NOAA Eastern Region Computer Programs
and Problems NWS ERCP - No. 29

ISENTROPIC PLOTTER

Charles D. Little
National Weather Service Forecast Office
Columbia, South Carolina

Scientific Services Division
Eastern Region Headquarters
February 1985

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)

16 RAMP: Stability Analysis Program. Hugh M. Stone, February 1984.(PB84 1614

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)

Eastern Region CP No. 29 February 1985

ISENTROPIC PLOTTER

PART A: PROGRAM INFORMATION AND INSTALLATION PROCEDURE

PROGRAM NAME: ISENT.SV

AAL ID:
REV NO.: 1.ØØ

FUNCTION: The program will compute or find the following data on an isentropic surface: pressure, condensation pressure, mixing ratio and wind speed and direction closest to the desired level. The input file is generated by the TTBBD.SV decoder and output is to NMCPLT4ØA.

PROGRAM INFORMATION:

Development Programmer:
Charles Little
Location: WSFO Columbia, SC
Phone: FTS 677-5501
Language: DG FORTRAN IV/Rev. 5.20
Save File Creation Date: 02/22/85
Running Time: Variable, depending on the number of stations, about 2 seconds per station.
Disk Space: ISENT.SV 52 RDOS Blocks
NMCPLT4ØA 3 RDOS Blocks
The size of the last file depends on the number of stations that are being decoded. The disk space given is for 37 decoded stations.

Maintenance Programmer:
Charles Little
Location: WSFO Columbia, SC
Phone: FTS 677-5501
Type: Standard
Revision Date:

PROGRAM REQUIREMENTS:

Program Files:

Name	Comments
ISENT.SV	

Data Files:

Name	Location	Read/Write	Comments
STDIR.MS	DPØ	R	
NMCPLT40A	DPØ	W	Stored to AFOS
CODED FILE	DPØ	R	Current raob data
TTBBSTAS.DB	DPØ	R	Raobs to be decoded

AFOS Products:

ID	Action	Comments
NMCPLT40A	Stored by program	Plotfile

LOAD LINE

RLDR ISENT ISN11 ISN22 ISN33 ISN44 ISN55 ISN66 ISN77 ISN88 ISN99 CFLTCVT BNSCH WMROF VAPFW TCONOF BG.LB UTIL.LB FORT.LB AFOSE.LB

PROGRAM INSTALLATION

1. NMCPLT4ØA must be in the database.
2. ISENT.SV, STDIR.MS must be on DPØ or linked to it.
3. TTBBD.SV must be run first.
4. The plotted product will have to have a map background of 2.

ISENTROPIC PLOTTER

PART B: PROGRAM EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: ISENT.SV

AAL ID:
REV NO.: 1.ØØ

PROGRAM EXECUTION

The first step is to run the program TTBBD which will decode all the raobs in the file TTBBSTAS.DB. Now ISENT may be run via the following commands:

RUN:ISENT XXX where

XXX is the isentropic surface you wish decoded and plotted in degrees Kelvin.

A macro will have to be created to run MODELUGF or PMOD and may be run right after the above command. An example of a macro to execute MODELUGF is

MODELUGF NMCPLT40A NMCGPHISN

An example of a macro to run PMOD is

PMOD 40A UAA.PM/O NA.PF/T
GENUTF XPLOT ISN

Either of these macros would store the plotted product into the graphic NMCGPHISN. The map background will have to be set to 2 on whatever key you choose. The above commands may be repeated any number of times for different isentropic surfaces, without running the TTBBD program again. TTBBD only needs to be run when new data has become available.

The /F switch may be used to enter a default filename. This command is:

RUN:ISENT XXX/F

The dasher will respond with ENTER FILENAME. The filename may be up to 12 characters long with the last or 13th character a carriage return. After the filename has been entered, the program will print FILENAME -- (entered filename). When entering the filename, the characters will not be echoed back on the dasher. After the filename has been printed on the dasher, the program will continue in a normal mode.

Output is to NMCPLT4ØA and it is stored into the AFOS database. This key was used because it was in our database and was not being used for any other plotfile. The output key for the plotted graphic will depend on how your macro is set to run MODELUGF or PMOD.

ERROR CONDITIONS

All error messages are sent to the ADM. Error messages 900 through 910 will cause the program to halt.

Messages	Meaning
900	Error in coded data file-rerun TTBBD
905	Error in opening TTBBSTAS.DB-check file
910	Error in opening coded data file-rerun TTBBD
920	No data for this station
930	Date or time error for this station
940	Error in BNSCH for this station
950	Theta not found for this station
960	Data formatting error for this station

NOAA EASTERN REGION COMPUTER PROGRAMS AND PROBLEMS - No. 29

ISENTROPIC PLOTTER

Charles D. Little
National Weather Service Forecast Office
Columbia, South Carolina

Scientific Services Division
Eastern Region Headquarters
February 1985

Editor's Note

This program presents all data important to isentropic analysis in a single display. The output can also be overlain with the contoured fields described in WRCP-No. 47.

ISENTROPIC PLOTTER

Charles D. Little
National Weather Service Forecast Office
Columbia, South Carolina

I. Introduction

This program will plot isentropic surfaces and will give the operational forecasters of the National Weather Service another method of analyzing upper air data for use in day to day forecasting. Isentropic charts are very useful in finding and following two of the most important meteorological variables: areas of vertical motion and moisture (Uccellini, 1976 and Saucier, 1955).

This program, ISENT, when run in conjunction with TTBBD (Gilhousen and Person, 1981) and PMOD, will create a graphic of the plotted isentropic data (Figure 1). It will read the data file created by TTBBD and compute the height of the desired isentropic surface in mbs, the difference between the pressure of the isentropic surface and the condensation pressure in mbs and the mixing ratio. Next, it will decode the reported winds closest to the isentropic surface. The data is then put into plotfile format and stored to the AFOS database. Execution of MODELUGF or PMOD on the plotfile will create the plotted graphic.

Saucier gives the following guidelines when choosing an isentropic surface to plot.

WINTER	290 - 295°K
SPRING	295 - 300°K
SUMMER	310 - 315°K
FALL	300 - 305°K

These ranges are only a guideline and will probably need to be altered for day to day use.

II. Methodology and Software Structure

ISENT is the main program and begins by reading the command line to obtain the isentropic surface to be decoded. It also creates and opens the file NMCPLT4ØA (Figure 2), which will contain the decoded data in plotfile format. Finally, after all the data has been decoded, the program stores the plotfile to the AFOS database and alerts the ADM on completion. If you do not wish to use the current data file from TTBBD, a /F switch will allow you to input the filename at the dasher.

ISN11 - This subroutine is used to compute the filename of the output file from TTBBD using the current date and time.

ISN22 - This subroutine drives most of the remaining routines. It opens and reads the file containing the decoded data from TTBBD. The routine also opens and reads TTBBSTAS.DB to obtain the 3 letter station identification for use in BNSCH. ISN22 drives the routines for computing isentropic data, obtaining the winds and formatting the data. Most of the error messages originate from this subroutine and are output to the ADM.

ISN33 - ISN33 gets the pixel locations for the North America map background (BØ2) and the station elevation for each station. This is done with the use of the BNSCH subroutine and the STDIR.MS file.

