK option and enter the correct filename.

2. The program will not be able to find the datafile and the status error "NO SUCH FILENAME -- TRY ANOTHER" will appear. Enter the correct filename.

ERROR CONDITIONS

SLOS produces two types of error responses: status errors/messages and halting errors. The status errors and messages will be displayed just above the LOCATIONS column. Halting errors will cause the program to stop executing and will replace the menu. All error messages are listed below along with the most likely cause.

STATUS ERRORS/MESSAGES

1. NONE (a message)	No errors
2. CURSOR MUST BE IN A BOX	The cursor is out of bounds - move it to a box and press ENTER CURSOR again.
3. FLAG SET -- TRY ANOTHER	This field has already been entered. If a mistake was made continue to enter data. When all data has been entered, the program will either plot the wrong graph or give Status Error 4.
4. ENTER THE TIME OF HIGH TIDE	Go to the column TIME OF HIGH TIDE and select a time.
5. TRBL CONVERTING TIDE DATA	Indicates an error in converting the time of high tide to binary. The error is likely in the SLOS.DT format under the TIME OF HIGH TIDE column.

HALTING ERRORS

1. TRBL WITH TIDE- SLOS.DT	Trouble converting the mean tide or departure data to binary. The error is likely in the last four lines of SLOS.DT.
2. TRBL WITH TIDE DATA	More than two STATUS ERROR 6's were recorded. Check the format of SLOS.DT.
3. ERROR READING COMMAND LN	Unable to convert data in the command line. Enter the correct command.
4. LANDFALL OUT OF BOUNDS	The time of landfall was less than 0 or greater than 24. Enter the correct command.

5. TRBL CONVERTING THE MENU	Can't compute the end of column data. Check the SLOS.DT format and spacing in the menu portion of the file.
6. BAD DATA FILE	Can't convert ASCII to binary in the datafile just entered. Check the datafile.
7. NO ERROR	Normal stop.
8. NOT VALID LANDFALL TIME	The time of landfall was not entered or was not four characters. Enter the correct command.
9. NAME/LOCATION MISMATCH	Name in datafile and name on menu do not match. Enter the correct command.
10. SLOS.DT-FLOOD PLAIN HGT	Height of the flood plain cannot be converted to binary. (Data on line 67 of SLOS.DT)
11. TRBL OPENING SLOS.DT	File most likely missing.
12. CANT READ # OF LOCATIONS	The number of locations in SLOS.DT can't be converted to binary.
13. TOO MANY LOCATIONS	The maximum number allowed is 24. Check SLOS.DT
14. TRBL READING DIRECTORY	Format error in SLOS.DT on line with the location/directory information.
15. CANNOT INIT DIRECTORY	Unable to initialize the directory listed in SLOS.DT. Check to see if the directory name is spelled correctly or if device is on-line (DP0/DP0F).

Messages from ADM	Meaning
none	

Dasher Messages	Meaning
none	

VI. Figures

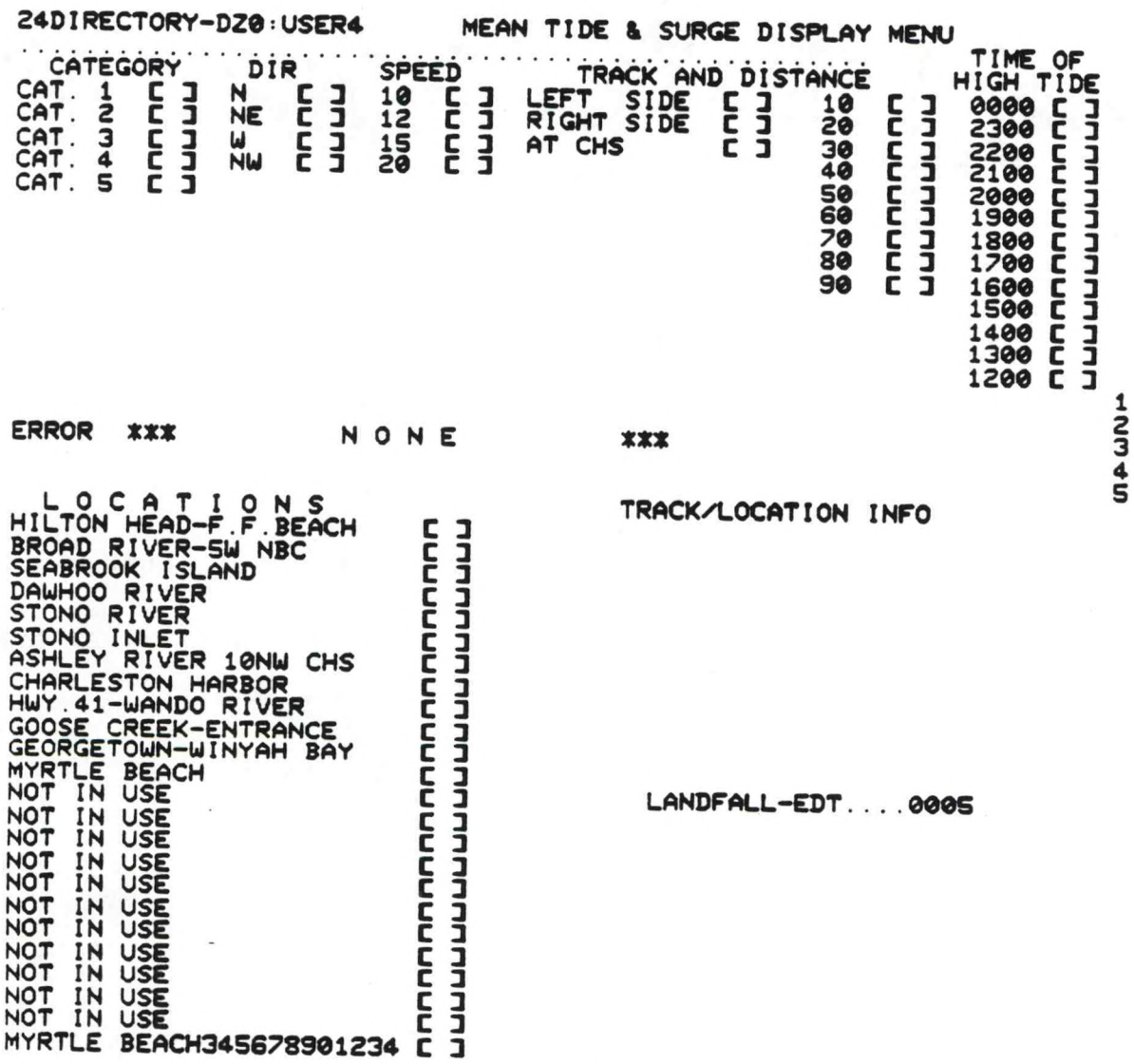


Figure 1. Sample SLOS menu

```
24DIRECTORY-DZ0:USER4        MEAN TIDE & SURGE DISPLAY MENU

   CATEGORY      DIR       SPEED          TRACK AND DISTANCE        TIME OF
                                                                   HIGH TIDE
CAT. 1  [ ]   N   [ ]   10  [ ]  LEFT  SIDE  [**]   10  [ ]   0000 [ ]
CAT. 2  [ ]   NE  [ ]   12  [**] RIGHT SIDE  [ ]    20  [**]  2300 [ ]
CAT. 3  [ ]   W   [ ]   15  [ ]  AT CHS      [ ]    30  [ ]   2200 [ ]
CAT. 4  [ ]   NW  [**]  20  [ ]                     40  [ ]   2100 [ ]
CAT. 5  [**]                                        50  [ ]   2000 [ ]
                                                    60  [ ]   1900 [ ]
                                                    70  [ ]   1800 [ ]
                                                    80  [ ]   1700 [ ]
                                                    90  [ ]   1600 [ ]
                                                              1500 [ ]
                                                              1400 [ ]
                                                              1300 [ ]
ANOTHER -- LOCATION    TRACK   [ ]    STOP   [ ]             1200 [ ]
                                                                    1
                                                                    2
ERROR  ***          N O N E          ***                            3
                                                                    4
                                                                    5
  L O C A T I O N S                 TRACK/LOCATION INFO
HILTON HEAD-F.F.BEACH      [ ]
BROAD RIVER-SW NBC         [ ]
SEABROOK ISLAND            [ ]       TRACK.....5NW12L20
DAWHOO RIVER               [ ]
STONO RIVER                [ ]
STONO INLET                [ ]
ASHLEY RIVER 10NW CHS      [**]      LOCATION.....ASHLEY RIVER 10NW CHS
CHARLESTON HARBOR          [ ]
HWY.41-WANDO RIVER         [ ]
GOOSE CREEK-ENTRANCE       [ ]       HIGH TIDE-EDT...2300
GEORGETOWN-WINYAH BAY      [ ]
MYRTLE BEACH               [ ]
NOT IN USE                 [ ]       LANDFALL-EDT....0005
NOT IN USE                 [ ]
NOT IN USE                 [ ]
NOT IN USE                 [ ]
NOT IN USE                 [ ]
NOT IN USE                 [ ]
NOT IN USE                 [ ]
NOT IN USE                 [ ]
NOT IN USE                 [ ]
NOT IN USE                 [ ]
MYRTLE BEACH345678901234   [ ]
```

Figure 2. Sample menu, filled out (no ** will appear in time of high tide column)

