IV. RAWINS

This section documents program RAWINS, that component of the PSU/NCAR modeling system which performs the objective analysis of surface and upper-air observations. The first section, Section A, discusses the purpose of RAWINS, and presents some of the methods by which RAWINS achieves that purpose. Section A outlines some of the techniques used by RAWINS without examining the source code itself. The second section, Section B, provides the user with some basic instructions for executing RAWINS for various cases. Section B should familiarize the reader with the input to and output from RAWINS, the user options for RAWINS, and the RAWINS script. The third section, Section C, discusses the details of the program code. Section C details the memory structure and the general program flow, and examines all of the key subprograms individually.

A. Purpose and Methods of RAWINS

RAWINS is part of the set of programs which comprise the "front end" of the PSU/NCAR modeling system. This set of programs, which includes programs TERRAIN, DATAGRID, RAWINS, and INTERP, prepares initial fields and other input data for the forecast model. The purpose of RAWINS is to enhance a coarse-grid analysis by objectively analyzing surface and upper-air observations. RAWINS bases its objective analyses on the output from DATAGRID. The objective analyses are then used as input to the program INTERP, which vertically interpolates the pressure-level analyses to σ levels to create the initial fields and boundary conditions for the forecast model.

RAWINS produces improved analyses by using a Cressman-type objective-analysis scheme (discussed in detail below) to successively nudge first-guess fields toward the raw observations. The first-guess fields for any individual analysis may come from one of several sources. Usually, program DATAGRID is executed to provide first-guess fields at the surface and at mandatory pressure levels. RAWINS then creates first-guess fields at user-specified nonmandatory pressure levels by vertically interpolating from the first-guess fields at the mandatory levels. If DATAGRID is not used, output from a previous execution of the forecast model may supply first-guess fields to RAWINS. Surface first-guess fields at intermediate FDDA times may be interpolated from surface first-guess fields at standard upper-air times (0000 and 1200 UTC). As an alternative, analyzed surface fields from one time period may serve as the first-guess fields for the next intermediate FDDA surface analysis. Finally, if no first-guess fields of suitable quality are available, RAWINS can create first-guess fields based on the raw observations themselves. This last option is not recommended unless there are no other sources for first-guess fields.

RAWINS improves the first-guess fields by nudging toward observations from historical archives or real-time observations available from Unidata. RAWINS has several objective data-checking procedures to remove suspect observations, thereby reducing the chances that erroneous observations will be incorporated into the final analyses. RAWINS also has options for the user to insert bogus data into the dataset and subjectively correct suspect observations.

1. Cressman-type Scheme for Objective Analysis

The purpose of objective analysis is to derive meaningful data at regularly spaced gridpoints from irregularly spaced observations. In the Cressman-type scheme used by RAWINS (Benjamin and Seaman, 1985), several successive scans nudge a first-guess field, defined at the mesoscale grid points, toward the observations. Each scan first compares a current-guess field to the observations, calculating a difference value between the current guess and the observation at each observation point. The gridpoints values are then adjusted according to an average of the difference values at nearby observation points, weighted by the distance from the gridpoint to the observations. The procedure is outlined in greater detail below.

The standard Cressman scheme (Cressman, 1959) assigns to each observation a circular region of influence with radius R. Each gridpoint which falls within a distance R of an observation is influenced by that observation. Any particular gridpoint may fall within the regions of influence of several observations. For each mesoscale gridpoint (i, j), the value of the current-guess field G_1 is interpolated from the gridpoints to the n observations which include gridpoint (i, j) in their regions of influence. A difference value D_k between the kth observation O_k and the current-guess G_{1_k} at the observation location is defined as

$$D_k = O_k - G_{1_k}. \tag{4.1}$$

A weighting coefficient W is also calculated for each observation, based on the distance d_{ijk} of the kth observation from the gridpoint (i, j):

$$W_{ijk} = \frac{R^2 - d_{ijk}^2}{R^2 + d_{ijk}^2}\ . \tag{4.2}$$

From (4.2) it can be seen that the weighting coefficient decreases with increasing distance from the gridpoint, and thus the closer an observation is to the gridpoint, the greater is its weight. The weighted average α of the difference values within distance R from gridpoint (i, j) is then calculated by

$$\alpha_{ij} = \frac{\sum_{k=1}^{n} W_{ijk}^2 D_k}{\sum_{k=1}^{n} W_{ijk}}, \tag{4.3}$$

for adjusting a current guess, and

$$\alpha_{ij} = \frac{\sum_{k=1}^{n} W_{ijk}^2 D_k}{\sum_{k=1}^{n} W_{ijk}^2}, \tag{4.4}$$

for creating a new analysis from observations; or for some applications,

$$\alpha_{ij} = \frac{1}{n} \sum_{k=1}^{n} W_{ijk} D_k, \tag{4.5}$$

for adjusting a current guess, and

$$\alpha_{ij} = \frac{\sum_{k=1}^{n} W_{ijk} D_k}{\sum_{k=1}^{n} W_{ijk}}, \tag{4.6}$$

for creating a new analysis from observations

The adjusted value G_2 of the analysis at gridpoint (i, j) is then :

$$G_{2_{ij}} = G_{1_{ij}} + \alpha_{ij}. \tag{4.7}$$

The procedure is then repeated several times, using the improved analysis G_2 rather than the earlier field G_1 as the current guess. Each successive scan decreases the radius of influence R, in order to analyze for features on a smaller scale.

The standard Cressman scheme with its circular region of influence around each observation works well for observed fields that have not been significantly deformed by the wind field. However, Benjamin and Seaman (1985) and others have noted that meteorological variables frequently exhibit along-flow streakiness. For example, low-level moist tongues and wind maxima in jet streaks tend to exhibit streaked patterns. The temperature field, on the other hand, usually does not exhibit a streaked pattern. In cases of streaked behavior, an analysis procedure with a circular region of influence will not reproduce the streaked features effectively. When the fields are deformed by straight flow, elliptical regions of influence (oriented with the major axis parallel to the streamlines) produce better analyses; and when the fields are deformed by curved flow, curved elliptical (banana-shaped) regions of influence (curved along the streamlines) produce better analyses. For this reason, RAWINS allows the region of influence around each observation point to assume one of three possible shapes: the circular shape of the standard Cressman scheme discussed above; an elliptical shape (Benjamin and Seaman, 1985); or a curved elliptical shape (the Banana scheme, also from Benjamin and Seaman).

The user may select any of three weighting schemes: Cressman, Elliptical, or Banana. If the user selects the Cressman scheme, the region of influence about all observations will be circular. For the elliptical scheme, the circular region of influence is used only for the temperature field (because temperature does not usually exhibit along-flow streakiness) and for observation points at which the wind speed is less than a certain critical value v_c (because the elliptical elongation would be small in such cases). The critical wind speed v_c depends on the pressure level of the observation and is defined by

$$v_c(p) = \begin{cases} 25.0 - 0.02p & \text{for } p \geq 500 \text{ mb}, \\ 15.0 & \text{for } p < 500 \text{ mb}. \end{cases} \tag{4.8}$$

The elliptical weighting function has a form similar to the circular weighting function:

$$W_{ijk} = \frac{R^2 - d_m^2}{R^2 + d_m^2} \ , \tag{4.9}$$

where d_m is defined by

$$d_m = \left(\frac{{x'}^2}{1 + \beta \left| \mathbf{V_k} \right|} + {y'}^2 \right)^{\frac{1}{2}} \ , \tag{4.10}$$

where β is the elongation of the ellipse, defined by

$$\beta = \frac{0.7778}{v_c}, \tag{4.11}$$

and x' and y' are defined as

$$x' = \frac{u_k (x_{ij} - x_k) + v_k (y_{ij} - y_k)}{|\mathbf{V}_k|}, \tag{4.12a}$$

$$y' = \frac{v_k (x_{ij} - x_k) - u_k (y_{ij} - y_k)}{|\mathbf{V}_k|}. \tag{4.12b}$$

Equation (4.9) defines a region of influence in the shape of an ellipse elongated in the direction of the wind at observation O_k by a factor of $(1 + \beta |\mathbf{V}_k|)^{\frac{1}{2}}$.

If the user selects the Banana scheme, again the circular function is used for temperature and for observation points at which the wind speed does not exceed the critical value v_c. The elliptical region of influence is used if the streamline radius of influence exceeds three times the initial radius of influence, because the curvature would be small. If the wind speed and curvature are great enough, the banana-shaped region of influence is used. Thus, RAWINS uses the more complex functions only at observation points for which the flow justifies the additional computations required.

The weighting function for the Banana scheme has the same form as that of the elliptical and circular schemes:

$$W_{ijk} = \frac{R^2 - d_m^2}{R^2 + d_m^2} . \tag{4.13}$$

In the Banana scheme, d_m is defined as

$$d_m = \left(\frac{{x_b'}^2}{1 + \beta |\mathbf{V}_k|} + {y_b'}^2 \right)^{\frac{1}{2}} , \tag{4.14}$$

where x_b' and y_b' are defined by

$$x_b' = |r_k| (\theta_k - \theta_{ij}) , \tag{4.15a}$$

$$y_b' = (|r_k| - r_{ij}) , \tag{4.15b}$$

where

θ_k is the angle from the positive x-direction to the line between the center of curvature and kth the observation point.

θ_{ij} is the angle from the positive x-direction to the line between the center of curvature and the gridpoint (i, j).

r_k is the radius of curvature of the streamline at the kth observation point.

r_{ij} is the distance from the center of curvature to the gridpoint (i, j).