ISN44 - This subroutine scans the decoded data for the desired isentropic level. If the surface is not found, that station will be recorded as "missing" in the plotfile output. If the level is found, the subroutine will then compute the following for the selected isentropic surface:

1. Pressure of the isentropic surface (mbs).
2. The difference between the pressure of the isentropic surface and the condensation pressure.
3. Mixing ratio (g/kg X 10) on the isentropic surface.
4. Temperature of the isentropic surface.
5. Dewpoint of the isentropic surface.
6. Temperature and dewpoint difference (Station circle will be filled in if 5 degrees or less).

Potential temperature (θ) is constant everywhere in a given isentropic surface, so it serves as a convenient identifier. θ is that temperature the air would have if brought dry adiabatically to a standard pressure of 1000 mbs. θ is defined by equation 1 in terms of P and T (Saucier, 1955).

$$\theta = (1000/P)^K * T \qquad (1).$$

where K is equal to R/c_p, θ and T are in degrees Kelvin and P is in mbs.

Equation 2 is used to find the pressure of the isentropic surface, P, when it occurs between two significant levels.

$$P - P_1 = ((P_2 - P_1)/(\theta_2 - \theta_1)) * (\theta - \theta_2) \qquad (2).$$

$$\theta_1 = (1000/P_1)^K * T_1 \qquad (2a).$$

$$\theta_2 = (1000/P_2)^K * T_2 \qquad (2b).$$

The pressure at P_1 is greater than the pressure at P_2 and P_N, T_N and θ_N are the pressure, temperature and potential temperature for their respective levels. P is the pressure of the desired isentropic surface, θ.

The condensation pressure, P_C, is the pressure a parcel on the isentropic surface would have if allowed to rise dry adiabatically, which corresponds to constant θ, until saturation occurred. The subroutine TCONOF (Doswell, et al, 1982) was used to find the temperature at the LCL. Once the temperature at the LCL is known, equation 1 may be rewritten as equation 3 and solved for pressure. In this case, the pressure is the pressure at which the air rising from the isentropic surface reaches saturation and T is the temperature at the LCL.

$$P_c = \left(\frac{T}{\theta}\right)^{1/K} * 1000 \qquad (3).$$

ISN55 - This subroutine computes the height of the isentropic surface using a routine based on thickness and is given by equation 4 (Hess, 1959).

$$dZ = \frac{R\overline{T}}{9.8} \quad Ln(P_2/P_1) \qquad (4).$$

where $\overline{T}$ is the average temperature in the layer between P_1 and P_2.

ISN66 - Scans the decoded data and selects the wind data closest to the isentropic level.

ISN77 - This subroutine is used to format data for output to the plotfile.

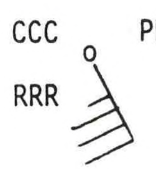
ISN88 - Subroutine used to code error messages.

ISN99 - This subroutine is used to format the date/time information for the plotfile.

CFLTCVT (Peroutka, 1981) - CFLTCVT is a modified version of FLTCVT 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.

WRMOF and VAPFW (Doswell, et al, 1982) - These are used to compute the mixing ratio.

The station plot model is shown below.



PPP is the pressure of the isentropic surface (mbs).

CCC is the difference, in mbs, between the pressure at the isentropic surface and the condensation pressure. It is the amount of lift that a parcel on the isentropic surface would need to reach saturation.

RRR is the mixing ratio in grams per kilogram (tens, units and tenths) on the isentropic surface.

The winds that are plotted are the observed winds closest to the isentropic surface.

The station circle is shaded if the dew point is within 5 degrees of the temperature.

III. Cautions and Restrictions

The program finds only the lowest location of a given isentropic surface.

IV. References

Berry, F.A., E. Bollay, and Norman R. Beers editors. 1945. Handbook of Meteorology. McGraw-Hill Book Company, New York, pp 1068.

Byers, H.R. 1944. General Meteorology. McGraw-Hill Book Company, New York, pp 387-405.

Doswell, C.A., J.T. Schaefer, D.W. McCann, T.W. Schlatter, and H.B. Wobus. 1982. Thermodynamic Analysis Procedures at the National Severe Storms Forecast Center, Preprints, 9th Conference Weather Forecasting and Analysis, Amer. Meteor. Soc., Seattle, pp 304-309.

Gilhousen, Dave, Person, Arthur. 1981. Significant Level Decoder, AAL ID : DBC002. Revision No.: 03.00 1984.

Hess, Seymour L. 1959. Introduction to Theoretical Meteorology. Holt, Rinehart and Winston, New York, pp 355.

Saucier, Walter J. 1955. Principles of Meteorological Analysis. The University of Chicago Press, Chicago pp 438.

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

Petterssen, Sverre. 1940. Weather Analysis and Forecasting. McGraw-Hill Book Company, New York, pp 505.

Uccellini, Louis W. 1976. Operational Diagnostic Applications of Isentropic Analysis, Source unknown.

V. Program Information and Procedures for Installation and Execution

Eastern Region CP No. 29 February 1985

ISENTROPIC PLOTTER

PART A: PROGRAM INFORMATION AND INSTALLATION PROCEDURE

PROGRAM NAME: ISENT.SV

AAL ID:
REV NO.: 1.ØØ

FUNCTION: The program will compute or find the following data on an isentropic surface: pressure, condensation pressure, mixing ratio and wind speed and direction closest to the desired level. The input file is generated by the TTBBD.SV decoder and output is to NMCPLT4ØA.

PROGRAM INFORMATION:

Development Programmer:
Charles Little
Location: WSFO Columbia, SC
Phone: FTS 677-5501
Language: DG FORTRAN IV/Rev. 5.20
Save File Creation Date: 02/22/85
Running Time: Variable, depending on the number of stations, about 2 seconds per station.
Disk Space: ISENT.SV 52 RDOS Blocks
NMCPLT4ØA 3 RDOS Blocks
The size of the last file depends on the number of stations that are being decoded. The disk space given is for 37 decoded stations.

Maintenance Programmer:
Charles Little
Location: WSFO Columbia, SC
Phone: FTS 677-5501
Type: Standard
Revision Date:

PROGRAM REQUIREMENTS:

Program Files:

Name	Comments
ISENT.SV	

Data Files:

Name	Location	Read/Write	Comments
STDIR.MS	DPØ	R	
NMCPLT40A	DPØ	W	Stored to AFOS
CODED FILE	DPØ	R	Current raob data
TTBBSTAS.DB	DPØ	R	Raobs to be decoded

AFOS Products:

ID	Action	Comments
NMCPLT40A	Stored by program	Plotfile

LOAD LINE

```
RLDR ISENT ISN11 ISN22 ISN33 ISN44 ISN55 ISN66 ISN77 ISN88
  ISN99 CFLTCVT BNSCH WMROF VAPFW TCONOF BG.LB UTIL.LB
  FORT.LB AFOSE.LB
```

PROGRAM INSTALLATION

1. NMCPLT4ØA must be in the database.
2. ISENT.SV, STDIR.MS must be on DPØ or linked to it.
3. TTBBD.SV must be run first.
4. The plotted product will have to have a map background of 2.

ISENTROPIC PLOTTER

PART B: PROGRAM EXECUTION AND ERROR CONDITIONS

PROGRAM NAME: ISENT.SV

AAL ID:
REV NO.: 1.ØØ

PROGRAM EXECUTION

The first step is to run the program TTBBD which will decode all the raobs in the file TTBBSTAS.DB. Now ISENT may be run via the following commands:

RUN:ISENT XXX where

XXX is the isentropic surface you wish decoded and plotted in degrees Kelvin.

