NOAA Eastern Region Computer Programs
and Problems NWS ERCP - No. 16

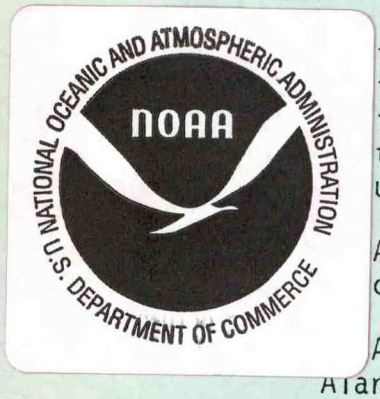
# RANP:
# Stability Analysis Plot Program

Hugh M. Stone
National Weather Service Eastern Region
Garden City, New York

Scientific Services Division
Eastern Region Headquarters
February 1984

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

National Weather Service, Eastern Region Computer Programs and Problems




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

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

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.

# RANP
# STABILITY ANALYSIS PLOT PROGRAM

Hugh M. Stone
Scientific Services Division
National Weather Service Eastern Region
Garden City, New York

## I. General Information

### A. Summary

New indicators of static stability, the energy indices EI1 and EI2, are now available (Stone, 1983) and may be computed with the AFOS application program RAN. These indices, formerly called B1 and B2, are based on the change of kinetic energy of a parcel as it moves upward through the atmosphere entraining environmental air during its ascent. Unlike the more traditional stability indices, i.e. Lifted, K, Showalter, etc., which use only a few levels of the sounding, the energy indices utilize all the information in significant level sounding (UJ1) and should provide a more accurate measure of static stability.

The RAN program computes the EI1 and EI2 indices, the equilibrium level, a variety of other traditional stability indices, and several other parameters, which can be derived from significant level raob data; the computation is done for a single raob station. For many purposes, the geographical distribution of these parameters is more important than a point value. The RANP program does the same computation as RAN, but instead of using a single raob station, the computation may be done for a group of stations specified in a list. The output consists of tabulated values for 12 parameters and a graphic with plotted values of two of the most important parameters, the EI1 index and the equilibrium level EL.

A switch has been provided so RANP can also be run for a single station specified in the RUN line. In this case, the output is identical to that provided by the RAN program and provides a detailed analysis of the significant level raob sounding. The RAN program may therefore be replaced by RANP.

RANP was developed to run on an Eclipse S/230 minicomputer. Language used is Data General FORTRAN IV.

## II. Application

### A. Program Description

The motivation for the development of the energy indices EI1 and EI2 (formerly B1 and B2) and the method of computation has already been given in ERCP No. 9 (Stone, 1983). Experience with these indices during the 1983 convective season suggested that the entrainment rate used in the computation was excessive. The EI1 index had a bias toward stable

conditions with convection occasionally beginning with values of EI1 as low as -80. Likewise, equilibrium levels seemed to be too low.

Changes in the entrainment rate have now been made to help correct these defects. The method of computation is still the same except that the entrainment rate for the energy index computation has been reduced from 100 percent to 60 percent. This means a 60 percent increase in mass of the parcel over a 500mb ascent rather than the doubling of mass used previously. We are still not sure this is the best entrainment rate and it may need to be changed again.

Changing the entrainment rate from 100 to 60 percent usually has the effect of making unstable indices more unstable, slightly stable indices become unstable, and very stable indices can go either way depending on the moisture distribution.

The equilibrium level is now computed separately using a zero entrainment rate. The interior portions of large thunderstorms that may produce severe weather are virtually unaffected by the entrainment process. Since the radar tops of large storms are compared to the equilibrium level to assess their potential for severe weather, we believe the zero entrainment rate is appropriate.

The EI2 index was previously defined (Stone, 1983) as the change in kinetic energy of a parcel as it moves from the level of maximum instability (PMAX) to the equilibrium level. This definition is still correct provided that the "equilibrium level" is computed with entrainment; it is not the equilibrium level EL that appears in the output.

The equilibrium level EL is first determined in pressure units then the mandatory level raob transmission (US1) is used to interpolate a value in feet, which makes it easier to compare to radar tops which are also reported in feet. If the mandatory levels are missing for any reason, the interpolation is done using the U.S. Standard Atmosphere, with the value followed by an "E" in the output to indicate estimated. The same procedure is used for tropopause height which is read from the mandatory level transmission and converted to feet for comparison to the equilibrium level. Both equilibrium level and tropopause height in hundreds of feet are given in the AFOS product WRKTPC (figure 1).

The following parameters are included in the AFOS product WRKTPC:

| | |
|---|---|
| PØ | surface pressure (mb) |
| PMAX | pressure (mb) of highest wet bulb potential temperature in the lowest 150mb of the sounding. Starting point of parcel ascent. |
| EL | Equilibrium level in (mb) and (Ft. X $10^2$) |
| TROP | Tropopause level (Ft. X $10^2$) |
| EI1 | Energy Index 1. Change in kinetic energy of parcel moving from level PMAX to 400mb level. (J/Kg X 10) |
| EI2 | Energy Index 2. Change in kinetic energy of parcel moving from level PMAX to the "equilibrium level computed with entrainment" (J/Kg X 10) |

| | |
|---|---|
| LI | Lifted Index ($^{o}C$) |
| KI | K Index ($^{o}C$) |
| SWI | Showalter Index ($^{o}C$) |
| CCL | Convective Condensation Level (mb) |
| ETCCL | Amount of low level heating required for surface parcel to ascend dry adiabatically to CCL level. (J/Kg X 10) |

Missing values indicated by "999."

Two of the most important parameters, the energy index EI1 and the equilibrium level EL in hundreds of feet, are plotted on the AFOS graphic NMCGPHEIS (figure 2). EI1 appears to the upper right of the station circle and EL to the upper left. Positive (unstable) values of EI1 are plotted with a solid station circle and negative (stable) values with an open station circle. The graphic EIS uses map background 2, North America. A zoom of 4:1 or more must be used. The raob stations used in the computation must be listed in the RDOS file STNS1 (figure 3).

If RANP is run for a single station, the output goes into AFOS products WRKTPA (figure 4) and WRKTPB (figure 5). These are the same products created by the program RAN and have the same format, which was explained in ERCP No. 9 (Stone, 1983).

B. Machine Requirements

Using a list of 32 raob stations, covering approximately the eastern third of the United States, the total runtime is usually around 8 minutes. Two FORTRAN channels are open during the program run. One RDOS channel is used at the end of the computation to insert headings and endings for the AFOS alphanumeric products. Disk space required for the program RANP.SV is 122 RDOS blocks with an accompanying overlay file RANP.OL occupying 28 RDOS blocks.

C. Database

Products that are referenced:

1. CCCSGLXXX : significant level raob soundings listed in file STNS1 or given in run line.

2. CCCMANXXX : corresponding mandatory level raob soundings. If these are not available for any reason, the program will still run, but a U.S. Standard Atmosphere will be assumed in the computation of equilibrium level in feet and tropopause height will be missing.

Files/products that are created:

1. INDEXX : temporary storage file for data that are eventually stored as AFOS products WRKTPA, WRKTPC, or WRKTPD. INDEXX is deleted at the end of the computation.
2. INDEXY : temporary storage for data that are eventually stored as AFOS product WRKTPB. INDEXY is deleted at the end of the computation.

3. HMSGPH.Ø1 : temporary storage file for graphics data that are eventually stored as AFOS product NMCGPHEIS. HMSGPH.Ø1 is deleted at the end of the computation.

D. Structure of the Software

RANP is the main program. The program checks the system clock to determine which raob data to process. After 00Z, the program uses 00Z data from the current date; likewise, after 12Z, 12Z data is used. Sufficient time must be allowed after these cutoff times for the new raob data to arrive in the database. The database may be checked prior to beginning the computation by using the global switch "C"; see Part III, Section B. The computation is done for each raob station listed in the file STNS1 or for the single station specified in the run line. If the computation is done for a single station, the date and time of the raob report are not checked for currency.

All of the computations are accomplished by a series of calls to various subroutines. Interpolations and extrapolations of temperature and dewpoint in all of the subroutines are done assuming a linear variation of the quantity with the logarithm of pressure. At the end of the computation, the alert light is turned on and all temporary files are deleted.

Sixteen of the subroutines (indicated by *) are also common to the program RAN (Stone, 1983), but most of them have been changed or corrected and are repeated here for reference.

The function of the various subroutines are as follows:

DECOS *

Reads the temperature portion of the UJ1 raob specified in the array JST, utilizing the AFREAD subroutine (Peroutka, 1981).

TEMP1 *

This subroutine called by DECOS for decoding temperature and dewpoint.

INDX1 *

Computes lifted index, K index, and Showalter index. When surface pressure is less than 850mb, K and Showalter index cannot be computed. This will be indicated by 999 on the output.

BNDX *

Determines pressure level PMAX that has the highest wet bulb potential temperature in the lowest 150mb of the sounding. If an identical maximum value is found at 2 levels, the lowest level (highest pressure) is selected for PMAX. A modified raob is created which has its base at level PMAX and an additional significant level is added at level PX = 400mb, if a significant level does not already exist there. If the raob terminates below level PX, but within 50mb of PX, a level PX is extrapolated, so that the index EI1 may still be computed.

RANN2 *

The principal subroutine, does all the energy area computations as the parcel is raised from the bottom to the top of the sounding. The first half of the subroutine raises a parcel along a dry adiabat from PMAX to the lifting condensation level (LCL). Since entrainment is allowed during the dry ascent, the LCL is usually slightly higher than it would be if there were no entrainment.

The second half of the subroutine continues a moist ascent above the LCL. Entrainment is continued all the way using the method proposed by Austin (1948).

A 50mb step is used for the parcel ascent, but if a significant level is present within the next 50mb, the step is reduced to terminate at that level. Steps are likewise shortened, if the energy area changes sign, or if the LCL is reached during the next 50 millibars. The 50mb step was selected because it gives sufficient accuracy and is economical in computer time. On rare occasions when the parcel temperature and environmental temperature are very close over a large depth of the atmosphere, the 50mb step may be unsatisfactory. In such a case, the computation is restarted using a 10mb step.

The stability indices EI1 (B1) and EI2 (B2) are computed after the ascent has been completed. The level of free convection is also determined here.

The subroutine RANN2 is called three times:

(1) The first call calculates the variable ETCCL, which represents the amount of low level heating required for a surface parcel to reach the convective temperature and then move dry adiabatically to the convective condensation level, CCL. No entrainment is allowed in the calculation of ETCCL.

(2) The second call computes equilibrium level with zero entrainment.

(3) The third call computes the energy indices, the lifting condensation level LCL, and the level of free convection LFC. Sixty percent entrainment is normally used here.

CCL1 *

Computes the convective condensation level CCL.

MODRB *

Called immediately after CCL1 is finished to modify the original raob so ETCCL can be computed.

WOBF SATLFT TCONOF WMROF DPTOF VAPFW *

All thermodynamic computations are done using these six subroutines from the National Severe Storms Forecast Center, MO. (Doswell, et al., 1982).

ITCVT *

Converts ASCII characters to integer values. This is a modification of function INTCVT in AFREAD.LB (Peroutka, 1981).

FTCVT *

Converts ASCII characters to floating point values. This is a modification of function FLTCVT in AFREAD.LB (Peroutka, 1981).

PULYR *

Determines all potential (convective) unstable layers and computes lapse rate of wet bulb potential temperature in the layers and amount of lifting required to achieve saturation at the base and top pressure of the layer.

TPA

Outputs data for single station analysis to INDEXX file, which is later stored as AFOS product WRKTPA.

TPB

Calls subroutine PULYR to determine potentially unstable layers with results output to INDEXY file, for subsequent transfer to AFOS product WRKTPB.

FTCV

This function is used for reading numerical data input by switches in the RUN line. It is a modification of function FLTCVT in AFREAD.LB (Peroutka, 1981).

DECOM

Reads the mandatory level raob data, US1, utilizing the AFREAD subroutine (Peroutka, 1981).

IVCK

This function, used by DECOS and DECOM, checks the date time group of the raob to assure that the correct version is being used.

WND

Used by DECOM for decoding winds of US1 raob data.

HEIGHT

Computes height of an arbitrary pressure by interpolating or extrapolating heights of mandatory levels.

JREAL

Rounds a floating point number to an integer.

STLOC

This uses the subroutine BNSCH to search the station directory file to find X and Y coordinates of each raob station for plotting on a graphic.

BNSCH

Subroutine for binary search of data in the station directory file. (This program written by Rich Thomas, AOD).

GPT

Creates a graphic displaying geographical distribution of EI1 and EL using subroutines given by MacDonald (1981).

ISCR

Converts a positive or negative integer of up to three digits to ASCII characters. This is used to convert EI1 and EL for plotting on graphics.

JSCR

Converts a positive two digit integer to ASCII characters. This is used to get date/time numbers for plotting on graphics.

MTITL

Makes date/time heading for graphics and converts number of month to three letter abbreviation in ASCII characters.

The complete program would not fit in memory without the use of overlays.

If the computation is done for the list of raob stations in file STNS1, the following overlays are used:

OVØ - Subroutines STLOC, BNSCH
OV1 - Subroutines GPT, ISCR, JSCR, MTITL.

These are used for creating the graphic EIS.

If the computation is done for a single raob station, the following overlays are used:

OV2 - Subroutines TPB, PULYR
OV3 - Subroutine TPA

These are used to output results to AFOS products WRKTPB and WRKTPA.

## III Program Reference

### A. Preparation

The following AFOS products must be added to the database (CCC is the local node):

CCCWRKTPA
CCCWRKTPB
CCCWRKTPC
CCCWRKTPD
NMCGPHEIS

The program RANP.SV, its overlay file RANP.OL, and the list of raob stations in file STNS1 must be on the main disk DPØ or DPØF with a link to DPØ.

The file STNS1 (figure 3) may contain up to 50 raob stations (significant level AFOS identifiers). The list is read using the FORMAT (5A2). If a list of more than 50 raob stations is needed, the parameter "NRAOB" must be changed in the main program RANP and the subroutines GPT and STLOC.

Significant level raob data for all stations specified in STNS1 must be in the database. Corresponding mandatory level data should also be in the database; if it is missing, the U.S. Standard Atmosphere is assumed for the computation of equilibrium level in feet.

### B. Initiating the Program

To run the program at the ADM to compute stability indices for the list of stations in file STNS1, type:

RUN:RANP

A switch is available for changing the entrainment rate from its basic value of 60 percent. For example, if you wished to compare values of EI1 to values of B1 which used 100 percent entrainment (Stone, 1983), you could type:

RUN:RANP 100/E

We recommend that this switch <u>not</u> be normally used.

A global switch "C" may be used to check the database for the availability of significant level raobs. If a new raob is not available, or if the raob is not in proper format, or if it cannot be read for any reason, a message indicating the trouble is output to AFOS product WRKTPD (figure 6). To check the database at the ADM console, type:

RUN:RANP/C

This check takes approximately 1½ minutes for 32 raob stations.

A global switch "S" may be used to obtain a detailed raob analysis for a single station. In this case, at the ADM, type:

RUN:RANP/S CCCSGLXXX

The local switch "E" for changing the entrainment rate may also be used in combination with the "S" switch. Running time for the single station computation is usually around 40 to 60 seconds.

C. Output

The alert light is turned on when the program is finished. Output goes to various AFOS products depending on the global switches used:

| SWITCH | OUTPUT |
|---|---|
| None | WRKTPC and EIS |
| C | WRKTPD |
| S | WRKTPA and WRKTPB |

D. Cautions or Restrictions on Use

If significant level raob data is missing, or not in proper format, or raob terminates below 450mb, or a dewpoint is missing below the 700mb level, the computation is not performed and a message indicating the trouble will appear in AFOS product WRKTPC or WRKTPA. This situation will be indicated on the EIS graphic by a station circle with an "M" inside it.

If mandatory level raob data are missing or not in proper format, a message indicating this is printed on the Dasher. A U.S. Standard Atmosphere is then assumed and the computation continues.

As the program works its way through the list of raob stations in file STNS1, numbers appear on the Dasher indicating which raob data is currently being processed, i.e. "9" would indicate that data for the ninth station on the list is currently being computed. A mistake in the significant level raob data, such as a completely unrealistic temperature for some level, can cause the program to hang up. This is a very rare event, but if it happens, the numbers printed on the Dasher indicate which raob is causing the trouble. The report can either be corrected or purged and the program restarted.

When printing any of the alphanumeric products WRKTPA, B, C, or D from the ADM using the command "PRINT:", an extraneous line of printing frequently appears on the PPM copy. This problem will hopefully be corrected by AFOS Version AOD 10.00. Meanwhile, a perfect copy can be obtained by using the command, "PRINT:WRKTPA", "B", "C", or "D".

E. Load Line

```
RLDR/P RANP DECOS TEMP1 RANN2 CCL1 MODRB INDX1 BNDX WOBF SATLFT TCONOF
 WMROF DPTOF VAPFW DECOM IVCK WND HEIGHT JREAL FTCV
 [TPB PULYR, TPA, STLOC BNSCH, GPT ISCR JSCR MTITL]
 OUT AFREAD.LB ITCVT FTCVT TOP.LB AG.LB UTIL.LB FORT.LB
```

F. Program Listings:

See pages 16 through 55.

Subroutine Index on page 55.

ACKNOWLEDGEMENT

Thanks to Cynthia Scott for programming advice, and Fortune Vilcko for the typing of this manuscript.

REFERENCES

Austin, J.M., 1948: A Note on Cumulus Growth in a Nonsaturated Environment, Journal of Meteorology, 5, 103-107.

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 Conf. Weather Forecasting and Analysis, Amer. Meteor. Soc., Seattle, 304-309.

MacDonald, A.E., 1981: AFOS Graphics Creation from FORTRAN, NOAA Western Region Computer Programs and Problems NWS WRCP-No. 18, National Weather Service, Salt Lake City, UT.

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

Stone, H.M., 1983: Stability Analysis Program, NOAA Eastern Region Computer Programs and Problems NWS ERCP-No. 9, National Weather Service, Garden City, NY

RAOB INDICES FOR 2/ 2/84 12Z DP = 50. EFF = 60. PX = 400.