The radius of curvature is defined according to the relative vorticity ζ_r of the first-guess field at the gridpoint (i_o, j_o) nearest the kth observation point (i, j). In natural coordinates,

$$r_k = \frac{\mathbf{V}_{i_o j_o}}{\zeta_{r_{i_o j_o}} + \frac{\partial |\mathbf{V}_{i_o j_o}|}{\partial n}}, \tag{4.16}$$

where

$$\zeta_{r_{i_o j_o}} = \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y}, \tag{4.17}$$

and n is the distance along a perpendicular to the left of the flow. In RAWINS, the derivatives in equations (4.16) and (4.17) are evaluated as finite differences over distances of three grid spaces:

$$\zeta_{r_{i_o j_o}} = \frac{v_{(i_o, j_o+3)} - v_{(i_o, j_o-3)}}{6\Delta x} - \frac{u_{(i_o+3, j_o)} - u_{(i_o-3, j_o)}}{6\Delta y}, \tag{4.18}$$

and

$$\frac{\partial |\mathbf{V}_{\mathrm{i_o j_o}}|}{\partial n} = \frac{\left(u_1^2 + v_1^2\right)^{\frac{1}{2}} - \left(u_2^2 + v_2^2\right)^{\frac{1}{2}}}{6\Delta s}, \tag{4.19}$$

where $\Delta s = \Delta x = \Delta y$, and u and v at points 1 and 2 are the wind components of the first-guess field at distances of $\pm 3\Delta s$ along the direction normal to the wind in the vicinity of the station. The calculations of equations (4.18) and (4.19) are performed from the first-guess field, not the current guess.

The i and j coordinates of the streamline center of curvature, i_c and j_c, are calculated according to

$$i_c = i_o + r_k \frac{u_{i_o j_o}}{|\mathbf{v}_{i_o j_o}|}, \tag{4.20a}$$

$$j_c = j_o - r_k \frac{v_{i_o j_o}}{|\mathbf{v}_{i_o j_o}|}. \tag{4.20b}$$

Because the center of curvature is located along the line perpendicular to the wind at point (i_o, j_o), the angle θ_k may be calculated by

$$\theta_k = \arctan\left(-\frac{u_{i_o j_o}}{v_{i_o j_o}}\right) + \pi. \tag{4.21}$$

The angle θ_{ij} may be calculated by

$$\theta_{ij} = \arctan\left(\frac{i - i_c}{j - j_c}\right). \tag{4.22}$$

The resulting region of influence defined by (4.11) is an ellipse curved along the streamlines.

2. Quality control for observations

RAWINS has a number of features for checking observations and removing suspect data. These features include three objective checks and a mechanism for the user to subjectively examine and remove suspect data.

a. Objective checks

The objective procedures perform quality checks on observations of temperature, wind, and surface pressure. No objective quality check is performed for the moisture observations.

Vertical check of sounding data

Each upper-air report read by RAWINS is checked for unrealistic spikes in the profiles of temperature and wind. For spikes in the wind profile, an attempt is made to correct the suspect observations by altering the most significant digit in either the speed or the direction. If no correction can be made, the suspect data point is deleted. Spikes are removed from the temperature profile by deleting the suspect data points. Additionally, superadiabatic lapse rates in the temperature profile are removed. Once the temperature profile has been corrected, the geopotential heights are recalculated according to the corrected temperature profile.

Check of observations against the first-guess field

For each observation of temperature, wind, and surface pressure, the observation is compared to the first-guess field. For each observation point, the first-guess field is interpolated to determine the value of the field at the observation point. The difference between the observation and the first guess is calculated, and if the difference value exceeds a certain maximum allowable difference (based on variable type, pressure level, and user specifications), the observation is flagged as bad data and removed from consideration.

Check of difference values against those at surrounding stations (buddy check)

For each observation point, the difference value between the observations and the first guess is calculated. The difference value at each observation point is then compared to an average of the difference values at nearby observation points. If the difference value of the observation differs from the average by more than a certain maximum allowable difference (depending on pressure level, variable type, and user specifications), then the observation is flagged as suspect, and removed from consideration.

b. Subjective check (Autobogus)

The subjective check of observations by the user is performed in two successive submittals of RAWINS. In the first submittal, each observation is checked against the first-guess field, as discussed above, and the suspect data are flagged. RAWINS then continues with the objective analysis, ignoring all flagged observations. The analyzed fields are then plotted with the suspect observations overlaid, allowing the user to examine the observations which RAWINS flagged as potentially bad data. Any of these suspect data which do not appear to the user to be erroneous are so noted by the user in an input file listing all the flagged data to the second RAWINS submittal. The second submittal then reanalyzes all fields, ignoring those flagged values which the user confirmed were erroneous. This procedure is referred to as *autobogusing.*

3. Bogus data

RAWINS provides options in addition to the autobogusing for the user to both delete data from the original set of observations and to insert bogus data into the dataset.

The provisions to delete data are useful when the user knows (typically from a previous RAWINS submittal) that certain specific observations, or even entire station reports, are erroneous. In those cases, the user may informs RAWINS (through an input file) which observations or station reports are erroneous, and RAWINS will not include them in the analysis. The provisions to insert bogus data are useful when the user has observations not archived, or feels that the objective analysis would be more accurate if bogus data at certain points were to be included. In this case, the user includes the locations and values of the bogus observations or station reports in an input file before submitting RAWINS. The details of both deletion and inclusion of data are discussed in the Section IV.B, *How to use RAWINS*, and in Appendices D and E.

B. How to use RAWINS

This section contains information which the user will need to successfully run RAWINS. The discussion is specific to running RAWINS on NCAR's Cray Y-MP (shavano) and assumes some familiarity with the UNICOS operating system and the C-shell script language. An example of a shell script used to submit program RAWINS to shavano is shown in Appendix C, and can be found in the file /u1/mesouser/Decks/Rawins/rawins.deck on shavano.

As with DATAGRID, the capabilities and procedures of RAWINS periodically change, which implies that detailed instructions have a limited period of usefulness.

RAWINS has many features that allow the user to tailor the processing of the observations and analyses to the requirements of a particular case. If the forecast model is to make use of its four-dimensional-data-assimilation (FDDA) capabilities, the user must activate the FDDA options in RAWINS, which produce surface analyses at intermediate times between the standard upper-air observation times (0000 and 1200 UTC). If the user knows in advance that certain observations are erroneous, or that certain observations have not been included in the archived files, the options for deletion of data or insertion of bogus data may be activated. If the user is concerned about the objective data-checking procedures of RAWINS and wants to subjectively check any observations which RAWINS considers erroneous, the autobogus options, which require two separate submittals of RAWINS, may be activated.

Five major categories of RAWINS submittals may be defined: the Standard RAWINS submittal, with none of the special options activated; the FDDA submittal, for producing fields to be used for FDDA; the Bogus submittal, for insertion, deletion, and modification of specific observations or station reports; the first Autobogus submittal (hereafter referred to as Autobogus 1), which examines the data and plots analyses for the subjective inspection of suspect observations; and the second Autobogus submittal (Autobogus 2), which completes the analysis begun in the Autobogus 1 submittal, ignoring those suspect data points which RAWINS determined were erroneous and incorporating the suspect observations which the user determined were valid.

Selecting options from one category of RAWINS submittal does not preclude using options from another. The FDDA options, Bogus options, and Autobogus options may all be activated in the same RAWINS submittal. The only exception is that the Autobogus 1 and Autobogus 2 options may not be combined in one submittal. The other distinctions among the categories are made merely for the convenience of the discussion to follow.

The following sections discuss the input, output, script variables, and Local Master Input File options for each category of RAWINS submittal.

1. Standard Case

The standard RAWINS submittal takes as input the Local Master Input File, the first-guess fields, the upper-air observations, and the surface observations (if any). The upper-air data are objectively checked and corrected, and upper-air data and first-guess fields are interpolated to the nonmandatory levels. All observations are objectively checked against the first-guess fields. No bogus data are included. The objective analysis is performed at the surface, at mandatory levels, and at nonmandatory levels. One file containing the improved analyses on the mesoscale grid is written, and one file containing the corrected rawinsonde reports is written. One plot file containing sounding plots is generated.

a. RAWINS input – Standard Case

Local Master Input File

The RAWINS Local Master Input File contains options related to the objective analysis. It is accessed by RAWINS as FORTRAN unit 14. The contents of the Local Master Input File are discussed later in Section IV.B.1.c.

First-guess fields

The first-guess fields input to RAWINS are generated by either program DATAGRID or a previous submittal of the forecast model. The first-guess fields are contained in a single file stored on the MSS. The file is retrieved from the MSS by the RAWINS script, and is accessed by RAWINS as FORTRAN unit 4. For each standard time period, the file contains a header record followed by all of the data records for that time. The header record contains variables from the Common Master Input File and the DATAGRID local Master Input File which define the contents and format of the subsequent data records. Each record holds the data for one variable, at one level, for one time period. For further details of the contents of the input file, see the discussion of the output from DATAGRID, Section III.B.4.a.

Upper-air observations

If the modeling system is used for historical simulations, archived observations are used in RAWINS. The files containing the upper-air observations are archived on the NCAR MSS. The user must determine the MSS filenames of the files which contain the data from the time period of the simulation. Catalogs listing the upper-air data archived in the MSS files may be found on shavano in the /u1/mesouser/catalog directory. The files containing the upper-air observations are retrieved from the MSS by the RAWINS script, and accessed by the RAWINS program beginning with FORTRAN unit 20.

If the modeling system is used in a real-time mode, observations are retrieved from UNIDATA. In this case, the RAWINS script copies files containing observations to the local directory.

Surface observations

The surface observations are also archived on the MSS. The user must determine the MSS filenames of the files containing the surface data covering the simulation period. The 0300, 0900, 1500, and 2100 UTC surface observations are stored in one set of files, while the 0000, 0600, 1200, and 1800 UTC surface observations are stored in a separate set of files. The files containing the 0300, 0900, 1500, and 2100 UTC surface observations also contain the 0000, 0600, 1200, and 1800 UTC ship and buoy observations. Catalogs listing the surface data archived in the MSS files may be found on shavano in the /u1/mesouser/catalog directory.