A macro will have to be created to run MODELUGF or PMOD and may be run right after the above command. An example of a macro to execute MODELUGF is

MODELUGF NMCPLT40A NMCGPHISN

An example of a macro to run PMOD is

PMOD 40A UAA.PM/O NA.PF/T
GENUTF XPLOT ISN

Either of these macros would store the plotted product into the graphic NMCGPHISN. The map background will have to be set to 2 on whatever key you choose. The above commands may be repeated any number of times for different isentropic surfaces, without running the TTBBD program again. TTBBD only needs to be run when new data has become available.

The /F switch may be used to enter a default filename. This command is:

RUN:ISENT XXX/F

The dasher will respond with ENTER FILENAME. The filename may be up to 12 characters long with the last or 13th character a carriage return. After the filename has been entered, the program will print FILENAME -- (entered filename). When entering the filename, the characters will not be echoed back on the dasher. After the filename has been printed on the dasher, the program will continue in a normal mode.

Output is to NMCPLT4ØA and it is stored into the AFOS database. This key was used because it was in our database and was not being used for any other plotfile. The output key for the plotted graphic will depend on how your macro is set to run MODELUGF or PMOD.

ERROR CONDITIONS

All error messages are sent to the ADM. Error messages 900 through 910 will cause the program to halt.

Messages	Meaning
900	Error in coded data file-rerun TTBBD
905	Error in opening TTBBSTAS.DB-check file
910	Error in opening coded data file-rerun TTBBD
920	No data for this station
930	Date or time error for this station
940	Error in BNSCH for this station
950	Theta not found for this station
960	Data formatting error for this station

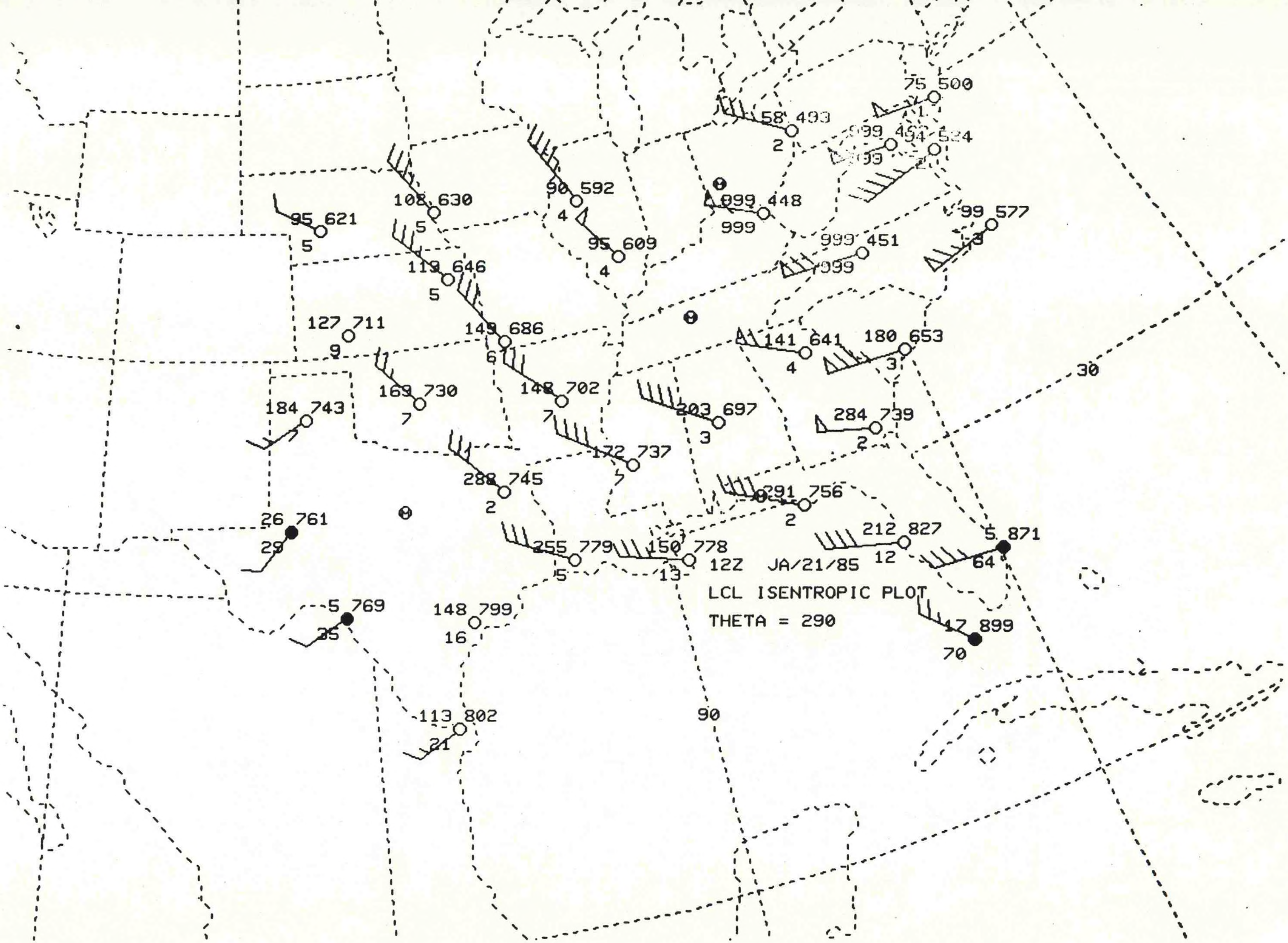


FIGURE 1. Plotted Data

```
NMCPLT40A0010200020481536285014 25+0975+1688122101851520
1260,250,00000Z,12Z  JA/21/85;
1260,230,00000Z,LCL ISENTROPIC PLOT;
1260,210,00000Z,THETA = 290 ;
1477,609,00000A,,,0,28068,451,999,999;
1433,530,00000A,,,0,30062,641,141,4;
1511,533,00000A,,,0,28074,653,180,3;
1366,475,00000A,,,0,31041,697,203,3;
1344,558,00000A,,,M;
1488,471,00000A,,,0,29052,739,284,2;
1044,477,00000A,,,0,24015,743,184,7;
1433,411,00000A,,,0,30045,756,291,2;
1166,235,00000A,,,0,24017,802,113,21;
1344,368,00000A,,,0,29036,778,150,13;
1366,665,00000A,,,M;
1077,544,00000A,,,0,99999,711,127,9;
1077,322,00000A,,,8,24011,769,5,35;
1566,306,00000A,,,8,32023,899,17,70;
1200,421,00000A,,,0,32031,745,288,2;
1577,632,00000A,,,0,26064,577,99,3;
1400,641,00000A,,,0,30055,448,999,999;
1499,697,00000A,,,0,28053,462,999,999;
1300,442,00000A,,,0,31039,737,172,7;
1055,627,00000A,,,0,30011,621,95,5;
1255,368,00000A,,,0,30030,779,255,5;
1244,492,00000A,,,0,32032,702,148,7;
1033,390,00000A,,,8,22011,761,26,29;
1133,490,00000A,,,0,32024,730,169,7;
1144,642,00000A,,,0,33033,630,108,5;
1588,378,00000A,,,8,28036,871,5,64;
1255,651,00000A,,,0,34047,592,90,4;
1422,708,00000A,,,0,31034,493,58,2;
1122,405,00000A,,,M;
1288,606,00000A,,,0,33052,609,95,4;
1511,382,00000A,,,0,29041,827,212,12;
1155,588,00000A,,,0,32034,646,119,5;
1200,539,00000A,,,0,33028,686,149,6;
1177,319,00000A,,,0,99999,799,148,16;
1399,418,00000A,,,M;
1533,693,00000A,,,0,26044,524,94,2;
1533,735,00000A,,,0,28049,500,75,1;
```