ENERGY UNITS : J/KG X 10 EL & TROP IN HNDS FT

| | STN | P0 | PMAX | EL(MB) | EL(FT) | TROP | EI1 | EI2 | LI | KI | SWI | CCL | ETCCL | STN |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | CAR | 995. | 845. | 845. | 47. | 286. | -146. | 999. | 26. | 8. | 11. | 640. | 111. | CAR |
| 2 | PWM | 1021. | 871. | 871. | 41. | 329. | -228. | 999. | 27. | -21. | 17. | 495. | 299. | PWM |
| 3 | ALB | 1017. | 907. | 907. | 32. | 358. | -290. | 999. | 33. | -24. | 20. | 485. | 332. | ALB |
| 4 | BUF | 995. | 845. | 845. | 50. | 364. | -217. | 999. | 22. | -23. | 16. | 633. | 164. | BUF |
| 5 | DAY | 982. | 832. | 832. | 54. | 380. | -164. | 999. | 17. | -11. | 12. | 616. | 127. | DAY |
| 6 | PIT | 979. | 829. | 829. | 55. | 366. | -207. | 999. | 19. | -26. | 17. | 554. | 211. | PIT |
| 7 | CHH | 1023. | 873. | 873. | 41. | 307. | -242. | 999. | 25. | -48. | 17. | 456. | 317. | CHH |
| 8 | ACY | 1026. | 876. | 876. | 41. | 376. | -295. | 999. | 29. | -25. | 20. | 504. | 323. | ACY |
| | WAL | NEW RAOB NOT AVAILABLE | | | | | | | | | | | | |
| 10 | IAD | 1016. | 866. | 866. | 44. | 372. | -265. | 999. | 27. | -18. | 18. | 536. | 293. | IAD |
| 11 | HTS | 991. | 850. | 850. | 49. | 999. | -199. | 999. | 18. | -17. | 13. | 576. | 171. | HTS |
| 12 | HAT | 1028. | 1014. | 1014. | 4. | 379. | -316. | 999. | 25. | -39. | 22. | 505. | 338. | HAT |
| 13 | GSO | 993. | 843. | 843. | 52. | 365. | -207. | 999. | 20. | -15. | 15. | 539. | 212. | GSO |
| 14 | CHS | 1024. | 1012. | 1012. | 4. | 371. | -279. | 999. | 19. | -24. | 16. | 558. | 219. | CHS |
| 15 | BNA | 999. | 849. | 849. | 50. | 377. | -192. | 999. | 16. | -22. | 13. | 574. | 164. | BNA |
| 16 | AHN | 994. | 844. | 844. | 52. | 359. | -204. | 999. | 17. | -29. | 15. | 552. | 184. | AHN |
| 17 | AYS | 1019. | 1000. | 1000. | 7. | 344. | -264. | 999. | 20. | -13. | 14. | 545. | 210. | AYS |
| 18 | BVE | 1021. | 1000. | 1000. | 4.E | 999. | -180. | 999. | 12. | 5. | 15. | 760. | 62. | BVE |
| 19 | JAN | 1011. | 861. | 861. | 44.E | 999. | -228. | 999. | 20. | -23. | 15. | 490. | 244. | JAN |
| 20 | CKL | 1006. | 856. | 856. | 46.E | 999. | -213. | 999. | 19. | -20. | 14. | 570. | 182. | CKL |
| 21 | AQQ | 1023. | 873. | 873. | 41.E | 999. | -222. | 999. | 18. | -26. | 14. | 521. | 211. | AQQ |
| 22 | TBW | 1020. | 929. | 929. | 24.E | 999. | -142. | 999. | 14. | -4. | 13. | 636. | 130. | TBW |
| 23 | EYW | 1019. | 869. | 869. | 42.E | 999. | -91. | 999. | 5. | 7. | 14. | 766. | 60. | EYW |
| 24 | PBI | 1020. | 1000. | 783. | 69.E | 999. | -105. | 0. | 6. | -1. | 6. | 715. | 114. | PBI |
| 25 | SSM | 989. | 839. | 839. | 49. | 346. | -160. | 999. | 21. | 7. | 14. | 637. | 148. | SSM |
| 26 | FNT | 988. | 904. | 904. | 31. | 369. | -202. | 999. | 19. | -11. | 12. | 649. | 130. | FNT |
| 27 | GRB | 986. | 836. | 836. | 50. | 372. | -184. | 999. | 26. | -31. | 16. | 543. | 200. | GRB |
| 28 | PIA | 987. | 837. | 821. | 56. | 355. | -117. | 0. | 16. | 5. | 8. | 644. | 92. | PIA |
| 29 | SLO | 993. | 843. | 843. | 50. | 376. | -138. | 999. | 15. | -0. | 11. | 598. | 134. | SLO |
| 30 | UMN | 961. | 811. | 811. | 60.E | 999. | -86. | 999. | 14. | 5. | 5. | 644. | 114. | UMN |
| 31 | LIT | 997. | 847. | 701. | 99.E | 999. | -115. | -5. | 20. | -5. | 7. | 522. | 222. | LIT |
| 32 | LCH | 1020. | 897. | 897. | 33.E | 999. | -133. | 999. | 14. | 19. | 11. | 582. | 205. | LCH |

Figure 1. Example of WRKTPC

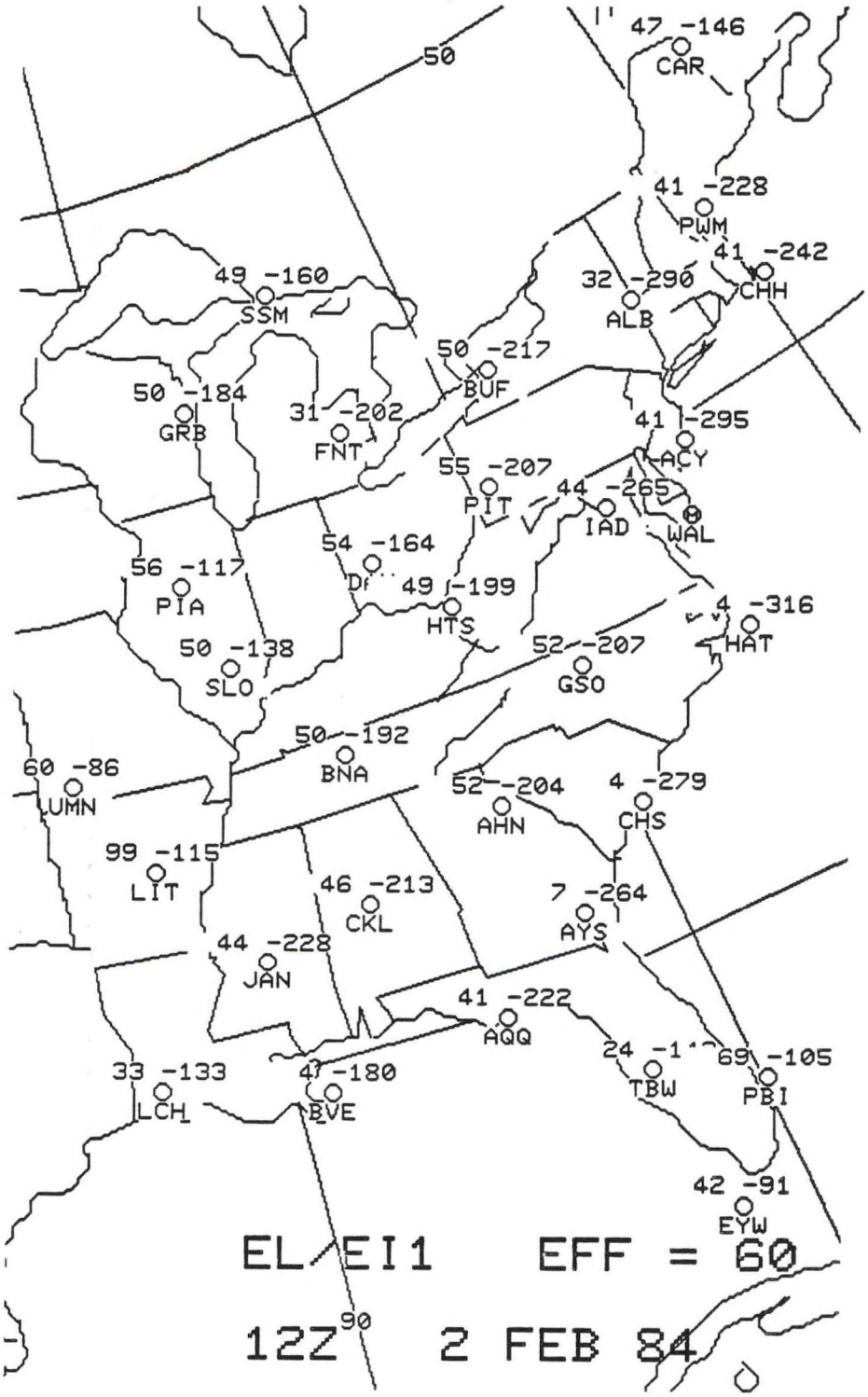



Figure 2. Example of EIS graphic.

```
PWMSGLCAR
PWMSGLPWM
ALBSGLALB
BUFSGLBUF
CLESGLDAY
PITSGLPIT
BOSSGLCHH
PHLSGLACY
WBCSGLWAL
WBCSGLIAD
CRWSGLHTS
RDUSGLHAT
RDUSGLGSO
CAESGLCHS
MEMSGLBNA
ATLSGLAHN
ATLSGLAYS
NEWSGLBVE
JANSGLJAN
BHMSGLCKL
BHMSGLAQQ
MIASGLTBW
MIASGLEYW
MIASGLPBI
ARBSGLSSM
ARBSGLFNT
MKESGLGRB
CHISGLPIA
CHISGLSLO
STLSGLUMN
LITSGLLIT
NEWSGLLCH
```

Figure 3. Example of file STNS1.

```
RAOB ANALYSIS FOR PBI     2/ 2/84   12Z          UNITS : J/KG X 10
  ASSUMED EFF =  60. PERCENT ENTRAINMENT PER 500MB ASCENT, FOR EI1, EI2 & EL60
  ASSUMED EFF =   0. PERCENT ENTRAINMENT PER 500MB ASCENT, FOR EL

P0 = 1020.    PTOP =  100.    PMAX (MAX INSTABILITY) = 1000.    PX =  400.
EL =  783. MB ( 69.E HND FT)      LCL =  898.    LFC =  865.
BASED ON PARCEL MVG FM LVL -PMAX-

EI2  =      0.  ENERGY PMAX TO EL60   EI1  =   -105.  ENERGY PMAX TO PX
EI2P =      3.  POSITIVE PART         EI1P =      3.  POSITIVE PART
EI2N =     -2.  NEGATIVE PART         EI1N =   -108.  NEGATIVE PART

P1         P2         ENERGY GAINED (LOST) IN LAYER
1000.      865.         -2.
 865.      784.          3.
 784.      100.       -814.
EX =   -106.

LI =   6.    KI =  -1.    SWI =   6.

CCL =  715.  ETCCL =   114.  CONV TEMP =  34.7 ( 94.5F )  WAVG =  7.73 G/KG

DEEPEST POT. UNSTABLE LYR : 1000. -  789. MB, TWLAPSE =   3.6  SEE WRKTPB
```

Figure 4. Example of WRKTPA

POTENTIAL (CONVECTIVE) UNSTABLE LAYERS FOR PBI 2/ 2/84 12Z

| P1 | P2 | DP | TWLAPSE | DP1 | DP2 |
|---|---|---|---|---|---|
| 429. | 392. | 37. | 3.3 | 22. | 160. |
| 615. | 453. | 162. | 0.2 | 11. | 36. |
| 1000. | 789. | 211. | 3.6 | 100. | 300. |

SIGNIFICANT LEVELS

| P | T | TD | TW (WET BULB POTENTIAL TEMP) |
|---|---|---|---|
| 100. | -70.3 | -100.3 | 30.7 |
| 108. | -72.7 | -102.7 | 28.7 |
| 126. | -65.3 | -95.3 | 28.2 |
| 259. | -48.3 | -78.3 | 18.2 |
| 300. | -39.7 | -69.7 | 17.8 |
| 349. | -32.5 | -62.5 | 16.7 |
| 360. | -32.9 | -62.9 | 15.6 |
| 392. | -28.7 | -58.7 | 15.0 |
| 429. | -22.9 | -26.0 | 16.2 |
| 453. | -20.1 | -25.1 | 15.8 |
| 500. | -14.9 | -16.3 | 16.1 |
| 615. | -4.3 | -5.4 | 16.1 |
| 621. | -4.7 | -13.7 | 13.7 |
| 630. | -4.1 | -23.1 | 12.2 |
| 729. | 6.4 | -23.6 | 11.6 |
| 775. | 5.6 | -24.4 | 9.1 |
| 789. | 0.6 | -29.4 | 5.8 |
| 805. | 2.4 | 2.0 | 11.9 |
| 850. | 5.8 | 5.5 | 12.7 |
| 1000. | 17.4 | 10.4 | 13.3 |
| 1015. | 15.8 | 11.1 | 12.4 |
| 1020. | 12.2 | 10.6 | 10.4 |

Figure 5. Example of WRKTPB

```
SGL REPORTS MISSING OR INCORRECT FORMAT,  2/ 2/84  12Z
   WAL  NEW RAOB NOT AVAILABLE
```