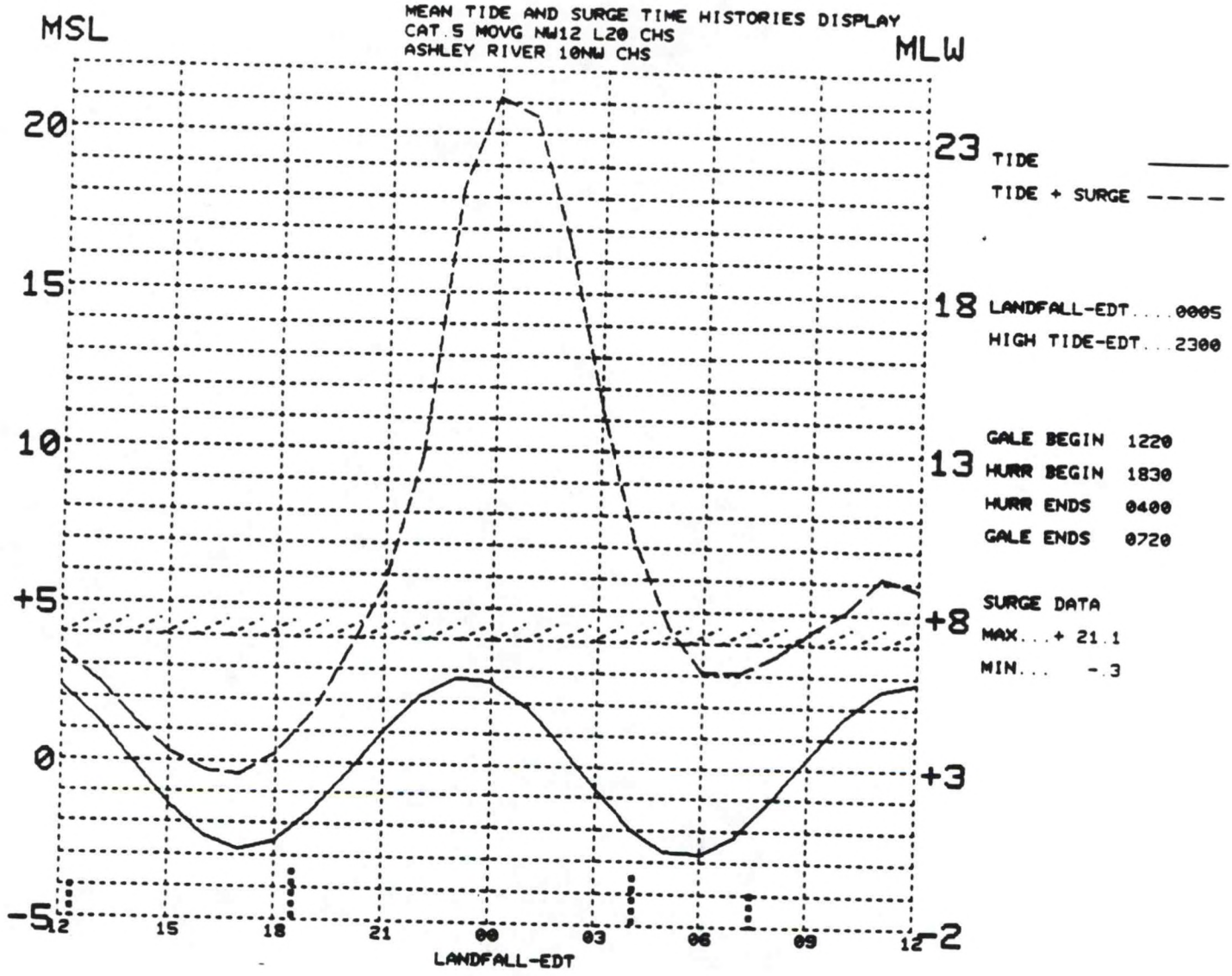


Figure 3. Graph produced from menu in Figure 2 (heights - feet, small bars = beginning and ending of 40 mph winds, large bars = beginning and ending of hurricane force winds, diagonal lines = height at which flooding begins)

Figure 4. Sample SLOS.DT

```
4NW12L20
HILTON HEAD-F.F.BEACH,13,14,14,15,15,15,16,17,20,20,13,-1,-27,-42,-32,-11,2,8,-3,-5,-7,-9,-7,-3,1,-950,-340,150,550
BROAD RIVER-SW NBC,10,10,10,10,11,12,13,13,13,12,10,9,7,5,4,3,2,0,0,0,2,3,4,5,6,-850,-250,310,710
SEABROOK ISLAND,15,16,17,17,19,21,25,31,40,54,62,70,59,44,27,16,9,8,7,6,5,4,3,2,1,-1050,-520,330,630
DAWHOO RIVER,10,10,10,10,11,12,13,14,17,18,21,27,34,42,39,36,32,28,19,16,14,9,8,7,6,-1000,-430,330,740
STONO RIVER,5,5,5,5,5,5,5,5,5,6,14,28,90,130,126,110,100,94,80,82,78,75,71,67,63,-1020,-450,400,720
STONO INLET,17,18,19,21,23,26,32,41,56,77,104,110,120,100,73,42,17,16,17,18,4,3,8,15,14,-1110,-540,330,630
ASHLEY RIVER 10NW CHS,11,12,13,16,19,22,25,30,35,43,52,122,160,169,149,115,99,70,56,40,43,39,31,31,20,-1020,-440,400,720
CHARLESTON HARBOR,17,10,19,21,23,27,33,43,55,77,110,142,157,144,99,43,16,17,16,14,4,3,2,1,0,-1050,-510,330,640
HWY.41-WANDO RIVER,10,10,10,10,11,12,13,14,17,20,23,33,63,80,97,94,85,76,73,69,66,67,60,56,54,-1030,-440,320,650
GOOSE CREEK-ENTRANCE,10,10,10,10,10,10,10,10,10,10,10,11,12,56,85,70,70,65,61,59,57,55,54,53,52,-1000,-420,350,720
GEORGETOWN-WINYAH BAY,12,13,14,15,16,18,19,22,25,31,37,44,50,52,51,47,42,33,28,25,24,22,20,19,18,-920,99,99,500
MYRTLE BEACH,19,20,20,21,22,24,27,30,32,33,35,35,33,30,20,9,0,1,14,23,18,8,3,7,14,-930,99,99,440
END
```

Figure 5. Sample datafile

MAIN PROGRAM

SLOS

SUBROUTINES

```
SLOS6  ----->  CFLTCVT

SLOS1  ----->  SLOS0  ----->  CFLTCVT
               SLOS8
               SLOSI
               SLOSX

SLOS2  ----->  SLOS3
               SLOS4
               SLOS5
               CURSR  ----->  RDCUR

SLOSG

SLOS7  ----->  SLOSX

SLOS9  ----->  SLOS3
               SLOSX
               CFLTCVT

SLOSX

SLOSA  ----->  SLOSB
               SLOSC  ----->  SLOSF
               SLOSD
               SLOSE

CURSR  ----->  RDCUR
```

LOAD LINE

```
RDLR/P SLOS SLOS<0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F,G,I,X> CFLTCVT CURSR
RDCUR SLOSREV EGR2.LB BG.LB UTIL.LB FORT.LB SLOS.LM/L
```

(assumes AFOSE.LB linked to alias SYS.LB)

Figure 6. Software structure and load line

VII. Appendices

APPENDIX A - EDITING SLOS.DT

When editing SLOS.DT (E:F/), be sure that the format and spacing of the file remains unchanged. The main program displays the menu directly from SLOS.DT. If the format has been changed, the program may not function correctly or graphs may not be properly labeled or plotted. The program also reads this file for certain parameters that will be needed to initialize disks or subdirectories. If the format is not correct, the program will halt. Listed below are the items that may be changed in SLOS.DT:

LINE 1 and 2	EDT - Change to the desired time zone. (The data is actually retrieved and plotted in terms of hours before and after landfall.)
LINE 6 and 19	CHS - Change to the 3 letter identifier of the basin.
LINE 14	(first two characters) - Change to the number of locations contained in each local datafile. Should be an even number.
LINE 14	DIRECTORY-DZO:USER4 - Change to the location of the local datafiles. This can be any directory or directory:subdirectory followed by two spaces. (If using DPO or DPOF the devices must READY so SLOS can access the files.)
LINES 17-29	THE MENU - The CATEGORY and TRACK columns will not have to be changed except to insert the basin ID on line 19 of the TRACK column. The remaining columns (DIR, SPEED and DISTANCE) may be changed, added to or deleted from, as needed, depending on the local datafiles. It is extremely important to keep the proper spacing and location of the data and brackets.
LINES 36-47	LOCATIONS - The name of each location in the datafile. The spelling MUST be the same on the menu as in the datafile.
LINE 67	XXX.X - XXX.X is height above MSL in feet at which flooding usually begins. This is just a rule of thumb threshold value and will be the same for all graphs. Enter 999.9 and this height will not be plotted.
LINE 68	XX.X,XX.X... - MEAN TIDAL RANGE (feet) above and below MSL. With a 0 (zero) entered no tide is plotted. Data on this line is for the

first 12 locations in the datafile and must match the sequence of these locations. Data must be entered for all 12 locations - enter a 0 if the range is not needed.

LINE 69

XXXX,XXXX,... - TIDE DEPARTURE (time) FROM KEY LOCATION - The average time, in minutes, of departure from the tide times at a known reference point. (The key location's time of high tide is the one entered when running the program.) A minus indicates high tide earlier than at the key location. Data on this line is for the first 12 locations in the datafile and must match the sequence of these locations. Data must be entered for 12 locations.

LINE 70 and 71

The same as line 68 and 69 but for the second 12 stations in the datafile.

(NOTE - The current tide tables for the East Coast of North and South America may be used to get an estimation for lines 68 - 71.)

A sample SLOS.DT appears in Figure 4.

APPENDIX B - CREATING THE DATAFILES

The datafiles for this program will have to be created locally and are limited in size to 7 RDOS blocks. Each file contains data for a different combination of storm strength, speed and track for up to 24 locations. The number of locations should be an even number (2, 4, 6, ..., 24) and the same locations must be used for each datafile. The format for a datafile is as follows:

LINE 1 - Same as the filename (for example, 5NW12L20) followed by a carriage return or a carriage return and line feed.

LINE 2 THORUGH 25 - Should contain the following:

1. NAME OF THE LOCATION - The spelling in this file and the spelling in the SLOS.DT file MUST be the same. The name may not exceed 24 characters.

2. 25 HOURLY SURGES - The values are coded in tenths of feet: 10 equals 1.0 feet. The surge values are for 12 hours before landfall to 12 hours after landfall.

3. WIND THRESHOLD - Time in hours and minutes before (-) and after the arrival of the storm. There are 4 wind threshold values -

   A. beginning of 40 mile an hour winds
   B. beginning of hurricane force winds
   C. ending of hurricane force winds
   D. ending of 40 mile an hour winds

   EXAMPLES -
   -810 is 8 hours and 10 minutes before landfall.
   645 is 6 hours and 45 minutes after landfall.
   99 is for no wind for this threshold.

4. After each line there should be a carriage return or carriage return and line feed.

LAST LINE - The word END

A sample datafile appears in Figure 5.