Normally, for the standard RAWINS submittal surface observations from only 0000 and 1200 UTC would be used. Thus, the files containing surface data at 0000, 0600, 1200, and 1800 UTC, and the files containing ship and buoy data at 0000, 0600, 1200, and 1800 UTC should be accessed. The files containing the land-station data are accessed beginning with FORTRAN unit 25. The files containing the ship and buoy data are accessed beginning with FORTRAN unit 30.

b. RAWINS output – Standard Case

Output analysis file

The main output from RAWINS is the file containing the final analyses on the mesoscale grid. This output file is written as FORTRAN unit 2. For each time period processed, it contains a header record followed by all final analyses for that time period. The header record contains information concerning the format of the subsequent data records as well as many other variables from the Common and Local Master Input Files. Each record contains all the data for the analysis of one variable, at one level, for one time period. The order of the output records is as follows:

- Record Header.
- Terrain height (at cross points).
- Land-use characteristics (at cross points).
- Map-scale factor (at cross points).
- Map-scale factor (at dot points).
- Coriolis parameter (at dot points).
- Latitude (at cross points).
- Longitude (at cross points).
- Latitude (at dot points).
- Longitude (at dot points).
- Snow cover (at cross points).
- Sea-level pressure (at dot points).
- Sea-level pressure (at cross points).
- Surface temperature (at cross points).
- Sea-surface temperature (at cross points).
- Pressure-level temperature (at cross points); one record for each level up to PTOP.
- Surface u (at dot points).
- Surface v (at dot points).
- Geopotential height, u, and v at pressure levels (at dot points); three records for each pressure level up to PTOP.
- Surface relative humidity (at cross points).
- Relative humidity at pressure-levels (at cross points); one record from each level up to PTOP.

Output sounding file

RAWINS outputs a file containing the upper-air station data to FORTRAN unit 11. The data in this file have been corrected by the objective check of vertical consistency, but have not been corrected by objective comparison to the first-guess fields or to surrounding data. Thus, the data in this file may not be exactly the data which RAWINS used for its analysis.

For each upper-air report processed, RAWINS writes a single record containing:

- the station identifier.
- the number of levels at which temperature and dewpoint are reported.
- the array of pressure levels at which temperature and dewpoint are reported.
- the temperature array.
- the dewpoint array.
- the number of levels at which wind data are reported.
- the array of pressures at the wind levels.
- the array of wind directions.
- the array of wind speeds.
- the station latitude, longitude, and elevation.
- the year, month, day, and hour processed.

Output sounding plots

RAWINS generated an NCAR-Graphics file, called SND.PLT, which contains Stüve diagrams of all the upper-air reports. The data plotted have been objectively checked for vertical consistency, but have not been objectively compared to the first-guess fields or surrounding data. Thus, some of the data plotted may not be used for the analysis.

Print Output

The print output from the Standard RAWINS submittal provides a description of the options activated, the input fields (used as a first guess for the objective analyses), and all upper-air reports accessed. This output is followed by samples of the first-guess fields and the corresponding final analyses. Next is printed a summary of all analyzed fields written to the output file (unit 2). Finally, any observations which RAWINS determined to be erroneous are printed, along with the reasons RAWINS flagged those observation as bad data.

c. RAWINS Local Master Input File – Standard Case

The following options in the Local Master Input File must be set for the standard RAWINS submittal.

NOBLND: Logical flag to activate the objective analysis. if NOBLND = .TRUE., the raw data are accessed and printed out, but the objective analysis *is not* performed. If NOBLND = .FALSE., the objective analysis *is* performed. NOBLND should be set to .F.

IFAC: Hollerith flag to select whether an entirely new analysis is to be generated from the raw observations (IFAC=6HANALYS), or whether first-guess fields are to be improved with the raw observations (IFAC=6HALTERS). If the first-guess fields are available and of acceptable quality, they should be used, since a good first-guess field generally results in a good final analysis. If, on the other hand, first-guess fields are of poor quality, or are not available, then new analyses may be generated.

IWTSCM: Hollerith flag to select the weighting scheme for the objective analysis (see Section IV.A.1). The options are:
- 6HCRESMN: (circular regions of influence),
- 6HELLIPS: (elliptical regions of influence),
- 6HBANANA: (banana-shaped regions of influence).

IWIND: Hollerith flag to select the surface first-guess wind field. The options are 6H1000MB to use the 1000 mb winds; or 6HSFCGEO to use the surface geostrophic winds, adjusted for friction, which were calculated in DATAGRID., The choice of IWIND depends on which field makes a better first guess; factors such as latitude and terrain elevation must be considered.

IWT: Logical flag to select the treatment of weights for the objective analysis (see Section IV.A.1).
- .TRUE.: weights are squared in the objective analysis.
- .FALSE.: weights are not squared in the objective analysis.

SMOOTH: Logical flag, set to .TRUE. to select smoothing of certain fields after the objective analysis.

IPOT: Not used in the current version of RAWINS. Ignore it.

UNIOBS: Logical flag, set to .TRUE. to select real-time observations from Unidata; set to .FALSE. to select archived observations.

ALRAWS: Logical flag indicating that all input sounding reports should be considered. Should always be set to .TRUE. in the current version of RAWINS; the .FALSE. option is not currently implemented.

NSTATN: Number of stations to include in RAWINS. NSTATN should always be set to 0, as this option is not currently implemented.

NSELIM: Used for Bogus submittal; should be set to .F. for standard submittal.

NBOGUS: Used for Bogus submittal; should be set to .F. for standard submittal.

KBOGUS: Used for Bogus submittal; should be set to .F. for standard submittal.

IUPPER: Not used in the current version of RAWINS. Ignore it.

ISFCS3: Logical flag; set it to .TRUE. if surface observations from 0300, 0900, 1500, and 2100 UTC should be accessed. For the standard RAWINS submittal, ISFCS3 is most likely set to .FALSE.

ISFCS6: Logical flag; set it to .TRUE. if land-station, ship, and buoy observations from 0000, 0600, 1200, and 1800 UTC should be accessed. For the standard RAWINS submittal, ISFCS6 is most likely set to .TRUE., but only the 0000 and 1200 UTC data are needed. If surface observations are not available or are not desired, ISFCS6 should be set to .FALSE.

F4D: Logical flag indicating whether or not FDDA fields should be generated. Set it to .FALSE. for the standard RAWINS submittal.

RWSUBM: Integer flag indicating whether this submittal of RAWINS is the only one expected (RWSUBM = 0), the first of two autobogus submittals (RWSUBM = 1), or the second of two autobogus submittals (RWSUBM = 2). This flag is set automatically by the RAWINS shell to the value of the script variable `Submit`.

BUDWGT: Determines the tolerance for the check of an observation's "error" against those of its neighbors (buddy check). A value of 1.0 gives an estimate of normal tolerance, a value greater than 1.0 increases the tolerance, while a value less than 1.0 decreases the tolerance. For example, a value of 2.0 would remove only highly suspect data, while a value of 0.5 would remove some good data along with all suspect data. A value of 0.0 deactivates the buddy check.

ISPRNT: Logical flag; set it to .TRUE. to print out the input surface observations. Should be set to .FALSE., or considerable output is generated.

ERRMXW: Value which determines the maximum difference allowed between the first-guess u and v wind components (interpolated to observations points) and the observations. For the ERRMXW, ERRMXT, and ERRMXP options (collectively referred to as the ERRMX options), there are no set guidelines for determining useful values. Smaller values will flag more observations. Generally, for poor first-guess fields, the ERRMX options should be set higher, so as to remove fewer observations which did not compare well to first-guess fields. Examination of the print output from a RAWINS submittal may give some indication of whether the ERRMX values were set appropriately. If too many values were flagged (and therefore not used), increase the ERRMX values and submit RAWINS again. If too few values were flagged, decrease the ERRMX values and submit RAWINS again. If the Buddy check is activated (option BUDWGT), then ERRMX options are set automatically within RAWINS.

ERRMXT: Value which determines the maximum difference allowed between the first-guess temperature field (interpolated to observation points) and the observations. See discussion of ERRMXW.

ERRMXP: Value which determines the maximum difference allowed between the first-guess sea-level pressure field (interpolated to observation points) and the observations. See discussion of ERRMXW.

IPLOT: Logical flag; set it to .TRUE. to generate Stüve plots of soundings.

IUINTVL: Time interval (hours) for sounding data. Should normally be set to 12.

d. RAWINS Script – Standard Case

The following script variables must be set for the standard RAWINS submittal:

`Submit`: Same as the Local Master Input File option RWSUBM. Set to 0 for the standard RAWINS submittal.

`INOBS`: Switch to indicate whether real-time (`INOBS = UNIOBS`) or archived (`INOBS = ARCHIVE`) observations are to be accessed.

`unidate`: The date for which real-time observations are to be retrieved (YYMMDDHH), if `INOBS` = `UNIOBS`.

`FDDAsw`: Switch to indicate whether FDDA fields are to be generated. Set to `NoFDDA` for the standard submittal.

`SFCsw`: Set to `SFC` if surface data are to be accessed, `NoSFC` if no surface data are to be accessed. Normally, if surface data are available they should be used.

`BOGUSsw`: Switch to indicate which bogus options are to be activated. Set to `NoBog` for the standard submittal.

`InRaobs`: Lists the MSS filenames of the files which hold the raw upper-air observations.

`InDatg`: Lists the MSS filename of the source of the input first-guess fields. `InDatg` is the MSS filename of either the DATAGRID output file or the model output file.

`InSfc3h`: Lists the MSS filenames of the files containing the surface observations from 0300, 0900, 1500, and 2100 UTC. These files also contain the ship and buoy observations from 0000, 0600, 1200, and 1800 UTC.

`InSfc6h`: Lists the MSS filenames of the files containing the surface observations from 0000, 0600, 1200, and 1800 UTC.