Figure 6. Example of WRKTPD

```
C   RANP.FR
C    THIS VERSION WORKS WITH C ,E AND S SWITCH.
C   "RANP" COMPUTES STABILITY INDICES FOR SET OF RAOB STATIONS,
C   SPECIFIED IN FILE: "STNS1" WITH
C   OUTPUT TO AFOS PRODUCT "WRKTPC" AND "NMCGPHEIS".
C
C   GLOBAL SWITCH "C" DOES CHECK OF DATABASE ONLY; "SGL" REPORTS THAT
C   CANNOT BE READ ARE LISTED IN "WRKTPD".
C
C   GLOBAL SWITCH "S" MAY BE USED FOR A SINGLE STATION COMPUTATION WITH
C   OUTPUT TO AFOS PRODUCTS "WRKTPA" & "WRKTPB".
C
C   ZERO PERCENT ENTRAINMENT ALWAYS USED FOR EQUILIBRIUM LEVEL.
C   SIXTY PERCENT ENTRAINMENT FOR ALL OTHER COMPUTATIONS, BUT MAY BE
C   CHANGED WITH LOCAL SWITCH "E".
C
C   FOLLOWING PARAMETER MUST EQUAL OR EXCEED NUMBER OF RAOB STNS TO BE
C   PROCESSED, AND MUST AGREE WITH ~NRAOB~ PARAMETER IN SUBROUTINES ~GPT~
C   AND 'STLOC'.
C
      PARAMETER NRAOB=50
C
      COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
      COMMON/T/RLCL,RLFC,EL,B2,B2P,B2N,IALL,B1,B1P,B1N,EX
      COMMON/CCL/PCCL,ETCCL,TS0,TSD0,L,TSCCL,TCCL,TDCCL,WAVG
      COMMON/G/PP(0:20),ET(20),TW(0:50),DP,EFF,KMOD,KK
      COMMON/GG/NJ,PPB(15),PPT(15),DELPP(15),DTWDP(15),DPB(15),DPT(15),
     1  PTMAX,PBMAX,TWLAPSE,DMAX    :   FOR PULYR SUBROUTINE
      COMMON/TT/PT(0:50),TST(0:50),TSDT(0:50)
      COMMON/V/JNOM,PX
      COMMON/H/IHDR1(11),KEY(5)
      COMMON/ZZZ/ZZ(0:12)
      DIMENSION IXX(NRAOB),IYY(NRAOB),JB(NRAOB),JEL(NRAOB),IST(NRAOB,2)
      DIMENSION H(0:12),T(0:12),TD(0:12),D(0:12),S(0:12)
      INTEGER SW(2),DAT(10)
      EXTERNAL OV0,OV1,OV2,OV3    :   OV0 & OV1 - NO SWITCH, OV2 & OV3  - S SWITCH
      DATA IHDR1/"    RKTPC000",177777K,177777K,"70",142600K,6412K/
      DATA KEY/"WRKTPC"/
C   "ZZ" HEIGHT OF STANDARD PRESSURE SURFACES, U. S. STANDARD ATMOSPHERE
      DATA ZZ/0.,111.,1457.,3012.,5574.,7185.,9164.,10363.,11784.,13608.,
     1  16180.,999.,0./
      IFD=10    :   OUTPUT DEVICE FOR ERROR MESSAGES FROM DECOM
      IFC=20    :   OUTPUT DEVICE FOR ERROR MSGS FM DECOS, BNDX, INDX1 SUBROUTINES
      CNVM=.032808399    :   CONVERSION FACTOR, M TO FT X 10-2
      CALL KFILL (KEY,IER)
      IHDR1(1)=KEY(1)
      IHDR1(2)=KEY(2)
      IEND=101603K    :   ENDING FOR AFOS PRODUCT
      DP=50.          :   50 MB STEP
      IALL=1          :   IALL=2 TO PRINT EVERY LVL IN RANN2 SUBROUTINE
      PX=400.         :   CUT-OFF PRESSURE FOR EI1 INDEX
      N=0     :    COUNTER FOR NUMBER OF RAOB STATIONS
      CALL FCOM (IC,IER)
      CALL COMCM (IC,DAT,11,SW,IER)    :   READING: RUN:RANP
      ICK=0
      IF (ISWSET(SW,"C")) ICK=1   :   ICK=1 DENOTES DATABASE CHECK ONLY
      IF (ISWSET(SW,"S")) ICK=2   :   ICK=2 DENOTES SINGLE STATION COMPUTATION
      IF (ICK.EQ.0) GO TO 45      :   ICK=0 DENOTES FULL COMPUTATION FOR LIST OF STATIONS
      IF (ICK.EQ.2) GO TO 50
      IHDR1(5)="D0"
```

```
      GO TO 41
45    CALL CFILW("HMSGPH.01",2,IER)    ;   CREATING GRAPHIC FILE
      IF (IER.NE.1) WRITE (10,35) IER
35    FORMAT (1H ,"IER = ",I4," CFILW, PROGRAM RANP, STATEMENT 35")
      GO TO 49
C   SINGLE STATION COMPUTATION
50    IHDR1(5)="A0"    ;   DENOTES INDEXX TO STORE IN WRKTPA
      DO 51 I=1,10
51    DAT(I)=0
      CALL COMCM(IC,DAT,11,SW,IER)    ;   READING CCCSGLXXX
      DO 52 I=1,5
52    JST(I)=DAT(I)
      JER=1    ;   INDICATOR THAT SINGLE STATION COMPUTATION COMPLETED OK
C
49    DO 43 I=1,10
43    DAT(I)=0
      CALL COMCM (IC,DAT,11,SW,IER)    ;   READING ENTRAINMENT RATE
      IF (ISWSET(SW,"E")) GO TO 40    ;   SPECIAL EFF HAS BEEN READ
      EFF0=60.    ;    NORMAL ENTRAINMENT RATE IN PERCENT
      GO TO 41
40    EFF0=FTCV(DAT,$44)
      WRITE (10,42) EFF0
42    FORMAT (1H ,"EFF0 = ",F5.0)
      GO TO 41
44    CALL FORKE ("RANP","EFF0",IER)
      CALL KLOSE(IC,IER)
      STOP
41    CONTINUE
      CALL KLOSE (IC,IER)
      CALL FOPEN (20,"INDEXX",300)
      CALL FGTIME (IHR,IMIN,ISEC)    ;   GET TIME
      CALL DATE (KDATE,IER)          ;   (MO,DY,YR) GET DATE
      KDATE(3)=KDATE(3)-1900    ;   MAKING 2 DIGIT YEAR
      KDAT=KDATE(2)    ;   SAVE ORIGINAL KDATE(2) FOR SINGLE STN COMPUTATION
      IHOUR=0
      KTIME=IHR*100+IMIN
C
      IF (KTIME.GT.1200) IHOUR=12    ;   DOES LATEST TIME
      IM=ICK+1
      GO TO (20,53,58),IM
58    WRITE (20,59)    ;   LEAVING ROOM FOR HEADER AT BGNG OF INDEXX FILE
59    FORMAT (12X,"            ")
      KDATE(2)=0    ;   SIGNIFIES NOT TO DO DATE/TIME CHECK
      GO TO 54
53    WRITE (20,29) (KDATE(I),I=1,3),IHOUR
29    FORMAT (12X,"            SGL REPORTS MISSING OR INCORRECT FORMAT, ",
     1  I2,"/",I2,"/",I2,2X,I2,"Z")
      GO TO 31
20    WRITE (20,21) (KDATE(I),I=1,3),IHOUR,DP,EFF0,PX
21    FORMAT (12X,"            RAOB INDICES FOR ",I2,"/",I2,"/",I2,
     1  2X,I2,"Z",3X,"DP = ",F4.0,3X,"EFF = ",F5.0,3X,"PX = ",F5.0
     2  /"<15><12>"," ENERGY UNITS : J/KG X 10",5X,"EL & TROP IN HNDS FT")
      WRITE (20,23)
23    FORMAT ("<15><12>",3X,"STN",3X,"P0",2X,"PMAX",1X,"EL(MB)",1X,"EL(FT)"
     1  ,1X,"TROP",3X,"EI1",3X,"EI2",3X,"LI",3X,"KI",2X,"SWI",2X,"CCL",1X,
     2  "ETCCL",1X,"STN")
      WRITE (20,27)
27    FORMAT ("<15><12>")    ;   BLANK LINE
31    CALL OPEN (22,"STNS1",1,IER)    ;   FILE STNS CONTAINS RAOB ID'S
      IF (IER.NE.1) STOP OPEN ERROR
```

```
5       READ (22,3,END=8) (JST(I),I=1,5)
3       FORMAT (5A2)
        N=N+1
        CALL TYPED(N)    ;   TYPES N ON DASHER TO MONITOR PROGRESS
        IF (N.EQ.20.OR.N.EQ.40.OR.N.EQ.60.OR.N.EQ.80) WRITE (10,48) ; NXT LINE
48      FORMAT (1H )
        IST(N,1)=JST(4)    ;   MAKING STATION ID FOR PLOTTING
        IST(N,2)=JST(5)
54      CALL DECOS (JST,KDATE(2),IHOUR,JNO,P,TS,TSD,IFC,$26)
        IF (ICK.EQ.1) GO TO 5    ;   SGL READ OK
        DO 76 I=0,JNO
        IF (TSD(I).NE.999.) GO TO 76
        IF (P(I).GE.700.) GO TO 7
        TSD(I)=TS(I)-30.   ;  IF DEWPT MISG ABV 700MB, ASSUME DRY
76      CONTINUE
        GO TO 6
7       WRITE (20,10) N,(JST(J),J=4,5),P(I)
10      FORMAT ("<15><12>",I2,1X,2A2," DEWPOINT MISSING AT P = ",F5.0)
        GO TO 26
6       KMOD=0    ;   FOR ETCCL CALCULATION
        CALL CCL1 (IFC,$26)
        CALL MODRB   ;  MODIFIES RAOB FOR ETCCL COMPUTATION
        CALL RANN2 (PT,TST,TSDT,JNOM,PX) ; CALLED FOR ETCCL ONLY, JNOM FM MODRB
        CALL INDX1 (RLI,RKI,RWI,$26,IFC)
        CALL BNDX (IFC,$26)   ;  MODIFY RAOB FOR MAX INSTABILITY
        EFF=0.     ;   ZERO ENTRAINMENT FOR EQUILIBRIUM LVL
        KMOD=2    ;   FOR NORMAL COMPUTATION WITH RANN2
        CALL RANN2 (PT,TST,TSDT,JNOM,PX)    ;   COMPUTE EL LVL, JNOM FROM BNDX
        EL0=EL     ;   SAVE EL WITH ZERO ENTRAINMENT RATE
        EFF=EFF0    ;   RESET ENTRAINMENT RATE FOR ALL OTHER COMPUTATIONS
        JEFF=JREAL(EFF)
        CALL RANN2 (PT,TST,TSDT,JNOM,PX)    ;   COMPUTE STABILITY INDICES
        EL=EL0     ;   USE EL WITH ZERO ENTRAINMENT
C
C   THIS SECTION USES MANDATORY LVLS TO GET "EL" AND "TROP" IN FEET.
        JST(2)=JST(2)-" S"+" M"    ;   MAKING ID FOR MANDATORY LVLS
        JST(3)="AN"
        IDECOM=1    ;   INDICATOR THAT MANDATORY LVLS USED FOR EL AND TROP
        KDATE(2)=KDAT   ;  RESETTING KDATE(2)
        CALL DECOM (JST,KDATE(2),IHOUR,H,T,TD,D,S,$9,IFD,PTROP)   ;  READ MANDATORY LVLS
17      IF (EL.NE.0.) GO TO 14
        EL1=999.    ;   EQUILIBRIUM LEVEL NOT COMPUTED
        GO TO 15
14      CONTINUE
        CALL HEIGHT (H,EL,EL1,$9)   ;  CONVERT EL TO METERS
        EL1=EL1*CNVM    ;   CONVERT M TO FT X 10-2
15      CONTINUE
        IF (PTROP.NE.999.) GO TO 13
        TROP=999.    ;   TROP NOT OBSERVED
        GO TO 16
13      CALL HEIGHT (H,PTROP,TROP,$30)    ;   GET TROP IN METERS
        TROP=TROP*CNVM    ;   CONVERT M TO FT X 10-2
        GO TO 16
30      WRITE (10,33) (JST(I),I=1,5),PTROP
33      FORMAT (1H ,5A2," PTROP = ",F8.0," ERROR RANP, STATEMENT 33")
        TROP=999.
        GO TO 16
C
C   CANNOT USE MANDATORY LVLS (NOT AVBL, TOO LARGE EXTRAPOLATION, ETC.)
9       IDECOM=0      ;   MANDATORY LVLS NOT USED
```

```
        DO 11 I=0,12
11      H(I)=ZZ(I)    ;   SUBSTITUTE U. S. STANDARD ATMOSPHERE
        PTROP=999.
        GO TO 17
C
C   OUTPUT FOLLOWS
16      CONTINUE
        IF (ICK.EQ.2) GO TO 56
        JB(N)=JREAL(B1)
        JEL(N)=JREAL(EL1)
        IF (IDECOM.EQ.1) GO TO 12    ;   MANDATORY LEVELS USED
        WRITE (20,28) N,(JST(I),I=4,5),P(0),PT(0),EL,EL1,TROP,B1,B2,RLI,RKI
     1  ,RWI,PCCL,ETCCL,(JST(I),I=4,5)    ;   NO MANDATORY LVLS USED
28      FORMAT ("<15><12>",I2,1X,2A2,F5.0,3F6.0,"E",F5.0,1X,2F6.0,3F5.0,
     1  2F5.0,1X,2A2)
        GO TO 5
12      WRITE (20,22) N,(JST(I),I=4,5),P(0),PT(0),EL,EL1,TROP,B1,B2,RLI,RKI
     1  ,RWI,PCCL,ETCCL,(JST(I),I=4,5)  ;  MANDATORY LVLS USED..TROP & EL
22      FORMAT ("<15><12>",I2,1X,2A2,F5.0,3F6.0,1X,F5.0,1X,2F6.0,3F5.0,
     1  2F5.0,1X,2A2)
        GO TO 5
26      CONTINUE
        JER=0    ;   INDICATES SINGLE STATION COMPUTATION NOT COMPLETED
        IF (ICK.EQ.2) GO TO 55
C   INSERT DUMMY VALUES FOR STAB INDICES FOR PLOT HERE!
        JB(N)=999
        JEL(N)=999
        GO TO 5
8       WRITE (10,48)    ;   NEXT LINE
        CALL CLOSE (22,IER)
        IF (IER.NE.1) STOP CLOSE ERROR
        GO TO 55
C
C   OUTPUT FOR SINGLE STATION ANALYSIS
56      CALL GCHN (ICHN,IER)
        IF (IER.NE.1) TYPE "GCHN ERROR FOR OVERLAY OV2 & 3, IER = ",IER
        CALL OVOPN (ICHN,"RANP.OL",IER)    ;   OPEN RANP.OL
        IF (IER.NE.1) TYPE "RANP.OL OPENING ERROR, IER = ",IER
        CALL OVLOD (ICHN,OV2,-1,IER)      ;   LOAD OV2
        IF (IER.NE.1) TYPE "OV2 LOADING ERROR, IER = ",IER
        CALL FOPEN (21,"INDEXY",300)
        CALL TPB    ;   COMP. OF POT UNSTBL LYRS AND OUTPUT TO CHANNEL 21
        CALL CLOSE (21,IER)
        CALL OVLOD (ICHN,OV3,-1,IER)      ;   LOAD OV3
        IF (IER.NE.1) TYPE "OV3 LOADING ERROR, IER = ",IER
        CALL TPA(JEFF,EL1,RLI,RKI,RWI,IDECOM)    ;   OUTPUT FOR SNGL STN RAOB ANALYSIS
        CALL KLOSE (ICHN,IER)
        IF (IER.NE.1) TYPE "KLOSE ERROR FOR ICHN, IER = ",IER
C
C
55      CALL CLOSE (20,IER)
        IF (IER.NE.1) TYPE "CHANNEL 20 CLOSE ERROR, IER = ",IER
C
C   INSERT HEADING AND ENDING ON INDEXX
        CALL GCHN (ICHN,IER)    ;   GET RDOS CHANNEL
        CALL OPENN (ICHN,"INDEXX",0,IER)
        CALL WRS (ICHN,IHDR1,22,IER)    ;   HEADER INSERTION
        CALL KLOSE (ICHN,IER)
        CALL GCHN (ICHN,IER)    ;   GET RDOS CHANNEL
        CALL OPENA (ICHN,"INDEXX",0,IER)    ;   OPEN FOR APPENDING
```

```
        CALL WRS (ICHN,IEND,2,IER)    ;   ENDING FOR AFOS PRODUCT
        CALL KLOSE (ICHN,IER)
        CALL FSTORE ("INDEXX",0,IER)    ;   STORE INTO WRKTPA,C, OR D
        GO TO (57,39,66),IM    ;    IM=ICK+1
C
C   INSERT HEADING AND ENDING ON INDEXY
66      IF (JER.EQ.0) GO TO 63
        IHDR1(5)="80"
        CALL GCHN (ICHN,IER)    ;   GET RDOS CHANNEL
        CALL OPENN (ICHN,"INDEXY",0,IER)
        CALL WRS (ICHN,IHDR1,22,IER)    ;   HEADER INSERTION
        CALL KLOSE (ICHN,IER)
        CALL GCHN (ICHN,IER)    ;   GET RDOS CHANNEL
        CALL OPENA (ICHN,"INDEXY",0,IER)    ;   OPEN FOR APPENDING
        CALL WRS (ICHN,IEND,2,IER)    ;   ENDING FOR AFOS PRODUCT
        CALL KLOSE (ICHN,IER)
        CALL FSTORE ("INDEXY",0,IER)    ;   STORE INTO WRKTPB
        CALL FORKP ("RANP","WRKTPA,WRKTPB",IER)
        GO TO 46
C
C   CREATE GRAPHIC EIS
57      CALL GCHN (ICHN,IER)
        IF (IER.NE.1) TYPE "GCHN ERROR FOR OVERLAY OV0 & 1, IER = ",IER
        CALL OVOPN (ICHN,"RANP.OL",IER)    ;   OPEN RANP.OL
        IF (IER.NE.1) TYPE "RANP.OL OPENING ERROR, IER = ",IER
        CALL OVLOD (ICHN,OV0,-1,IER)      ;   LOAD OV0
        IF (IER.NE.1) TYPE "OV0 LOADING ERROR, IER = ",IER
        CALL STLOC(N,IXX,IYY,IST)
        CALL OVLOD (ICHN,OV1,-1,IER)      ;   LOAD OV1
        IF (IER.NE.1) TYPE "OV1 LOADING ERROR, IER = ",IER
        CALL GPT(N,IXX,IYY,JB,JEL,IST,IHOUR,KDATE,JEFF)    ;   CREATE AFOS GRAPHIC
        CALL KLOSE (ICHN,IER)
        IF (IER.NE.1) TYPE "KLOSE ICHN, IER = ",IER
        CALL FORKP ("RANP","WRKTPC & EIS",IER)    ;   TURN ON ALERT LIGHT
        GO TO 46
39      CALL FORKP ("RANP","WRKTPD",IER)
63      DO 68 I=1,1000
68      TIMEWASTE=1./2.    ;   DELAY FOR CALL TO DFILW
46      DO 61 I=1,1000
        CALL DFILW ("INDEXX",IER)    ;   DELETE INDEXX FILE
        IF (IER.EQ.1) GO TO 62
61      CONTINUE
        IF (IER.NE.1) TYPE "INDEXX FILE NOT DELETED, IER = ",IER
62      GO TO (32,38,60),IM    ;    IM=ICK+1
60      IF (JER.EQ.1) GO TO 67
        CALL FORKP ("RANP","WRKTPA",IER)
        STOP
67      DO 64 I=1,1000
        CALL DFILW ("INDEXY",IER)    ;   DELETE INDEXY FILE
        IF (IER.EQ.1) GO TO 65
64      CONTINUE
        IF (IER.NE.1) TYPE "INDEXY FILE NOT DELETED, IER = ",IER
65      STOP
32      DO 37 I=1,1000
        CALL DFILW ("HMSGPH.01",IER)    ;   DELETE GRAPHIC FILE
        IF (IER.EQ.1) GO TO 38    ;   THIS LOOP IS NECESSARY, FOR SLOW CLOSING
37      CONTINUE
        IF (IER.NE.1) WRITE (10,36) IER
36      FORMAT (1H ,"IER = ",I4,"  HMSGPH.01 NOT DELETED - RANP, STATEMENT 36")
38      STOP
```

```
      END
                                    *


                                    *
      SUBROUTINE DECOS (JST,IDATE,IHOUR,JNO,P,TS,TSD,IFC,Q)
C   DECODES RAOB SIGNIFICANT LEVELS UP TO 100MB
C   JST...AFOS IDENTIFIER
C   IDATE/IHOUR IS THE DATE/HOUR WANTED FOR DECODING.
C   IDATE=0 MEANS DATE AND HOUR NOT TO BE TESTED.
C   JNO...NO. OF SIGNIFICANT LEVELS DECODED.
C   P...PRESSURE, TS...TEMPERATURE, TSD...DEWPOINT.
C   IFC...OUTPUT DEVICE, Q...ABNORMAL ERROR RETURN STATEMENT NUMBER.
      DIMENSION JST(5),P(0:50),TS(0:50),TSD(0:50)
      DIMENSION IOUT(40)
      INTEGER Q
      CALL AFREAD (1,JST,$100)
      CALL AFREAD (2,IOUT,$50,$125)
C   TEST FOR NEW RAOB CODE FORMAT
15    IF (IOUT(4).EQ."TT".AND.IOUT(5).EQ."BB") GO TO 9
      GO TO 10   ;   OLD RAOB CODE
9     IF (IOUT(6).EQ." 5".OR.IOUT(6).EQ." 6".OR.IOUT(6).EQ." 7".
     1  OR.IOUT(6).EQ." 8") GO TO 11
      K=-5           ;   DOUBLE SPACE AFTER TTBB
      K1=-3
      K2=-2
      GO TO 12
11    K=-6           ;   SINGLE SPACE AFTER TTBB
      K1=-3
      K2=-2
      GO TO 12
C   OLD RAOB CODE FORMAT
10    K=0
      IF (IOUT(6).EQ." U".AND.IOUT(7).EQ."J1") K=4
      K1=K/2
      K2=K1
      IF (IOUT(9+K1).EQ." 5".OR.IOUT(9+K1).EQ." 6".OR.IOUT(9+K1).EQ." 7".
     1  OR.IOUT(9+K1).EQ." 8") GO TO 12  ;  TESTING FOR DOUBLE SPACE AFTER TTBB
      K=K-1
12    JDATE=ITCVT(18+K,2,$900)-50
      JHOUR=ITCVT(20+K,2,$900)
      IF (IDATE.EQ.0) GO TO 13
      IVCHECK=IVCK(IDATE,IHOUR,JDATE,JHOUR)
      GO TO (13,14,133,135),IVCHECK
14    CALL PRVRF(IER)
      IF (IER.NE.1) GO TO 135
      CALL AFREAD (3,JST,$102)
      CALL AFREAD (2,IOUT,$50,$125)  ;  READ 1ST LINE OF PRVS VERSION
      GO TO 15
C
C   TEST FOR MISSING RAOB 10142 ETC.
13    IIA=ITCVT(30+K,4,$900)
      IIB=ITCVT(34+K,1,$900)
      IIIA=ITCVT(36+K,3,$900)
      IF (IIA.EQ.5151.AND.IIB.EQ.5.AND.IIIA.EQ.101) GO TO 126
C
      JNO=0
      DO 1 I=0,2
      I1=6*I
      I2=I1+I1
```

```
      II=ITCVT(30+K+I2,2,$900)
      IJJ=I*11
      IF (IJJ.NE.II) GO TO 127  ;  TEST FOR IMPROPER FORMAT
      IF (IOUT(17+K1+I1).EQ."//".OR.IOUT(17+K1+I1).EQ."/ ") GO TO 1 ; SIG LVL PRS MISG
      P(JNO)=FTCVT(32+K+I2,3,$901)
      IF (P(JNO).LT.100.) P(JNO)=P(JNO)+1000.
      TS(JNO)=FTCVT(36+K+I2,3,$901)
      IF (IOUT(20+K1+I1).EQ."//".OR.IOUT(20+K2+I1).EQ."/ ") GO TO 2  ;  DEWPT MISG
      TSD(JNO)=FTCVT(39+K+I2,2,$901)
      GO TO 3
2     TSD(JNO)=999.
3     CALL TEMP1(TS(JNO),TSD(JNO))  ;  TEMPERATURE DECODER
      JNO=JNO+1
1     CONTINUE
C   FIRST LINE OF RAOB IS FINISHED HERE
C   STATEMENT 4 STARTS 2ND AND SUBSEQUENT LINES
      JNO=JNO-1
      IJK=22
4     CALL AFREAD (2,IOUT,$50,$125)
      DO 5 I=0,4
      I1=6*I
      I2=I1+I1
      IJK=IJK+11
      IF (IJK.EQ.110) IJK=11