VIII. Source Code

```
C   SLOS.FR                    REV 01.00
C   OCT 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      THE PROGRAM READS DATA FROM SLOS.DT AND DISPLAYS A MENU. WHEN THE
C      CORRECT DATA HAS BEEN ENTERED - THE PROGRAM WILL READ AND PLOT A
C      GRAPH OF THE DESIRED DATA.
C   CHANNELS - VIA GCHN:
C      IC  - TO GDM
C      IZ  - TO IFILE - DATA FILE
C      IO  - TO SLOS.DT
C   RLDR:
C      SLOS SLOS0 SLOS1 SLOS2 SLOS3 SLOS4 SLOS5 SLOS6 SLOS7 SLOS8 SLOS9
C      SLOSA SLOSB SLOSC SLOSD SLOSE SLOSF SLOSG SLOSX SLOSI CFLTCVT CURSR RDCURC      EGR2.LB BG.LB UTIL.LB FORT.LB AFOSE.LB
C   EXITS:
C      SLOS
C      SLOS1
C      SLOS7
C      SLOS9
C
C   NOTE -  FORT/X SLOS(2,5,7,A,G,I,X) FOR SOME OUTPUT AT THE DASHER
C
C
C
      COMMON/COL/MENUD(-8:8),MENUP(-8:8),MENUB(200)
      COMMON/BARDG/IOUTU(90)
      COMMON MENU(500),KERR(400),NAME(400),IDATA06(22,9),LNT(12)
      DIMENSION IFILE(10),LOCATN(20),SRG(-2:50),IBT(5),ITM(2),ITD(2)
      DIMENSION TIDMD(30,2)
      DIMENSION IDIR(20),LDIR(20),MDIRT(12),LNK(20),IDATA(7168),IDATA1(30,2)
      IHITD=-99              ;TM OF HI TIDE
      LRR=1                  ;CONTROL VRBL
      IHALT=0                ;WL STOP PROGRM IF NOT ZERO
      IXE=300                ;LOCTN TO RIT ERR MSG
      IYE=2100               ;LOCTN TO RIT ERR MSG
      ICN=0                  ;DFLT CONSOL
      ISC=2                  ;DFLT SCN
      CALL SLOS6 (ICN,ISC,LNDFL,IHALT,IFT,ITM)
      CALL GCHN (IC,IER)
      CALL OPENN (IC,'%GDM',0,IER)
      DO 5 I=1,3
      CALL EFCLR (IC,ICN,ISC,I)
  5   CONTINUE
      CALL SLOS1 (IC,ICN,ISC,IXE,IYE,IHALT,ITM,ITD,ICTY,IDIR,LDIR,NU,IFT
     *,FB,TIDMD)
 10   CALL SLOS2 (IC,ICN,ISC,IXE,IYE,IFILE,LOCATN,NROW,LDIR,NU,LNK,LRR)
      IF (IHITD.EQ.-99) GOTO 50
 11   CALL DLINK (IFILE,LNK,IER)
      CALL SLOS6 (IFILE,IDATA,IDATA1,IER)              ;RD DATAFL
      IF(IER.NE.1) GOTO 900
      CALL DULNK (IFILE,IER)
 15   CALL SLOS7 (IC,ICN,ISC,IXE,IYE,SRG,NROW,LOCATN,IDATA,IDATA1,TIDMD)
      CALL SLOSA (IC,ICN,LNDFL,IFT,IHITD,LOCATN,SRG,FB)  ;PLT DATA
      CALL EFSTXT (MENU(97),IC,300,2250,0,0,0,ICN,ISC,3,24)
 19   CALL CURSR (IC,ICN,ISC,I,J,IER)
      IF(I.GE.2570.AND.I.LE.2665.AND.J.LE.2080.AND.J.GT.2240) GOTO 35
      IF(I.GE.1420.AND.I.LE.1508.AND.J.GE.2248.AND.J.LE.2280) GOTO 20
      IF(I.GE.1932.AND.I.LE.2020.AND.J.GE.2248.AND.J.LE.2280) GOTO 997
      GOTO 30
      CALL EFSTXT (KERR(21),IC,IXE,IYE,0,0,0,ICN,ISC,3,20)
```

```
      GOTO 19
   20 CALL EFCLR(IC,ICN,ISC,2)                          ;ANOTHER TRACK
      CALL EFCLR(IC,ICN,ISC,1)
      CALL EFSTXT (KERR(1),IC,IXE,IYE,0,0,0,ICN,ISC,3,20)
      GOTO 40
   30 LRR=-9999                                         ;SAME TRACK-ANOTHER LOCATN
      CALL EFCLR (IC,ICN,ISC,1)
      CALL SLOS2 ( IC,ICN,ISC,IXE,IYE,IFILE,LOCATN,NROW,LDIR,MU,LNK,LRR)
      LRR=1
      GOTO 15
   35 CALL SLOS9 (IC,ICN,ISC,IXE,IYE,IHITD,J,ITD)       ;TIME OF HI TIDE
      GOTO 19
  900 CALL EFSTXT (KERR(201),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;CANT FIND FL
      CALL EFCLR (IC,ICN,ISC,1)
      CALL EFCLR (IC,ICN,ISC,2)
   40 CALL DULNK (IFILE,IER)
      GOTO 10
   50 CALL EFSTXT (KERR(281),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;1ST TIDE
      CALL CURSR (IC,ICN,ISC,I,J,IER)
      CALL SLOS9 (IC,ICN,ISC,IXE,IYE,IHITD,J,ITD)
      CALL EFSTXT (KERR(1),IC,IXE,IYE,0,0,0,ICN,ISC,3,20)
      GOTO 11
  997 IHALT=221                                         ;NORMAL STOP
      CALL SLOSX (IC,ICN,ISC,IHALT)
      END
```

```
      SUBROUTINE SLOS0 (FB,ICTY,IBUF,IHALT)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      CONVERTS DATA ON LAST LINE OF SLOS.DT TO BINARY. COMPUTES LOWER LIMITS
C      FOR EACH COLUMN IN THE MENU.
C   ARGUMENT LIST:
C      FB    - FEET ABV MSL TO PLOT HASH MARKS TO INDICATE FLOODG BEGINS
C      ICTY  - # OF CITIES IN DATAFILE
C      IBUF  - BUFFER WITH DATA FROM SLOS.DT
C      IHALT - POINTER TO ERROR MESSAGE-KERR/MENUQ-GREATER THAN 400 MENUQ
C
C
C
      COMMON MENU(500),KERR(400),NAME(400),IDATADS(22,9),LNT(12)
      DIMENSION IBUF(65),IWK(65)
      COMMON/QARDQ/IOUTU(90)
      CALL UNPACK (IBUF,15,IOUTU)
      FB=CFLTCVT(1,5)
      IF(FB.EQ.999999.) GOTO 90
      LNT(1)=5          ;CAT
      LNT(4)=3          ;TRACK
      LNT(7)=13         ;TIDE
      DO 20 K=1,10
      IF(IDATADS(K,1).EQ.000040K) GOTO 25
   20 CONTINUE
      GOTO 95
   25 LNT(2)=K-1        ;DIR
      DO 30 K=1,10
      IF(IDATADS(K,3).EQ.000040K) GOTO 35
   30 CONTINUE
      GOTO 95
   35 LNT(3)=K-1        ;SPD
      DO 40 K=1,15
      IF(IDATADS(K,6).EQ.000040K) GOTO 45
   40 CONTINUE
      GOTO 95
   45 LNT(5)=K-1        ;DIST
      LNT(6)=ICTY
      RETURN
   90 IHALT=301   ;TRBL IN CFLTCVT
      RETURN
   95 IHALT=161   ;TRBL RDG MENU
      RETURN
      END
```

```
      SUBROUTINE SLOS1 (IC,ICN,ISC,IXE,IYE,IHALT,ITM,ITD,ICTY,IDIR,LDIR,NU
     *,IFT,FB,TIDWD)
C                              REV 01.00
C  OCT 1987                    CHUCK LITTLE          WSFO CAE FTS 677-5501
C  FORTRAN IV/REV 5.57         DG ECLIPSE (S230)     RDOS/REV 6.19
C  PURPOSE:
C     READS SLOS.DT FOR DATA USED IN DISPLAY-MENU AND ERRORS. DISPLAYS
C     THE MENU OR HALTS THE PROGRAM IF ERRORS ARE DETECTED.
C  ARGUMENT LIST:
C     IC,ICN,ISC - CHANNEL TO GDM, CONSOLE, SCREEN
C     IXE,IYE    - PIXELS TO PLOT STATUS MESSAGES
C     IHALT      - POINTER TO ERROR MESSAGE-KERR/MENUQ-GREATER THAN 400 MENUQ
C     ITM,ITD    - TIME OF LANDFALL,TIME OF HI TIDE ASCII
C     ICTY       - # OF LOCATIONS IN DATAFILE
C     IDIR       - DIR/SUBDIR FOR INIT
C     LDIR       - DIR - LOCATION OF DATAFILE - UNPACKED
C     NU         - # OF BYTES IN LDIR
C     IFT        - TIME 12 HRS BEFORE LANDFALL
C     FB         - FEET MSL TO PLOT FLOOD BEGIN LINE
C     TIDWD      - TIDE DATA FROM SLOS.DT
C  EXITS:
C     SLOSX
C
C
      COMMON/CDL/MENUD(-8:8),MENUP(-8:8),MENUQ(200)
      DIMENSION IBUF (65),ITM(2),ITD(2),IXX(65),IDIR(20),LDIR(20),TIDWD(30,2)
      COMMON MENU(500),KERR(400),NAME(400),IDATADS(22,9),LNT(12)
      IOV=3
      CALL GCHN (IQ,IER)
      CALL OPENN (IQ,'SLOS.DT',0,IER)
      IF(IER.NE.1) GOTO 900     ;TRBL OPNG SLOS.DT
      NWD=34
      K0=0
      IX=300
      IY=3000
      NO=1
      DO 5 J=1,13               ;RD GRAPH LABLS
      CALL RDL (IQ,MENU(NO),IBT,IER)
      NO=NO+24
  5   CONTINUE
      ITX=IFT/100
      IF(ITX.EQ.0) ITX=24
      ITX=ITX+12+264
      DO 10 J=1,22              ;RD MENU
      CALL RDL (IQ,IBUF,IBT,IER)
      IF(J.EQ.1) CALL SLOS0 (IBUF,IBT,IDIR,ICTY,LDIR,IHALT,NU,IER)
      IF(IER.EQ.0) GOTO 910
      IF(J.LT.4.OR.J.GT.16) GOTO 8
      L=J-3
      CALL UNPACK (IBUF,62,IXX)
      IDATADS(L,1)=IXX(14)
      IDATADS(L,2)=IXX(15)
      IDATADS(L,3)=IXX(23)
      IDATADS(L,4)=IXX(24)
      IDATADS(L,7)=IXX(50)
      IDATADS(L,8)=IXX(51)
      IDATADS(L,5)=MENU(ITX)
      IDATADS(L,6)=IXX(50)
      IBUF(30)=MENU(ITX)
      ITX=ITX-1
```

```
  9   CALL EFSTXT (IBUF,IC,IX,IY,K0,K0,K0,ICN,ISC,ICV,NWD)
      IY=IY-50
 10   CONTINUE
      IY=1900
      NO=1
      NWD=14
      DO 20 J=1,12              ;RD LOCATIONS
      CALL RDL (IQ,NAME(NO),IBT,IER)
      DO 15 K=1,2

      CALL EFSTXT (NAME(NO),IC,IX,IY,K0,K0,K0,ICN,ISC,ICV,NWD)
      NO=NO+14
      IY=IY-50
 15   CONTINUE
 20   CONTINUE
      DO 30 J=1,3           ;RD 3 BLANK LINES
      CALL RDL (IQ,IBUF,IBT,IER)
 30   CONTINUE
      NO=1
      DO 40 J=1,16          ;RD ERROR MSG
      CALL RDL (IQ,KERR(NO),IBT,IER)
      NO=NO+20
 40   CONTINUE
      MENU(33)=ITM(1)
      MENU(34)=ITM(2)
      CALL EFSTXT (KERR(1),IC,IXE,IYE,K0,K0,K0,ICN,ISC,ICV,20)
      CALL EFSTXT (MENU (25),IC,1700,1300,K0,K0,K0,ICN,ISC,ICV,10)
      CALL RDL (IQ,IBUF,IBT,IER)   ;RD FLD PLN HGT
      CALL SLOS0 (FB,ICTY,IBUF,IHALT)
      IF(IHALT.NE.0) GOTO 910
      CALL SLOS1 (TIDMD,IQ,IHALT)  ;RD TIDE DATA
      IF(IHALT.NE.0) GOTO 910
      CALL KLOSE (IQ,IER)
      RETURN
900   IHALT=401
910   CALL KLOSE (IQ,IER)
      CALL SLOSX (IC,ICN,ISC,IHALT)
      END
```

```
      SUBROUTINE SLOS2 (IC,ICN,ISC,IXE,IYE,IFILE,LOCATN,NROW,LDIR,NU,LNK,
     $LRR)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C     READS CURSOR LOCATION ON SCREEN, COMPUTES IF IN A KNOWN COLUMN,
C     DISPLAY STATUS MESSAGE AS NEEDED
C   ARGUMENT LIST:
C     IC,ICN,ISC - CHANNEL TO GDM,CONSOLE,SCREEN
C     IXE,IYE    - PIXELS TO PLOT STATUS MESSAGES
C     IFILE      - NAME OF DESIRED DATAFILE
C     LOCATN     - NAME OF DESIRED LOCATION
C     NROW       - LINE NO IN DATAFILE OF DESIRED LOCATION
C     LDIR       - DIR - LOCATION OF DATAFILE - UNPACKED
C     NU         - # OF BYTES IN LDIR
C     LNK        - LINK DIR:FILENAME -LDIR+IFILE  PACKED
C     LRR        - INDICATOR FOR LOCATION CHANGE ONLY
C
C
C
      DIMENSION IWRK(20),LOCATN(20),IFILE(10),IFLAG(7),LDIR(20),LNK(20)
      COMMON MENU(500),KERR(400),NAME(400),IDATA0S(22,9),LNT(12)
      IB='**'
      DO 5 I=1,6              ;SET DONE FLGS
      IFLAG(I)=0
    5 CONTINUE
   10 CALL CURSR (IC,ICN,ISC,I,J,IER)
X     TYPE 'X,Y',I,J
      LFLAG=0
      IF(I.LE.676.AND.I.GE.596)   LFLAG=1  ;COL 1  CAT
      IF(I.LE.1000.AND.I.GE.924)  LFLAG=2  ;COL 2  DIR
      IF(I.LE.1324.AND.I.GE.1244.AND.J.GT.2350) LFLAG=3 ;COL 3 SPD
      IF(I.LE.1936.AND.I.GE.1860) LFLAG=4  ;COL 4  TRACK
      IF(I.LE.2300.AND.I.GE.2220) LFLAG=5  ;COL 5  DIST
      IF(I.LE.1300.AND.I.GE.1204.AND.J.LT.2250) LFLAG=6 ;COL 6 LOCATN
      IF(LFLAG.NE.0) GOTO 50
   90 CALL EFSTXT (KERR(21),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;ERR IN BOX
      GOTO 10
   50 IF(LFLAG.EQ.6) GOTO 600
      IF(IFLAG(LFLAG).EQ. LFLAG) GOTO 900   ;FLG ALREADY SET
      CALL EFSTXT (KERR(1),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;CLR ERR
      CALL SLOS3 (IROW,J,2850,2880,LFLAG)                 ;FIND ROW
      IF(IROW.EQ.0) GOTO 90                               ;ERR IN BOX
      GOTO (700,700,700,100,700),LFLAG
  100 IF(IROW.EQ.3) IFLAG(5)=5                            ;AT POINT
      GOTO 700
  600 IF(IFLAG(LFLAG).EQ.LFLAG) GOTO 900
      CALL EFSTXT (KERR(1),IC,IXE,IYE,0,0,0,ICN,ISC,3,20)
      CALL SLOS3 (IROW,J,1900,1930,LFLAG)
      IF(IROW.EQ.0) GOTO 90
      IFLAG(LFLAG)=LFLAG
      NROW=IROW
      CALL SLOS4 (IROW,LFLAG,IWRK,LOCATN)
      CALL EFSTXT (MENU(49),IC,1700,1600,0,0,0,ICN,ISC,1,20) ;LOCATN
      CALL EFSTXT (IB,IC,I,J,0,0,0,ICN,ISC,1,1)
      IF(LRR.EQ.-9999) RETURN                  ;SAME TRACK-NEW CITY
      GOTO 755
  700 IFLAG(LFLAG)=LFLAG
      CALL SLOS4 (IROW,LFLAG,IWRK,LOCATN)
      CALL EFSTXT (IB,IC,I,J,0,0,0,ICN,ISC,1,1)
  755 DO 760 I=1,6                             ;CK DONE FLAGS
      IF(IFLAG(I).NE.I) GOTO 10
  760 CONTINUE
      CALL SLOS5 (IWRK,IFILE,LDIR,NU,LNK)
      CALL EFSTXT (MENU(73),IC,1700,1750,0,0,0,ICN,ISC,2,13)
      RETURN
  900 CALL EFSTXT (KERR(41),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;FLG ALRDY SET
      GOTO 10
      END
```

```
      SUBROUTINE SLOS3 (IROW,J,M,N,LFLAG)
C                              REV 01.00
C  SEP 1987                    CHUCK LITTLE          WSFO CAE FTS 677-5501
C  FORTRAN IV/REV 5.57         DG ECLIPSE (S230)     RDOS/REV 6.19
C  PURPOSE:
C     COMPUTES A COL NUMBER IF DATA IS VALID OTRW ERROR MSG SENT TO PGM
C     VIA IROW
C  ARGUMENT LIST:
C     IROW  - ROW NUMBER - IF 0 CURSOR NOT IN A KNOWN LOCATION
C     J     - Y VALUE OF CURSOR IN PIXELS
C     M     - MIN ALLOWED Y VALUE OF 1ST BOX
C     N     - MAX ALLOWED Y VALUE OF 1ST BOX
C     LFLAG - CURRENT COL NUMBER
C
C
      COMMON MENU(500),KERR(400),NAME(400),IDATA0S(22,9),LNT(12)
      IF(LFLAG.EQ.0) GOTO 30
      K0=LNT(LFLAG)
      LN=M
      LN=M
      IROW=0
      DO 10 K=1,K0
      IF(J.GE.LN.AND.J.LE.LN) IROW=K
      IF(IROW.NE.0) GOTO 20
      LN=LN-50
      LN=LN-50
   10 CONTINUE
   20 RETURN
   30 IROW=0
      RETURN
      END



      SUBROUTINE SLOS4 (IROW,ICOL,IWRK,LOCATN)
C                              REV 01.00
C  SEP 1987                    CHUCK LITTLE          WSFO CAE FTS 677-5501
C  FORTRAN IV/REV 5.57         DG ECLIPSE (S230)     RDOS/REV 6.19
C  PURPOSE:
C     DECODE DATA CHOOSEN FROM THE MENU - COLS 1 - 5 AND THE LOCATIONS
C
C  ARGUMENT LIST:
C     IROW   - ROW # OF CURSOR
C     ICOL   - COL # OF CURSOR
C     IWRK   - WORK ARRAY WITH PARTIAL FILENAME OF DATAFILE
C     LOCATN - NAME OF LOCATION
C
C
C
      DIMENSION LOCATN (20),IWRK(20),IWRK1(10)
      COMMON MENU(500),KERR(400),NAME(400),IDATA0S(22,9)
      GOTO (300,400,500,100,200,600), ICOL
  100 GOTO (101,102,103), IROW
  101 IWRK(6)=000114K           ;LS   COL 4 CNTR
      GOTO 999
  102 IWRK(6)=000122K           ;RS
      GOTO 999
  103 IWRK(6)=000060K           ;AT POINT
      IWRK(7)=000060K
      IWRK(8)=000060K
      GOTO 999
  200 IWRK(7)=IDATA0S(IROW,7)   ;COL 5    DIST
      IWRK(8)=IDATA0S(IROW,8)
      GOTO 999
  300 K=1                       ;COL 1 CAT
      CALL UBMDEC (IROW,IWRK1,K)
      IWRK(1)=IWRK1(6)
      GOTO 999
  400 IWRK(2)=IDATA0S(IROW,1)   ;COL 2 DIR
      IWRK(3)=IDATA0S(IROW,2)
      GOTO 999
  500 IWRK(4)=IDATA0S(IROW,3)   ;COL 3 SPD
      IWRK(5)=IDATA0S(IROW,4)
      GOTO 999
  600 NWD=((IROW-1)*14)+1       ;COL 6 NAME/LOCATN
      DO 610 J=1,12
      LOCATN(J)=NAME(NWD)
      MENU(56+J)=LOCATN(J)
      NWD=NWD+1
  610 CONTINUE
  999 RETURN
      END
```

```
      SUBROUTINE SLOSS (IWRK,IFILE,LDIR,NU,LNK)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      TRANSFER DATA IN IWRK INTO IFILE - REMOVING ANY BLANKS AND PACKING.
C      FINISH CREATING DIR:FILNAME FOR LINK TO DATA FILE.
C   ARGUMENT LIST:
C      IWRK   - ARRAY HOLDING FILENAME OF DESIRED DATAFILE
C      IFILE  - NAME OF DESIRED DATAFILE
C      LDIR   - DIR - LOCATION OF DATAFILE - UNPACKED
C      NU     - # OF BYTES IN LDIR
C      LNK    - LINK DIR:FILENAME - LDIR+IFILE PACKED
C
C
C
      COMMON MENU(500),KERR(400),NAME(400),IDATA$S(22,9)
      DIMENSION IWRK(20),IFILE(10),IX(30),IZ(6),LDIR(20),LNK(20)
      DO 1 I=1,NU
      LNK(I)=LDIR(I)
    1 CONTINUE
      LNK(NU)=000072K
      NX=NU+1
      IZ(6)=000015K
      NB=0
      IWRK(NB+1)=000040K
      IWRK(NB+2)=000040K
      GOTO 50
    5 DO 10 J=1,NB
      IF(IWRK(J).EQ.000040K) GOTO 20
   10 CONTINUE
      CALL PACK (IWRK,10,IFILE)
      DO 15 J=1,5
      MENU(80+J)=IFILE(J)
      IZ(J)=IFILE(J)
   15 CONTINUE
      GOTO 70
   20 DO 25 K=J,NB
      IWRK(K)=IWRK(K+1)
   25 CONTINUE
      NB=NB-1
      GOTO 5
   50 CALL UNPACK (MENU(121),24,IX)
      IX(5)=IWRK(1)
      IX(12)=IWRK(2)
      IX(13)=IWRK(3)
      IX(14)=IWRK(4)
      IX(15)=IWRK(5)
      IX(17)=IWRK(6)
      IX(18)=IWRK(7)
      IX(19)=IWRK(8)
      CALL PACK (IX,24,MENU(121))
      GOTO 5
   70 DO 75 I=1,10
      LNK(NX)=IWRK(I)
      NX=NX+1
      IF(IWRK(I+1).EQ.000040K) GOTO 80
   75 CONTINUE
   80 LNK(NX)=000000K
      CALL PACK (LNK,NX,LNK)
X     CALL SPCHR (' ** CURRENT DIR/LINK ENTRY --)',IER)
X     CALL SPCHR (LNK,IER)
X     CALL SPCHR ('(15)',IER)
      RETURN
      END
```

```
      SUBROUTINE SLOS6 (ICN,ISC,LNDFL,IHALT,IFT,ITN)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      READS THE COMMAND LINE FOR THE TIME OF LANDFALL,CONSOLE AND SCREEN
C      TO WRITE THE MENU.
C   ARGUMENT LIST:
C      ICN   - IF /C COMPUTE NEW CONSOLE - DEFAULT = 0
C      ISC   - IF /S COMPUTE NEW SCREEN  - DEFAULT = 2
C      LNDFL - TIME OF LANDFALL
C      IHALT - POINTER TO ERROR MESSAGE-KERR/MENUB-GREATER THAN 400 MENUB
C      IFT   - TIME 12 HRS BFR LANDFALL
C      ITN   - TIME OF LANDFALL - ASCII
C
      COMMON MENU(500),KERR(400),NAME(400),IDATADS(22,9)
      DIMENSION IDA(20),ISW(2),ITN(2)
      COMMON/GARDG/IOUTU(90)
      LNDFL=-1
      IFT=0
      CALL FCOM (II,IER)
      DO 50 J=1,6
      CALL COMCM (II,IDA,N,ISW,IER)
      IF(IER.EQ.9) GOTO 80
      CALL UNPACK (IDA,N,IOUTU)
      N=N-1
      IF(ISWSET(ISW,"C")) GOTO 10 ;CONSOL
      IF(ISWSET(ISW,"S")) GOTO 20 ;SCREEN
      IF(ISWSET(ISW,"L")) GOTO 30 ;LANDFALL TIME
      GOTO 50
 10   CN=CFLTCVT(1,N)
      IF(CN.EQ.999999.) GOTO 903  ;ERROR FM CFLTCVT
      ICN=CN
      GOTO 50
 20   SC=CFLTCVT(1,N)
      IF(SC.EQ.999999.) GOTO 903
      ISC=SC
      GOTO 50
 30   IF(N.NE.4) GOTO 903         ;TIME NOT 4 CHARACTERS
      X=CFLTCVT(1,N)
      IF(X.EQ.999999.) GOTO 903
      LNDFL=X
      IF(LNDFL.GT.2400.OR.LNDFL.LT.0) GOTO 904  ;LANDFALL TIME OUT BOUNDS
      IFT=LNDFL-1200
      IF(IFT.LT.0) IFT=LNDFL+1200
      ITN(1)=IDA(1)               ;TN OF LNDFL
      ITN(2)=IDA(2)
 50   CONTINUE
 80   IF(LNDFL.EQ.-1) GOTO 901
 85   RETURN
901   IHALT=241  ;NO LNDFL TIME
      GOTO 85
903   IHALT=121  ;TROL READING COMMAND LINE
      GOTO 50