FIGURE 2. Plotfile

```
MAIN PROGRAM
    ISENT


SUBROUTINES
    ISN11
    ISN99                  CFLTCVT
    ISN22 -----------> ISN33 ------> BNSCH
                       ISN44 ------> ISN55 ------->WMROF/VAPFW
                       ISN66         TCONOF
                       ISN77
                       ISN88


LOAD LINE

RLDR ISENT ISN11 ISN22 ISN33 ISN44 ISN55 ISN66 ISN77 ISN88
     ISN99 CFLTCVT BNSCH WMROF VAPFW TCONOF BG.LB UTIL.LB FORT.LB
     AFOSE.LB
```

FIGURE 3. Software Structure and Load Line

VI. Source Listings

```
C     PROGRAM ISENT                                   REV 01.00
C     JAN 1985                 CHUCK LITTLE           WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20      DG ECLIPSE (S230)      RDOS/REV 6.18
C
C     LOAD LINE: RLDR ISENT ISN11 ISN22 ISN33 ISN44 ISN55 ISN66 ISN77 ISN88
C                    ISN99 CFLTCVT BNSCH WMROF VAPFW TCONOF BG.LB UTIL.LB
C                    FORT.LB AFOSE.LB
C
C     PURPOSE:
C        THIS PROGRAM WILL COMPUTE OR FIND THE FOLLOWING DATA ON AN ISENTROPIC
C        SURFACE -- PRESSURE, CONDENSATION PRESSURE, MIXING RATIO, WIND
C        SPEED/DIRECTION CLOSEST TO THE ISENTROPIC LEVEL. THE INPUT FILE IS
C        GENERATED BY THE TTBBD.SV DECODER. OUTPUT IS TO NMCPLT40A.
C
C     EXITS
C        STOP - ERROR 900 - ERROR IN IFILE (DECODED FILE FROM TTBBD.SV)
C        STOP - ERROR 905 - ERROR IN TTBBSTAS.DB
C        STOP - ERROR 910 - ERROR IN OPENING IFILE
C
C
C
C
C
C
C
C
C
C
C
C
C
      COMMON/CDL/MONTH(15),IRR(10)
      COMMON/QARDQ/IOUTU(90),I
      DIMENSION IC(30),IDT(3),ITM(3),IZTM(25),IDQ(20),ISW(2),IFILE(15)
      COMMON/A/IBUF(110)
      DATA IBUF/'NMCPLT40A000',177777K,177777K,2400K,142600K/
      DATA IBUF(11)/'NMCPLT40A00102000204815362850142S+0975+1688 '/
      DATA IBUF(33)/6412K,'1260,250,00000Z,'/
      DATA IBUF(42)/'1260,230,00000Z,LCL ISENTROPIC PLOT; '/
      DATA IBUF(70)/6412K,'1260,210,00000Z,THETA = '/
      DATA IBUF(90)/' ;',6412K,'END;',101603K/
      IDELTA=0
      CALL FCOM (II,IER)
      CALL COMCM (II,IDQ,N,ISW,IER)
      CALL COMCM (II,IDQ,N,ISW,IER)
      IF(ISWSET(ISW,'F')) IDELTA=1
      CALL DATE (IDT,IER)
      CALL TIME (ITM,IER)
      IH=0
      IF(ITM(1).GE.12) IH=12
      CALL UNPACK (IDQ,3,IOUTU)
      IF (IDELTA.EQ.1) GOTO 20
      CALL ISN11(IDT,ITM,IFILE,IH)
10    CALL ISN99 (ITM,IZTM,IC,IFILE)
      CALL DELET ('NMCPLT40A',IER)
      CALL CRAND ('NMCPLT40A',IER)
      CALL GCHN (ICY,IER)
      CALL OPENN (ICY,'NMCPLT40A',0,IER)
      CALL WRS (ICY,IBUF,63,IER)
      CALL WRS (ICY,IZTM,12,IER)
```

```
      CALL WRS (ICY,IBUF(33),18,IER)
      CALL WRS (ICY,IC,16,IER)
      CALL WRS (ICY,IBUF(42),36,IER)
      CALL WRS (ICY,IBUF(70),26,IER)
      CALL WRS (ICY,IDQ,3,IER)
      CALL WRS (ICY,IBUF(90),4,IER)
      CALL ISN22 (IH,IDT,ITM,ICY,IFILE)
      CALL FORKP ('ISENT','NMCPLT40A',IER)
      CALL WRS (ICY,IBUF(92),6,IER)
      CALL KLOSE (ICY,IER)
      CALL FSTORE ('NMCPLT40A',0,IER)
      STOP
C
C  DEFAULT FILENAME--
C
  20  CALL SPCHR (' ENTER FILENAME<15>',IER)
      DO 30 I=1,13
      CALL GCHAR (ICHR,IER)
      IF(ICHR.EQ.000015K) GOTO 40
      IFILE(I)=ICHR
  30  CONTINUE
      CALL SPCHR (' *** ERROR IN FILENAME ***<15>',IER)
      STOP
  40  IFILE(I)=000000K
      CALL PACK (IFILE,I,IFILE)
      CALL SPCHR (' FILENAME -- ',IER)
      CALL SPCHR (IFILE,IER)
      GOTO 10
      END
```

```
      SUBROUTINE ISN11 (IDT,ITM,IFILE,IH)
C                               REV 01.00
C     OCT 1984                  CHUCK LITTLE        WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20       DG ECLIPSE (S230)   RDOS/REV 6.13
C     PURPOSE:
C        TO COMPUTE THE NAME OF THE FILE PRODUCED FROM TTBBD.SV -- MMDDYYU.HH
C           MM -- MONTH       DD -- DATE              YY -- YEAR
C           U. -- CONSTANT   HH -- CYCLE TIME (00/12Z)
C
C     ARGUMENT LIST
C        IDT        - INTEGER DATE
C        ITM        - INTEGER TIME
C        IFILE      - DECODED FILENAME
C        IH         - CYCLE TIME (00Z/12Z)
C
C
C
      DIMENSION ITM(3),IDT(3),IBQ(10),IFILE(15),IWORK(15)
      COMMON/CDL/MONTH(15),IRR(10)
      DATA MONTH/'JAFEMAAPMYJUJYAUSEOCNODE'/
      MO=IDT(1)
      IFILE(1)=MONTH(MO)
      CALL UNPACK (IFILE,2,IWORK)
      K=1
      IF(IDT(2).GE.10) GOTO 20
      IWORK(3)=000060K
      CALL UBNDEC (IDT(2),IBQ,K)
      IWORK(4)=IBQ(6)
      GOTO 30
  20  CALL UBNDEC (IDT(2),IBQ,K)
      IWORK(3)=IBQ(5)
      IWORK(4)=IBQ(6)
  30  K=1
      IDT(3)=IDT(3)-1900
      CALL UBNDEC (IDT(3),IBQ,K)
      IWORK(5)=IBQ(5)
      IWORK(6)=IBQ(6)
      IWORK(7)=000125K
      IWORK(8)=000056K
      IF(IH.EQ.0) IWORK(9)=000060K
      IF(IH.EQ.0) IWORK(10)=000060K
      IF(IH.EQ.12) IWORK(9)=000061K
      IF(IH.EQ.12) IWORK(10)=000062K