      IJK1=I2+1
      IF (IOUT(I1+2).EQ."//") GO TO 5  ;  SIG LVL PRESSURE MISG
      I515=ITCVT(I2+1,3,$900)
      IF (I515.EQ.515) GO TO 53  ;  TEST FOR 51515 101XX GROUP ENDING MSG
      JI=ITCVT(IJK1,2,$900)
      IF (IJK.NE.JI) GO TO 128  ;  TEST FOR IMPROPER FORMAT
      IF (JNO.EQ.50) GO TO 51
      JNO=JNO+1
      P(JNO)=FTCVT(I2+3,3,$901)
      IF (P(JNO).LT.100.) P(JNO)=P(JNO)+1000.
      TS(JNO)=FTCVT(I2+7,3,$901)
      IO=IOUT(I1+6)
      IF (IO.EQ."/ ".OR.IO.EQ."/=") GO TO 6
      TSD(JNO)=FTCVT(I2+10,2,$901)
      GO TO 7
6     TSD(JNO)=999.
7     CALL TEMP1(TS(JNO),TSD(JNO))
      IF (IO.EQ."/=".OR.IO.EQ."0=".OR.IO.EQ."1=".OR.IO.EQ."2=".OR.IO.EQ.
     1  "3=".OR.IO.EQ."4=".OR.IO.EQ."5=".OR.IO.EQ."6=".OR.IO.EQ."7=".OR.IO.
     2  EQ."8=".OR.IO.EQ."9=") GO TO 53  ;  TEMPERATURES FINISHED
5     CONTINUE
      GO TO 4   ;   RETURNS TO 4 TO DO 3RD AND SUBSEQUENT LINES
53    CONTINUE
C   IF WIND DATA REQUIRED, READ IT HERE...
      RETURN
51    WRITE (IFC,52) (JST(I),I=4,5),P(JNO)
52    FORMAT ("<15><12>",1X,2A2,1X,"51 SIGNIFICANT LEVELS HAVE BEEN DECOD
     1  ED, LEVELS ABOVE ",F5.0,"MB DISREGARDED.")
      RETURN
50    WRITE (IFC,54) (JST(I),I=4,5)
54    FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 50 - DECOS")
      RETURN Q
100   WRITE (IFC,55) (JST(I),I=4,5)
55    FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 100 - DECOS")
      RETURN Q
102   WRITE (IFC,103) (JST(I),I=4,5)
```

```
103     FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 102 - DECOS")
        RETURN Q
125     WRITE (IFC,56) (JST(I),I=4,5)
56      FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 125 - DECOS")
        RETURN Q
126     WRITE (IFC,57) (JST(I),I=4,5)
57      FORMAT ("<15><12>",3X,2A2," STATION MISSING - DECOS")
        RETURN Q
127     WRITE (IFC,129) (JST(I),I=4,5),IJJ,II
129     FORMAT ("<15><12>",3X,2A2," IMPROPER FORMAT (1ST LINE) LOOKING FOR:
     1  ",I3," FOUND: ",I3," DECOS")
        RETURN Q
128     WRITE (IFC,130) (JST(I),I=4,5),IJK,JI
130     FORMAT ("<15><12>",3X,2A2," IMPROPER FORMAT (2ND + LINE) LOOKING FOR:
     1  ",I3," FOUND: ",I3," DECOS")
        RETURN Q
900     WRITE (IFC,131) (JST(I),I=4,5)
131     FORMAT ("<15><12>",3X,2A2," SGL RAOB ERROR - SUBROUTINE ITCVT")
        RETURN Q
901     WRITE (IFC,132) (JST(I),I=4,5)
132     FORMAT ("<15><12>",3X,2A2," SGL RAOB ERROR - SUBROUTINE FTCVT")
        RETURN Q
133     WRITE (IFC,134) (JST(I),I=4,5)
134     FORMAT ("<15><12>",3X,2A2," NEW RAOB NOT AVAILABLE")
        RETURN Q
135     WRITE (IFC,136) (JST(I),I=4,5)
136     FORMAT ("<15><12>",3X,2A2," DESIRED VERSION NOT FOUND - DECOS")
        RETURN Q
        END
                                *


                                *
        SUBROUTINE TEMP1 (T,TD)
C   COMPUTES + OR - TEMPERATURE, AND COMPUTES DEWPOINT
        TT=AMOD(T,2.)
        IF (TT.EQ.1.) T=-T
        T=T*.1
        IF (TD.EQ.999.) RETURN
        IF (TD.LE.50.) GO TO 1
        TD=T-(TD-50.)
        RETURN
1       TD=T-TD*.1
        RETURN
        END
                                *


                                *
C   SUBROUTINE ~RANN2~ COMPUTES ENERGY AREAS ON THERMODYNAMIC DIAGRAM,
C   USING PARCEL METHOD, WITH SELECTED ENTRAINMENT RATE AND PRESSURE STEP
C   JNOJ = NO. OF LEVELS IN RAOB: PA,TSA,TSDA
C   PX = PRESSURE LEVEL ENDING "B1" INDEX COMPUTATION
C   IMPORTANT...KMOD MUST BE PROPERLY SET, BEFORE THIS SUBROUTINE IS CALLED.
C   IF KMOD = -1 BELOW STATEMENT 108, THEN CCL MODIFIED RAOB IS USED.
        SUBROUTINE RANN2 (PA,TSA,TSDA,JNOJ,PX)
        COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
        COMMON/G/PP(0:20),ET(20),TW(0:50),DP,EFF,KMOD,KK
        COMMON/T/RLCL,RLFC,EL,B2,B2P,B2N,IALL,B1,B1P,B1N,EX
        COMMON/CCL/PCCL,ETCCL,TS0,TSD0,L,TSCCL,TCCL,TDCCL,WAVG
```

```
      DIMENSION PA(0:50),TSA(0:50),TSDA(0:50)
      THETA(T,P2,P1)=T*(P2/P1)**.2857142 ; DRY ADIABATIC (T,P1) TO (THETA,P2)
      KDP=0   ;   KDP RESET TO 1, IF 2ND PASS THRU RANN2, WITH REDUCED DP
      DPSAVE=DP   ;   SAVES ORIGINAL DP, PASSED THRU COMMON/G
      R=287.04  ;  GAS CONSTANT FOR DRY AIR .. J/KG PER DEG K
      R=R*.1    ;  SCALING ENERGY UNITS
      EF1=.00002*EFF  ;  ENTRAINMENT FACTOR PER MILLIBAR
      KMOD=KMOD-1   ;    KMOD= -1, WHEN OPERATING ON CCL MODIFIED RAOB
108   IF (KMOD.EQ.-1) GO TO 106
      TS0=TSA(0)
      TSD0=TSDA(0)
106   IF (TS0.NE.TSD0) GO TO 92
      TC=TS0  ;  PARCEL INITIALLY SATURATED
      RLCL=PA(0)
      GO TO 107
92    TC=TCONOF(TS0,TSD0)   ;  CONDENSATION TEMP
      PC=PA(0)*((TC+273.16)/(TS0+273.16))**(1./.2857142) ; COND. PRES.
      IF (KMOD.EQ.-1) PC=PCCL ;  PC COMPUTED ABOVE IS NOT EXACTLY PCCL
107   TH=THETA(TS0+273.16,1000.,PA(0))-273.16  ;  POT. TMP DEG C
      WTH=WOBF(TH)
      WTC=WOBF(TC)
      THW=TH-WTH+WTC  ;  EQUIV WET BULB POT TMP (DEG C)
C   LIFT DRY ADIABATICALLY UNTIL TP=TC AT PRESSURE PC
      DO 7 I=1,20
7     ET(I)=0.
      DT1=0.
      IF (KMOD.EQ.-1) DT1=TS0-TSA(0)
      J=0
      JJ=0
      JK=0
      KJ=0
      EN=0.
      EP=0.
      P1=PA(J)
      PP(0)=PA(0)
      KK=1
      KKK=0
      TP=TSA(J)
      IF (KMOD.EQ.-1) TP=TS0
      IF (TSDA(J).EQ.999.) TSDA(J)=TSA(J)-30. ; IF MISG, ASSUME DRY
      WP=WMROF(P1,TSDA(J))
      IF (IALL.EQ.2) WRITE (10,86)
86    FORMAT (1H ,"P1",8X,"P2",10X,"TE",13X,"TP",13X,"DT1",12X,"DT2",
     1  12X,"E")
      IF (TS0.EQ.TSD0) GO TO 15  ;  PARCEL INITIALLY SATURATED
13    P2=P1-DP
      MJ=0
      IF (PC-P2) 3,4,4
4     P2=PC
      RLCL=PC  ;   LIFTING CONDENSATION LVL
3     IF (PA(J+1)-P2) 5,6,6
6     P2=PA(J+1)
      J=J+1
      MJ=1
      KJ=1
30    PLOG1=ALOG(PA(J)/PA(J+1))
      FACTORT=(TSA(J)-TSA(J+1))/PLOG1
      IF (TSDA(J+1).EQ.999.) TSDA(J+1)=TSA(J+1)-30. ; IF MISG, ASSUME DRY
      FACTORD=(TSDA(J)-TSDA(J+1))/PLOG1
      KJ=1
```

```
5       IF (KJ.EQ.0) GO TO 30  ;  INSURES FACTORT,D COMPUTED 1ST TIME THRU
        IF (JJ.EQ.0) TP0=TP        ;   SAVE ORIGINAL TP
        IF (JJ.EQ.1) TP=TP0  ; RESETS TP TO ORIGINAL VALUE, IF P2 ADJUSTED
        TP=TP+273.16    ;   CONVERT TO DEG K
        TP=THETA (TP,P2,P1)-273.16 ;DRY ADIABATIC LIFT P1 TO P2 DEG C
        PLOG2=ALOG(P2/PA(J+1))
        TE=TSA(J+1)+PLOG2*FACTORT ; ENVIRONMENTAL TEMP AT P2
        DP1=P1-P2
        IF (KMOD.EQ.-1) GO TO 42  ;  NO ENTRAINMENT BELOW CCL LEVEL
        IF (EFF.EQ.0.) GO TO 42  ;   EFF=0. FOR NO ENTRAINMENT
        EF=EF1*DP1
        TP=(TP+273.16+EF*(TE+273.16))/(1.+EF)-273.16  ;  DEG C
        TDE=TSDA(J+1)+PLOG2*FACTORD  ; DEG C
        WE=WMROF(P2,TDE)  ;  G/KG MIXING RATIO OF ENVIRONMENT
        WP=(WP+EF*WE)/(1.+EF)  ; MIXING RATIO OF PARCEL AFTER MIXING
        X=.0200*(TP-12.5+7500./P2)      ;  CORRECTION FOR NON-IDEAL GAS
        WFW=1.+.0000045*P2+.00140*X*X  ;  CORRECTION FOR NON-IDEAL GAS
        E2=WP*.001*P2/((WP*.001+.62197)*WFW) ; VAPOR PRES (MB) OF PARCEL
        ES2=VAPFW(TP)    ;  SATURATION VAPOR PRES OF PARCEL
        ES=ES2-E2
        IF (ES) 40,40,41  ;  GOES TO 40, IF PARCEL SATURATED
41      IF (ES.LE..01) GO TO 40  ;  CLOSE ENOUGH FOR SATURATION
        TDP=DPTOF(E2)    ;  DEWPOINT OF PARCEL AFT MXG (DEG C)
        TC=TCONOF (TP,TDP)
        PC=P2*((TC+273.16)/(TP+273.16))**(1./.2857142)
        GO TO 42
40      TC=TP
        PC=P2
        RLCL=PC
C   SINCE LCL HAS BEEN CHANGED, NEW -THW- IS ALSO REQUIRED
        TH=THETA(TC+273.16,1000.,PC)-273.16  ;  POT TEMP  DEG C
        WTH=WOBF(TH)
        WTC=WOBF(TC)
        THW=TH-WTH+WTC
42      DT2=TP-TE
        IF (JJ.EQ.0) GO TO 96
        TI=DT1-DT2
        JJ=0
        JK=1
        IF (TI) 10,10,11
96      IF (KMOD.NE.-1) GO TO 14
        IF (P2.GT.PC) GO TO 14
        KKK=1
        GO TO 11    ;   LAST STEP IN COMPUTING CCL ENERGY
14      IF (JK.NE.1) GO TO 66 ; JK=1, IF PREVIOUS PASS WAS A CROSSING PT
        JK=0
        IF (DT2) 10,10,11
66      IF (DT2) 8,8,9
8       IF (DT1) 10,10,12
9       IF (DT1) 12,11,11
C   GOES TO 12 IF DRY ADIABAT CROSSES ENVIRONMENTAL TEMP
12      P2=P1-ABS(DT1)/(ABS(DT1)+ABS(DT2))*DP1 ; APPROX PRES WHERE DT2=0.
        IF (KK.LE.20) GO TO 75
        TYPE "ET DIMENSION EXCEEDS 20"
        GO TO 110
75      KKK=1
        JJ=1
        IF (MJ.EQ.0) GO TO 5
C   MJ=1 MEANS J, WHICH HAS JUST BEEN SET AT STATEMENT 6, MUST BE RESET
C    TO INTERPOLATE PROPERLY
```

```
      J=J-1
      MJ=0
      GO TO 30
10    E=.5*(DT2+DT1)*ALOG(P1/P2)
      EN=EN+E
      IF (IALL.EQ.2) WRITE (10,85) P1,P2,TE,TP,DT1,DT2,E
85    FORMAT (1H ,2F10.3,5E15.6)
      P1=P2
      DT1=DT2
      IF (KKK.EQ.0.AND.P2.NE.PA(JNOJ)) GO TO 62
      ET(KK)=EN*R    ;   CONVERTS TO J/KG UNITS
      PP(KK)=P2
      KK=KK+1
      EN=0.
      KKK=0
62    IF (P2.EQ.PC) GO TO 15  ;  PARCEL SATURATED
      GO TO 13
11    E=.5*(DT2+DT1)*ALOG(P1/P2)
      EP=EP+E
      IF (IALL.EQ.2) WRITE (10,85) P1,P2,TE,TP,DT1,DT2,E
      P1=P2
      DT1=DT2
      IF (KKK.EQ.0.AND.P2.NE.PA(JNOJ)) GO TO 63
      ET(KK)=EP*R    ;   CONVERTS TO J/KG UNITS
      PP(KK)=P2
      KK=KK+1
      EP=0.
      KKK=0
63    IF (P2.EQ.PC) GO TO 15  ;  PARCEL SATURATED
      GO TO 13
C
C   LIFT PARCEL ALONG SATURATION ADIABATIC
C
15    CONTINUE
      IF (KMOD.NE.-1) GO TO 84
      DT1=0.
      ETCCL=ET(1)
      RETURN    ;   REMOVE, IF FULL COMPUTATION OF CCL MODIFIED SOUNDING IS DESIRED
      KK=1  ;  KK SET FROM 2 BACK TO 1, CCL ENERGY HAS JUST BEEN COMPUTED
84    JJ=0
      JK=0
      ISTOP=0
      KKK=0
24    P2=P1-DP
      MJ=0
      IF (PA(J+1)-P2) 16,17,17
17    P2=PA(J+1)
      IF (PA(J+1).GT.PA(JNOJ)) GO TO 25
      ISTOP=1
      GO TO 16
25    J=J+1
      MJ=1
88    PLOG1=ALOG(PA(J)/PA(J+1))
      FACTORT=(TSA(J)-TSA(J+1))/PLOG1
      IF (TSDA(J+1).EQ.999.) TSDA(J+1)=TSA(J+1)-30. ; IF MISG, ASSUME DRY
      FACTORD=(TSDA(J)-TSDA(J+1))/PLOG1
      KJ=1
16    IF (KJ.EQ.0) GO TO 88
      IF (JJ.EQ.0) THW0=THW        ; SAVE ORIGINAL THW
      IF (JJ.EQ.1) THW=THW0 ;RESETS THW TO ORIGINAL VALUE, IF P2 ADJUSTED
```

```
        TP=SATLFT (THW,P2)  ;  TEMP OF PARCEL AT P2 ON ~THW~ WET ADIABAT
        PLOG2=ALOG(P2/PA(J+1))
        TE=TSA(J+1)+PLOG2*FACTORT  ;  ENVIRONMENTAL TEMP AT P2
        DP1=P1-P2
        IF (EFF.EQ.0.) GO TO 67  ;  EFF=0. FOR NO ENTRAINMENT
        TDE=TSDA(J+1)+PLOG2*FACTORD ; ENVIRONMENTAL DEWPT AT P2
        WE=WMROF (P2,TDE)  ;  MIXING RATIO (G/KG) OF ENVIRONMENT
        WP=WMROF (P2,TP)  ;  MIXING RATIO OF SATURATED PARCEL
        EF=EF1*DP1
        WP=(WP+EF*WE)/(1.+EF)
        TP=(TP+273.16+EF*(TE+273.16))/(1.+EF)-273.16
        X=.0200*(TP-12.5+7500./P2)      ;  CORRECTION FOR NON-IDEAL GAS
        WFW=1.+.0000045*P2+.00140*X**X  ;  CORRECTION FOR NON-IDEAL GAS
        E2=WP*.001*P2/((WP*.001+.62197)*WFW) ; VAPOR PRES (MB) OF PARCEL
        TDP=DPTOF(E2)  ;  DEWPT OF PARCEL AFT MXG
        IF (TDP.GT.TP) TDP=TP
        TC=TCONOF(TP,TDP)
        TH=THETA(TP+273.16,1000.,P2)-273.16 ;  POT TEMP DEG C
        WTH=WOBF(TH)
        WTC=WOBF(TC)
        THW=TH-WTH+WTC  ;  EQUIV WET BULB POT TEMP (DEG C)
        TP=SATLFT(THW,P2)  ;  PARCEL TEMP AFT EVAPORATING LIQUID WATER
67      DT2=TP-TE
C   IF ADDITIONAL INFORMATION ON LEVELS IS NEEDED,  INSERT PRINT STATEMENT HERE
        IF (JJ.EQ.0) GO TO 23  ;  JJ=1 IF NEW P2 HAS BEEN COMPUTED FOR CROSSOVER.
        TI=DT1-DT2
        JJ=0
        JK=1
        IF (TI) 20,20,22
23      IF (JK.NE.1) GO TO 65 ; JK=1, IF PREVIOUS PASS WAS A CROSSING PT
        JK=0
C
C   IN CASE SAT. ADIABAT INTERSECTS ENVIRONMENTAL TEMP IN 2 PLACES CREATING
C   A VERY SMALL POSITIVE AREA, THIS AREA WILL BE IGNORED (STATEMENT 101).
        CHECK=DT2*ET(KK-1)  ;  USUALLY NEGATIVE
        IF (CHECK.LT.0.) GO TO 100
        IF (DT2) 101,102,102
101     EN=ET(KK-1)
        KK=KK-1
        TYPE "STATEMENT 101 USED IN RANN2"
        GO TO 20
C   IF STATEMENT 102 IS USED, PRESSURE STEP IS REDUCED AND ENTIRE COMPUTATION
C   IS REPEATED.  THIS OCCURS WHEN DT2 CHANGES SIGN SEVERAL TIMES IN A SHORT
C   PRESSURE DISTANCE.  THIS SHOULD BE A VERY RARE OCCURENCE
102     DP=10.   ;   REDUCE PRESSURE STEP TO 10MB.
        IF (KDP.EQ.0) GO TO 109
        GO TO 110   ;   RANN2 CANNOT BE COMPLETED WITH REDUCED PRESSURE STEP
109     KDP=KDP+1
        TYPE "STATEMENT 102 USED IN RANN2"
        GO TO 108   ;   REPEAT ENTIRE ENERGY CALCULATION WITH 10MB PRES STEP
C
100     IF (DT2) 20,20,22
65      IF (DT2) 18,18,19
18      IF (DT1) 20,20,21
19      IF (DT1) 21,22,22
C   GOES TO 21 IF WET ADIABAT CROSSES ENVIRONMENTAL TEMP
21      P2=P1-ABS(DT1)/(ABS(DT1)+ABS(DT2))*DP1
        IF (KK.LE.20) GO TO 76
        TYPE "ET DIMENSION EXCEEDS 20"
        GO TO 110
```

```
76      KKK=1
        JJ=1
        IF (MJ.EQ.0) GO TO 16
C   MJ=1 MEANS J, WHICH HAS JUST BEEN SET AT STATEMENT 25, MUST BE RESET
C    TO INTERPOLATE PROPERLY
        J=J-1
        MJ=0
        GO TO 88
20      E=.5*(DT2+DT1)*ALOG(P1/P2)
        EN=EN+E
        IF (IALL.EQ.2) WRITE (10,85) P1,P2,TE,TP,DT1,DT2,E
        P1=P2
        DT1=DT2
        IF (P2.NE.PX) GO TO 99
        EX=EN*R  ;  SUBTOTAL FOR ENERGY AREA ENDING AT PX
        KX=KK
99      IF (KKK.EQ.0.AND.P2.NE.PA(JNOJ)) GO TO 60
        ET(KK)=EN*R   ;   CONVERTING TO J/KG UNITS
        PP(KK)=P2
        KK=KK+1
        EN=0.
        KKK=0
60      IF (ISTOP.EQ.1.AND.P1.EQ.PA(JNOJ)) GO TO 26
        GO TO 24
22      E=.5*(DT2+DT1)*ALOG(P1/P2)
        EP=EP+E
        IF (IALL.EQ.2) WRITE (10,85) P1,P2,TE,TP,DT1,DT2,E
        P1=P2
        DT1=DT2
        IF (P2.NE.PX) GO TO 104
        EX=EP*R  ;  SUBTOTAL FOR ENERGY AREA ENDING AT PX
        KX=KK
104     IF (KKK.EQ.0.AND.P2.NE.PA(JNOJ)) GO TO 61
        ET(KK)=EP*R   ;   CONVERTING TO J/KG UNITS
        PP(KK)=P2
        KK=KK+1
        EP=0.
        KKK=0
61      IF (ISTOP.EQ.1.AND.P1.EQ.PA(JNOJ)) GO TO 26
        GO TO 24
26      CONTINUE
C
C   KK = NUMBER OF ENERGY AREAS IN SOUNDING + 1
        KK1=KK-1
        KK2=KK-2
        KK3=KK-3
        EL=0.
        B2=999.
        B2P=999.
        B2N=999.  ;   999 DENOTES THAT VARIABLE IS UNDEFINED
        RLFC=0.
C   DETERMINE "LFC" LEVEL
        IF (KK.EQ.2.AND.ET(1).GT.0.) RLFC=PP(0)
        IF (KK.EQ.3.AND.ET(1).LT.0.) RLFC=PP(1)
        IF (KK.GE.4.AND.ET(1).LT.0.) RLFC=PP(1)
        IF (KK.GE.4.AND.ET(1).GT.0.) RLFC=PP(2)
C   IN ALL OTHER CASES RLFC IS UNDEFINED...RLFC=0.
C
        IF (ET(KK1).GT.0.) GO TO 70  ;  HIGHEST AREA IS +, NO INDICES COMPUTED
C   "EL" LEVEL DETERMINED HERE
```

```
      EL=PP(KK2)
C   COMPUTE ENERGY INDICES BELOW EL LEVEL
      IF (KK.EQ.2) GO TO 70  ;  ONLY ONE LAYER, ALL NEGATIVE
      B2=0.
      B2P=0.
      B2N=0.
      IF (ET(1).LT.0.) GO TO 58
      DO 74 I=1,KK2,2
74    B2P=B2P+ET(I)
      IF (KK.GT.3) GO TO 68
      B2N=0.
      GO TO 69
68    DO 73 I=2,KK3,2
73    B2N=B2N+ET(I)
      GO TO 69
58    DO 91 I=1,KK3,2
91    B2N=B2N+ET(I)
      DO 103 I=2,KK2,2
103   B2P=B2P+ET(I)
69    B2=B2P+B2N
70    CONTINUE
C
C   COMPUTE B1 INDEX (ENERGY AREAS ENDING AT PX)
      KX1=KX-1
      B1=0.
      B1P=0.
      B1N=0.
      DO 105 I=1,KX1
      IF (ET(I).LT.0.) B1N=B1N+ET(I)
      IF (ET(I).GT.0.) B1P=B1P+ET(I)
105   CONTINUE
      IF (EX.LT.0.) B1N=B1N+EX
      IF (EX.GT.0.) B1P=B1P+EX
      B1=B1P+B1N
      RETURN
C   GOES TO 110, IF RANN2 CANNOT BE COMPLETED DUE TO MANY SIGN CHANGES OF DT2
C   OVER A SMALL PRESSURE INTERVAL, OR TOO MANY ENERGY AREAS (KK.GT.20).
110   EL=0.
      B1=999.
      B1P=999.
      B1N=999.
      B2=999.
      B2P=999.
      B2N=999.
      WRITE (10,111) (JST(I),I=4,5)
111   FORMAT (1H ,2A2," RANN2 SUBROUTINE DID NOT COMPLETE.")
      DP=DPSAVE   ;   RESTORE DP TO ORIGINAL VALUE, IF IT WAS CHANGED.
      RETURN
      END
                                *


                                *
      SUBROUTINE CCL1 (IFC,Q)