904   IHALT=141  ;LNDFL OUT OF BOUNDS
      GOTO 50
      END
```

```
      SUBROUTINE SLOS7 (IC,ICN,ISC,IXE,IYE,SRG,NROW,LOCATN,IDATA,IDATA1,TIDWD)
C                              REV 01.00
C   OCT 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      CONVERTS ASCII TO BINARY IN THE ARRAY - IDATA - FOR THE SELECTED
C      LOCATION.
C   ARGUMENT LIST:
C      IC,ICN,ISC - CHANNEL TO GDM,CONSOLE,SCREEN
C      IXE,IYE    - PIXELS TO PLOT STATUS MESSAGES
C      SRG        - ARRAY CONTAINING DATA FOR ONE LOCATION
C      NROW       - LINE NO IN DATAFILE OF DESIRED LOCATION
C      LOCATN     - NAME OF DESIRED LOCATION
C      IDATA      - UNPACKED DATAFILE
C      IDATA1     - POINTERS FOR EACH ROW OF DATA IN IDATA
C      TIDWD      - TIDE DATA
C   EXITS:
C      SLOSX -
C
      COMMON MENU(500),KERR(400),NAME(400),IDATA8S(22,9)
      DIMENSION SRG(-2:50),LOCATN(20),IDATA(7168),IDATA1(30,2)
      DIMENSION IWK(42),IX(20),TIDWD(30,2)
      COMMON/QARDQ/IOUTU(90)
      CALL UNPACK (LOCATN,30,IWK)
      NROW=NROW+1
      IB=IDATA1(NROW,1)
      IE=IDATA1(NROW,2)
      NU=1
      DO 20 J=IB,IE
      IF(IDATA(J).EQ.000054K) GOTO 22   ;1ST COMMA
      IF(IDATA(J).NE.IWK(NU)) GOTO 920   ;NAME/LOCATION MISMATCH
X      KCHR=IWK(NU)
X      CALL PCHAR (KCHR,IER)
      NU=NU+1
  20  CONTINUE
  22  K=J
      NO=-1
  25  JSTRT=K+1
      JEND=JSTRT+6
      JCHR=0
      DO 40 K=JSTRT,JEND
      IF(IDATA(K).EQ.000054K.OR.IDATA(K).EQ.000015K.OR.IDATA(K).EQ.
     * 000012K) GOTO 50
      JCHR=JCHR+1
      IOUTU(JCHR)=IDATA(K)
  40  CONTINUE
  50  NO=NO+1
      SRG(NO)=CFLTCVT(1,JCHR)
      IF(SRG(NO).EQ.999999.) GOTO 900
      IF(IDATA(K).EQ.000015K.OR.IDATA(K).EQ.000012K) GOTO 75
      GOTO 25
  75  SRG(-2)=TIDWD(NROW,1)
      SRG(-1)=TIDWD(NROW,2)
      RETURN
 900  IHALT=101
      GOTO 990
 920  IHALT=261
 990  CALL SLOSX (IC,ICN,ISC,IHALT)
      END
```

```
      SUBROUTINE SLOSB (IBUF,IBT,IDIR,ICTY,LDIR,IHALT,NU,IER0)
C                               REV 01.00
C  SEP 1987                     CHUCK LITTLE          VSFO CAE FTS 677-5501
C  FORTRAN IV/REV 5.57          DG ECLIPSE (S230)     RDOS/REV 6.19
C  PURPOSE:
C     COMPUTES # OF CITIES IN DATAFILE AND READS DIRECTORY INFO FROM
C     SLOS.DT
C  ARGUMENT LIST:
C     IBUF - BUFFER WITH DATA FROM SLOSH.DT
C     IBT  - # OF BYTES IN IBUF
C     IDIR - DIR/SUBDIR FOR INIT
C     ICTY - # OF CITIES IN DATAFILE
C     LDIR - DIR - LOCATION OF DATAFILE - UNPACKED
C     IHALT- POINTER TO ERROR MSG-KERR/MENUB-GREATER THAN 400 MENUB
C     NU   - # OF BYTES IN LDIR
C     IER0 - ERROR 1 = 0000, 0 WILL HALT PGM IN SLOS1
C
C
      COMMON MENU(500),KERR(400),NAME (400),IDATA06(22,9),LNT(12)
      COMMON/BARDQ/IOUTU(90)
      DIMENSION IDIR(20),LDIR(20),IBUF(60),IWRK(30)
      DO 5 I=1,20
      IDIR(I)=000040K
      LDIR(I)=000040K
    5 CONTINUE
      CALL UNPACK (IBUF,IBT,IOUTU)
      CTY=CFLTCVT(1,2)
      IF(CTY.EQ.999999.) GOTO 900       ; BAD DATA CONVERSN
      IF(CTY.GT.24.) GOTO 910           ;TOO MANY CITIES
      ICTY=CTY                          ;NO OF CITIES IN DATAFILE
      DO 10 I=9,25
      IF(IOUTU(I).EQ.000052K) GOTO 20   ;LK FR -
   10 CONTINUE
      GOTO 920                          ;TROL READING INPUT DIR
   20 NO=1
      NU=1
      J=I+1
      DO 25 I=J,40
      IWRK(NO)=IOUTU(I)
      LDIR(NU)=IOUTU(I)
      IF(IWRK(NO).EQ.000040K) GOTO 30   ;DONE
      NO=NO+1
      NU=NU+1
      IF(LDIR(NU-1).EQ.000072K) NU=1    ;REMOV : IF PRESENT
   25 CONTINUE
      GOTO 920
   30 NO=NO+1
      IWRK(NO)=000000K
      CALL PACK (IWRK,NO,IDIR)
      CALL INIT (IDIR,0,IER)
      IF(IER.EQ.1) GOTO 40  ;GD CALL
      IF(IER.EQ.40) GOTO 40 ;DIR ALRDY IN SYS
      GOTO 930              ;CANT INIT
   40 LDIR(NU)=000000K
      IER0=1
      RETURN
  900 IHALT=421
      GOTO 999
  910 IHALT=441
      GOTO 999
  920 IHALT=461
      GOTO 999
  930 IHALT=481
  999 IER0=0
      RETURN
      END
```

```
      SUBROUTINE SLOS9 (IC,ICN,ISC,IXE,IYE,IHITD,J,ITD)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      COMPUTE TIME OF HI TIDE FROM MENU & DISPLAY STATUS MSGS/ERRORS.
C   ARGUMENT LIST:
C      IC,ICN,ISC - CHANNEL TO GDM,CONSOLE,SCREEN
C      IXE,IYE    - POSITION OF ERROR MSGS
C      IHITD      - TIME OF HI TIDE - IN THE 12 HRS BFR LANDFALL
C      J          - PIXEL LOCATION OF CURSOR
C      ITD        - TIME OF HI TIDE - ASCII
C   EXITS:
C      SLOSX
C
C
      COMMON/BARDB/IOUTU(90)
      COMMON MENU(500),KERR(400),NAME(400),IDATA0S(22,9),LNT(12)
      DIMENSION ITD(2),IXX(4),IA(4)
      IX=0
      LFLAG=7
    5 CALL SLOS3 (IRON,J,2850,2800,LFLAG)
      IF(IRON.EQ.0) GOTO 91
      IA(1)=IDATA0S(IRON,5)
      CALL UNPACK(IA,2,IXX)
      IOUTU(1)=IXX(1)
      IOUTU(2)=IXX(2)
      IOUTU(3)=000040K
      IOUTU(4)=000040K
      X=CFLTCVT(1,4)
      IF(X.EQ.999999.) GOTO 90
      CALL PACK (IOUTU,4,ITD)
      MENU(9)=ITD(1)
      MENU(10)=ITD(2)
      CALL EFSTXT (MENU(1),IC,1700,1450,0,0,0,ICN,ISC,3,10)
      IHITD=X
      RETURN
   90 NUM=101
      CALL EFSTXT (KERR(NUM),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;TRBL TIDE
      IF(IX.EQ.2) GOTO 97
      GOTO 95
   91 CALL EFSTXT (KERR(21),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;ERR IN BOX
   95 CALL CURSR (IC,ICN,ISC,1,J,IER)
      CALL EFSTXT (KERR(1),IC,IXE,IYE,0,0,0,ICN,ISC,3,20) ;CLR ERR
      IX=IX+1
      GOTO 5
   97 IHALT=01
      CALL SLOSX (IC,ICN,ISC,IHALT)
      END
```

```
      SUBROUTINE SLOSA (IC,ICN,LNDFL,IFT,INITD,LOCATN,SRG,FB)
C                              REV 01.00
C  SEP 1987                    CHUCK LITTLE          WSFO CAE FTS 677-5501
C  FORTRAN IV/REV 5.57         DG ECLIPSE (S230)     RDOS/REV 6.19
C  PURPOSE:
C     DRIVER ROUTINE FOR DISPLAYING COMPUTED DATA/GRAPH OF TIDE/SURGE
C  ARGUMENT LIST:
C     IC,ICN - CHANNEL TO GDN, CONSOLE
C     LNDFL  - TIME OF LANDFALL
C     IFT    - TIME 12 HRS BEFOR LANDFALL
C     INITD  - TIME OF HI TIDE
C     LOCATN - NAME OF LOCATION
C     SRG    - ARRAY WITH DATA FOR ONE LOCATION
C     FB     - FT MSL TO PLOT FLOOD BEGIN LINES
C
C
C
C
      COMMON MENU(500),KEYR(400),NAME(400),IDATABS(22,9)
      DIMENSION SRG(-2:50)
      DIMENSION IFILE(10),LOCATN(20),TIDE(0:60),TSURGE(0:60),IB(10)
      ISC=1                    ;SCREEN FOR GRAPH
      ML=SRG(-2)+0.5           ;MLW
      IQ=0
      IOV=3
      DO 5 I=1,3
      CALL EFCLR (IC,ICN,ISC,I)
    5 CONTINUE
      ITM=IFT/100
      IF(ITM.EQ.0) ITM=24
      IYB=200                  ;Y BASE
      IYT=2800                 ;Y TOP
      IXL=200                  ;LEFT SIDE  X
      IXR=2800                 ;RIGHT SIDE X
      DX=(IXR-IXL)/8.          ;DIST BTWN 3 HRS
      IXM=4*DX+IXL             ;MID PT OF X
      IX=IXL
      ITMY=IYB-50
      DO 15 J=1,9
      CALL EFLINE (IC,IX,IYB,IX,IYT,0,0,ICN,ISC,IOV) ;PLT TIM LN
      ITMX=IX-25
      ITZ=ITM+264
      CALL EFTXT (MENU(ITZ),IC,ITMX,ITMY,0,0,0,ICN,ISC,IOV,1) ;LABEL TIM
      IX=DX*J+IXL
      ITM=ITM+3
   15 CONTINUE
      CALL SLOSB(INITD,IFT,IYMAX,IYMIN,MOD,TIDE,TSURGE,SRGMX,SRGMN,SRG)
      CALL SLOSC(IC,ICN,ISC,IQ,IYB,IYT,IXL,IXR,DX,MOD,IYMAX,IYMIN,TIDE,TSURGE
     *,FB,ML)
      CALL SLOSD(LNDFL,DX,IXL,IXM,IYB,IC,ICN,ISC,IQ,SRG)
      IX=3000
      CALL EFTXT (MENU(241),IC,IX,2550,IQ,IQ,IQ,ICN,ISC,IOV,2);LABEL GRAPH
      CALL EFTXT (MENU(243),IC,IX,2450,IQ,IQ,IQ,ICN,ISC,IOV,6)
      CALL EFTXT (MENU(25),IC,IX,2100,IQ,IQ,IQ,ICN,ISC,IOV,10)
      CALL EFTXT (MENU(1),IC,IX,2000,IQ,IQ,IQ,ICN,ISC,IOV,10)
      CALL EFTXT (MENU(169),IC,IX,1700,IQ,IQ,IQ,ICN,ISC,IOV,8)
      CALL EFTXT (MENU(177),IC,IX,1600,IQ,IQ,IQ,ICN,ISC,IOV,8)
      CALL EFTXT (MENU(193),IC,IX,1500,IQ,IQ,IQ,ICN,ISC,IOV,8)
      CALL EFTXT (MENU(201),IC,IX,1400,IQ,IQ,IQ,ICN,ISC,IOV,8)
      CALL EFTXT (MENU(25),IC,1350,75,IQ,IQ,IQ,ICN,ISC,IOV,6)
      CALL EFTXT (MENU(145),IC,1200,2960,IQ,IQ,IQ,ICN,ISC,IOV,23)
      CALL EFTXT (MENU(121),IC,1200,2900,IQ,IQ,IQ,ICN,ISC,IOV,12)
      CALL EFTXT (LOCATN,IC,1200,2840,IQ,IQ,IQ,ICN,ISC,IOV,12)
      CALL EFTXT (MENU(249),IC,100,2860,3,IQ,IQ,ICN,ISC,IOV,2)
      CALL EFTXT (MENU(251),IC,2700,2860,3,IQ,IQ,ICN,ISC,IOV,2)
      CALL EFLINE (IC,3400,2564,3730,2564,IQ,IQ,ICN,ISC,1)
      CALL EFLINE (IC,3400,2464,3730,2464,IQ,IQ,ICN,ISC,2)
      CALL SLOSE (SRGMX,IB)
      DO 20 J=1,3
      MENU(224+J)=IB(J)
   20 CONTINUE
      CALL SLOSE (SRGMN,IB)