`OutRawAnl`: MSS filename to which the final analyses from RAWINS are written.

`OutRaobs`: MSS filename to which the corrected upper-air data are written.

`OutVPPlt`: MSS filename to which the sounding plot file is written.

2. FDDA case

The FDDA RAWINS submittal is similar to the Standard case, except that it generates an extra file containing the surface fields at standard upper-air times (0000 and 1200 UTC) and intermediate FDDA times (usually 0300, 0600, 0900, 1500, 1800, and 2100 UTC). Surface first-guess fields at these intermediate times are generated within RAWINS.

a. RAWINS input – FDDA case

The input files for a FDDA RAWINS submittal are the same as for the standard case, except that surface data *must* be accessed. The files containing the land-station data at 0300, 0900, 1500, and 2100 UTC are accessed beginning with FORTRAN unit 20. The files containing the land-station data at 0000, 0600, 1200, and 1800 UTC are accessed beginning with FORTRAN unit 25. The files containing the ship and buoy data at 0000, 0600, 1200, and 1800 UTC are accessed beginning with FORTRAN unit 30.

b. RAWINS output – FDDA case

The output from the FDDA submittal is the same as that from the standard submittal, with three additional files. The surface observations are written to FORTRAN unit 60, after all error checks and corrections have been performed. The upper-air observations at mandatory and nonmandatory levels are written to FORTRAN unit 61, after all error checks and corrections have been performed. Finally, all surface fields at the standard upper-air times and at the intermediate FDDA times are written to FORTRAN unit 39. The output to unit 39 for each time period consists of a header record followed by eight data records. The header record contains information which must be passed to other program units and information which defines the contents and format of the subsequent data records. The data records contain:

- Surface u (at dot points).
- Surface v (at dot points).
- Surface temperature (at cross points).
- Surface mixing ratio (at cross points).
- p^* (at cross points).
- Relative humidity (at cross points).
- Sea-level pressure (at cross points).
- A data density array, indicating the number of observations within each grid box.

Print output

The print output from the FDDA submittal is the same as that from the standard case, with the addition of samples of surface first-guess fields and final analyses at intermediate FDDA times.

c. RAWINS Local Master Input File – FDDA case

For the FDDA RAWINS submittal, the options to be set in the Local Master Input File are the same as those for the standard RAWINS submittal, with the following exceptions and additional options.

ISFCS3: Logical flag, set it to .TRUE. to indicate that surface observations at 0300, 0900, 1500, and 2100 are to be accessed. Probably set to .TRUE. for a FDDA submittal, unless the FDDA time interval (option INTF4D) is 6 h.

ISFCS6: Logical flag, set it to .TRUE. to indicate that surface observations at 0000, 0600, 1200, and 1800 UTC are to be accessed. Set to .TRUE. for a FDDA submittal. ISFCS6 or both ISFCS3 and ISFCS6 must be set to .TRUE. for a FDDA submittal.

F4D: Logical flag, set it to .TRUE. to indicate that this submittal is for a FDDA case.

INTF4D: Time interval (in hours) for the FDDA surface fields. Set to either 3 or 6 h, depending on the availability of surface observations.

LAGTEM: Logical flag to select the option for FDDA surface first-guess fields. LAGTEM = .FALSE. means that FDDA first-guess fields are generated by interpolating temporally from the analyzed surface fields available at the standard upper-air times. LAGTEM = .TRUE. means that a 3 h lag time is used for the current first-guess surface fields; *i.e.*, the analyzed fields from 3 h earlier are used as first-guess fields. LAGTEM may not be activated for a FDDA time interval greater than 3 h (option INTF4D).

d. RAWINS script – FDDA case

The script variables for the FDDA case are the same as those for the Standard case, with the following exceptions:

`FDDAsw`: Switch to indicate that FDDA fields are to be generated; should be set to `FDDA` for the FDDA submittal.

`Out4dSfc`: MSS filename to which the FDDA file output from RAWINS is written.

`OutSobs`: MSS filename to which the surface observations (for observations nudging) are written.

`OutUobs`: MSS filename to which the upper-air observations (for observations nudging) are written.

Also note that `InSfc3h` and `InSfc6h` must be set appropriately for the script to retrieve the needed surface files.

3. Bogus case

The bogus options allow the user to subjectively modify the observational data set to improve the quality of the final analyses. The Kbogus option allows the user to modify existing station reports in order to delete individual erroneous observations, insert bogus values for missing observations, or change the values of erroneous observations. The Nbogus options allow the user to delete entire station reports and insert bogus station reports. Bogus submittals can be tricky, so all output should be examined carefully for problems.

a. RAWINS input – Bogus case

The input for a Bogus RAWINS submittal is the same as that for the standard case, with the addition of two files, called `nbogus` and `kbogus` in the RAWINS script. The `nbogus` file contains data for the insertion of bogus station reports and the removal of specific station reports, and is accessed by RAWINS as FORTRAN unit 13. The `kbogus` file contains data for modifying existing observations, and is accessed by RAWINS as FORTRAN unit 12. Examples of the `nbogus` and `kbogus` files and details of creating these files are found in Appendices D and E.

b. RAWINS output – Bogus case

The output from a Bogus RAWINS submittal is the same as that from the Standard submittal. The bogus station reports may be printed out if the print option BOGPRT (in subroutine SETUP) is activated. Since bogus submittals can be confusing, all output should be examined carefully to insure that the bogus data are incorporated into the dataset as expected.

c. RAWINS Local Master Input File – Bogus case

For the Bogus RAWINS submittal, the options to be set in the Local Master Input File are the same as for the standard RAWINS submittal, with the following exceptions.

NSELIM(18): Logical flag, set to .TRUE. to indicate that specific station reports are to be removed from consideration for a particular time period (up to 18 time periods). The `nbogus` file must list the station identifiers (see Appendix D). Set it to .FALSE. if no specific stations are to be deleted.

NBOGUS(18): Logical flag, sit to .TRUE. if bogus station reports are to be included for a particular time period (up to 18 time periods). The `nbogus` file must contain the new station reports (see Appendix D).

KBOGUS(18): Logical flag, set to .TRUE. if modification to existing observations are to be made for a particular time period (up to 18 time periods). The `kbogus` file must contain the instructions for modifications (see Appendix E).

d. RAWINS script – Bogus case

The script variables for the Bogus case are the same as for the Standard case, with the following exception:

`BOGUSsw`: Switch to indicate which bogus options are activated. Set to `Konly` if existing observations are to be modified (Kbogus). Set to `Nonly` if bogus station reports are to be included (Nbogus). Set to `KandN` if both existing observations are to be modified, and bogus station reports are to be included.

4. Autobogus 1 case

The autobogus options allow the user to subjectively screen any suspect data that RAWINS finds. RAWINS checks the observations by comparing them to interpolated values of the first-guess fields at the observation points. Any observations that differ from the first-guess field by more that a certain maximum allowable difference (determined by the ERRMX values from the Local Master Input File) are considered suspect and are flagged. The buddy check is automatically deactivated. RAWINS performs the objective analysis, not considering any of the suspect data. The analyzed fields are then plotted out, and any suspect observations for each analysis are superimposed on the plotted fields. An autobogus correction file is output which contains all of the suspect observations and their station identifiers. The user examines the analyzed fields and the suspect data, determines if any of the suspect observations are in fact good observations, and sets a flag in the autobogus file for observations which are to be considered for the final analysis. This autobogus file is then used as input to the second autobogus submittal.

a. RAWINS input – Autobogus 1 case

The input for the Autobogus 1 submittal of RAWINS is the same as that for the Standard case.

b. RAWINS output – Autobogus 1 case

The output from the Autobogus 1 submittal is the same as that for the Standard case, with the addition of two files: the autobogus plot file and the autobogus correction file.

Autobogus plots

The Autobogus 1 case generates an NCAR-Graphics file, called `AB.PLT` in the RAWINS script. This file contains plots of the analyzed fields, with the suspect data overlaid. The user may examine these plots to determine if the suspect data should actually be ignored or should be considered in the objective analysis. It is also helpful to examine the sounding plots.

Autobogus correction file (output)

The autobogus output file is called `rawab.out` in the script, and is written out as FORTRAN unit 40. An example of such a file is shown in Appendix F. The file includes the station identifier and pressure level index of suspect observations, the values of the possibly erroneous data, and a flag (set to "F" in the output). This file is to be edited and used as input for the second Autobogus submittal. See the discussion of Autobogus 2 input for further details.

Print output

The print output is the same as the standard case, with the addition of a printout of the autobogus file values.

c. RAWINS Local Master Input File – Autobogus 1 case

For the Autobogus 1 submittal, the options to be set in the Local Master Input File are the same as for the Standard RAWINS submittal, with the following exceptions and additional options:

RWSUBM: Must be set to 1, to indicate the first of two submittals. This is set automatically by the shell script.

ERRMXW: Should be set to a lower tolerance than for the standard case, so that more suspect observations are flagged.

ERRMXT: Should be set to a lower tolerance than for the standard case, so that more suspect observations are flagged.

ERRMXP: Should be set to a lower tolerance than for the standard case, so that more suspect observations are flagged.

ABFLAG: Logical flag, set to .TRUE. to produce autobogus plots.

ABOVER: Logical flag, set to .TRUE. to superimpose suspect observations on the autobogus plots.

d. RAWINS script – Autobogus 1 case

The script variables for the Autobogus 1 case are the same as those for the Standard case, with the following exceptions:

`Submit`: Must be set to 1, to indicate that this submittal is the first of two autobogus submittals.

`MapCol`: Flag indicating whether black-and-white (`BW`) or color (`CO`) maps should be generated.

`ConCol`: Flag indicating whether black-and-white (`BW`) or color (`CO`) contours should be drawn on the maps.