C   COMPUTES CCL AND CONVECTIVE TEMPERATURE
      COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
      COMMON/CCL/PCCL,ETCCL,TS0,TSD0,L,TSCCL,TCCL,TDCCL,WAVG
      INTEGER Q
      THETA(T,P2,P1)=T*(P2/P1)**.2857142 ; DRY ADIABATIC (T,P1) TO (THETA,P2)
      DP1=100.    ;   AVERAGES MIXING RATIO OVER FIRST ~DP1~ MBS.
```

```
      WSUM=0.
      J=0
      P1=P(0)
      TDE1=TSD(0)
      W1=WMROF(P(0),TDE1)
      PFINISH=P1-DP1
3     P2=PFINISH
      IF (P(J+1)-P2) 1,2,2
2     P2=P(J+1)
      TDE2=TSD(J+1)    ;   ENVIRONMENT DEWPT AT P2
      J=J+1
      GO TO 9
1     PLOG1=ALOG(P(J)/P(J+1))
      FACTORD=(TSD(J)-TSD(J+1))/PLOG1
      PLOG2=ALOG(P2/P(J+1))
      TDE2=TSD(J+1)+FACTORD*PLOG2 ;  ENVIRONMENT DEWPT AT P2
9     W2=WMROF(P2,TDE2)              ;  MIXING RATIO AT P2
      PLOG3=ALOG(P1/P2)
      W=.5*(W1+W2)*PLOG3             ;  AVG MIX RATIO IN LYR P1-P2
      WSUM=WSUM+W
      P1=P2
      W1=W2
      IF (P2.GT.PFINISH) GO TO 3
C   COMPUTE AVG VALUES FOR FIRST ~DP1~ MBS.
      PLOG4=ALOG(P(0)/PFINISH)
      WAVG=WSUM/PLOG4
C   DETERMINE LAYER CONTAINING CCL, CHECKING FROM TOP OF ATMOS DOWNWARD
      DO 4 I=0,JNO
      II=JNO-I
      WS=WMROF(P(II),TS(II))
      IF (WS-WAVG) 4,5,6
4     CONTINUE
      WRITE (10,8) (JST(I),I=4,5)
8     FORMAT (1H ,2A2," ERROR IN CCL1")
      STOP
5     PCCL=P(II)
      TCCL=TS(II)
      TDCCL=TSD(II)
      TSCCL=THETA(TCCL+273.16,P(0),PCCL)-273.16  ;  CONVECTIVE TEMP DEG C
      L=JNO+1
      JJNO=JNO-1
      RETURN   ;   CCL LEVEL IS ALSO A RAOB SIGNIFICANT LEVEL
6     J=II+1
      JJNO=JNO
      IF (J.EQ.(JNO+1)) GO TO 32   ;   SHORT RAOB
C   MXG RATIO INTERSECTS ENVIRONMENTAL TEMP BTWN P(J) AND P(J-1)
C   THIS LAYER WILL BE SUBDIVIDED UNTIL SATURATION VAPOR PRESSURE AT
C   MIDPOINT OF LAYER IS SUFFICIENTLY CLOSE (.01 G/KG) TO WAVG.
C   THIS DETERMINES THE CCL LEVEL.
      P1=P(J-1)    ;   BOTTOM
      P2=P(J)      ;   TOP
      T1=TS(J-1)   ;   BOTTOM
      T2=TS(J)     ;   TOP
31    ALOG1=ALOG(P1/P2)
      PM=.5*(P1+P2)    ;   MIDPOINT PRESSURE
      ALOG2=ALOG(PM/P2)
      TPM=T2+(T1-T2)/ALOG1*ALOG2  ;  MIDPOINT TEMPERATURE
      WSM=WMROF(PM,TPM)    ;   MIDPOINT SATURATION MIXING RATIO
      IF(ABS(WSM-WAVG).LE..01)  GO TO 29   ;   TEST FOR TOLERANCE
      IF (WSM-WAVG) 28,29,30
```

```
28      P2=PM
        T2=TPM
        GO TO 31
30      P1=PM
        T1=TPM
        GO TO 31
29      PCCL=PM       ;   CCL PRESSURE
        TCCL=TPM      ;   CCL TEMPERATURE
C   COMPUTE DEWPOINT AT CCL LEVEL
        ALOG1=ALOG(P(J-1)/P(J))
        ALOG2=ALOG(PM/P(J))
        TDCCL=TSD(J)+(TSD(J-1)-TSD(J))/ALOG1*ALOG2
        IF (TDCCL.GT.TCCL) TDCCL=TCCL  ;  CORRECTION FOR DEWPOINT EXCEEDING
     1  TEMPERATURE BY SMALL AMT
        TSCCL=THETA(TCCL+273.16,P(0),PCCL)-273.16  ;  CONVECTIVE TEMP DEG C
        L=J             ;     INDEX NUMBER OF ADDED CCL LEVEL
        RETURN
32      WRITE (IFC,33) (JST(I),I=4,5),P(JNO)
33      FORMAT ("<15><12>",3X,2A2,"RAOB TERMINATES TOO SOON, P(JNO) = ",
     1  F5.0," CCL1")
        RETURN Q
        END
                                *


                                *
C   THIS SUBROUTINE TO BE CALLED AFTER ~CCL1~ IS CALLED
        SUBROUTINE MODRB
        COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
        COMMON/CCL/PCCL,ETCCL,TS0,TSD0,L,TSCCL,TCCL,TDCCL,WAVG
        COMMON/TT/PT(0:50),TST(0:50),TSDT(0:50)
        COMMON/V/JNOM,PX
C   MODIFY ORIGINAL RAOB FOR SOLAR HEATING BLO CCL
        TS0=TSCCL                  ;  SFC TEMP RESET
        IF (L.EQ.(JNO+1)) GO TO 3  ;  CCL LEVEL IS A RAOB SGFNT LEVEL
C   MOVE SGFNT LVLS ABV PCCL UP ONE LVL
        I=JNO
1       TST(I+1)=TS(I)
        TSDT(I+1)=TSD(I)
        PT(I+1)=P(I)
        I=I-1
        IF (I.GE.L) GO TO 1
C   ONE ADDITIONAL LVL ADDED AT PCCL
        TST(L)=TCCL
        TSDT(L)=TDCCL
        PT(L)=PCCL
C   COMPLETE THE RAOB BELOW THE CCL LEVEL
3       LL=L-1
        DO 2 I=0,LL
        TST(I)=TS(I)
        TSDT(I)=TSD(I)
2       PT(I)=P(I)
C   MODIFY TSD(0) TO CONFORM TO WAVG, AVG MIXING RATIO IN LOWEST 100 MBS
        X=.0200*(TSDT(0)-12.5+7500./PT(0))  ; NON-IDEAL GAS CORRECTION
        WFW=1.+.0000045*PT(0)+.00140*X*X    ; NON-IDEAL GAS CORRECTION
        E2=:001*WAVG*PT(0)/((WAVG*.001+.62197)*WFW)  ;  VAPOR PRESSURE
        TSD0=DPTOF(E2)
        JNOM=JJNO+1
        RETURN
        END
```

```
*
                                        *
      SUBROUTINE INDX1 (RLI,RKI,RWI,Q,IFC)
C   COMPUTES LIFTED INDEX, K INDEX, AND SHOWALTER INDEX
C   IF SFC PRES LESS THAN 850MB, K AND SHOWALTER SET = 999.
C   IFC DENOTES OUTPUT DEVICE FOR ERROR MSG FM THIS SUBROUTINE
      COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
      DIMENSION PL(3),TL(3),TDL(3)
      INTEGER Q
      THETA(T,P2,P1)=T*(P2/P1)**.2857142 ; DRY ADIABATIC (T,P1) TO (THETA,P2)
      IHI=0   ;   INDICATOR FOR SFC PRESSURE GREATER THAN 850MB.
      PL(1)=850.
      PL(2)=700.
      PL(3)=500.
      DP1=50.    ;   AVERAGES OVER FIRST ~DP1~ MBS.
      WSUM=0.
      THSUM=0.
      J=0
      P1=P(0)
      TE1=TS(0)
      TDE1=TSD(0)
      TH1=THETA(TE1+273.16,1000.,P(0))    ;   POT TEMP
      W1=WMROF(P(0),TDE1)
      PFINISH=P1-DP1
3     P2=PFINISH
      IF (P(J+1)-P2) 1,2,2
2     P2=P(J+1)
      J=J+1
1     PLOG1=ALOG(P(J)/P(J+1))
      FACTORT=(TS(J)-TS(J+1))/PLOG1
      FACTORD=(TSD(J)-TSD(J+1))/PLOG1
      PLOG2=ALOG(P2/P(J+1))
      TE2=TS(J+1)+FACTORT*PLOG2    ;  ENVIRONMENT TEMP AT P2
      TDE2=TSD(J+1)+FACTORD*PLOG2 ;  ENVIRONMENT DEWPT AT P2
      TH2=THETA(TE2+273.16,1000.,P2) ;  POT TEMP AT TE2,P2
      W2=WMROF(P2,TDE2)              ;  MIXING RATIO AT P2
      PLOG3=ALOG(P1/P2)
      TH=.5*(TH1+TH2)*PLOG3          ;  AVG POT TEMP IN LYR P1-P2
      W=.5*(W1+W2)*PLOG3             ;  AVG MIX RATIO IN LYR P1-P2
      THSUM=THSUM+TH
      WSUM=WSUM+W
      P1=P2
      TH1=TH2
      W1=W2
      IF (P2.GT.PFINISH) GO TO 3
C   COMPUTE AVG VALUES FOR FIRST ~DP1~ MBS.
      PLOG4=ALOG(P(0)/PFINISH)
      THAVG=THSUM/PLOG4
      WAVG=WSUM/PLOG4
      PPARCEL=P(0)-.5*DP1
      TPARCEL=THETA(THAVG,PPARCEL,1000.)-273.16  ;  DEG C
      X=.0200*(TPARCEL-12.5+7500./PPARCEL) ;  NON-IDEAL GAS CORRECTION
      WFW=1.+.0000045*PPARCEL+.00140*X*X   ;  NON-IDEAL GAS CORRECTION
      E2=.001*WAVG*PPARCEL/((WAVG*.001+.62197)*WFW)  ;  VAPOR PRES (MB)
      TDPARCEL=DPTOF(E2)
      TC=TCONOF(TPARCEL,TDPARCEL)
      TH=THAVG-273.16  ;  POT TEMP DEG  C
      WTH=WOBF(TH)
```

```
      WTC=WOBF(TC)
      THW=TH-WTH+WTC  ;  EQUIV WET BULB POT TEMP (DEG C)
      TP500=SATLFT(THW,500.)
C   GET TEMP AND DEWPT AT 850,700,500 MBS
      DO 5 J=1,3
      DO 4 I=0,JNO
      IF (PL(J)-P(I)) 4,6,7
4     CONTINUE
      WRITE (IFC,10) (JST(I),I=4,5),P(JNO)
10    FORMAT ("<15><12>",3X,2A2," RAOB TERMINATES TOO SOON, P(JNO) = ",F5.0)
      RETURN Q
6     TL(J)=TS(I)
      TDL(J)=TSD(I)
      GO TO 5
7     IF (J.NE.1) GO TO 8
      IF (I.NE.0) GO TO 8
      IHI=1
      GO TO 5   ;   SFC PRESSURE LESS THAN 850MB
8     FACTOR=ALOG(PL(J)/P(I))/ALOG(P(I-1)/P(I))
      TL(J)=TS(I)+FACTOR*(TS(I-1)-TS(I))
      TDL(J)=TSD(I)+FACTOR*(TSD(I-1)-TSD(I))
5     CONTINUE
      RLI=TL(3)-TP500                          ;  LIFTED INDEX
      IF (IHI.EQ.0) GO TO 9   ;   COMPUTE K AND SHOWALTER INDICES
      RKI=999.    ;    K INDEX MISG
      RWI=999.    ;    SHOWALTER INDEX MISG
      RETURN
9     RKI=(TL(1)-TL(3))+TDL(1)-(TL(2)-TDL(2))  ;  K INDEX
C   COMPUTE SHOWALTER INDEX
      TC=TCONOF(TL(1),TDL(1))
      TH=THETA(TL(1)+273.16,1000.,850.)-273.16  ;  DEG C
      WTH=WOBF(TH)
      WTC=WOBF(TC)
      THW=TH-WTH+WTC  ;  EQUIV WET BULB POT TEMP
      TP=SATLFT(THW,500.)
      RWI=TL(3)-TP     ;   SHOWALTER INDEX
      RETURN
      END
                              *


                              *
      SUBROUTINE BNDX (IFC,Q)
C   DETERMINES LEVEL OF MAXIMUM INSTABILITY IN LOWER 150MBS OF RAOB,
C   ADJUSTS ORIGINAL RAOB, SO LEVEL OF MAX INSTABILITY IS FIRST SGFNT
C   LEVEL AND ADDS ADDITIONAL PRES LEVEL PX, IF PX IS NOT A SGFNT LEVEL.
C   IF RAOB TERMINATES BELOW PX, IT IS EXTRAPOLATED TO PX, IF TOP LEVEL IS WITHIN 50 MBS
C   IFC DENOTES OUTPUT DEVICE FOR ERROR MSG FM THIS SUBROUTINE.
      COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
      COMMON/TT/PT(0:50),TST(0:50),TSDT(0:50)
      COMMON/V/JNOM,PX
      DIMENSION PB(2),TB(2),TDB(2)
      INTEGER Q
      THETA(T,P2,P1)=T*(P2/P1)**.2857142  ; DRY ADIABATIC (T,P1) TO (THETA,P2)
      DP2=150.
C   GET TEMP AND DEWPT AT P(0)-DP2 AND PX
      PB(1)=P(0)-DP2
      PB(2)=PX
      DO 5 J=1,2
      DO 4 I=0,JNO
```

```
      IF (PB(J)-P(I)) 4,6,7
4     CONTINUE
      I=I-1
      IF (J.EQ.2.AND.(P(JNO)-PB(2)).LT.50.) GO TO 7 ; EXTRAPOLATES, IF WITHIN 50MBS
      WRITE (IFC,3) (JST(I),I=4,5),P(JNO)
3     FORMAT ("<15><12>",3X,2A2," RAOB TERMINATES TOO SOON, P(JNO) = ",F5.0," BNDX")
      RETURN Q
6     TB(J)=TS(I)
      TDB(J)=TSD(I)
      GO TO 5
7     FACTOR=ALOG(PB(J)/P(I))/ALOG(P(I-1)/P(I))
      TB(J)=TS(I)+FACTOR*(TS(I-1)-TS(I))
      TDB(J)=TSD(I)+FACTOR*(TSD(I-1)-TSD(I))
5     CONTINUE
C   FIND LARGEST POTENTIAL WET BULB TEMPERATURE IN FIRST DP2 MBS
      THWMAX=-1000.
      II=0
      DO 1 I=0,JNO
      IF (P(I)-PB(1)) 8,10,10
10    TC=TCONOF (TS(I),TSD(I))
      TH=THETA(TS(I)+273.16,1000.,P(I))-273.16 ; DEG C
      WTH=WOBF(TH)
      WTC=WOBF(TC)
      THW=TH-WTH+WTC  ;  WET BULB POTENTIAL TEMPERATURE
      IF (THW-THWMAX) 1,1,2
2     THWMAX=THW
      II=I
      PMAX=P(I)
      TMAX=TS(I)
      TDMAX=TSD(I)
1     CONTINUE
8     IF (P(I-1).EQ.PB(1)) GO TO 9
      TC=TCONOF (TB(1),TDB(1))
      TH=THETA(TB(1)+273.16,1000.,PB(1))-273.16  ;  DEG C
      WTH=WOBF(TH)
      WTC=WOBF(TC)
      THW=TH-WTH+WTC  ;  WET BULB POT TEMP
      IF (THW-THWMAX) 9,9,12
12    THWMAX=THW
      II=I-1
      PMAX=PB(1)
      TMAX=TB(1)
      TDMAX=TDB(1)
9     CONTINUE
C   MODIFY RAOB SO LOWEST LEVEL HAS MAXIMUM WET BULB POTENTIAL TEMPERATURE
      PT(0)=PMAX
      TST(0)=TMAX
      TSDT(0)=TDMAX
      JNOO=JNO-II
      DO 11 J=1,JNOO
      PT(J)=P(J+II)
      TST(J)=TS(J+II)
      TSDT(J)=TSD(J+II)
11    CONTINUE
      DO 14 J=1,JNOO
      IF (PB(2)-PT(J)) 14,17,16
14    CONTINUE
      J=JNOO+1
      GO TO 20  ;  EXTRAPOLATE RAOB
16    I=JNOO
```

```
18      TST(I+1)=TST(I)  ;  MOVE ALL LEVELS ABOVE PB(2) UP 1 LEVEL
        TSDT(I+1)=TSDT(I)
        PT(I+1)=PT(I)
        I=I-1
        IF (I.GE.J) GO TO 18  ;  J SET IN DO 14 LOOP
20      TST(J)=TB(2)  ;   ADD TB(2) LEVEL
        TSDT(J)=TDB(2)
        PT(J)=PB(2)
        JNOM=JNOO+1
        GO TO 19
17      JNOM=JNOO
19      CONTINUE
        RETURN
        END
                                *


                                *
        SUBROUTINE DECOM (IST,IDATE,IHOUR,Z,T,TD,D,S,Q,IFD,PTROP)
C   DECODES MANDATORY LVL RAOB DATA UP TO AND INCLUDING MAX WND (77/66 GRP)
C   SUBROUTINE SEARCHES FOR SPECIFIED DATE (IDATE) & HOUR (IHOUR)
C   IST...AFOS IDENTIFER, IFD...OUTPUT DEVICE
C   Q...ABNORMAL ERROR RETURN STATEMENT NUMBER
C   Z...(0 = SFC PRES), (1,2,...10 = HGTS), (11 = TROP DATA), (12 = MAX WND)
C   T...TEMPERATURE, TD...DEWPOINT, D...WND DIR, S...WND SPEED
C   T(12) = LOWER WIND SHEAR, TD(12) = UPPER WIND SHEAR
C    PTROP = Z(11) IS TROP PRESSURE FROM "88" GROUP, THIS REDUNDANCY FOR
C    BENEFIT OF RANP PROGRAM!
C
C   MISSING DATA INDICATED AS FOLLOWS:
C   HEIGHT      Z(1,...,10)        ~ -999.
C   PRESSURE    Z(11,12)           ~  999.
C   TEMPERATURE T & TD(1,...,11)   ~  999.
C   WND SHEAR   T & TD(12)         ~  999.
C   WND         D & S(0,...,12)    ~  -99.
C
        INTEGER Q
        DIMENSION IST(5),Z(0:12),T(0:12),TD(0:12),D(0:12),S(0:12),IOUT(40)
        COMMON/SI/IS(0:12)
        DATA IS/99,00,85,70,50,40,30,25,20,15,10,88,77/
        KS=0   ;   INDICATOR FOR LAST LEVEL OF WIND DATA REPORTED
        CALL AFREAD (1,IST,$100)
        CALL AFREAD (2,IOUT,$50,$125)   ;   READ 1ST LINE
7       LC=1    ;    LINE COUNTER
        IF (IOUT(4).EQ."TT".AND.IOUT(5).EQ."AA") GO TO 3  ;  NEW RAOB FORMAT
        GO TO 4   ;   OLD RAOB FORMAT
C
C   NEW RAOB FORMAT
3       IF (IOUT(6).EQ." 5".OR.IOUT(6).EQ." 6".OR.IOUT(6).EQ." 7".OR.
     1  IOUT(6).EQ." 8") GO TO 1   ;   TESTING FOR SINGLE SPACE AFT TTAA
        K=-5        ;    DOUBLE SPACE AFTER TTAA
        K1=-3
        K2=-2
        GO TO 2
1       K=-6        ;    SINGLE SPACE AFTER TTAA
        K1=-3
        K2=-2
        GO TO 2
C
C   OLD RAOB FORMAT
```

```
4       K=0
        IF (IOUT(6).EQ." U".AND.IOUT(7).EQ."S1") K=4
        K1=K/2
        K2=0
        IF (IOUT(9+K1).EQ." 5".OR.IOUT(9+K1).EQ." 6".OR.IOUT(9+K1).EQ." 7".
     1  OR.IOUT(9+K1).EQ." 8") GO TO 2  ;  TESTING FOR SINGLE SPACE AFT TTAA
        K=K-1
2       IDT=ITCVT (18+K,2,$900)-50    ;   DATE
        IHR=ITCVT (20+K,2,$900)       ;   HOUR
        IVCHECK=IVCK(IDATE,IHOUR,IDT,IHR)   ;   TESTING IDT & IHR FOR COR VERSION
        GO TO (5,10,116,114),IVCHECK
10      CALL PRVRF (IER)
        IF (IER.NE.1) GO TO 114
        CALL AFREAD (3,IST,$102)
        CALL AFREAD (2,IOUT,$50,$125)   ;   READ 1ST LINE OF PREVIOUS VERSION
        GO TO 7
C   TEST FOR MISSING RAOB 10142 ETC.
5       IIA=ITCVT (30+K,4,$900)
        IIB=ITCVT (34+K,1,$900)
        IIIA=ITCVT (36+K,3,$900)
        IF (IIA.EQ.5151.AND.IIB.EQ.5.AND.IIIA.EQ.101) GO TO 110
        KKK=IOUT(11+K1)     ;   INDICATOR FOR LAST LEVEL OF WND DATA
        KKK2=IOUT(11+K2)    ;   2ND INDICATOR FOR LAST LEVEL OF WND DATA
        KSS=1     ;    SOME WIND DATA AVAILABLE
        IF (KKK.EQ."0/".OR.KKK.EQ."2/".OR.KKK2.EQ."/ ") KSS=0 ; NO WND DATA AVBL
        IF (KSS.EQ.1) KS=ITCVT(22+K,1,$900) ; READ INDICATOR FOR LAST LVL OF WND
C
C   BEGIN READING SFC PRES GRP
        K=29+K  ;   SET CHARACTER INDEX
        JC=-1     ;   SET LEVEL INDEX
        KC=4    ;   SET GROUP INDEX
        LC=1    ;   LINE COUNTER
        GO TO 19
C
18      KC=0    ;    SET GROUP INDEX
        CALL AFREAD (2,IOUT,$50,$125)   ;   READ 2ND AND SUBSEQUENT LINES
        LC=LC+1   ;   LINE COUNTER
        IF (KG.EQ.1.AND.JC.LT.12) GO TO 20   ;   READ TEMP/DEWPT GRP NEXT
        IF (KG.EQ.1.AND.JC.EQ.12) GO TO 26   ;   READ MAX WND GRP
        IF (KG.EQ.2.AND.KSS.EQ.1.AND.(IZ.GE.KS.OR.IZ2.EQ.88)) GO TO 26
C
C   READ HEIGHT GROUP
C
19      KG=1
        KC=KC+1
        JC=JC+1
        GO TO 81    ;    DELETE THIS LINE FOR TEST, LINE 088
C   NEXT THREE STATEMENTS FOR TEST ONLY
        JD=JC-1     ;     TEMPO TEST !!!!!!!!!!!
        IF (JD.GE.0) WRITE (IFD,80) LC,KC,K,JC,Z(JD),T(JD),TD(JD),D(JD),S(JD)
80      FORMAT (1H ,4I3,F8.0,2F9.1,2F8.0)   ;   TEMPO TEST
81      CONTINUE
        IZ=ITCVT(1+K,1,$900)   ;   INDICATOR FOR PRES LEVEL
        IZ2=ITCVT(1+K,2,$900)  ;   2ND INDICATOR FOR PRES LEVEL
        IF (IZ2.EQ.IS(JC)) GO TO 27   ;   NORMAL
        IF (IZ2.EQ.66) GO TO 27       ;   NORMAL, 66 ENCODED INSTEAD OF 77
        IF (IZ2.EQ.51) GO TO 79       ;   77 GROUP NOT REPORTED, 51515 READ
        IF (IZ2.EQ.88) GO TO 29       ;   SOME LEVELS MISG
        IF (IZ2.EQ.00.AND.(JC.EQ.11.OR.JC.EQ.12)) GO TO 34 ; 88 OR 77 GRP ON NXT LINE (WAL)
        GO TO 112                     ;   FORMAT ERROR
```

```
27      Z(JC)=FTCVT(3+K,3,$901)  ;   HEIGHT (OR SFC, TROP, MAX WND PRES)
        IF (JC.EQ.11.AND.Z(JC).EQ.999.) GO TO 69   ;   TROP NOT OBSERVED
        IF (JC.EQ.12.AND.Z(JC).EQ.999.) GO TO 69   ;   MAX WND NOT OBSERVED
        IF (JC.EQ.12) GO TO 21   ;   READ MAX WND GRP
        IF (KC.LT.10) GO TO 20
        K=-6
        GO TO 18
C
29      DO 33 I=JC,10   ;   SETTING LVLS MISG (SHORT RAOB)
        Z(I)=-99.9   ;   WILL BE CHANGED TO -999., STATEMENT 63
        T(I)=999.
        TD(I)=999.
        D(I)=-99.
33      S(I)=-99.
        JC=11
        GO TO 27
C
34      KG=3   ;   GO TO NEXT LINE TO READ 88 OR 77 GRP (WAL)