      DO 21 J=1,3
      MENU(230+J)=IB(J)
   21 CONTINUE
      CALL EFTXT (MENU(217),IC,IX,1200,IQ,IQ,IQ,ICN,ISC,IOV,5)
      CALL EFTXT (MENU(222),IC,IX,1100,IQ,IQ,IQ,ICN,ISC,IOV,6)
      CALL EFTXT (MENU(228),IC,IX,1000,IQ,IQ,IQ,ICN,ISC,IOV,6)
X     TYPE ' ---MAX/MIN SURGE--ML-',SRGMX,SRGMN,ML
      RETURN
      END
```

```
      SUBROUTINE SLOSB (IHITD,IFT,IYMAX,IYMIN,NOD,TIDE,TSURGE,SRGMX
     *,SRGMN,SRG)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      COMPUTES NORMAL TIDE BASED ON DATA IN IFILE, ADDS TIDE & SURGE,
C      AND FINDS MAX/MIN STORM SURGE
C   ARGUMENT LIST:
C      IHITD       - TIME OF HI TIDE
C      IFT         - TIME 12 HRS BEFOR LANDFALL
C      IYMAX,IYMIN - TOP/BOTTOM (FT) OF GRAPH
C      NOD         - NO OF DIVISIONS-VERT
C      TIDE        - ARRAY WITH TIDE VALUES
C      TSURGE      - ARRAY WITH TIDE+SURGE VALUES
C      SRGMX,SRGMN - MAX/MIN TIDE+SURGE
C      SRG         - ARRAY WITH DATA FOR ONE LOCATION
      DIMENSION TIDE(0:60),TSURGE(0:60),SRG(-2:50)
      TC=SRG(-1)        ;TID DEP FROM REF POINT--DATAFL
      TR=SRG(-2)        ;RANGE OF TIDE--DATAFL
      HI=IHITD+TC
      DT=HI-IFT
      IF(DT.LT.0) DT=2400.-IFT+HI
      DO 10 J=0,48
      TIDE(J)=COS(((2500.-DT)/2500.)*12.5663706+J*.251327412)*TR
 10   CONTINUE
      IYMAX=10
      IYMIN=-5
      SRGMX=0.0
      SRGMN=10.0
      DO 20 J=0,24
      TSURGE(J)=TIDE(J*2)+(SRG(J)/10.)
      IF(TSURGE(J).GT.IYMAX) IYMAX=TSURGE(J)+1.0
      IF(TSURGE(J).LT.IYMIN) IYMIN=TSURGE(J)-1.0
      IF(TSURGE(J).GT.SRGMX) SRGMX=TSURGE(J)
      IF(TSURGE(J).LT.SRGMN) SRGMN=TSURGE(J)
 20   CONTINUE
      NOD=IYMAX+IABS(IYMIN)
      RETURN
      END
```

```
      SUBROUTINE SLOSC (IC,ICN,ISC,I0,IYB,IYT,IXL,IXR,DX,NOD,IYMAX,IYMIN,
     * TIDE,TSURGE,FB,ML)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      PLOT - MST LINES (FT) ON GRAPH, TIDE, TIDE AND SURGE, FLOOD BEGIN
C      LINES AND LABEL MST LINES
C   ARGUMENT LIST:
C      IC,ICN,ISC  - CHANNEL TO GDR, CONSOLE, SCREEN
C      I0          - USED FOR GRAPHIC ROUTINE - 0
C      IYB,IYT     - BOTTOM-TOP OF GRAPH-PIXELS
C      IXL,IXR     - RIGHT-LEFT OF GRAPH-PIXELS
C      DX          - DIFF IN PIXELS FOR 3 HRS-DIR FOR 1 HR RTND
C      NOD         - NO OF DIVISION- VERT
C      IYMAX,IYMIN - TOP-BOTTOM (FT) OF GRAPH
C      TIDE        - ARRAY WITH TIDE VALUES
C      TSURGE      - ARRAY WITH TIDE+SURGE VALUES
C      FB          - FT MSL TO PLOT FLOOD BEGIN LINES
C      ML          - DIFF BTWN MLW AND MSL
C
C
C
      DIMENSION TIDE(0:60),TSURGE(0:60),IW(2)
      COMMON/CDL/MENUO(-8:8),MENUP(-8:8)
      DATA MENUO(-8)/'40353025201510-5 0+5101520253035540'/
      DATA MENUP(-8)/'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX'/
      IZERO=IABS(IYMIN)       ;ZERO MSL
      Z=NOD
      DY=(IYT-IYB)/Z          ;SCALE
      NODO=NOD+1              ;NO OF HORIZ LINES
      IY=IYB
      IOV=3
      DO 10 J=1,NODO    ;PLT MST LINES-OVLY 3
      CALL EFLINE (IC,IXL,IY,IXR,IY,I0,I0,ICN,ISC,IOV)
      IY=DY*J+IYB
      IF(J.EQ.IZERO) GOTO 9  ;GET PIXEL LOCTN OF ZERO LINE
      GOTO 10
    9 IX0=IXL
      IY0=IY
   10 CONTINUE
      DX=DX/3.                ;FOR HRLY PLOTS
      IOV=1
      DO 30 J=0,24            ;PLT TIDE-OVLY 1
      IY=IY0+TIDE(J*2)*DY
      IX=DX*J+IXL
      IF(J.EQ.0) GOTO 25
      CALL EFLINE (IC,IX,IY,IX1,IY1,I0,I0,ICN,ISC,IOV)
   25 IX1=IX
      IY1=IY
   30 CONTINUE
      IOV=2
      DO 40 J=0,24          ;PLT TIDE+SURGE-OVLY 2
      IY=IY0+TSURGE(J)*DY
      IX=DX*J+IXL
      IF(J.EQ.0) GOTO 35
      CALL EFLINE (IC,IX,IY,IX1,IY1,I0,I0,ICN,ISC,IOV)
   35 IX1=IX
      IY1=IY
   40 CONTINUE
```

```
      IOV=3
      IF(FB.GT.100) GOTO 51 ;PLT HASH FOR FLOOD PLN
      ISP=100               ;PIXELS BETWEEN LINES
      LOOP=(IXR-IXL)/ISP
      IY=IYO+FB*DY
      IY1=IYO+((FB+0.5)*DY)
      IX=IXL
      IX1=IXL+ISP
      DO 50 J=1,LOOP
      CALL EFLINE (IC,IX,IY,IX1,IY1,IO,IO,ICN,ISC,IOV)
      IX=IX1
      IX1=IX1+ISP
50    CONTINUE
51    ILO=IYMIN/5.      ;LABEL HGT LINES MLW/MSL
      IHI=IYMAX/5.
      IF(ILO.LT.-8) ILO=-8
      IF(IHI.GT.8) IHI=8
      CALL SLOSF (ML,ILO,IHI)
      IX=IXL-145
      IZ=IXR+25
      DO 60 I=ILO,IHI
      IY=IYO+(I*5*DY)-40
      IW(1)=MENUD(I)
      CALL EFTXT (IW,IC,IX,IY,3,10,10,ICN,ISC,IOV,1)
      IW(1)=MENUP(I)
      CALL EFTXT (IW,IC,IZ,IY,3,10,10,ICN,ISC,IOV,1)
60    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SLOSD (LNDFL,DX,IXL,IXN,IYB,IC,ICN,ISC,IQ,SRG)
C                              REV 01.00
C  SEP 1987                    CHUCK LITTLE          WSFO CAE FTS 477-5501
C  FORTRAN IV/REV 5.57         DG ECLIPSE (S230)     RDOS/REV 6.19
C  PURPOSE:
C     COMPUTES/PLOTS THE TIME OF BEGINNING/ENDING OF HURR/GALE WNDS
C  ARGUMENT LIST:
C     LNDFL      - TIME OF LANDFALL
C     DX         - DIFF IN PIXELS FOR 1 HR
C     IXL        - LEFT SIDE OF GRAPH - PIXELS
C     IXN        - MID POINT OF X AXIS - PIXELS
C     IYB        - BASE OF GRAPH - PIXELS
C     IC,ICN,ISC - CHANNEL TO GDN, SCREEN, CONSOLE
C     IQ         - USER FOR GRAPHIC ROUINE - 0
C     SRG        - ARRAY WITH DATA FOR ONE LOCATION
C
C
C
      DIMENSION IWRK(5),IB(6),SRG(-2:50)
      COMMON MENU(500),KERR(400),NAME(400),IDATADS(22,9)
      IOV=3
      DO 90 I=1,4
      IF(I.EQ.1) ND=175
      IF(I.EQ.2) ND=183
      IF(I.EQ.3) ND=199
      IF(I.EQ.4) ND=207
      ITIM=LNDFL
      IWRK(1)=000060K
      IWRK(2)=000060K
      IWRK(3)=000060K
      IWRK(4)=000060K
      TH=SRG(I+24)/100.
      IF(TH.LT.0) TH=(AMOD(SRG(I+24),100.))-.5
      IF(TH.GE.0) TH=(AMOD(SRG(I+24),100.))+.5
      IHH=TH
      IMM=TH
      IH=ITIM/100
      IM=0
      IF(SRG(I+24).EQ.99.) GOTO 20   ;NONE
      IF(SRG(I+24).LT.0.)  GOTO 30   ;BFR LNDFL
      IF(SRG(I+24).GE.0.)  GOTO 40   ;LNDFL OR AFT
20    MENU(ND)='ND'
      MENU(ND+1)='NE'
      GOTO 90
30    IH=IH-1
      IM=60
40    IWH=IH+IHH
      IWM=IM+IMM
      IF(IWM.LT.60) GOTO 45
      IWH=IWH+1
      IWM=IWM-60
45    IF(IWH.LT.0) IWH=IWH+24
      IF(IWH.GE.24)IWH=IWH-24
      ITIM=(IWH*100)+IWM
      K=1
      CALL UBINDEC (ITIM,IB,K)
      IF(ITIM.GE.1000) GOTO 55
      IF(ITIM.GE.100)  GOTO 56
      IF(ITIM.GE.10)   GOTO 57
      GOTO 58
```

```
 55   IWRK(1)=IB(3)
 56   IWRK(2)=IB(4)
 57   IWRK(3)=IB(5)
 58   IWRK(4)=IB(6)
      CALL PACK (IWRK,4,IB)
      MENU(NO)=IB(1)
      MENU(NO+1)=IB(2)
      TH=IHH+IMN/60.
      IY=IYB+100
      GOTO (70,75,79,80),I
 70   IX=(IXM+TH*DX)-5
      GOTO 86
 75   IY=IY+50      ;HUR BARB
      GOTO 70
 79   IY=IY+50
 80   IX=(IXM+TH*DX)+5
 86   DO 88 J=1,10,2
      CALL EFLINE (IC,IX,IY,IX,IYB,IB,IB,ICW,ISC,IDV)
      IX=IX+3
 88   CONTINUE
 90   CONTINUE
      RETURN
      END



      SUBROUTINE SLOSE (SRB,IB)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      CONVERT SRB TO ASCII AND PUT INTO ARRAY IB.
C   ARGUMENT LIST:
C      SRB - MAX/MIN SURGE VALUE
C      IB  - ASCII MAX/MIN SURGE VALUE-PACKED
C
C
      DIMENSION IB(10),IWRK(10)
      ISB=SRB*10.
      DO 5 J=1,6
      IWRK(J)=000040K
   5  CONTINUE
      IWRK(1)=000053K
      IWRK(4)=000060K
      K=1
      CALL UBADEC (ISB,IB,K)
      IF(ISB.GE.100) GOTO 10
      IF(ISB.GE.10) GOTO 20
      IF(ISB.GE.0) GOTO 30
      IWRK(1)=000040K
      IWRK(2)=IB(3)
  10  IWRK(3)=IB(4)
  20  IWRK(4)=IB(5)
  30  IWRK(5)=000056K
      IWRK(6)=IB(6)
      CALL PACK (IWRK,6,IB)
      RETURN
      END
```

```
      SUBROUTINE SLOSF (ML,ILO,IHI)
C                              REV 01.00
C   SEP 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      TO GENERATE ASCII CODE FOR MLW LABELS
C   ARGUMENT LIST:
C      ML  - DIFF BTWN MLW AND MSL
C      ILO - LOWEST HGT LINE TO LABEL
C      IHI - HIGHEST HGT LINE TO LABEL
C
C
C
      DIMENSION IX(10),IWK(10)
      COMMON/CDL/MENUD(-8:8),MENUP(-8:8)
      DO 40 I=ILO,IHI
      MLW=I*5+ML
      K=1
      CALL UBMDEC (MLW,IX,K)
      IF(MLW.GE.10.OR.MLW.LE.-10) GOTO 10
      IWK(1)=000040K
      IWK(2)=IX(6)
      IF(MLW.LT.0) IWK(1)=000055K
      IF(MLW.GT.0) IWK(1)=000053K
      GOTO 20
 10   IWK(1)=IX(5)
      IWK(2)=IX(6)
 20   CALL PACK (IWK,2,IX)
      MENUP(I)=IX(1)
 40   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SLOSG (IFILE,IDATA,IDATA1,IER)
C                              REV 01.01