      IWORK(11)=000000K
      CALL PACK (IWORK,11,IFILE)
      RETURN
      END
```

```
      SUBROUTINE ISN22 (IH,IDT,ITM,ICY,IFILE)
C                                REV 01.00
C     JAN 1985                   CHUCK LITTLE          WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20        DG ECLIPSE (S230)     RDOS/REV 6.18
C     PURPOSE:
C        OPENS FILES (STDIR.MS-TTBBSTAS.DB & IFILE--CODED FILENAME FROM TTBBD)
C        READS DATA-MAKES CALLS TO PROCESS AND FORMAT DATA-WRITES TO PLOT FILE
C        PROCESS ERRORS
C
C     ARGUMENT LIST
C        IH          - CYCLE TIME (00/12Z)
C        IDT         - INTEGER DATE
C        ITM         - INTEGER TIME
C        ICY         - CHANNEL TO NMCPLT40A -- PLOTFILE
C        IFILE       - CODED FILENAME
C
      COMMON/QARDQ/IOUTU(90),I
      DIMENSION JUNK (10),ITIME(10),NOLVLS(10),IPRES(31),ITEMP(31),PRES(31)
      DIMENSION IDP(31),IWHT(31),IDIR(31),ISP(31),IFILE(15),TEMP(31),DPT(31)
      DIMENSION IX(20),ISTN(10),IDT(3),ITM(3),IB(3),IAD(2),A(15),IA(15)
      DIMENSION IOUT(100)
      IFLDP=1
      IFLD=6
      IAD(1)=0
      IAD(2)=0
      THETA=CFLTCVT(1,3)
      CALL GCHN (MI,IER)
      CALL OPEN (MI,IFILE,2,IER)
      IF (IER.NE.1) GOTO 910
      READ BINARY (MI,ERR=900,END=500) (ITIME(I),I=1,9)
      IDD=(ITIME(6)*100)+ITIME(5)
      CALL GCHN (KO,IER)
      CALL OPENN (KO,'STDIR.MS',0,IER)
      CALL RDS (KO,IB,6,IER)
      CALL GCHN (LO,IER)
      CALL OPENN (LO,'TTBBSTAS.DB',0,IER)
      IF(IER.NE.1) GOTO 905
      READ BINARY (MI,ERR=900,END=500) (JUNK(I),I=1,7)
    5 READ BINARY (MI,ERR=900,END=500) (NOLVLS(I),I=1,8)
      NC=8
      CALL RDS (LO,IX,11,IER)
      CALL UNPACK (IX(4),3,ISTN)
      ISTN(4)=000040K
      ISTN(5)=000000K
      ISTN(6)=000000K
      CALL PACK (ISTN,6,ISTN)
      NO=ITIME(9)
      READ BINARY (MI,ERR=900,END=500) (IPRES(I),I=1,NO)
      READ BINARY (MI,ERR=900,END=500) (ITEMP(I),I=1,NO)
      READ BINARY (MI,ERR=900,END=500) (IDP(I),I=1,NO)
      READ BINARY (MI,ERR=900,END=500) (IWHT(I),I=1,NO)
      READ BINARY (MI,ERR=900,END=500) (IDIR(I),I=1,NO)
      READ BINARY (MI,ERR=900,END=500) (ISP(I),I=1,NO)
      CALL ISN33 (KO,IB,IFLDP,IFLD,IAD,ISTN,IX,IER)
      IA(1)=IX(1)
      IA(2)=IX(2)
      IF(IER.NE.1) GOTO 940
      IF(NOLVLS(5).EQ.-999) GOTO 920
      IF(NOLVLS(4).NE.IDD) GOTO 930
      DO 20 I=1,NO
```

```
      PRES(I)=IPRES(I)
      TEMP(I)=(ITEMP(I)/10.)+273.16
      DPT(I)=(IDP(I)/10.)+273.16
      IDIR(I)=(IDIR(I)/10)*10
   20 CONTINUE
      CALL ISN44 (THETA,A,IA,PRES,TEMP,DPT,IX,IER)
      IF(IER.NE.1) GOTO 950
      CALL ISN66 (IA,A,IWHT,IDIR,ISP)
      CALL ISN77 (IA,IOUT,NC,M,IER)
      IF(IER.NE.1) GOTO 960
      CALL WRS (ICY,IOUT,M,IER)
C
      GOTO 5
  500 CALL KLOSE (MI,IER)
      CALL KLOSE (LO,IER)
      CALL KLOSE (KO,IER)
      RETURN
  900 CALL FORKE ('ISENT',' ERROR 900 ',IER) ;ERR IN TTBBD.SV DATA FL
      GOTO 500
  905 CALL FORKE ('ISENT',' ERROR 905 ',IER)  ;TTBBSTAS.DB
      GOTO 500
  910 CALL FORKE ('ISENT',' ERROR 910 ',IER)  ;TTBBD.SV DATAFL
      GOTO 500
  920 IR=2  ;NO DATA
      GOTO 990
  930 IR=3  ;RONG DAT/TIM
      GOTO 990
  940 IR=4  ;TRBL IN STDIR.MS
      GOTO 990
  950 IR=5  ;THETA NOT FND
      GOTO 990
  960 IR=6   ;TRBL IN FORMATG DATA
  990 CALL ISN88 (IR,ISTN)
      NC=2
      CALL ISN77 (IA,IOUT,NC,M,IER)
      CALL WRS (ICY,IOUT,M,IER)
      GOTO 5
      END
```

```
      SUBROUTINE ISN33 (KO,IB,IFLDP,IFLD,IAD,JST,IX,IER)
C                              REV 01.00
C     OCT 1984                 CHUCK LITTLE          WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20      DG ECLIPSE (S230)     RDOS/REV 6.13
C     PURPOSE:
C        FIND PIXEL & HEIGHT ABOVE SEA LEVEL FOR STATION IN JST AND MAP
C        BACKGROUND B02
C
C     ARGUMENT LIST
C        KO         - CHANNEL FOR READING
C        IB         - BNSCH VARIABLES
C        IFLDP      -   '     '
C        IFLD       -   '     '
C        IAD        -   '     '
C        JST        - STATION TO LOOK FOR - 3 LETTER ID
C        IX         - IX(1)= X PIXEL
C                     IX(2)= Y PIXEL
C                     IX(3)= HGT ABV MSL (M)
C        IER        - ERROR (IER) -- 1 FOR GOOD CALL -- 0 FOR BAD CALL
C
C
      DIMENSION JST(3),IB(3),IAD(2),IC1(14),IC2(14),IC3(14),IA(14),IX(10)
      JST(3)=20040K
      CALL BNSCH (KO,IB(1),IB(2),IB(3),IFLDP,IFLD,JST,IAD,IC1,IC2,IC3,IC)
      IF(IC.EQ.0) GOTO 20
      GOTO (1,2,3),IC
    1 IX(1)=IC1(8)
      IX(2)=IC1(9)
      IX(3)=IC1(5)
      GOTO 10
    2 IX(1)=IC2(8)
      IX(2)=IC2(9)
      IX(3)=IC2(5)
      GOTO 10
    3 IX(1)=IC3(8)
      IX(2)=IC3(9)
      IX(3)=IC3(5)
   10 IER=1
      RETURN
   20 IER=0
      RETURN
      END
```

```
      SUBROUTINE ISN44 (THETA,A,IA,PRES,TEMP,DP,IX,IER)