        K=0
        JC=JC-1
        GO TO 18
C
C   READ TEMPERATURE/DEWPOINT GROUP
C
20      KG=2
        KC=KC+1
        KL=7+K
        KLL=(KL+5)/2
        KLM=KLL-2
        IF (IOUT(KLM).EQ." /".OR.IOUT(KLM).EQ."//") GO TO 22  ;  TEMP MISG
        T(JC)=FTCVT(KL,3,$901)    ;    READ TEMPERATURE
        IF (IOUT(KLL).EQ."/ ".OR.IOUT(KLL).EQ."//") GO TO 30  ;  DEWPT MISG
        TD(JC)=FTCVT(10+K,2,$901)  ;   READ DEWPOINT
        GO TO 31
30      TD(JC)=999.                ;   DEWPOINT MISSING
31      CALL TEMP1 (T(JC),TD(JC))
        GO TO 23
22      T(JC)=999.   ;   GOES HERE, IF TEMP AND DEWPT BOTH MISG
        TD(JC)=999.
23      CONTINUE
        IF ((KSS.EQ.1.AND.(IZ.GE.KS.OR.IZ2.EQ.00)).OR.IZ2.EQ.88.OR.IZ2.EQ.
     1  99) GO TO 21   ;   TEST WHETHER OR NOT TO READ WND GRP
        IF (KC.EQ.10) GO TO 28
        K=K+12
        D(JC)=-99.   ;   WND MISG
        S(JC)=-99.
        GO TO 19     ;   SKIP WNDS
28      K=0
        D(JC)=-99.
        S(JC)=-99.
        GO TO 18     ;   SKIP WNDS
C
C   READ WIND GROUP
C
21      IF (KC.LT.10) GO TO 26
        K=-12
        GO TO 18
26      KG=3
        KC=KC+1
        IF (JC.EQ.12.AND.KC.GT.1) K=K-6   ;   READING 77 WND GRP
```

```
      KLK=(13+K)/2+1
      IF (IOUT(KLK).EQ."//") GO TO 24   ;   WIND MISSING
      D(JC)=FTCVT(13+K,2,$901)   ;   READ WND DIR
      S(JC)=FTCVT(15+K,3,$901)   ;   READ WND SPEED
      CALL WND (D(JC),S(JC))
      GO TO 25
24    D(JC)=-99.   ;   WND MISSING
      S(JC)=-99.
25    CONTINUE
      IF (IZ2.EQ.77.OR.IZ2.EQ.66) GO TO 71  ;  MAX WND READ, DO WND SHEAR NXT
      IF (KC.LT.10) GO TO 40
      K=0
      GO TO 18
40    CONTINUE
      K=K+18
      GO TO 19
79    Z(JC)=999.   ;   MAX WND GROUP NOT REPORTED
69    T(JC)=999.   ;   GOES HERE, IF TROP OR MAX WND NOT OBSERVED
      TD(JC)=999.
      D(JC)=-99.
      S(JC)=-99.
      IF (JC.EQ.12) GO TO 73   ;   FINISHED
      K=K+6
      GO TO 19                 ;   READ 77 GROUP
C
C   READ 77 WIND SHEAR GROUP
71    CONTINUE
      IF (KC.LT.10) GO TO 72
      CALL AFREAD (2,IOUT,$50,$125)   ;   READ NXT LINE FOR WND SHEAR
      K=-18
72    IFOUR=ITCVT(19+K,1,$900)
      IF (IFOUR.NE.4) GO TO 112
      KMK=(20+K)/2
      IF (IOUT(KMK).NE."4/") GO TO 76   ;   READ LOWER WND SHEAR
      T(JC)=999.                        ;   LOWER WND SHEAR MISSING
      GO TO 77
76    T(JC)=FTCVT(20+K,2,$901)   ;   LOWER WND SHEAR
77    IF (IOUT(KMK+2).NE."/ ") GO TO 78  ;   READ UPPER WND SHEAR
      TD(JC)=999.                        ;   UPPER WND SHEAR MISSING
      GO TO 73                           ;   FINISHED
78    TD(JC)=FTCVT(22+K,2,$901)  ;   UPPER WND SHEAR
C
C   SECONDARY WND MAXIMUM, IF ANY,(2ND 77 OR 66 GROUP) IS NOT DECODED
C
C   DECODE PRESSURE/HEIGHT VALUES: Z(0)/Z(1)....Z(10)
C
73    IF (Z(0).LT.100.) Z(0)=Z(0)+1000.   ;   SFC PRES
      IF (Z(1).LE.500.) GO TO 60
      Z(1)=-(Z(1)-500.)   ;   1000MB LEVEL BLO SEA LEVEL
      WRITE (IFD,61) (IST(I),I=4,5),Z(1)
61    FORMAT ("<15><12>",3X,2A2," 1000MB LVL BLO SEA LEVEL, Z(1) = ",F5.0)
60    Z(2)=Z(2)+1000.    ;    850MB
      IF (Z(3).GT.500.) GO TO 64
      Z(3)=Z(3)+3000.
      GO TO 63
64    Z(3)=Z(3)+2000.    ;    700MB
63    DO 62 I=4,10
62    Z(I)=Z(I)*10.   ;   500 TO 100 MB
      DO 74 I=7,10
      IF (Z(I).EQ.-999.) GO TO 75   ;   LEVELS MISSING
```

```
74      Z(I)=Z(I)+10000.              ;   250 TO 100 MB
75      CONTINUE
        PTROP=Z(11)
        RETURN
C
C   ERROR RETURNS
900     WRITE (IFD,65) (IST(I),I=4,5)
65      FORMAT ("<15><12>",3X,2A2," ERROR IN ITCVT - DECOM")
        RETURN Q
901     WRITE (IFD,66) (IST(I),I=4,5)
66      FORMAT ("<15><12>"3X,2A2," ERROR IN FTCVT - DECOM")
        RETURN Q
50      WRITE (IFD,51) (IST(I),I=4,5)
51      FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 50 - DECOM")
        RETURN Q
100     WRITE (IFD,101) (IST(I),I=4,5)
101     FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 100 - DECOM")
        RETURN Q
102     WRITE (IFD,103) (IST(I),I=4,5)
103     FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 102 - DECOM")
        RETURN Q
125     WRITE (IFD,126) (IST(I),I=4,5)
126     FORMAT ("<15><12>",3X,2A2," AFREAD ERROR 125 - DECOM")
        RETURN Q
110     WRITE (IFD,111) (IST(I),I=4,5)
111     FORMAT ("<15><12>",3X,2A2," STATION MISSING - DECOM")
        RETURN Q
112     WRITE (IFD,113) (IST(I),I=4,5),IS(JC)
113     FORMAT ("<15><12>",3X,2A2," FORMAT ERROR AT LEVEL ",I2," - DECOM")
        RETURN Q
114     WRITE (IFD,115) (IST(I),I=4,5)
115     FORMAT ("<15><12>",3X,2A2," DESIRED VERSION NOT FOUND - DECOM")
        RETURN Q
116     WRITE (IFD,117) (IST(I),I=4,5)
117     FORMAT ("<15><12>",3X,2A2," NEW RAOB NOT AVBL - DECOM")
        RETURN Q
        END
                                *


                                *
        FUNCTION IVCK (IDATE,IHOUR,IDT,IHR)
C   CHECKS DATE/TIME GROUP TO GET DESIRED VERSION
C   IDATE,IHOUR...DATE/HOUR WANTED.  IDT,IHR...DATE/HOUR OF CURRENT VERSION
C   OUTPUT IS AN INTEGER WITH VALUE 1 TO 4:
C   IVCK = 1   CURRENT VERSION IS WANTED
C          2   PREVIOUS VERSION IS WANTED
C          3   NEW VERSION IS NOT AVAILABLE
C          4   VERSION WANTED IS TOO FAR BACK, CANNOT BE RETRIEVED
C   DESIGNED TO RETRIEVE VERSIONS UP TO ABOUT 10 DAYS IN THE PAST.
        IDTCHECK=IDATE-IDT
        IF (IDTCHECK) 1,3,2
1       IVCK=2
        IF (IDTCHECK.LT.-10) IVCK=4
        IF (IDTCHECK.LT.-20) IVCK=3
        RETURN
2       IVCK=2
        IF (IDTCHECK.LT.20) IVCK=4
        IF (IDTCHECK.LT.10) IVCK=3
        RETURN
```

```
3       IHRCHECK=IHOUR-IHR
        IF (IHRCHECK) 4,5,6
4       IVCK=2
        RETURN
5       IVCK=1
        RETURN
6       IVCK=3
        RETURN
        END
                                *


                                *
        SUBROUTINE WND (D,S)
C   WIND DECODE...D= DIRECTION  S= SPEED
        IF (S.LT.500.) GO TO 1
        D=D*10.+5.
        S=S-500.
        RETURN
1       D=D*10.
        RETURN
        END
                                *


                                *
        SUBROUTINE HEIGHT (ZZ,PRES,HGT,Q)
C   GIVES HGT OF PRES SFC ACCORDING TO HEIGHTS OF STANDARD LVLS IN "ZZ" ARRAY
        INTEGER Q
        DIMENSION ZZ(0:12)
        COMMON/PS/SP(10)
        DATA SP/1000.,850.,700.,500.,400.,300.,250.,200.,150.,100./
        DO 1 I=1,10
        IF (PRES-SP(I)) 1,2,3
1       CONTINUE
        I=I-1   ;   "I" IS INCREASED TO 11, WHEN "DO 1" LOOP IS FINISHED
        IF ((SP(10)-PRES).LE.50.) GO TO 3   ;   EXTRAPOLATE UPWARD
        RETURN Q   ;   PRES IS NOT WITHIN RANGE
2       HGT=ZZ(I)   ;  PRES IS A STANDARD PRESSURE SFC
        RETURN
3       IF (I.EQ.1) I=I+1
        SPC=SP(I-1)-PRES
        IF (SPC.LT.-100.) RETURN Q ; IF PRES MORE THAN 1100MB, DON'T EXTRAPOLATE
        IF (ZZ(I).EQ.-999..AND.SPC.GT.50.) RETURN Q  ;  50MB LIMIT ON UPWARD EXTRAPOLATION
        IF (ZZ(I-1).EQ.-999.) RETURN Q  ;  CANNOT CONTINUE, LWR LVL MISG
        IF (ZZ(I).EQ.-999.) I=I-1   ;   EXTRAPOLATES UPWARD IF WITHIN 50MB
        HGT=ZZ(I)+(ZZ(I-1)-ZZ(I))/ALOG(SP(I-1)/SP(I))*ALOG(PRES/SP(I))
        RETURN
        END
                                *


                                *
        FUNCTION JREAL (R)
C   ROUNDS REAL "R" TO INTEGER VALUE
        RA=ABS(R)
        JREAL=RA   ;   TRUNCATES POSITIVE R TO INTEGER
        RD=RA-JREAL   ;   DECIMAL PORTION
        IF (RD.GE..5) JREAL=JREAL+1
        IF (R.LT.0.) JREAL=-JREAL   ;   CHANGE TO ORIGINAL SIGN
```

```
        RETURN
        END
                                *


                                *
        FUNCTION FTCV(DAT,Q)
C
C   THIS FUNCTION IS FOR USE IN READING NUMERICAL DATA INPUT BY SWITCHES
C   ASCII CHARACTERS IN "DAT" ARE UNPACKED, SCANNED, AND INTERPRETED
C   AS REAL NUMBERS.  IF NO DECIMAL POINT IS DETECTED, IT IS ASSUMED
C   TO FOLLOW THE LAST NUMERAL IN THE FIELD.  THE SCAN BEGINS
C   WITH CHARACTER IBGN.   N CHARACTERS ARE SCANNED.
C   ABNORMAL RETURN TO STATEMENT ~Q~.
C   THIS IS A MODIFICATION OF FUNCTION FLTCVT IN AFREAD.LB
C
        DIMENSION IOUTU(20)
        INTEGER Q,DAT(10)
        LOGICAL NEG
        IBGN=1
        CALL UNPACK(DAT,20,IOUTU)
C
C   DETERMINE NUMBER OF CHARACTERS TO READ
        N=0
        DO 1 I=1,20
        IF (IOUTU(I).EQ.0) GO TO 2
1       N=N+1
C
2       CONTINUE
        FTCV=0.
        NEG=.FALSE.
        IEND=IBGN+N-1
100     IF (IOUTU(IEND).NE.32) GO TO 200
        IF (IEND.EQ.IBGN) RETURN
        IEND=IEND-1
        GO TO 100
200     DO 250 I=IBGN,IEND
        IF (IOUTU(I).NE.32) GO TO 300
250     CONTINUE
        RETURN
300     IF (IOUTU(I).EQ.43) GO TO 400
        IF (IOUTU(I).NE.45) GO TO 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) GO TO 700
        FTCV=FTCV*10+IOUTU(I)-48
600     CONTINUE
        IF (NEG) FTCV=-FTCV
        RETURN
700     IF (IOUTU(I).NE.46) GO TO 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) GO TO 800
        FTCV=FTCV+(IOUTU(I)-48)/DIV
        DIV=DIV*10.
```

```
750     CONTINUE
        IF (NEG) FTCV=-FTCV
        RETURN
800     RETURN Q
        END
                                *


                                *
        OVERLAY OV2
        SUBROUTINE TPB
C   COMPUTATION OF POTENTIAL UNSTABLE LYRS AND OUTPUT FOR WRKTPB
        COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
        COMMON/G/PP(0:20),ET(20),TW(0:50),DP,EFF,KMOD,KK
        COMMON/GG/NJ,PPB(15),PPT(15),DELPP(15),DTWDP(15),DPB(15),DPT(15),
     1  PTMAX,PBMAX,TWLAPSE,DMAX
        THETA(T,P2,P1)=T*(P2/P1)**.2857142 ; DRY ADIABATIC (T,P1) TO (THETA,P2)
C
C   COMPUTE WET BULB POTENTIAL TEMP AT ALL SGFNT LEVELS ABV SFC
        DO 78 I=0,JNO
        TC=TCONOF(TS(I),TSD(I))   ;  CONDENSATION TEMPERATURE
        TH=THETA(TS(I)+273.16,1000.,P(I))-273.16   ;  POT TEMP DEG C
        WTH=WOBF(TH)
        WTC=WOBF(TC)
78      TW(I)=TH-WTH+WTC   ;  WET BULB POT TEMP - DEG C
C
        CALL PULYR   ;  DETERMINES POTENTIAL (CONVECTIVE) UNSTABLE LAYERS
        WRITE (21,1) (JST(I),I=4,5),(KDATE(I),I=1,3),IHOUR
1       FORMAT (12X,"            POTENTIAL (CONVECTIVE) UNSTABLE LAYERS FOR "
     1  ,2A2,4X,I2,"/",I2,"/",I2,3X,I2,"Z")
        WRITE (21,2)
2       FORMAT ("<15><12>",3X,"P1",8X,"P2",8X,"DP",4X,"TWLAPSE",6X,"DP1",7X,"DP2")
        M=NJ+1
        DO 8 I=1,NJ
        J=M-I
        WRITE (21,7) PPB(J),PPT(J),DELPP(J),DTWDP(J),DPB(J),DPT(J)
7       FORMAT ("<15><12>",1X,F5.0,2F10.0,F10.1,2F10.0)
8       CONTINUE
        WRITE (21,5)
5       FORMAT ("<15><12>")    ;   BLANK LINE
        WRITE (21,77)
77      FORMAT ("<15><12>","SIGNIFICANT LEVELS",
     1  /"<15><12>",4X,"P",9X,"T",9X,"TD",8X,"TW (WET BULB POTENTIAL TEMP)")
        M=JNO
        DO 79 I=0,M
        J=M-I
        WRITE (21,80) P(J),TS(J),TSD(J),TW(J)
80      FORMAT ("<15><12>",1X,F5.0,3F10.1)
79      CONTINUE
        WRITE (21,5)   ;  ENDING WITH BLANK LINE
        RETURN
        END
                                *


                                *
        OVERLAY OV3
        SUBROUTINE TPA(JEFF,EL1,RLI,RKI,RWI,IDECOM)
C   OUTPUT FOR SINGLE STATION RAOB ANALYSIS FOR WRKTPA
        COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
```

```
      COMMON/G/PP(0:20),ET(20),TW(0:50),DP,EFF,KMOD,KK
      COMMON/GG/NJ,PPB(15),PPT(15),DELPP(15),DTWDP(15),DPB(15),DPT(15),
     1  PTMAX,PBMAX,TWLAPSE,DMAX
      COMMON/T/RLCL,RLFC,EL,B2,B2P,B2N,IALL,B1,B1P,B1N,EX
      COMMON/CCL/PCCL,ETCCL,TS0,TSD0,L,TSCCL,TCCL,TDCCL,WAVG
      COMMON/TT/PT(0:50),TST(0:50),TSDT(0:50)
      COMMON/V/JNOM,PX
      WRITE (20,54) (JST(I),I=4,5),(KDATE(I),I=1,3),IHOUR
54    FORMAT (1H ,"RAOB ANALYSIS FOR ",2A2,4X,I2,"/",I2,"/",I2,3X,I2,"Z",
     1  11X,"UNITS : J/KG X 10")
      IF (JEFF.GE.100) GO TO 45
      WRITE (20,71) EFF,JEFF
71    FORMAT ("<15><12>",2X,"ASSUMED EFF = ",F4.0," PERCENT ENTRAINMENT
     1 PER 500MB ASCENT, FOR EI1, EI2 & EL",I2)
      GO TO 46
45    WRITE (20,72) EFF,JEFF
72    FORMAT ("<15><12>",2X,"ASSUMED EFF = ",F4.0," PERCENT ENTRAINMENT
     1 PER 500MB ASCENT, FOR EI1, EI2 & EL",I3)
46    WRITE (20,73)
73    FORMAT ("<15><12>",2X,"ASSUMED EFF =   0. PERCENT ENTRAINMENT
     1 PER 500MB ASCENT, FOR EL")
      WRITE (20,5)
5     FORMAT ("<15><12>")  ;  PUTTING IN BLANK LINE
      WRITE (20,4) P(0),P(JNO),PT(0),PX
4     FORMAT ("<15><12>","P0 = ",F5.0,4X,"PTOP = ",F5.0,4X,"PMAX (MAX INSTABILITY) = ",
     1  F5.0,4X,"PX = ",F5.0)
C
      IF (IDECOM.EQ.0) GO TO 27
      WRITE (20,50) EL,EL1,RLCL,RLFC
50    FORMAT ("<15><12>","EL = ",F5.0," MB (",F4.0," HND FT)      LCL = ",
     1  F5.0,4X,"LFC = ",F5.0/"<15><12>","BASED ON PARCEL MVG FM LVL ~PMAX~")
      GO TO 26
27    WRITE (20,51) EL,EL1,RLCL,RLFC
51    FORMAT ("<15><12>","EL = ",F5.0," MB (",F4.0,"E HND FT)      LCL = ",
     1  F5.0,4X,"LFC = ",F5.0/"<15><12>","BASED ON PARCEL MVG FM LVL ~PMAX~")
26    CONTINUE
      WRITE (20,5)
      IF (JEFF.GE.100) GO TO 47
      WRITE (20,56) B2,JEFF,B1,B2P,B1P,B2N,B1N
56    FORMAT ("<15><12>","EI2  = ",F7.0,2X,"ENERGY PMAX TO EL",I2,3X,"EI1  = ",F7.0,2X,
     1  "ENERGY PMAX TO PX"/
     2  "<15><12>","EI2P = ",F7.0,2X,"POSITIVE PART",9X,"EI1P = ",F7.0,2X,"POSITIVE PART"/
     3  "<15><12>","EI2N = ",F7.0,2X,"NEGATIVE PART",9X,"EI1N = ",F7.0,2X,"NEGATIVE PART")
      GO TO 48
47    WRITE (20,57) B2,JEFF,B1,B2P,B1P,B2N,B1N
57    FORMAT ("<15><12>","EI2  = ",F7.0,2X,"ENERGY PMAX TO EL",I3,2X,"EI1  = ",F7.0,2X,
     1  "ENERGY PMAX TO PX"/
     2  "<15><12>","EI2P = ",F7.0,2X,"POSITIVE PART",9X,"EI1P = ",F7.0,2X,"POSITIVE PART"/
     3  "<15><12>","EI2N = ",F7.0,2X,"NEGATIVE PART",9X,"EI1N = ",F7.0,2X,"NEGATIVE PART")
48    WRITE (20,5)
      WRITE (20,59)
59    FORMAT ("<15><12>","P1",9X,"P2",9X,"ENERGY GAINED (LOST) IN LAYER")
      KK1=KK-1
      DO 53 I=1,KK1
      J=I-1
      WRITE (20,52) PP(J),PP(I),ET(I)
52    FORMAT ("<15><12>",F5.0,5X,F5.0,5X,F7.0)
53    CONTINUE
      WRITE (20,100) EX
100   FORMAT ("<15><12>","EX = ",F7.0)
```

```
      WRITE (20,5)
      WRITE (20,81) RLI,RKI,RWI
81    FORMAT ("<15><12>","LI = ",F4.0,4X,"KI = ",F4.0,4X,"SWI = ",F4.0)
      WRITE (20,5)
      TSCCLF=1.8*TSCCL+32.  ;  CONV TEMP IN DEG F
      WRITE (20,93) PCCL,ETCCL,TSCCL,TSCCLF,WAVG
93    FORMAT ("<15><12>","CCL = ",F5.0,2X,"ETCCL = ",F6.0,2X,"CONV TEMP = "
     1  ,F5.1," (",F5.1,"F )  WAVG = ",F5.2," G/KG")
      WRITE (20,5)
      IF (DMAX.GT.0.) GO TO 9
      WRITE (20,10)
10    FORMAT ("<15><12>","DEEPEST POT. UNSTABLE LYR : NONE")
      GO TO 11
9     WRITE (20,12) PBMAX,PTMAX,TWLAPSE
12    FORMAT ("<15><12>","DEEPEST POT. UNSTABLE LYR : ",F5.0," - ",F5.0,
     1  " MB, TWLAPSE = ",F5.1,"  SEE WRKTPB")
11    WRITE (20,5)  ;  ENDING WITH A BLANK LINE
      RETURN
      END
                              *


                              *
      SUBROUTINE PULYR
C   COMPUTATION OF POTENTIAL (CONVECTIVE) UNSTABLE LAYERS, WITH LAPSE RATE
C   OF WET BULB POTENTIAL TEMPERATURE AND AMOUNT OF LIFT REQUIRED FOR SATURATION
      COMMON/S/JST(5),KDATE(3),IHOUR,JNO,JJNO,P(0:50),TS(0:50),TSD(0:50)
      COMMON/G/PP(0:20),ET(20),TW(0:50),DP,EFF,KMOD,KK
      COMMON/GG/NJ,PPB(15),PPT(15),DELPP(15),DTWDP(15),DPB(15),DPT(15),
     1  PTMAX,PBMAX,TWLAPSE,DMAX
      DMAX=0
      NJ=0
      IT=-1
      MK=0
      JNNO=JNO-1
      DO 1 I=0,JNNO
      IF ((TW(I)-TW(I+1)).LE.0.) GO TO 2  ;  GOES TO 2, IF STABLE
      GO TO 4
2     IF (MK.EQ.0) GO TO 1  ;  MK=0 INITIALLY, OR IF PREVIOUS LYR STABLE
      GO TO 3  ;  GOES TO 3, WHEN TOP OF UNSTABLE LYRS IS REACHED
C   DETERMINING INDICES OF UNSTABLE LYR, IT = TOP, IB = BOTTOM
4     IF (I.GT.IT) IB=I
      IT=I+1
      MK=1
      GO TO 1
3     NJ=NJ+1
      PPT(NJ)=P(IT)
      PPB(NJ)=P(IB)
      DELPP(NJ)=P(IB)-P(IT)
      DTWDP(NJ)=(TW(IB)-TW(IT))/DELPP(NJ)*100.