`OutABPlt`: MSS filename to which the autobogus plots are written.

`OutAB`: MSS filename to which the autobogus file is written.

5. Autobogus 2 case

The second of the two autobogus submittals may be run only after the Autobogus 1 submittal. The second submittal again reads all observations from the input volumes. No error-checking is performed on any input files in the Autobogus 2 submittal; the error-checking has already been performed by the first submittal. When the fields are analyzed, the suspect observations which the user confirmed were erroneous are not considered.

a. RAWINS input – Autobogus 2 case

The input for the Autobogus 2 submittal of RAWINS is the same as that for the Standard case, with the addition of a file containing the autobogus corrections. This file, called `autobog` in the RAWINS script, is accessed by the program as FORTRAN unit 10. The autobogus correction file was originally output from the Autobogus 1 submittal, and will have been edited by the user before being used as input to the Autobogus 2 submittal. An example of an autobogus correction file may be found in Appendix F.

When the Autobogus correction file was written by the Autobogus 1 submittal, the flag for each observation in the file was set to "F", indicating erroneous data. For each suspect observation which the user subsequently decides is actually correct data, the flag in the autobogus file for that observation is changed from "F" to "T". The flag for any observations that the user determines should be ignored remains set to "F".

The user may also change the value of the flagged observation, for inclusion of a new value in the final analysis. In that case, the "F" flag should be changed to "B", and the bogus value should be substituted for the erroneous value in the file.

b. RAWINS output – Autobogus 2 case

The output from the Autobogus 2 submittal is the same as that from the standard submittal, with the exception of the sounding plots, which have already been created by the Autobogus 1 submittal.

Print output

The print output from the Autobogus 2 submittal is the same as that from the standard case, with the exception of erroneous observations.

c. RAWINS Local Master Input File – Autobogus 2 case

For the Autobogus 2 submittal, the options so be set in the Local Master Input File must be the same as for the Autobogus 1 submittal, with the following exceptions:

RWSUBM: Must be set to 2, to indicate that this submittal is the second of two autobogus submittals. RWSUBM is set automatically by the RAWINS script to the value of script variable `Submit`.

ABFLAG: Should be set to .FALSE.

ABOVER: Should be set to .FALSE.

d. RAWINS script – Autobogus 2 case

The script variables for the Autobogus 2 case are the same as for the Standard case, with the following exceptions:

`Submit`: Must be set to 2 to indicate that this submittal is the second of two RAWINS submittals.

`OutAB`: MSS file to which the user-modified `autobog` file is written.

6. Summary

For easy reference, the complete lists of input files, output files, Local Master Input File options, and script variables are given in Tables 4.1 through 4.3.

Table 4.1 RAWINS Local Master Input File

Variable	Options and description	
NOBLEND	.F.	Objective analysis *is* performed
	.T.	Objective analysis *is not* performed
IFAC	6HANALYS	Generate first-guess fields from observations
	6HALTERS	Use first-guess fields supplied to the program
IWTSCM	6HCRESMN	Use circular radius of influence for objective analysis
	6HELLIPS	Use elliptical radius of influence
	6HBANANA	Use banana-shaped radius of influence
IWIND	6H1000MB	Use 1000 mb winds as first-guess surface winds
	6HSFCGEO	Use the adjusted surface geostrophic winds
IWT	.T.	Weights are squared in the objective analysis
	.F.	Weights are not squared in the objective analysis
SMOOTH	.T.	T, p_o, and RH fields are smoothed after objective analysis
	.F.	Fields are not smoothed
IPOT	Unused. Ignore it	
UNIOBS	.T.	Use real-time observations from Unidata
	.F.	Use historical observations
RWSUBM	0	The only RAWINS submittal (*i.e.* no autobogus)
	1	The first of two RAWINS submittals (*i.e.* autobogus 1)
	2	The second of two RAWINS submittals (*i.e.* autobogus 2)
IUINTVL	Time interval (hours) for upper-air data. Normally 12	
ALRAWS	Option not incorporated. Must be set to T	
NSTATN	Option not incorporated. Must be set to 0	
IUPPER	Unused. Ignore it	
ISFCS3	.T.	Acquire surface data from 0300, 0900, 1500, and 2100 UTC
	.F.	Do not acquire surface data from 0300, 0900, 1500, 2100 UTC
ISFCS6	.T.	Acquire surface data from 0000, 0600, 1200, and 1800 UTC
	.F.	Do not acquire surface data from 0000, 0600, 1200, and 1800 UTC
F4D	.T.	Create surface FDDA files at intermediate times
	.F.	Do not create surface FDDA files
INTF4D	Time interval (hours) for surface FDDA times. Either 3 or 6	
LAGTEM	.T.	Use a 3 hr lag time for intermediate FDDA first-guess fields
	.F.	Interpolate temporally for intermediate FDDA first-guess fields
NSELIM	.T.	Delete specified stations for a given time period
	.F.	Do not delete any stations for a given time period
NBOGUS	.T.	Include bogus station reports
	.F.	Do not include bogus station reports
KBOGUS	.T.	Include individual bogus points
	.F.	Do not include individual bogus points
ISPRNT	.T.	Print surface observations
	.F.	Do not print surface observations
ERRMXW	Maximum difference allowed (first-guess − obs) for wind values	
ERRMXT	Maximum difference allowed (first-guess − obs) for temperature values	
ERRMXP	Maximum difference allowed (first-guess − obs) for pressure values	
BUDWGT	Weighting parameter that varies the tolerance of the buddy check	
	$= 1$	Estimate of normal tolerance
	< 1	Decrease tolerance (remove more data)
	> 1	Increase tolerance (remove fewer data)
IPLOT	.T.	Plot soundings
	.F.	Do not plot soundings
ABFLAG	.T.	Plot gridded analyses for autobogus
	.F.	Do not plot gridded analyses
ABOVER	.T.	Overlay suspect observations on plots for autobogus
	.F.	Do not overlay suspect observations on plots

Table 4.2a Input files for program RAWINS

Input File	Description
Unit 4:	First-guess fields
Unit 10:	Autobogus correction file (input)
Unit 12:	Kbogus file (correct existing observations)
Unit 13:	Nbogus file (bogus station reports)
Unit 14:	Local Master Input File
Units 15 - 19:	Upper-air observations
Units 20 - 24:	Surface observations (0300, 0900, 1500, 2100 UTC
Units 25 - 29:	Surface observations (0000, 0600, 1200, 1800 UTC
Units 30 - 34:	Ship and buoy observations

Table 4.2b Output files from program RAWINS

Input File	Description
Unit 2:	Output analyses
Unit 11:	Upper-air observations
Unit 39:	Surface FDDA analyses
Unit 40:	Autobogus correction file (output)
Unit 60:	Surface observations for FDDA
Unit 61:	Upper-air observations for FDDA
SND.PLT:	Sounding plot file
AB.PLT:	Autobogus plot file

Table 4.3 RAWINS Script Variables

Variable	Options (if any) and description	
SUBMIT	0	This is the only RAWINS submittal (no autobogus)
	1	This is the first of two RAWINS submittals (Autobogus 1)
	1	This is the second of two RAWINS submittals (Autobogus 2)
INOBS	ARCHIVE	Use historical observations
	UNIOBS	Use real-time observations from Unidata
FDDAsw	FDDA	Analyze surface fields at FDDA times
	NoFDDA	Do not analyze surface fields at FDDA times
SFCsw	SFC	Access and analyze surface observations
	NoSFC	Do not access and analyze surface observations
BOGUSsw	NoBOG	No bogus options are used
	Konly	Kbogus options (individual points) are used
	Nonly	Nbogus options (station reports) are used
	KandN	Both Kbogus and Nbogus options are used
MapCol	BW	Autobogus maps are plotted on black-and-white fiche
	CO	Autobogus maps are plotted on color film
ConCol	BW	Black-and-white contours are plotted
	CO	Color contours are plotted
InRaobs	MSS filename(s) of the upper-air observations	
InDatg	MSS filename of the volume of first-guess fields	
InSfc3h	MSS filename(s) of the 0300, 0900, 1500, and 2100 UTC surface observations. These files also contain ship and buoy data at 0000, 0600, 1200, and 1800 UTC.	
InSfc6h	MSS filename(s) of the 0000, 0600, 1200, and 1800 UTC surface observations	
unidate	Eight-character date/time of the real-time observations	
OutRawAnl	MSS filename of file to hold RAWINS gridded analyses	
OutRaobs	MSS filename of file to hold input observations	
OutVPPlt	MSS filename of file to hold sounding plots	
Out4dSfc	MSS filename of file to hold output surface analyses at FDDA times	
OutABPlt	MSS filename of file to hold plots of analyzed fields (from Autobogus 1)	
OutAB	MSS filename to which autobogus listing is written (from Autobogus 1)	

C. RAWINS Code

The RAWINS code is comprised of a main program and over seventy-five subprograms. The main program, called RAWINS, manages the subroutines which perform the data ingest, objective quality checks, objective analysis, plotting, output, and other steps in the execution of the program. Many large blocks of memory are shared among the various program units. The following three sections discuss the memory structure, main program, and subprograms of RAWINS.

1. Memory structure

Like DATAGRID and most other programs of the PSU/NCAR Modeling System, RAWINS uses variable parameterized dimensions. Four of the PARAMETER statements may be easily modified by the user in the RAWINS script:

PARAMETER (IMX = ii, JMX = jj, LMX = ll)

PARAMETER (IMXC = ic, JMXC = jc)

PARAMETER (IRB = nb)

PARAMETER (IRS = ns)

The first two of these four statements are common to several of the programs of the modeling system; IMX and JMX and must be the same among all the programs for one case. LMX may vary among some programs. The second two parameter statements are used only in RAWINS. The parameters set by these four statements are:

IMX, JMX:	Dimensions of expanded grid.
LMX:	The total number of vertical levels, mandatory and nonmandatory.
IMXN:	The number of nested-grid gridpoints in the I (y) direction.
JMXN:	The number of nested-grid gridpoints in the J (x) direction.