C   OCT 1987                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57        DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      READ THE INPUT FILE =IFILE= INTO THE ARRAY IDATA. DATA IN IDATA IN
C      UNPACKED FORMAT.
C   ARGUMENT LIST:
C      IFILE  - NAME OF DESIRED DATAFILE
C      IDATA  - UNPACKED DATAFILE
C      IDATA1 - POINTERS FOR EACH ROW OF DATA IN IDATA
C      IER    - ERROR TYPE - 1 FOR GOOD CALL  - 0 FOR ERROR
C
C
      DIMENSION IBUF(3584),IDATA(7168),IDATA1(30,2),IX(200),IFILE(10)
      DO 5 I=1,30
      IDATA1(I,1)=1
      IDATA1(I,2)=1
    5 CONTINUE
      IBLK=0
      NBLK=7
      CALL GCHN (IZ,IER)
      CALL OPENN (IZ,IFILE,0,IER)
      IF(IER.NE.1) GOTO 900        ;CANT FIND FILE
      CALL RDB (IZ,IBUF,IBLK,NBLK,IER)
      IBT=NBLK*512
      CALL UNPACX (IBUF,IBT,IDATA)
      K=1
      LNNO=1
      IBS=1
      IBE=IBT
      IDATA1(LNNO,1)=1       ;BNG 1ST LN
      DO 60 J=IBS,IBE
      IF(LNNO.EQ.29) GOTO 65
      IF(IDATA(K).EQ.000203K) GOTO 65
      IF(IDATA(K).EQ.000012K.OR.IDATA(K).EQ.000015K) GOTO 40
      GOTO 59
   40 IDATA1(LNNO,2)=K-1                ;EOL
   41 IF(IDATA(K+1).EQ.000012K.OR.IDATA(K+1).EQ.000015K) GOTO 45
      K=K+1
      GOTO 55
   45 IF(IDATA(K+2).EQ.000012K.OR.IDATA(K+2).EQ.000015K) GOTO 50
      K=K+2
      GOTO 55
   50 K=K+2
      GOTO 41
   55 LNNO=LNNO+1             ;NEW LINE
      IDATA1(LNNO,1)=K       ;BOL
      IF(IDATA(K).EQ.000105K.AND.IDATA(K+1).EQ.000116K.AND.IDATA(K+2)
     * .EQ.000104K) GOTO 65
   59 K=K+1
   60 CONTINUE
   65 IER=1
      CALL KLOSE (IZ,IER)
      RETURN
  900 IER=0
      CALL KLOSE (IZ,IER)
      RETURN
      END
```

```
      SUBROUTINE SLOSI (TIDMD,IB,IHALT)
C                          REV 01.00
C   OCT 1987               CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV/REV 5.57    DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      READS SLOS.DT FOR THE MEAN TIDE & AND TIME OF DEPARTURE FROM A
C      KNOWN POINT.
C   ARGUMENT LIST:
C      TIDMD - ARRAY WITH TIDE DATA
C      IB    - CHANNEL TO SLOS.DT
C      IHALT - POINTER TO ERROR MESSAGE-KERR/MENUB-GREATER THAN 400 MENUB
C
C
C
      DIMENSION TIDMD (30,2),IBUF(65)
      COMMON/QARDQ/ IOUTU(90)
      N1=2
      N2=13
      DO 60 L=1,2
      DO 50 K=1,2
      CALL RDL (IB,IBUF,IBT,IER)
      CALL UNPACK (IBUF,IBT,IOUTU)
      NS=1
      DO 30 M=N1,N2
      TIDMD(M,K)=CFLTCVT(NS,4)
      NS=NS+5
      IF(TIDMD(M,K).EQ.999999.) GOTO 900
 30   CONTINUE
 50   CONTINUE
      N1=14
      N2=25
 60   CONTINUE
      RETURN
900   IHALT=61
      RETURN
      END
```

```
      SUBROUTINE SLOSX (IC,ICN,ISC,IHALT)
C                             REV 01.00
C   SEP 1987                  CHUCK LITTLE          WSFO CAE FTS 677-5501
C   FORTRAN IV.REV 5.57       DG ECLIPSE (S230)     RDOS/REV 6.19
C   PURPOSE:
C      PROCESS ALL ERROR MESSAGES THAT WILL HALT THE PROGRAM
C   ARGUMENT LIST:
C      IC,ISC,ICN - CHANNEL TO GDN,CONSOLE AND SCREEN
C      IHALT      - POINTER TO ERROR MESSAGE-KERR/MENUB-GREATER THAN 400 MENUB
C
C
      COMMON/COL/MENUB(-8:8),MENUP(-8:8),MENUB(200)
      COMMON MENU(500),KERR(400),NAME(400),IDATADB(22,9),LNT(12)
      DATA MENUB(1) /'PROGRAM STOP!!  TRBL OPENING SLOS.DT   !'/
      DATA MENUB(21)/'PROGRAM STOP!!  CANT READ # OF LOCATIONS'/
      DATA MENUB(41)/'PROGRAM STOP!!  TOO MANY LOCATIONS     !'/
      DATA MENUB(61)/'PROGRAM STOP!!  TRBL READING DIRECTORY !'/
      DATA MENUB(81)/'PROGRAM STOP!!  CANNOT INIT DIRECTORY  !'/
X     TYPE 'SLOSX,IHALT',IHALT
      IOA=1
      K0=0
      DO 10 I=1,3
      CALL EFCLR (IC,ICN,ISC,I)
 10   CONTINUE
      IXE=500
      IYE=2000
      IF(IHALT.GE.400) GOTO 30
      CALL EFSTXT (KERR(IHALT),IC,IXE,IYE,3,K0,K0,ICN,ISC,IOV,20)
 15   CALL KLOSE (IC,IER)
      CALL EXIT
 30   IHALT=IHALT-400
      CALL EFSTXT (MENUB(IHALT),IC,IXE,IYE,3,K0,K0,ICN,ISC,IOV,20)
      GOTO 15
      END
```

```
      REAL FUNCTION CFLTCVT (IBGN,N)
      COMMON/QARDQ/IOUTU(80)
C     THIS FUNCTION IS USED WITH THE SUBROUTINE AFREAD. ASCII CHARACTERS IN
C     THE CURRENT LINE ARE SCANNED AND INTERPRETED AS REAL NUMBERS. IF NO
C     DECIMAL POINT IS DETECTED, IT IS ASSUMED TO FOLLOW THE LAST NUMERAL IN
C     THE FIELD. THE SCAN BEGINS WITH CHARACTER IBGN. N CHARACTERS ARE
C     SCANNED.
C     ----------FLTCVT CHANGED TO CFLTCVT----------
C     STATEMENT NUMBER 800 AND THE FOLLOWING STATEMENT WERE CHANGED TO ALLOW
C     ERROR PROCESSING.
C
      LOGICAL NEG
      CFLTCVT=0.
      NEG=.FALSE.
      IEND=IBGN+N-1
  100 IF(IOUTU(IEND).NE.32) GOTO 200
      IF(IEND.EQ.IBGN) RETURN
      IEND=IEND-1
      GOTO 100
  200 DO 250 I=IBGN,IEND
      IF(IOUTU(I).NE.32) GOTO 300
  250 CONTINUE
      RETURN
  300 IF(IOUTU(I).EQ.43) GOTO 400
      IF(IOUTU(I).NE.45) GOTO 500
      NEG=.TRUE.
  400 I=I+1
  500 J=I
      DO 600 I=J,IEND
      IF(IOUTU(I).EQ.32) IOUTU(I)=48
      IF(IOUTU(I).LT.48.OR.IOUTU(I).GT.57) GOTO 700
      CFLTCVT=CFLTCVT*10+IOUTU(I)-48
  600 CONTINUE
      IF(NEG)CFLTCVT=-CFLTCVT
      RETURN
  700 IF(IOUTU(I).NE.46) GOTO 800
      J=I+1
      DIV=10.
      DO 750 I=J,IEND
      IF(IOUTU(I).EQ.32) IOUTU(I)=48
      IF(IOUTU(I).LT.48.OR.IOUTU(I).GT.57) GOTO 800
      CFLTCVT=CFLTCVT+(IOUTU(I)-48)/DIV
      DIV=DIV*10
  750 CONTINUE
      IF(NEG)CFLTCVT=-CFLTCVT
      RETURN
  800 CFLTCVT=999999.
      RETURN
      END
```

```
      SUBROUTINE CURSR(RCHN,CON,CHNL,X,Y,IER)
C
      INTEGER RCHN,CON,CHNL,X,Y,IBUF(6)
C
C *** SETUP FUNCTION WORD
C
      IBUF(1) = 120000K
      IBUF(1) = IOR(IBUF(1),ISHFT(CON,8))
      IBUF(1) = IOR(IBUF(1),ISHFT(CHNL,6))
C
C *** SETUP CHANNEL ID AND CONSOLE NUMBER
C
      IBUF(2) = 0K
      IBUF(2) = IOR(IBUF(2),ISHFT(CHNL,14))   ;CHANNEL NUMBER
      IBUF(2) = IOR(IBUF(2),ISHFT(CON,8))     ;CONSOLE NUMBER
      IBUF(2) = IOR(IBUF(2),RCHN)             ;RDOS CHANNEL
C
C *** READ CURSOR
C
      CALL RDCUR(IBUF,IER)
      IF(IER.NE.1) RETURN
C
C *** SEPERATE ZOOM VALUE
C
      IZOOM = ISHFT(IBUF(3),-12)
      IF(IZOOM.EQ.0) GO TO 10
C
C *** TAKE OUT ZOOM VALUE TO GET X
C
      CALL ICLR(IBUF(3),14)
      CALL ICLR(IBUF(3),13)
      CALL ICLR(IBUF(3),12)
C
C *** CALCULATE X,Y IN IMAGE COORDINATES
C
      X = (IBUF(3) - 2047)/(IZOOM + 1) + 2047 - IBUF(5)
      Y = (IBUF(4) - 1535)/(IZOOM + 1) + 1535 - IBUF(6)
      RETURN
C
   10 X = IBUF(3)
      Y = IBUF(4)
      RETURN
C
      END
```

```
;*************************************************************
;                                        RDCUR.SR
;     SUBROUTINE(IBUF,IER)
;     DIMENSION IBUF(6)
;     FORTRAN SUBROUTINE TO READ ENTER CURSOR BUTTON
;     ON GDN OF AFOS SYSTEM. TOTALLY INDEPENDANT OF
;     AFOS RUNNING..
; INPUT DATA
;     IBUF(1) - FUNCTION WORD FOR INTERROGATE CURSOR
;               I.E. 120100  LOCATION OF CURSOR ON
;               CHANNEL 1 CONSOLE 0
;     IBUF(2) - CONSOLE ID ...BITS 4-7
;               CHANL NUM ...BITS 1-3
;               RDOS CHANNEL NUM ...BITS 8-15
; RETURNED FROM CALL
;     IBUF(2) - NUMBER OF WORDS ALWAYS 4
;     IBUF(3) - ZOOM ...BITS 1-3
;               X LOCATION OF CURSOR ...BITS 4-15
;     IBUF(4) - Y LOCATION OF CURSOR ...BITS 4-15
;     IBUF(5) - X TRANSLATION ...BITS 4-15
;     IBUF(6) - Y TRANSLATION ...BITS 4-15
; FOR DEFINITION OF ABOVE VARIABLES SEE ...PROGRAMMER'S
;   REFERENCE GUIDE FOR GDN AGN     FACC.
; PROGRAM WRITTEN BY                   GERY GROSFELD - MSD
;                                      DON LAURINE - SLC/RFC
;                        12/19/79
;*************************************************************
       .TITL RDCUR
       .ENT  RDCUR
       .EXTD .CPYL,.FRET
       .NREL
       .DIO  .RDC=31400
       IBUF=-167      ;SET UP STACK
       IER=IBUF+1
       FS.=IER-IBUF+1
       FS.
RDCUR:
       JSR @.CPYL     ;COPY ADDRESS TO STACK
       LDA 2,IBUF,3    ;ADDRESS OF IBUF
       MOVZL 2,0       ;OFFSET TO BYTE POINTER
       LDA 2,1,2       ;LOAD CHANNEL AND CONSOLE ID
       LDA 1,CNT       ;# BYTES PASSED BACK FROM .RDC

       STA 3,FPTR      ;SAVE STACK POINTER

       .SYSTM
       .RDC 77 ;CALL READ CURSOR ROUTINE
       JMP RDERR

       LDA 3,FPTR
       LDA 2,C1        ;NO ERROR SO PASS IER=1
       STA 2,@IER,3

       JSR @.FRET      ;RETURN

RDERR:
       LDA 3,FPTR
       STA 2,@IER,3    ;PASS BACK RDOS ERROR

       JSR @.FRET      ;RETURN

C1:    000001
CNT:   10.
FPTR:  0
       .END
```