      IF(DELPP(NJ).LE.DMAX) GO TO 5
      DMAX=DELPP(NJ)
      TWLAPSE=DTWDP(NJ)
      PTMAX=PPT(NJ)
      PBMAX=PPB(NJ)
5     TC=TCONOF (TS(IB),TSD(IB))  ;  CONDENSATION TEMPERATURE
      PC=P(IB)*((TC+273.16)/(TS(IB)+273.16))**(1./.2857142) ; COND PRESSURE
      DPB(NJ)=P(IB)-PC  ;  AMT OF LIFT REQUIRED FOR BOTTOM SATURATION
      TC=TCONOF (TS(IT),TSD(IT))
      PC=P(IT)*((TC+273.16)/(TS(IT)+273.16))**(1./.2857142)
```

```
        DPT(NJ)=P(IT)-PC   ;  AMT OF LIFT REQUIRED FOR TOP SATURATION
        MK=0
1       CONTINUE
        RETURN
        END
                                *


                                *
        OVERLAY OV0
        PARAMETER NRAOB=50
C   'NRAOB' MUST AGREE WITH SAME PARAMETER IN RANP AND GPT!!!
C   THIS SUBROUTINE FINDS X AND Y COORDINATES FOR PLOTTING ON MAP
C   BACKGROUND 2 BY SEARCHING STATION DIRECTORY FILE.
C   THREE LETTER STATION IDENTIFIER (PACKED) MUST BE SUPPLIED IN ARRAY 'IST'.
        SUBROUTINE STLOC (N,IXX,IYY,IST)
        DIMENSION IXX(NRAOB),IYY(NRAOB),IST(NRAOB,2),JST(3),IB(3),IAD(2),
     1  IC1(14),IC2(14),IC3(14)
        IFLDP=1
        IFLD=6
        IAD(1)=0
        IAD(2)=0
        JST(3)=20040K    ;    DOUBLE SPACE
        CALL GCHN (ICHN,IER)
        CALL OPENR (ICHN,"STDIR.MS",0,IER)
        IF (IER.NE.1) TYPE "ERROR IN OPENING 'STDIR.MS' - STLOC, IER = ",IER
        CALL RDS (ICHN,IB,6,IER)    ;    READ FIRST 3 WORDS FROM FILE
        IF (IER.NE.1) TYPE "READ ERROR IN 'STDIR.MS' - STLOC, IER = ",IER
        DO 4 I=1,N
        JST(1)=IST(I,1)
        JST(2)=IST(I,2)    ;    STN IDENTIFIER, USED TO SEARCH 'STDIR.MS'
        CALL BNSCH(ICHN,IB(1),IB(2),IB(3),IFLDP,IFLD,JST,IAD,IC1,IC2,IC3,IC)
        IF (IC.EQ.0) GO TO 5
        GO TO (1,2,3), IC
1       IXX(I)=2*IC1(8)
        IYY(I)=2*IC1(9)
        GO TO 4
2       IXX(I)=2*IC2(8)
        IYY(I)=2*IC2(9)
        GO TO 4
3       IXX(I)=2*IC3(8)
        IYY(I)=2*IC3(9)
        GO TO 4
5       IXX(I)=0
        IYY(I)=0    ;    X & Y COORDINATES ZERO, IF STATION NOT FOUND
        WRITE (10,6) (JST(J),J=1,3)
6       FORMAT (1H ,3A2,"NOT FOUND IN 'STDIR.MS' FILE")
4       CONTINUE
        CALL KLOSE (ICHN,IER)
        IF (IER.NE.1) TYPE "CHANNEL NOT CLOSED - STLOC, IER = ",IER
        RETURN
        END
                                *


                                *
        OVERLAY OV1
        PARAMETER NRAOB=50
C   PARAMETER ~NRAOB~ MUST AGREE WITH SIMILAR PARAMETER IN RANP
        SUBROUTINE GPT(N,IXX,IYY,JB,JEL,IST,IHOUR,KDATE,JEFF)
```

```
C   THIS VERSION FOR USE ON MAP BACKGROUND 2
C   THIS SUBROUTINE PLOTS A GRAPHIC (EL/B1)
C   INSERTS ENTRAINMENT RATE INTO HEADING OF GRAPHIC
      DIMENSION IXX(NRAOB),IYY(NRAOB),JB(NRAOB),JEL(NRAOB),IST(NRAOB,2)
     1 ,KDATE(3)
      DIMENSION ISC(5),JS(3),IT(15)
      COMMON/TITLE/JT(12)
      DATA JT/"EL/EI1   EFF ="/    ;    FIRST LINE OF TITLE
      MAP=2
      DO 10 I=1,N
      IX=IXX(I)
      IY=IYY(I)
      JDAT=JB(I)
      IF (JDAT.EQ.999) GO TO 1
      CALL ISCR(ISC,JDAT,-1)   ;   CONVERT B1 TO ASCII
      IYOF=12   ;   Y OFFSET FOR PLOTTING
      CALL TEXT (ISC,IX,IY,1,2, 5,IYOF)    ;   PLOT B1
      JDAT=JEL(I)
      CALL ISCR(ISC,JDAT,+1)   ;   CONVERT EL1 TO ASCII
      CALL TEXT (ISC,IX,IY,1,2,-40,IYOF)   ;   PLOT EL1
      ISC(2)=14   ;   STATION SYMBOL, CLEAR
      GO TO 2
1     ISC(2)=5    ;   MISSING DATA
2     ISC(1)=22K  ;   START SPECIAL SYMBOLS
      ISC(3)=21K  ;   END SPECIAL SYMBOLS
      ISC(4)=0
      ISC(5)=0
      IF (JB(I).GT.0.AND.JB(I).NE.999) ISC(2)=3   ;   STATION SYMBOL, OVERCAST
      CALL TEXT (ISC,IX,IY,1,1,0,0)   ;   PLOT STATION SYMBOL
      JS(1)=IST(I,1)
      JS(2)=IST(I,2)
      JS(3)=0
      CALL TEXT (JS,IX,IY,1,1,-7,-10)   ;   PLOT STATION ID
10    CONTINUE
      CALL MTITL(IHOUR,KDATE,IT)   ;   MAKE DATE/TIME GRP TITLE
C   PRINT TITLE IN LOWER RIGHT CORNER OF GRAPHIC
      CALL ISCR(ISC,JEFF,-1)   ;   CONVERT ENTRAINMENT RATE TO ASCII
      DO 3 I=8,12
3     JT(I)=ISC(I-7)
      CALL TEXT (JT,2600,550,3,1,0,0)   ;   FIRST LINE OF TITLE
      CALL TEXT (IT,2600,450,3,1,0,0) ; DATE/TIME LINE OF TITLE
      CALL UTF("NMCGPHEIS","HMSGPH.01")
      RETURN
      END
                              *


                              *
      SUBROUTINE ISCR (ISC,JDAT,KSHIFT)
C   CONVERTS JDAT TO ASCII
C   SET KSHIFT = -1 FOR SHIFT LEFT, +1 FOR SHIFT RIGHT
      DIMENSION ISC(5)
      ISC(1)=32      ;   SPACE
      IF (JDAT.LT.0) ISC(1)=45   ;   NEGATIVE SIGN
      JDAT=IABS(JDAT)   ;   USE ABSOLUTE VALUE OF JDAT
      IF (JDAT.LT.1000) GO TO 2   ;   NORMAL
      JDAT=888                    ;   888 DENOTES: NUMBER TOO LARGE
2     ISC(2)=JDAT/100    ;    HUNDREDS DIGIT
      IS=JDAT-ISC(2)*100
      ISC(3)=IS/10       ;    TENS DIGIT
```

```
        ISC(4)=IS-10*ISC(3)    ;   UNITS DIGIT
        IF (ISC(2).NE.0) GO TO 3    ;   FINISHED, 3 DIGIT NUMBER IS PLOTTED
        IF (ISC(3).NE.0) GO TO 6    ;   2 DIGIT NUMBER IS PLOTTED
        IF (KSHIFT.EQ.1) GO TO 1
C   CONVERT 1 DIGIT NUMBER TO ASCII AND SHIFT LEFT
        ISC(2)=ISC(4)+48
        ISC(3)=32    ;   SPACE
        ISC(4)=32    ;   SPACE
        GO TO 5
C   CONVERT 1 DIGIT NUMBER TO ASCII AND SHIFT RIGHT
1       ISC(4)=ISC(4)+48
        ISC(3)=ISC(1)    ;   SHIFT SIGN
        ISC(1)=32        ;   SPACE
        ISC(2)=32        ;   SPACE
        GO TO 5
6       IF (KSHIFT.EQ.1) GO TO 7
C   CONVERT 2 DIGIT NUMBER TO ASCII AND SHIFT LEFT
        ISC(2)=ISC(3)+48
        ISC(3)=ISC(4)+48
        ISC(4)=32        ;   SPACE
        GO TO 5
C   CONVERT 2 DIGIT NUMBER TO ASCII AND SHIFT RIGHT
7       ISC(4)=ISC(4)+48
        ISC(3)=ISC(3)+48
        ISC(2)=ISC(1)    ;   SHIFT SIGN
        ISC(1)=32        ;   SPACE
        GO TO 5
C   CONVERT 3 DIGIT NUMBER TO ASCII
3       ISC(2)=ISC(2)+48
        ISC(3)=ISC(3)+48
        ISC(4)=ISC(4)+48
5       CONTINUE
        ISC(5)=0    ;    SET TO ZERO FOR TEXT SUBROUTINE
        RETURN
        END
                                *


                                *
        SUBROUTINE JSCR (KN,KS)
C   CONVERTS POSITIVE 2 DIGIT INTEGER TO ASCII
C   IF NEGATIVE INTEGER IS ENTERED, 99 IS RETURNED
C   THIS SUBROUTINE FOR GETTING ASCII DATE/TIME NUMBERS FOR PLOTTING
        DIMENSION KS(2)
        IF (KN.GE.0) GO TO 1    ;   NORMAL
        KS(1)=57     ;      9
        KS(2)=57     ;      9
        WRITE (10,2) KN
2       FORMAT (1H ,"NEGATIVE NUMBER, KN = ",I4,"  IN SUBROUTINE JSCR...
     1  MISTAKE IN DATE/TIME GROUP OF GRAPHIC")
        RETURN
1       CONTINUE
        KS1=KN/10              ;   TENS DIGIT
        KS(1)=KS1+48           ;   TENS DIGIT CONVERTED TO ASCII
        KS(2)=KN-10*KS1+48     ;   UNITS DIGIT CONVERTED TO ASCII
        IF (KS(1).EQ.48) KS(1)=32    ;  SUBSTITUTE SPACE FOR ASCII ZERO
        RETURN
        END
                                *
```

```
                                    *
      SUBROUTINE MTITL (IHOUR,KDATE,IT)
C   THIS SUBROUTINE RETURNS DATE/TIME GROUP "IT" FOR GRAPHIC TITLE
      DIMENSION KDATE(3),IT(15),KS(2)
C   GET ASCII TIME
      IT(1)=48    ;   0
      IT(2)=48    ;   0
      IF (IHOUR.EQ.0) GO TO 13
      IT(1)=49    ;   1
      IT(2)=50    ;   2
13    IT(3)=90    ;   Z
      IT(4)=32    ;   SPACE
      IT(5)=32    ;   SPACE
C   GET ASCII DATE
      CALL JSCR (KDATE(2),KS)
      IT(6)=KS(1)
      IT(7)=KS(2)
      IT(8)=32    ;   SPACE
      IT(12)=32   ;   SPACE
C   GET ASCII YEAR
      CALL JSCR (KDATE(3),KS)
      IT(13)=KS(1)
      IT(14)=KS(2)
      IT(15)=0    ;   MUST BE ZERO FOR TEXT SUBROUTINE
C
C   GET 3 LETTER MONTH ABBREVIATION
C
      KGO=KDATE(1)
      GO TO (1,2,3,4,5,6,7,8,9,10,11,12) KGO
1     IT(9)=74    ;   JAN
      IT(10)=65
      IT(11)=78
      GO TO 14
2     IT(9)=70    ;   FEB
      IT(10)=69
      IT(11)=66
      GO TO 14
3     IT(9)=77    ;   MAR
      IT(10)=65
      IT(11)=82
      GO TO 14
4     IT(9)=65    ;   APR
      IT(10)=80
      IT(11)=82
      GO TO 14
5     IT(9)=77    ;   MAY
      IT(10)=65
      IT(11)=89
      GO TO 14
6     IT(9)=74    ;   JUN
      IT(10)=85
      IT(11)=78
      GO TO 14
7     IT(9)=74    ;   JUL
      IT(10)=85
      IT(11)=76
      GO TO 14
8     IT(9)=65    ;   AUG
      IT(10)=85
```

```
        IT(11)=71
        GO TO 14
9       IT(9)=83    ;   SEP
        IT(10)=69
        IT(11)=80
        GO TO 14
10      IT(9)=79    ;   OCT
        IT(10)=67
        IT(11)=84
        GO TO 14
11      IT(9)=78    ;   NOV
        IT(10)=79
        IT(11)=86
        GO TO 14
12      IT(9)=68    ;   DEC
        IT(10)=69
        IT(11)=67
14      CONTINUE
        RETURN
        END
                                *


                                *
        FUNCTION ITCVT(IBGN,N,Q)
C
C   THIS FUNCTION IS USED WITH SUBROUTINE AFREAD.  ASCII
C   CHARACTERS IN THE CURRENT LINE ARE SCANNED AND INTERPRETED
C   AS INTEGERS.  THE SCAN BEGINS WITH CHARACTER IBGN AND N
C   CHARACTERS ARE SCANNED.  ABNORMAL RETURN TO STATEMENT ~Q~.
C   THIS IS A MODIFICATION OF FUNCTION INTCVT IN AFREAD.LB
C
        COMMON/QARDQ/IOUTU(80)
        INTEGER Q
        LOGICAL NEG
        ITCVT=0
        NEG=.FALSE.
        IEND=IBGN+N-1
100     IF (IOUTU(IEND).NE.32) GO TO 200
        IF (IEND.EQ.IBGN) RETURN
        IEND=IEND-1
        GO TO 100
200     DO 250 I=IBGN,IEND
        IF (IOUTU(I).NE.32) GO TO 300
250     CONTINUE
        RETURN
300     IF (IOUTU(I).EQ.43) GO TO 400
        IF (IOUTU(I).NE.45) GO TO 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) GO TO 800
        ITCVT=ITCVT*10+IOUTU(I)-48
600     CONTINUE
        IF (NEG) ITCVT=-ITCVT
        RETURN
800     RETURN Q
        END
```

```
*
*
      FUNCTION FTCVT(IBGN,N,Q)
C
C   THIS FUNCTION IS USED WITH SUBROUTINE AFREAD.  ASCII
C   CHARACTERS IN THE CURRENT LINE ARE SCANNED AND INTERPRETED
C   AS REAL NUMBERS.  IF NO DECIMAL POINT IS DETECTED, IT IS ASSUMED
C   TO FOLLOW THE LAST NUMERAL IN THE FIELD.  THE SCAN BEGINS
C   WITH CHARACTER IBGN.   N CHARACTERS ARE SCANNED.
C   ABNORMAL RETURN TO STATEMENT ~Q~.
C   THIS IS A MODIFICATION OF FUNCTION FLTCVT IN AFREAD.LB
C
      COMMON/QARDQ/IOUTU(80)
      INTEGER Q
      LOGICAL NEG
      FTCVT=0.
      NEG=.FALSE.
      IEND=IBGN+N-1
100   IF (IOUTU(IEND).NE.32) GO TO 200
      IF (IEND.EQ.IBGN) RETURN
      IEND=IEND-1
      GO TO 100
200   DO 250 I=IBGN,IEND
      IF (IOUTU(I).NE.32) GO TO 300
250   CONTINUE
      RETURN
300   IF (IOUTU(I).EQ.43) GO TO 400
      IF (IOUTU(I).NE.45) GO TO 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) GO TO 700
      FTCVT=FTCVT*10+IOUTU(I)-48
600   CONTINUE
      IF (NEG) FTCVT=-FTCVT
      RETURN
700   IF (IOUTU(I).NE.46) GO TO 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) GO TO 800
      FTCVT=FTCVT+(IOUTU(I)-48)/DIV
      DIV=DIV*10.
750   CONTINUE
      IF (NEG) FTCVT=-FTCVT
      RETURN
800   RETURN Q
      END
*
*
      FUNCTION WOBF(T)
COMPUTE BY DOUBLE ASYMPTOTIC APPROXIMATION
CONSIDER SEPARATELY IF .GT. OR .LE. 20 DEG.
```

```
CENT. FOR ALL TEMPS...THETW=THETA-WOBF(THETA)+WOBF(TEMPCON)
CENT. FOR ALL TEMPS...THETM=THETA-WOBF(THETA)+WOBF(TEMP)
        X=T-20.0
        IF(X) 10,10,20
10      CONTINUE
CURVE FIG FOR COOL TEMPERATURE RANGE
        POL=1.000+X*(-8.8416605E-3+X*(1.4714143E-4+X*(-9.6719890E-7
     1  +X*(-3.2607217E-8+X*(-3.8598073E-10)))))
        POL=POL*POL
        WOBF=15.130/(POL*POL)
        RETURN
20      CONTINUE
CURVE FIT FOR WARMER TEMPERATURES
        POL=1.000+X*(3.6182989E-3+X*(-1.3603273E-5+X*(4.9618922E-7
     1  +X*(-6.1059365E-9+X*(3.9401551E-11+X*(-1.2588129E-13
     2  +X*(1.6688280E-16)))))))
        POL=POL*POL
        WOBF=29.930/(POL*POL)+0.9600*X-14.800
        RETURN
        END
                                *


                                *
        FUNCTION SATLFT (THM,P)
COMPUTES TEMPERATURE (DEG C) WHERE THETA MOIST (DEG C) CROSSES P (MB)
CONSIDER THE EXPONENTIAL FOR POTENTIAL TEMPERATURE AS ROCP
        ROCP=0.28571428
        IF(ABS(P-1000.0)-0.0010) 100,100,200
100     SATLFT=THM
        RETURN
200     PWRP=(P/1000.0)**ROCP
COMPUTE TEMPERATURE OF DRY ADIABATIC LIFT FOR FIRST GUESS
        TONE=(THM+273.16)*PWRP-273.16
CONSIDER PSEUDO-ADIABAT, EW1, THROUGH TONE AT P.
COMPUTE EONE=EW1-THM
        EONE=WOBF(TONE)-WOBF(THM)
        RATE=1.0
        GO TO 330
300     CONTINUE
CONTRIBUTION TO ITERATION IS CHANGE IN T
CORRESPONDING TO CHANGE IN E
        RATE=(TTWO-TONE)/(ETWO-EONE)
        TONE=TTWO
        EONE=ETWO
330     CONTINUE
COMPUTE ESTIMATED SATLIFT, TTWO
        TTWO=TONE-EONE*RATE
CONSIDER PSEUDO-ADIABAT, EW2, THROUGH TTWO AT P.
COMPUTE ETWO=EW2-THM
        ETWO=(TTWO+273.16)/PWRP-273.16
        ETWO=ETWO+WOBF(TTWO)-WOBF(ETWO)-THM
CORRECTION TO TTWO IS EOR
        EOR=ETWO*RATE
        IF(ABS(EOR)-0.1000) 400,400,300
400     SATLFT=TTWO-EOR
        RETURN
        END
                                *
```

```
                                *
      FUNCTION TCONOF(TEMP,DEWPT)
COMPUTES CONDENSATION TEMPERATURE (DEGREES CENT) BY LIFTING
      S=TEMP-DEWPT
CONSIDER TEMP AND DEWPT TO BE LIKE UNITS (C OR K)
      T=TEMP
      IF(100.-TEMP) 4,5,5
4     T=TEMP-273.16
COMPUTE CURVE FIT IN MOST EFFICIENT MANNER
5     DLT=S*(1.2185+0.001278*T+S*(-0.002190+11.73E-6*S-5.20E-6*T))
      TCONOF=T-DLT
      RETURN
      END
                                *


                                *
      FUNCTION WMROF(P,TD)
COMPUTE MIXING RATIO (G/KG)...DEWPOINT (DEGREES C OR K)...PRESSURE (MB)
      T=TD
      IF (100.-T) 3,4,4
3     T=T-273.16
CURVE FIT CORRECTION FOR NON-IDEAL GAS
4     X=0.0200*(T-12.5+7500.0/P)
      WFW=1.+0.0000045*P+0.00140*X*X
COMPUTE ACCORDING TO STANDARD FORMULA
      FWESW=WFW*VAPFW(T)
      WMROF=621.97*(FWESW/(P-FWESW))
      RETURN
      END
                                *


                                *
      FUNCTION DPTOF(EW)
COMPUTE DEWPOINT, DPT, IN DEGREES C GIVEN WATER VAPOR PRESSURE (MB)
CREATE TOLERANCE TO DEGREE DESIRED
      TOL=0.00010
      IF (EW-0.21382876E-09) 20,20,30
20    DPTOF=-10000.
      RETURN
30    IF (1013.0-EW) 20,100,100
CREATE GUESS BY INVERTING TETEN~S FORMULA
100   X=ALOG(EW/6.1078)
      BOT=17.269388-X
      DPTOF=(237.3*X)/BOT
      BOT=BOT*EW
      DELTM=0.
200   EDP=VAPFW(DPTOF)
CORRECT GUESS BY DERIVATIVE OF TEMPERATURE WITH RESPECT TO VAPOR PRES.
CALCULATED FROM INVERSE OF TETEN~S FORMULA
      DTDE=(DPTOF+237.3)/BOT
      DELT=DTDE*(EW-EDP)
      DPTOF=DPTOF+DELT
CHECK THAT ITERATION IS NOT IN AN ENDLESS CYCLE, A RARE SITUATION
C   IF NEEDED, CHANGE ~TOL~ AND EXIT
      DM=DELT-DELTM
      IF(ABS(DM).GE.1.E-7) GO TO 10 ; IF DM VERY SMALL, ITERATION IS ENDLESS
      TOL=ABS(DELT)
```

```
        TYPE "TOLERANCE (TOL) IN DPTOF CHANGED TO ",TOL," (NORMAL TOL = .00010)"
10      DELTM=-DELT
CHECK TO SEE IF ANSWER CLOSE ENOUGH, IF NOT ITERATE OVER CORRECTION
        IF (ABS(DELT)-TOL) 300,300,200
CHANGE SO DEWPOINT IS ALWAYS LESS THAN THE TEMP.
COMPATIBILITY WITH TOL IS FORCED
300     DPTOF=DPTOF-TOL
        RETURN
        END
                                *


                                *
        FUNCTION VAPFW(T)
COMPUTE SATURATION VAPOR PRESSURE OVER WATER, VAPFW, IN MBS.
CONSIDER T(TEMPERATURE) IN DEGREES C OR DEGREES K.
        X=T
        IF (100.0-X) 3,4,4
3       X=X-273.16
CURVE FIT FOR RANGE -50 < T < 100 DEGREES C.
4       POL = 0.99999683  E-00 + X *(-0.90826951  E-02 +
     1    X *(0.78736169  E-04 + X *(-0.61117958  E-06 +
     2    X *(0.43884187  E-08 + X *(-0.29883885  E-10 +
     3    X *(0.21874425  E-12 + X *(-0.17892321  E-14 +
     4    X *(0.11112018  E-16 + X *(-0.30994571  E-19)))))))))
        POL=POL*POL
        POL=POL*POL
        VAPFW=6.107800/(POL*POL)
        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
```