IMXC, JMXC: The dimensions of the unexpanded grid.

IRB: The maximum number of upper-air reports that may be processed.

IRS: The maximum number of surface reports that may be processed, + IRB.

Two other internal PARAMETER statements are shared among many of the RAWINS program modules:

PARAMETER (JCR1 = IMX * JMX * (6 * LMX + 20) + (IRB / 2) * 1000 + 2000)

PARAMETER (IHS = JMX + 1, IMKS = IMX * (JMX + 1) + 1, ICR1 = JCR1)

These PARAMETER statements most likely do not need to be modified by the user. However, the JCR1 parameter is an empirical approximation for the size of the D1 storage array; it does not always work. In this case, the JCR1 parameter may be set to some large constant, perhaps on the order of 1.5 million.

Most of the in-core storage is used by several key arrays. Some of these are scratch arrays, used by many different variables during the program's execution. The arrays are discussed in detail below:

(a) D1(ICR1) – A large storage array which, over the course of the program's execution, holds all of the upper-air observations, surface analyses, and pressure-level analyses.

(b) DATAH, CORRH, both dimensioned by (ID1, ID2, ID3) – These arrays hold upper-air sounding data of up to ID1 = 5 variable types. The variables represented by index ID1 are:
    1: u
    2: v
    3: T
    4: RH
    5: height.
    A maximum of ID2 = 55 vertical levels, either mandatory or significant levels, can be stored for each station. The arrays can hold up to ID3 = IRB stations, where IRB is the maximum number of upper-air stations, as defined in the PARAMETER statements. DATAH holds the mandatory-level data, and CORRH holds significant-level data.

(c) DATAS(5, 2, IRS) – This array holds the surface station data of up to 5 variable types. The data from a maximum of IRS stations can be stored, where IRS is defined in the PARAMETER statements. When the second dimension of DATAS is equal to 1, DATAS holds latitude, longitude, and elevation data for each station; when the second dimension is equal to 2, DATAS holds the meteorological data from each station.

(d) HORZH(IH1, IH2, IH3) – This array holds up to IH3 = 5 horizontal slabs of data (IH1 = IMX, IH2 = JMX+1) on the mesoscale grid. The HORZH array is used mostly for working space.

(e) SLAB1, SLAB2, SLAB3, COR, NS, SUM, SUM2, LAND, SNOW, SFCPR, SCR, PDIF12, all dimensioned by (IMX, JMX) – These arrays each hold one horizontal slab of data on the mesoscale grid for a particular variable. Most of these arrays are defined for general work space; LAND, SNOW, SFCPR, and PDIF12 are defined for specific variables.

(f) SLAB4, LANDC, both dimensioned by (IMXC, JMXC) – These arrays hold one horizontal slab of data on the non-expanded mesoscale grid.

(g) RSOND(5, IRS), BUFD(5, IRS), BUFUPR(5, IRB) – Buffer arrays, providing working space to hold observational data from the D1 storage array. BUFUPR is used in transferring the data from D1 to RSOND and BUFD.

(h) ISTNUM(3, IRS), RANG(3, IRS), IFOUND(3, IRS), IKLAT(2, IRS), IKLON(2, IRS), IKELEV(2, IRS) – These arrays hold accounting information about the surface and upper-air station reports. ISTNUM holds the station identifiers; IFOUND holds a flag identifying the reports as real or bogus; RANG holds an angle used for converting wind data from meteorological coordinates to map coordinates; IKLAT, IKLON, and IKELEV hold the station latitudes, longitudes, and elevations.

RAWINS has many named common blocks used to pass specific categories of variables among program units. They are:

/A/,/B/,/C/: Hold miscellaneous variables, 1-d arrays, and 2-d arrays.

/D/: Holds mostly the identification codes for the meteorological fields.

/MEMCOR/: Holds variables and arrays pertaining to memory storage, including the large storage array D1.

/HEDMIF/: Holds variables extracted from the DATAGRID record headers, and variables written to the RAWINS record headers.

/WT/: Holds arrays for the objective analysis.

/MXER/: Holds information concerning observations removed by the maximum-error check.

/BUD/: Holds information concerning observations removed by the buddy check.

/ADP/: Holds variables used for the reading and unpacking the archived observational data.

/SFCEL/: Holds elevations of the surface stations.

/SCA/: Holds the number of scans for the objective analysis.

/XYLL/: Holds variables used in the conversion from x-y coordinates to latitude-longitude coordinates.

/XYLLON/: Holds variables used in the conversion from x-y coordinates to latitude-longitude coordinates.

/LAMSTF/: Holds variables used in conversion from x-y coordinates to latitude-longitude coordinates.

/MAP4AB/: Holds variables used for plotting the autobogus maps.

/RECINT/: Holds variables used for plotting the autobogus maps.

/PTS/: Holds variables used for plotting the autobogus maps.

Some temporary peripheral storage is also used. Data are stored on units 45, 98, and 99. Unit 45 temporarily holds surface first-guess fields (interpolated temporally from standard times) at intermediate FDDA times. Units 98 and 99 temporarily store data for the autobogus output plots.

RAWINS was originally designed to execute on computer systems with smaller memory capacities than are commonly available today. The memory structure was designed such that the in-core storage array D1 could be easily replaced by a peripheral storage device such as a disk. Most of the major arrays could be written to the external device during the program's execution, and transferred to in-core storage only as needed. Horizontal slabs of data were broken into strips, so that only portions of slabs needed to be accessed. In-core storage requirements could then be minimized. Due to the availability of large-memory machines, the peripheral storage and stripping features are now obsolete. The structures that manage the peripheral storage and strip the data, however, remain in the code. Most noticeable are the portions of the code that manage the writing to and reading from the D1 storage array: subroutines WDISK, RDISK, SLBFIL, FILSLB, HRZFIL, and SAVFIL.

2. Main Program Design

The main program, called RAWINS, manages the subroutines which perform the data ingest, objective quality checks, objective analysis, output, and other steps in the execution of the program. Most of the processing occurs in two main DO-loops: the first loop (DO-loop 35) cycles through the number of standard time periods (*i.e.*, IFILES from the Common Master Input File); the second loop (DO-loop 45) cycles through the number of FDDA times in each standard time period. The flow structure for the main program is shown in Fig. 4.1.

The execution of RAWINS may be broken down into several smaller tasks, each of which is executed by a number of subroutines. The tasks are described below; the subroutines related to the tasks are listed in parentheses below each task's name.

Miscellaneous Initialization Task

(SETUP, SETUPS)

Initialization of variables occurs throughout the program, but much of it is accomplished in two subroutines, SETUP and SETUPS. Subroutine SETUP is called once during the execution of the program, and initializes a number of variables which are used as constants throughout the program. Subroutine SETUPS, on the other hand, is called once for each time period processed (standard and intermediate) during program execution. SETUPS initializes and re-initializes variables which must be reset for each time period.

Fig. 4.1 Flow structure for program RAWINS

PROGRAM RAWINS

- Read the Local Master Input File.
- Initialize miscellaneous variables.
- Read the header record from the DATAGRID output volume (containing first-guess fields).
- Call subroutine **HEDRIN** to retrieve the parameters from arrays MIF, MRF, and MLF, which have been read from the DATAGRID output volume.
- Do 35 IFILE = 1, IFILES (Loop over all standard time periods).
    - Read the header record from the DATAGRID output (except for the first time through loop 35, as the first header record has already been read.
    - Calculate the time interval for FDDA fields, and the number of FDDA fields to be generated for each standard time period.
    - DO 45 IFDA = 1, IFDATL (Loop over all FDDA times in the time period)
        - Call subroutine **SETUP** to initialize miscellaneous variables and check consistency among parameters. SETUP is called only once during the program's execution.
        - Call subroutine **SETUPS** to reset certain memory variables and arrays.
        - If the first-guess fields need to be created for FDDA, then
            - Call subroutine **INHRZ2** to read surface first-guess fields at 0000 and 1200 UTC, interpolate temporally to the FDDA times, and write them to the temporary volume FORTRAN unit 45. INHRZ2 is called only once during the program's execution.
        - Endif
        - If a standard upper-air time is being processed (*i.e.*, IFDA = 1), then
            - Call subroutine **INHRZ1** to read the surface and upper-air fields for the particular time period being processed.
        - Endif.
        - Call subroutine **GETRAW** to read the sounding data, surface data, and bogus station reports from the various input files.
        - Call subroutine **BOGPTS** to correct specific observations in existing surface station reports (kbogus or autobogus).
        - Call subroutine **SFCBLN** to perform the objective analysis for the surface fields.
        - If a standard upper-air time is being processed (*i.e.*, IFDA = 1), then

          - Call subroutine **PSFC** to calculate surface pressure from the final analysis of sea-level pressure.
          - Calculate the difference between sea-level pressure and surface pressure.
          - Call Subroutine **SIGDAT** to interpolate upper-air observations to nonmandatory levels and create sounding plots.
          - Loop over all five variables at each nonmandatory pressure level
            - Call subroutine **NEWPLV** to generate first-guess fields at nonmandatory levels. NEWPLV is called once for each variable at each nonmandatory level.
          - Endloop.
          - Call subroutine **BOGPTS** to correct specific observations in existing sounding reports.
          - Call subroutine **BLEND** to perform the pressure-level analyses
          - Call subroutine **OUTAP** to write the final analyses to the output volume (FORTRAN unit 2).
        - Endif.
        - Print the observations that have been removed by the buddy check.
        - Print the observations that have been removed by the ERRMX check.
        - Call subroutine **OUTFDA** to write the FDDA output file (FORTRAN unit 39).
    - Endloop 45 (FDDA loop)
- Endloop 35 (Standard-time loop)
- Read and print the autobogus file.
- Call subroutine **PLOTAB** to generate the autobogus plots.