Eastern Region Computer Programs and Problems (Continued)

43 AEX - Automatic Program Execution. Harold H. Opitz, June 1988. (PB 88 231121/AS)

44 TURB: Turbulence Forecasting for Small/Medium and Large Aircraft. Steven J. Naglic, July 1988. (PB 88 246368/AS)

NOAA SCIENTIFIC AND TECHNICAL PUBLICATIONS

The National Oceanic and Atmospheric Administration was established as part of the Department of Commerce on October 3, 1970. The mission responsibilities of NOAA are to assess the socioeconomic impact of natural and technological changes in the environment and to monitor and predict the state of the solid Earth, the oceans and their living resources, the atmosphere, and the space environment of the Earth.

The major components of NOAA regularly produce various types of scientific and technical information in the following kinds of publications:

PROFESSIONAL PAPERS—Important definitive research results, major techniques, and special investigations.

CONTRACT AND GRANT REPORTS—Reports prepared by contractors or grantees under NOAA sponsorship.

ATLAS—Presentation of analyzed data generally in the form of maps showing distribution of rainfall, chemical and physical conditions of oceans and atmosphere, distribution of fishes and marine mammals, ionospheric conditions, etc.

TECHNICAL SERVICE PUBLICATIONS—Reports containing data, observations, instructions, etc. A partial listing includes data serials; prediction and outlook periodicals; technical manuals, training papers, planning reports, and information serials; and miscellaneous technical publications.

TECHNICAL REPORTS—Journal quality with extensive details, mathematical developments, or data listings.

TECHNICAL MEMORANDUMS—Reports of preliminary, partial, or negative research or technology results, interim instructions, and the like.

Information on availability of NOAA publications can be obtained from:

NATIONAL TECHNICAL INFORMATION SERVICE
U. S. DEPARTMENT OF COMMERCE
5285 PORT ROYAL ROAD
SPRINGFIELD, VA 22161