C                               REV 01.00
C     JAN 1985                  CHUCK LITTLE        WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20       DG ECLIPSE (S230)   RDOS/REV 6.18
C     PURPOSE:
C        COMPUTES PRESSURE--TEMPERATURE--DEWPOINT--MIXING RATIO--HEIGHT--AND
C        CONDENSATION PRESSURE FOR THE SELECTED THETA LEVEL
C
C     ARGUMENT LIST
C        THETA      - ISENTROPIC SURFACE DEGREES KELVIN
C        A          - ONE LINE OF DATA FOR PLOTFILE-REAL
C        IA         - ONE LINE OF DATA FOR PLOTFILE-INTEGER
C        PRES       - INFO ON PRESSURE
C        TEMP       - INFO ON TEMPERATURE
C        DP         - INFO ON DEW POINT
C        IX         - INFO ON HEIGHT ABOVE SEA LEVEL (IX(3))
C        IER        - THETA NOT FOUND IER = 0 -- IER = 1 FOR THETA FOUND
C
C
C
      DIMENSION A(15),IA(15),PRES(31),TEMP(31),DP(31),IX(10)
      ZKAPPA=0.28750
      I=1
 10   IF(PRES(I+1).LT.-500) GOTO 90             ;THETA NOT FOUND
      AB=((1000./PRES(I))**ZKAPPA)*(TEMP(I))
      B=((1000./PRES(I+1))**ZKAPPA)*(TEMP(I+1))
      IF(AB.LT.THETA.AND.B.GT.THETA) GOTO 20
      IF(AB.EQ.THETA) GOTO 30
      IF(B.EQ.THETA) GOTO 25
      I=I+1
      GOTO 10
 20   DS=((PRES(I+1)-PRES(I))/(B-AB))*(THETA-AB)
      A(1)=PRES(I)-ABS(DS)                      ;PRES (MBS)
      A(2)=(((A(1)/1000.)**ZKAPPA)*THETA)       ;TEMP (DEG K)
      IF(DP(I).LT.-500..OR.DP(I+1).LT.-500.) GOTO 24
      IF(DP(I).EQ.DP(I+1)) GOTO 23              ;DP'S EQL
      A(3)=DP(I)+(DS/((PRES(I+1)-PRES(I))/(DP(I+1)-DP(I)))) ;DP
      GOTO 50
 23   A(3)=DP(I)
      GOTO 50
 24   A(3)=-600.                                ;MSG DP
      GOTO 55
 25   I=I+1
 30   A(1)=PRES(I)                              ;THETA AT SIG LVL
      A(2)=TEMP(I)
      A(3)=DP(I)
 50   IF(A(3).LT.-500.) GOTO 55
      P=A(1)
      T=A(2)
      TD=A(3)
      A(4)=(WMROF (P,TD))*10.                   ;MXG RATIO (G/KG X 10)
      A(10)=A(1)-(((((TCONOF(T,TD))+273.16)/THETA)**(1.0/ZKAPPA))*1000.0
      GOTO 56
                                                ;A(10) = COND PRES - THETA PRES
 55   A(4)=999.                                 ;MSG MXG RATIO
      A(10)=999.                                ;MSG CONDENSATION PRESSURE
 56   CALL ISN55 (A,I,PRES,TEMP,DP)
      A(7)=(A(6)+(IX(3)*3.2808))/100.           ;HGT ABV MSL (HNDS OF FT)
      A(8)=(A(5)+IX(3))/1000.                   ;HGT ABV MSL (KM)
      A(9)=(10.04*(A(2)))+(98.*A(8))            ;STREAM FUNCTION
      IA(6)=(A(1)+.5)                           ;PRES
```

```
      IF(IA(6).GE.1000) IA(6)=IA(6)-1000
      IA(7)=(A(10)+.5)                        ;CONDENSATION PRESSURE
      IF(IA(7).GE.1000) IA(7)=IA(7)-1000
      IA(8)=(A(4)+.5)                         ;MXR RATIO
      IA(9)=(A(9)+.5)                         ;STREAM FUNCTION
      IA(3)=0
      TTD=(A(2)-A(3))
      IF(TTD.LE.5.0) IA(3)=8                   ;DP DEP 5 OR LESS
      IA(10)=IX(3)                             ;HGT ABV MSL (M)
      IER=1                                    ;THETA FND
      RETURN
90    IA(3)=9
      IER=0                                    ;THETA NOT FND
      RETURN
      END
```

```
      SUBROUTINE ISW55 (A,I,PRES,TEMP,DP)
C                             REV 01.00
C     JAN 1985                CHUCK LITTLE        WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20     DG ECLIPSE (S230)   RDOS/REV 6.18
C     PURPOSE:
C        COMPUTES THE HEIGHT OF THE ISENTROPIC LEVEL USING THICKENSS
C
C     ARGUMENT LIST
C        A          - ONE LINE OF DATA FOR PLOTFILE-REAL
C        I          - NUMBER OF PRESSURE SURFACES BETWEEN THE GROUND & THETA
C        PRES       - INFO ON PRESSURE
C        TEMP       - INFO ON TEMPERATURE
C        DP         - INFO ON DEWPOINT
C
C
C
      DIMENSION A(15),PRES(31),TEMP(31),DP(31)
      J=1
      ZT=0.
      ZS=29.2898
      IF(I.EQ.J) GOTO 20
 10   TA=(TEMP(J)+TEMP(J+1))/2.0
      IF(DP(J).LT.-500..OR.DP(J+1).LT.-500.) GOTO 15
      TD=(DP(J)+DP(J+1))/2.
      AP=(PRES(J)+PRES(J+1))/2.
      W=(WMROF(AP,TD))/1000.
      TA=((1.+1.609*W)/(1.+W))*TA
 15   TB=ABS(ALOG(PRES(J+1)/PRES(J)))
      ZT=ZT+(ZS*TA*TB)
      J=J+1
      IF(I.EQ.J) GOTO 20
      GOTO 10
 20   TA=(A(2)+TEMP(J))/2.
      IF(DP(J).LT.-500..OR.A(3).LT.-500.) GOTO 25
      TD=(A(3)+DP(J))/2.
      AP=(A(1)+PRES(J))/2.
      W=(WMROF (AP,TD))/1000.
      TA=((1.+1.609*W)/(1.+W))*TA
 25   TB=ABS(ALOG(A(1)/PRES(J)))
      ZT=ZT+(ZS*TA*TB)
      A(5)=ZT         ;HGT ABV STN (M)
      A(6)=ZT/.3048   ;HGT ABV STN (FT)
      RETURN
      END
```

```
      SUBROUTINE ISN66 (IA,A,IWHT,IDIR,ISP)
C                             REV 01.00
C     OCT 1984                CHUCK LITTLE        WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20     DG ECLIPSE (S230)   RDOS/REV 6.18