End PROGRAM RAWINS

Fig 4.1: Continued

Header Record Task

(HEDRIN, HEDROT)

The header record, written out before each set of output files for each time period, includes three important arrays: MIF, MRF, and MLF. MIF, MRF, and MLF contain miscellaneous integer, real, and logical values (respectively) which are passed among the various components of the PSU/NCAR modeling system. In RAWINS, the arrays are initialized by reading the DATAGRID output.

Subroutine HEDRIN extracts data from the MIF, MRF, and MLF arrays so that the data may be used in RAWINS. Subroutine HEDROT modifies certain data and adds data to the MIF, MRF, and MLF arrays, so that pertinent information may be passed from RAWINS to subsequent components of the modeling system.

Read First-guess Fields

(INHRZ1, INHRZ2, RTAPE)

Three subroutines perform the task of reading first-guess fields from FORTRAN unit 4: subroutines INHRZ1, INHRZ2, and RTAPE.

INHRZ1 reads terrestrial and meteorological fields from unit 4. For most of the fields, INHRZ1 first calls RTAPE to read the field from the file to a two-dimensional array, and then calls SLBFIL to transfer the field to the D1 storage array (See D1 Storage Task, below). Subroutine RTAPE merely executes a READ statement; RTAPE also contains an option for printing a sample of the field (using subroutine OUTPT, described later under the Output Task).

INHRZ2 reads the first-guess fields from unit 4 directly (without calling RTAPE) to create intermediate FDDA first-guess fields. All fields except surface first-guess fields are immediately discarded, and surface first-guess fields at intermediate FDDA times are generated by temporal interpolation from standard times.

Retrieval of Raw Observations

(GETRAW, GETUNIOB, SFSCVT, SFUAOB, RDADP, PROUPR, PROSFC, MANADP, SIGADP, WPPADP, WZZADP, SFCADP, LOGCMP, NTENS, UNIUP, UNISF, INACCT, SAVSTN)

The routines which perform the reading, unpacking, and organizing of the archived surface and upper-air data sets are all controlled by subroutine GETRAW. If real-time observations have been requested, GETRAW calls GETUNIOB to access the observations and write them to an external file. GETUNIOB in turn calls SFSCVT for surface observations, and SFUAOB for upper-air observations. GETRAW then proceeds to the main data-processing cycle.

For the data processing, GETRAW calls RDADP to read a packed surface or upper-air station report from the archives (or from the file of real-time observations), and retrieve header variables indicating the time, location, type etc. of report. The report itself is read into a buffer array.

If the report for an upper-air data, GETRAW calls PROUPR. PROUPR in turn calls subroutines MANADP, SIGADP, WPPADP, and WZZADP for historical data (or UNIUP for real-time observations), to extract data from the packed record. The four subroutines MANADP, SIGADP, WPPADP, and WZZADP are similar in their processing. Each uses functions LOGCMP and NTENS to extract the data. Subroutine UNIUP, on the other hand, reads data which have been preprocessed by subroutine GETUNIOB.

If the report is for surface data, GETRAW calls PROSFC, which in turn calls SFCADP for historical data (or UNISF for real-time observations) to extract the data from the packed record. SFCADP uses functions LOGCMP and NTENS to extract the data. UNISF reads data which have been preprocessed by subroutine GETUNIOB.

Once the data have been retrieved in either PROUPR or PROSFC, INACCT is called to update accounting variables for station reports, and SAVSTN is called to perform miscellaneous transformations of the data to make the data usable in the objective analysis procedure.

Finally, GETRAW calls FILSLB (discussed later in the D1 Storage Task) to transfer observations to the D1 storage array.

GETRAW also manages subroutines PROBGU and PROBGS to process bogus data (See Bogus Data task).

Bogus Data Task

(BOGPTS, PROBGU, PROBGS)

There are several features related to the processing of bogus data. The bogus data may be located in up to three separate files: the kbogus file, which contains modifications to existing station reports; the nbogus file, which contains entire bogus station reports as well as station identifiers for reports which are to be ignored; and the autobogus correction file (input), which contains the subjective instructions for keeping, replacing, or deleting observations found to be suspect by the ERRMX check in subroutine GETERR (Data Quality Control Task).

The bogus corrections to existing reports are incorporated into the observational data set by subroutine BOGPTS. BOGPTS reads and applies the modifications as specified in the kbogus and autobogus correction files. The bogus station reports (in the nbogus file) are incorporated into the data set by subroutines PROBGU (for upper-air reports) and PROBGS (for surface reports) in a manner similar to the real reports processed in GETRAW (see Retrieval of Raw Observations Task).

The station reports which are to be deleted from the data set are read from the nbogus file by subroutine SETUPS and removed from consideration by subroutine GETRAW.

Surface Pressure Task

(PSFC, SEAPRS)

Two subroutines convert between surface pressure and sea-level pressure. Subroutine PSFC derives surface pressure from the final analyses of sea-level pressure. Surface pressure is used for interpolating data to new pressure levels and for adjusting superadiabatic lapse rates. Subroutine SEAPRS converts upper-air station reports of surface pressure to sea-level pressure. This conversion is performed because the pressure field is objectively analyzed not as surface pressure, but as the smoother, more spatially correlated field of sea-level pressure. SEAPRS also converts bogus sea-level pressure reports to surface pressure.

Nonmandatory-level Interpolation Task

(NEWPLV, SIGDAT)

The first-guess fields are available from the DATAGRID output only at the surface and at mandatory levels. However, the user may specify that analyses are to be performed at certain nonmandatory levels as well as mandatory levels. Subroutine NEWPLV is called to create first-guess fields at these nonmandatory levels by interpolating from bracketing mandatory levels. NEWPLV is called once for each variable field at each nonmandatory level.

The upper-air stations report data at mandatory levels and significant levels. For mandatory-level analyses, observations are generally available at the level to be analyzed. The significant-level data, however, are not reported at the same levels at all upper-air stations. For nonmandatory level analyses, the observations are most likely not available at the levels analyzed. For each upper-air report, the data must be interpolated from bracketing significant and mandatory levels to the specified nonmandatory levels. Subroutine SIGDAT performs this task. SIGDAT also performs some quality-control on each upper-air report (see Data Quality Control Task). Subroutine SIGDAT is called once during each time period to interpolate all upper-air reports for that time to nonmandatory levels.

Data Quality Control Task

(SORT, BUDDY, GETERR)

RAWINS has a number of features to objectively examine the data and remove or modify suspect observations. These objective checks are performed within various subroutines.

Each sounding report is examined by subroutine SORT, which checks the temperature and wind profiles for unrealistic spikes, and attempts to correct any bad data in the wind reports by altering the report. Temperature data which cause spikes and any wind data which cause spikes and cannot be corrected are deleted. An attempt to replace the deleted value with a vertically interpolated value is later made in subroutine SIGDAT.

Subroutine BUDDY compares the difference value (observed – first-guess) at each observation location to the values at nearby stations. Any observation for which the difference value exceeds a weighted average of the difference values at nearby stations is flagged as erroneous and ignored during the objective analysis.

Subroutine GETERR, an important step in the Objective Analysis Task also performs an important quality-control task. In GETERR, each observation is compared directly to the first-guess fields. Any observation for which the difference value (observed − first-guess) exceeds certain maximum allowable differences (the ERRMX parameters) are flagged as erroneous and ignored for the objective analysis.

The ERRMX parameters, ERRMXW, ERRMXT, and ERRMXP, have their base values set by the user in the RAWINS Local Master Input File, and vary according to time, level, and source of the first-guess fields, as described in Tables 4.4 through 4.6.

Objective Analysis Task

(BLEND, SFCBLN, SETANA, GETERR, ANALMN, SMTHER, BARNES)

Seven subroutines deal directly with the objective analysis. Subroutines BLEND (for pressure levels) and SFCBLN (for surface) are the top-level routines. These two routines cycle through all of the two-dimensional fields which must be analyzed. BLEND and SFCBLN access the first-guess fields; each calls subroutine SETANA.

SETANA manages two other routines. Its basic tasks are to determine the number of scans needed for the successive Cressman-type scheme and to execute subroutines GETERR and ANALMN for each scan. SETANA is called for each field, at each level.

Subroutine GETERR calculates the difference value (observation − first-guess) at each observation location. GETERR performs some quality-control tasks as well (See Quality Control Task) so that the analysis may ignore suspect observations.

Subroutine ANALMN uses the difference values calculated by GETERR to calculate correction values (α_{ij} of equations 4.3 through 4.6) at each gridpoint. ANALMN applies the correction values to the current guess to create an improved field.

Subroutine SMTHER, which is not in the hierarchy below BLEND and SFCBLN, smoothes certain fields after the final analysis, in order to dampen short-wavelength components.

Table 4.4 ERRMXW Adjustments

Case	Adjustments
Standard time, temporal interpolation in DATAGRID to generate missing first-guess fields	ERRMXW = ERRMXW $\times$ 1.5
FDDA intermediate time, first pass through objective analysis	ERRMXW = ERRMXW $\times$ 1.33
p > 1000 mb	No further adjustments
1000 > p > 500 mb	ERRMXW = ERRMXW $\times$ 1.25
p < 500 mb	ERRMXW = ERRMXW $\times$ 1.5

ERRMXW adjustments. Initial value of ERRMXW from the Local Master Input File. If BUDWGT is used, ERRMXW = 14 initially.