## INDEX OF SUBROUTINES

EASTERN REGION CP No. 16

# RANP
# STABILITY ANALYSIS PLOT PROGRAM

## PART A: PROGRAM INFORMATION AND INSTALLATION PROCEDURE (RANP)

PROGRAM NAME: RANP

AAL ID:
REVISION NO. 1.00

PURPOSE: Uses significant level raob data to compute the energy indices EI1 and EI2 and equilibrium level EL along with several other thermodynamic parameters.

PROGRAM INFORMATION:

Development Programmer:
Hugh M. Stone
Location: ERH Garden City, NY
Phone: (FTS) 649-5443
Language: FORTRAN IV
Date: 2/84
Running Time:
Single station, 40 to 60 seconds
32 stations, about 8 minutes

Maintenance Programmer:
Same

Type: Standard
Revision Date: NA

Disk Space:

| | | | |
|---|---|---|---|
| Program Files | - | 122 | RDOS blocks |
| Overlay Files | - | 28 | RDOS blocks |
| Data Files | - | 3 | RDOS blocks |

PROGRAM REQUIREMENTS:

Program Files: RANP.SV and RANP.OL

Data Files:

| Name | DP Location | R/W | Comments |
|---|---|---|---|
| STNS1 | DPØ | Read | List of raob stations (not to exceed 50) |
| INDEXX | DPØ | Write | Temporary |
| INDEXY | DPØ | Write | Temporary |
| HMSGPH.01 | DPØ | Write | Temporary |

AFOS Products:

| ID | ACTION | COMMENTS |
|---|---|---|
| CCCSGLXXX | Input | List from file STNS1 or specified in RUN line. |
| CCCMANXXX | Input | Not necessary, but used, if available, to get tropopause and to convert EL units from pressure to feet. |
| WRKTPA | Output | Complete stability analysis for a single station. |
| WRKTPB | Output | Listing of significant levels and convectively unstable layers for a single station. |
| WRKTPC | Output | Tabular listing of energy indices for list specified in file STNS1. |
| WRKTPD | Output | Listing of missing or unuseable SGL raob reports. ("C" switch) |
| NMCGPHEIS | Output | Graphic with plotted values of EI1 and EL. |

LOAD LINE:

```
RLDR/P RANP DECOS TEMP1 RANN2 CCL1 MODRB INDX1 BNDX WOBF SATLFT TCONOF
 WMROF DPTOF VAPFW DECOM IVCK WND HEIGHT JREAL FTCV
 [TPB PULYR, TPA, STLOC BNSCH, GPT ISCR JSCR MTITL]
 OUT AFREAD.LB ITCVT FTCVT TOP.LB AG.LB UTIL.LB FORT.LB
```

PROGRAM INSTALLATION:

1. Add CCCWRKTPA, B, C, & D to database.
   Add NMCGPHEIS to database and assign map background 2.

2. RANP.SV, RANP.OL, and STNS1 should be on DPØ or DPØF with link to DPØ.

# RANP
# STABILITY ANALYSIS PLOT PROGRAM

## PART B: PROGRAM EXECUTION AND ERROR CONDITION (RANP)

PROGRAM NAME: RANP

### PROGRAM EXECUTION

1. To run program for a single station, at ADM type:

   RUN:RANP/S CCCSGLXXX

2. To check the database prior to running program for raobs in file STNS1, at ADM type:

   RUN:RANP/C

3. To run program for list of raobs in file STNS1, at ADM type:

   RUN:RANP

A local switch "/E" is available for changing the entrainment rate from its basic value of 60 percent; it may be added to the end of the RUN line in options 1 and 3 above. However, use of E switch is not recommended.

### ERROR CONDITIONS

Error condition messages, if any, are output to the dasher or ADM and the alert light is turned on at the ADM.

# 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**