C     PURPOSE:
C        FIND THE WIND GROUP NEAREST THE SELECTED THETA LEVEL
C
C     ARGUMENT LIST
C        IA          - ONE LINE OF DATA FOR PLOTFILE-INTEGER
C        A           - ONE LINE OF DATA FOR PLOTFILE-REAL
C        IWHT        - INFO ON HEIGHT OF WIND DATA
C        IDIR        - INFO ON DIRECTION OF WIND DATA
C        ISP         - INFO ON WIND SPEED
C
C
C
      DIMENSION IA(15),A(15),IWHT(31),IDIR(31),ISP(31)
      I=1
   10 IF(IWHT(I).LT.-600) GOTO 40
      IF(A(7).GT.IWHT(I).AND.A(7).LT.IWHT(I+1)) GOTO 20
      IF(A(7).EQ.IWHT(I)) GOTO 25
      IF(A(7).EQ.IWHT(I+1)) GOTO 30
      I=I+1
      GOTO 10
   20 D1=A(7)-IWHT(I)
      D2=IWHT(I+1)-A(7)
      IF(D2.LT.D1) GOTO 30
   25 IA(4)=IDIR(I)
      IA(5)=ISP(I)
      GOTO 60
   30 IA(4)=IDIR(I+1)
      IA(5)=ISP(I+1)
      GOTO 60
   40 IA(4)=999
      IA(5)=99
   60 RETURN
      END
```

```
      SUBROUTINE ISN77 (IA,IOUT,NC,M,IER)
C                              REV 01.00
C    JAN 1985                  CHUCK LITTLE        WSFO CAE FTS 677-5501
C    FORTRAN IV/REV 5.20       DG ECLIPSE (S230)   RDOS/REV 6.18
C    PURPOSE:
C       PUTS INTEGERS INTO PACKED FORMAT FOR ONE LINE OF THE PLOTFILE
C
C    ARGUMENT LIST
C       IA          - ONE LINE OF DATA FOR THE PLOTFILE-INTEGER
C       IOUT        - ONE LINE OF DATA FOR THE PLOTFILE-PACKED ASCII
C       NC          - NUMBER OF TIMES TO GO THROUGH LOOP
C       M           - NUMBER OF BYTES IN IOUT
C       IER         - 1 FOR GOOD CALL -- 0 FOR BAD CALL
C
      DIMENSION IOUT(50),IBQ(20),IA(20)
      M=1
      DO 81 J=1,NC
      K=1
      GOTO (5,5,20,10,15,5,5,5),J
  5   IF(IA(J).LT.10) GOTO 20
      IF(IA(J).LT.100) GOTO 30
      IF(IA(J).LT.1000) GOTO 40
      IF(IA(J).LT.10000) GOTO 50
      GOTO 60
 10   IF(IA(J).LT.100) GOTO 70
      GOTO 40
 15   IF(IA(J).LT.10) GOTO 71
      IF(IA(J).LT.100) GOTO 30
      GOTO 40
 20   CALL UBNDEC (IA(J),IBQ,K)
      IOUT(M)=IBQ(6)
      IOUT(M+1)=000054K
      M=M+2
      GOTO 80
 30   CALL UBNDEC (IA(J),IBQ,K)
      IOUT(M)=IBQ(5)
      IOUT(M+1)=IBQ(6)
      M=M+2
      IF(J.EQ.4) GOTO 80    ;FOR WIND DIR--NO COMMA
      IOUT(M)=000054K
      M=M+1
      GOTO 80
 40   CALL UBNDEC (IA(J),IBQ,K)
      IOUT(M)=IBQ(4)
      IOUT(M+1)=IBQ(5)
      IOUT(M+2)=IBQ(6)
      M=M+3
      IF(J.EQ.4) GOTO 80     ;FOR WIND DIR--NO COMMA
      IOUT(M)=000054K
      M=M+1
      GOTO 80
 50   CALL UBNDEC (IA(J),IBQ,K)
      IOUT(M)=IBQ(3)
      IOUT(M+1)=IBQ(4)
      IOUT(M+2)=IBQ(5)
      IOUT(M+3)=IBQ(5)
      IOUT(M+4)=000054K
      M=M+5
      GOTO 80
 60   IER=0
```

```
      RETURN
70    IOUT(M)=000060K
      M=M+1
      GOTO 30
71    IOUT(M)=000060K
      M=M+1
      GOTO 20
75    IOUT(M)=000060K
      IOUT(M+1)=000060K
      IOUT(M+2)=000060K
      IOUT(M+3)=000060K
      IOUT(M+4)=000060K
      IOUT(M+5)=000101K
      IOUT(M+6)=000054K
      IOUT(M+7)=000054K
      IOUT(M+8)=000054K
      M=M+9
      GOTO 81
80    IF(J.EQ.2) GOTO 75
81    CONTINUE
      IF(NC.EQ.2) IOUT(M)=000115K
      IF(NC.EQ.2) IOUT(M+1)=000054K
      IF(NC.EQ.2) M=M+2
      IOUT(M-1)=000073K
      IOUT(M)=000015K
      IOUT(M+1)=000012K
      M=M+1
      CALL PACK (IOUT,M,IOUT)
      IER=1
      RETURN
      END
```

```
      SUBROUTINE ISN88 (IR,ISTN)
C                             REV 01.00
C     OCT 1984                CHUCK LITTLE        WSFO CAE FTS 677-5501
C     FORTRAN IV/REV 5.20     DG ECLIPSE (S230)  RDOS/REV 6.18
C     PURPOSE:
C
C       SENDS NON HALTING ERROR MESSAGES TO ADM
C
C     ARGUMENT LIST
C         ISTN       - CONTAINS THE 3 LETTER STATION ID
C         IR         - THE LINE NUMBER FOR CORRESPONDING ERROR
C
      DIMENSION ISTN(10)
      COMMON/CDL/MONTH(15),IRR(10)
      DATA IRR/'ERR-XX0-XXXX'/
      GOTO (10,20,30,40,50,60),IR
 10   RETURN
 20   IRR(3)='92'
      GOTO 70
 30   IRR(3)='93'
      GOTO 70
 40   IRR(3)='94'
      GOTO 70
 50   IRR(3)='95'
      GOTO 70
 60   IRR(3)='96'
 70   IRR(5)=ISTN(1)
      IRR(6)=ISTN(2)
      CALL FORKE ('THETA',IRR,IER)
      RETURN
      END
```

```
      SUBROUTINE ISN99 (ITM,IZTM,IC,IFILE)
C                             REV 01.00
C     OCT 1984                CHUCK LITTLE        WSFO CAE FTS 677-5501
C     FORTRAN IV/ REV. 5.20   DG ECLIPSE (S230)   RDOS/REV 6.18
C     PURPOSE:
C        COMPUTE DATE/TIME DATA FOR PLOTFILE HEADER AND PLOTFILE LABEL LINE
C        FROM DATA IN IFILE AND ITM
C
C     ARGUMENT LIST
C        ITM        - INTEGER TIME
C        IZTM       - ASCII TIME/DATE FOR PLOTFILE HEADER
C        IC         - ASCII TIME/DATE FOR PLOTFILE
C        IFILE      - CODED FILENAME FOR DATA FROM TTBBD
C
      DIMENSION IDT(3),ITM(3),IA(20),IZTM(25),IC(20),IFILE(15),IM(20)
      CALL GCHN (J0,IER)
      CALL OPEN (J0,IFILE,2,IER)
      READ BINARY(J0) (IA(I),I=1,9)
      M0=IA(7)
      CALL UNPACK (IA,8,IM)
      IA(1)=IM(1)
      IA(2)=IM(2)
      IA(3)=000132K
      IA(4)=000040K
      IA(5)=000040K
      IA(6)=IM(3)
      IA(7)=IM(4)
      IA(8)=000057K
      IA(9)=IM(5)
      IA(10)=IM(6)
      IA(11)=000057K
      IA(12)=IM(7)
      IA(13)=IM(8)
      IA(14)=000073K
      IA(15)=000015K
      IA(16)=000012K
      CALL KLOSE (J0,IER)
      IZTM(5)=000060K
      IZTM(9)=000060K
      IZTM(11)=000060K
      I=1
      CALL UBNDEC (ITM(1),IM,I)
      IZTM(10)=IM(6)
      IF(ITM(1).GE.10) IZTM(9)=IM(5)
      I=1
      CALL UBNDEC (ITM(2),IM,I)
      IZTM(12)=IM(6)
      IF(ITM(2).GE.10) IZTM(11)=IM(5)
      I=1
      CALL UBNDEC (M0,IM,I)
      IZTM(6)=IM(6)
      IF(M0.GE.10) IZTM(5)=IM(5)
      IZTM(1)=IA(1)
      IZTM(2)=IA(2)
      IZTM(3)=IA(9)
      IZTM(4)=IA(10)
      IZTM(7)=IA(12)
      IZTM(8)=IA(13)
      CALL PACK (IZTM,12,IZTM)
      CALL PACK (IA,16,IC)
      RETURN
      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 BNSCH(ICHN,NREC,LREC,ISTAR,IFLDP,IFLD,ITEST,
     1   IAD,IC1,IC2,IC3,IC)
C     BINARY SEARCH ROUTINE:
C
C     PROGRAMMER - RICH THOMAS SXB,ISL,SDO 11/79
C
C     ICHN=CHANNEL WHICH FILE HAS BEEN OPENNED TO
C     NREC=NUMBER OF RECORDS
C     LREC=LENGTH OF EACH RECORD (BYTES)
C     ISTAR=BYTE OF FIRST RECORD (0=BEGINNING)
C     IFLDP=WORD POINTER TO FIELD IN RECORD
C     IFLD=LENGTH OF FIELD IN BYTES
C     ITEST=ARRAY CONTAINING TEST FIELD
C     IAD=RETURNED TWO WORD ARRAY CONTAINING ADDRESS ITEST RECORD
C       SHOULD BEGIN AT-
C       IC= 1,2,3 IN SECOND WORD INDICATING RECORD WAS FOUND AND
C       IS IN ARRAY IC1,IC2, OR IC3
C     THOSE THREE ARRAYS SHOULD BE DIMENSIONED LREC/2 WORDS
      DIMENSION ITEST(1),IC1(1),IC2(1),IC3(1),IAD(2)
      DIMENSION IAD1(2),IAD2(2),IAD3(2)
      DIMENSION D1(2),D2(2)
      INTEGER D1,D2
      IC=0
      IAD1(1)=0
      IAD1(2)=ISTAR
      CALL SPOS(ICHN,IAD1,IER)
      CALL ERROR(IER,'I1')
      CALL RDS(ICHN,IC1,LREC,IER)
      CALL ERROR(IER,'RDS - IC1')
      D2(1)=0
      D2(2)=LREC
      CALL DSUB(D2,D2,IAD1)
      CALL DMPY(D1,NREC,LREC)
      CALL DSUB(IAD2,D1,D2)
      CALL SPOS(ICHN,IAD2,IER)
      CALL ERROR(IER,'I2')
      CALL RDS(ICHN,IC2,LREC,IER)
      CALL ERROR(IER,'RDS-IC2')
      CALL BCOMP(IC1(IFLDP),ITEST,IFLD,IER1)
      IF(IER1.GT.1)GO TO 100
      CALL BCOMP(IC2(IFLDP),ITEST,IFLD,IER2)
      IF(IER2.NE.2)GO TO 125
    5 CALL DSUB(D1,IAD2,IAD1)
      CALL DDVD(INC,IR,D1,LREC)
      IF(INC.GE.32767)GO TO 900
      IF(INC.LT.1)GO TO 150
      INC=(INC-1)/2+1
      CALL DMPY(D1,INC,LREC)
      CALL DADD(IAD3,IAD1,D1)
      CALL SPOS(ICHN,IAD3,IER)
      CALL ERROR(IER,'I5')
      CALL RDS(ICHN,IC3,LREC,IER)
      CALL ERROR(IER,'I6')
      CALL BCOMP(IC3(IFLDP),ITEST,IFLD,IER3)
      IF(IER3.EQ.1)GO TO 50
      IF(IER3.EQ.2)GO TO 60
      IF(IER3.NE.3)GO TO 900
      IAD(1)=IAD3(1)
      IAD(2)=IAD3(2)
      IC=3
```

```
      RETURN
   50 IAD1(1)=IAD3(1)
      IAD1(2)=IAD3(2)
      GO TO 5
   60 IAD2(1)=IAD3(1)
      IAD2(2)=IAD3(2)
      IF(INC.EQ.1)GO TO 150
      GO TO 5
  100 IAD(1)=IAD1(1)
      IAD(2)=IAD1(2)
      IF(IER1.NE.3)GO TO 101
      IC=1
      IAD(1)=IAD1(1)
      IAD(2)=IAD1(2)
  101 RETURN
  125 D1(1)=0
      D1(2)=LREC
      CALL DADD(IAD,D1,IAD2)
      IF(IER2.NE.3)GO TO 126
      IAD(1)=IAD2(1)
      IAD(2)=IAD2(2)
      IC=2
  126 RETURN
  150 IAD(1)=IAD3(1)
      IAD(2)=IAD3(2)
      RETURN
  900 CALL ERROR(IER3,'IER3')
      IER=2
      CALL ERROR(IER,'TOO MANY RECORDS IN FILE')
      STOP
      END
```

```
      FUNCTION WMROF (P,TD)
C
C  COMPUTE MIXING RATIO (G/KG)--DPT (DEG C)--PRES (MBS)
C
      T=TD
      IF(100.-T) 3,4,4
    3 T=T-273.16
    4 X=0.0200*(T-12.5+7500./P)
      WFW=1+0.0000045*P+0.00140*X*X
      FWESW=WFW*VAPFW(T)
      WMROF=621.97*(FWESW/(P-FWESW))
      RETURN
      END
```

```
      FUNCTION VAPFW(T)
C
C COMPUTE SATURATION VAPOR PRESSURE OVER WATER--VAPFW IN MBS--TMP (DEG C)
C
      X=T
      IF(100.-X) 3,4,4
    3 X=X-273.16
    4 POL=0.99999683 E-00 + X*(-0.90826951 E-02 +
     * X* (0.78736169 E-04 + X*(-0.61117958 E-06 +
     * X* (0.43884187 E-08 + X*(-0.29883885 E-10 +
     * X* (0.21874425 E-12 + X*(-0.17892321 E-14 +
     * X* (0.11112018 E-16 + X*(-0.30994571 E-19)))))))))
      POL=POL*POL
      POL=POL*POL
      VAPFW=6.107800/(POL*POL)
      RETURN
      END
```

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)

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

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:

ENVIRONMENTAL SCIENCE INFORMATION CENTER (D822)
ENVIRONMENTAL DATA AND INFORMATION SERVICE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
U.S. DEPARTMENT OF COMMERCE

6009 Executive Boulevard
Rockville, MD 20852