Table 4.5 ERRMXT Adjustments

	Case	Adjustments
Surface Station	FDDA intermediate time, first pass through objective analysis	ERRMXT = ERRMXT $\times$ 1.33
	Buoy or Ship	ERRMXT = ERRMXT $\times$ 1.5
	Land station, near sunrise; Elevation $\leq$ 500 m	ERRMXT = ERRMXT $\times$ 1.25
	Land station, near sunrise; Elevation>500 m	ERRMXT = ERRMXT $\times$ 1.75
	Land station, near sunset	ERRMXT = ERRMXT $\times$ 1.5
	Any other surface station	No further adjustments
Upper-air station	$p < 700$ mb	ERRMXT = ERRMXT $\times$ 0.85
	$700 < p < 300, p = 200, p = 100$ mb	ERRMXT = ERRMXT $\times$ 0.65
	ECMWF first-guess fields, by vertical interpolation (*i.e.*, $p = 150$, 250, or 400 mb)	No further adjustments

ERRMXT adjustments. Initial value of ERRMXT from the Local Master Input File. If BUDWGT is used, ERRMXT = 15 before adjustments.

Table 4.6 ERRMXP Adjustments

Case	Adjustments
Standard time, temporal interpolation in DATAGRID to generate missing first-guess fields	ERRMXP = ERRMXP × 1.5
FDDA intermediate time, first pass through objective analysis	ERRMXP = ERRMXP × 1.33
All others	No further adjustments

ERRMXP adjustments. Initial value of ERRMXP from the Local Master Input File. If BUDWGT is used, ERRMXP = 8 before adjustments.

Subroutine SFCBLN has provisions (currently bypassed in the subroutine code) for creating surface first-guess fields from observations, using a Barnes analysis (Barnes; 1964, 1973) in subroutine BARNES. If the user wishes to activate the Barnes analysis, the SFCBLN code must be modified.

Output Task

(OUTPT, OUTAP, WTAPE, OUTFDA, PLOTAB, PLSOND)

The primary output from RAWINS is performed by six subroutines. The output is of three different types: print output of samples of the two-dimensional analyses (subroutine OUTPT), output of plots (subroutines PLOTAB and PLSOND), and output to permanent files (subroutines OUTAP, WTAPE, and OUTFDA).

Subroutine OUTPT performs the basic task of printing out a sample of a horizontal slab of data. OUTPT is called at most levels for fields used as first guesses for the objective analyses, and also for final analysis.

Subroutine PLSOND generates a Stüve plot for each input upper-air station (*e.g.*, Fig. 4.2). Subroutine PLOTAB generates horizontal plots of the final analyses of wind, temperature, and sea-level pressure, with suspect observations (not used in the analyses) superimposed (*e.g.*, Fig. 4.3). PLOTAB and PLSOND each call many other subroutines to accomplish their tasks using NCAR-Graphics packages. These other subroutines are not discussed in this document.

Subroutine OUTAP is called to write the final analyses and other fields (at standard

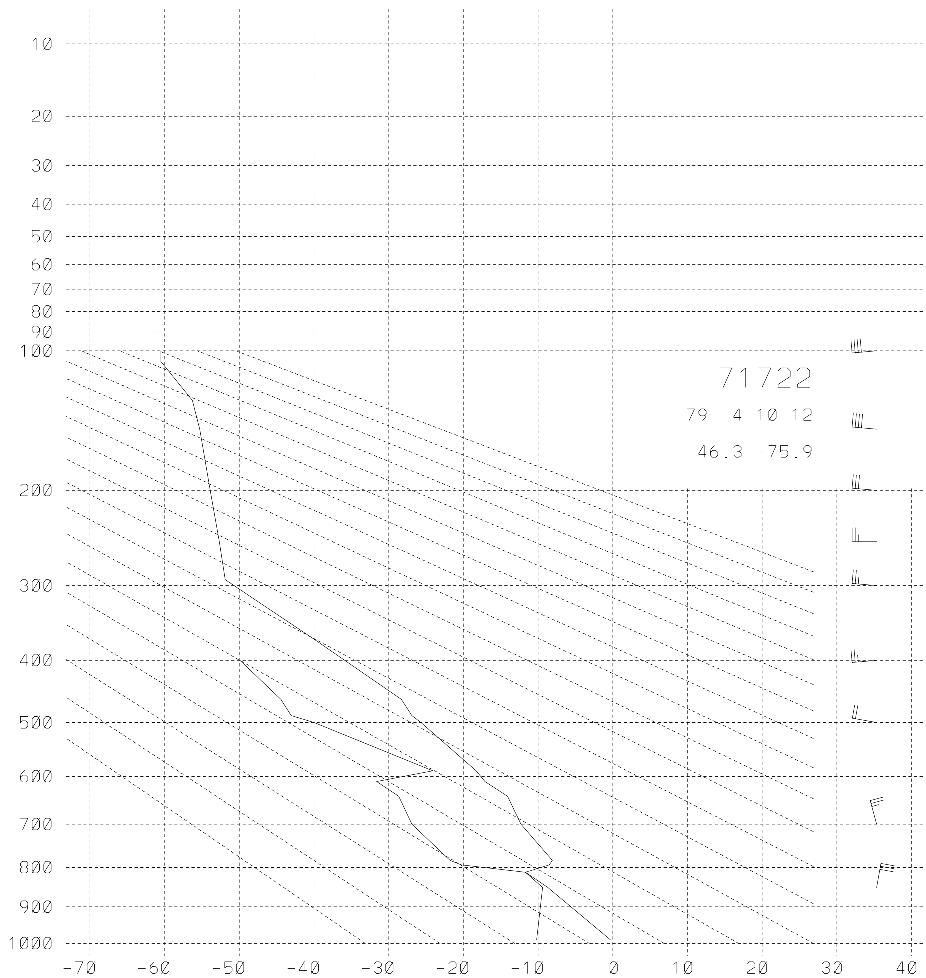

Fig. 4.2: Sample sounding plot (Stüve diagram), showing vertical distribution of temperature, dewpoint, and wind. The sloping dashed lines are lines of constant potential temperature. The vertical pressure coordinate is scaled by p^{κ}, where $\kappa = \frac{2}{7}$. Plots are generated by RAWINS and written to the file SND.PLT.

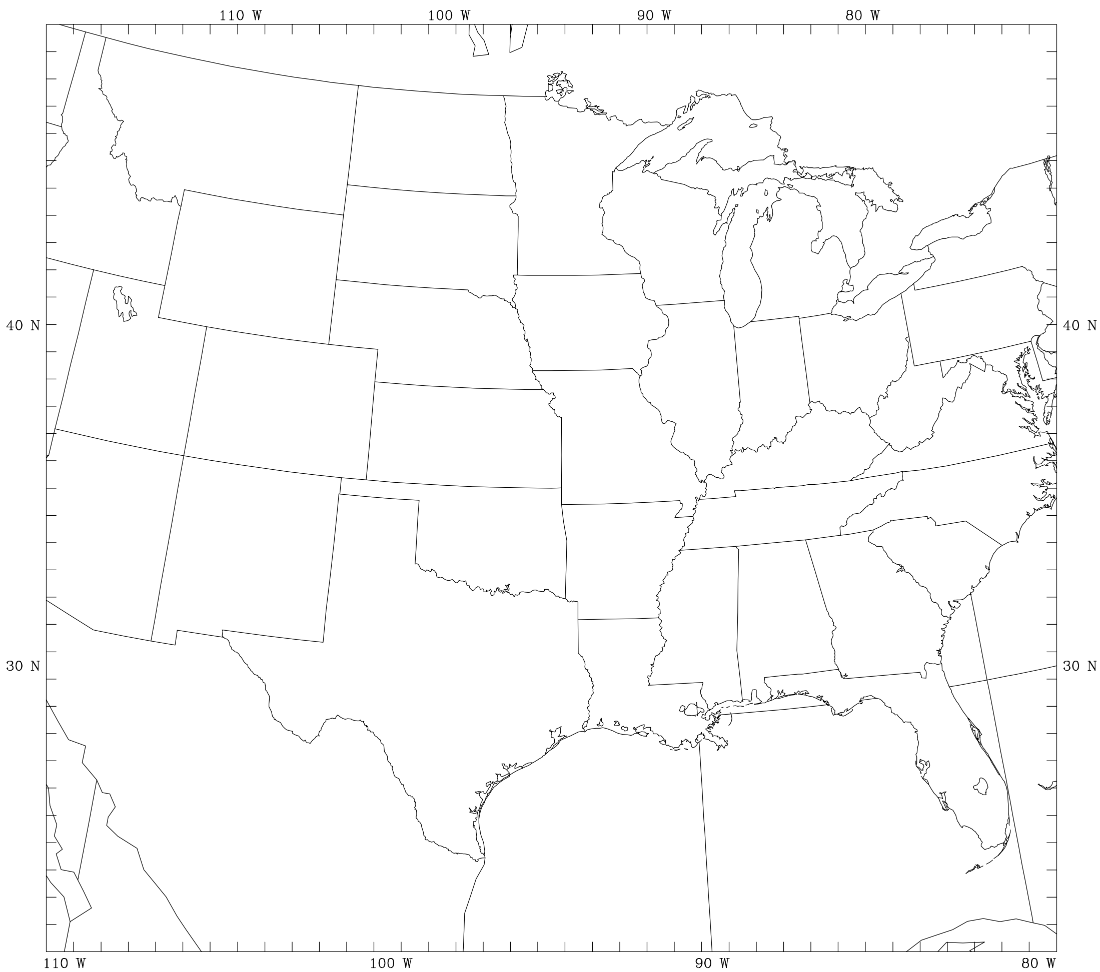

Fig. 4.3: Sample autobogus plot (generated by RAWINS) showing sea-level pressure (mb) analysis and observation values that are flagged in the autobogus file.

upper-air times) to the primary output file. OUTAP accesses each field and calls WTAPE to write the data to the output file. Subroutine OUTFDA performs the same task in a similar manner for the surface fields at FDDA times.

Subroutine WTAPE performs the actual transfer of data by a write statement. Each record contains the final analysis of one variable, at one time, for one level. WTAPE has an option to print a sample of the field (with subroutine OUTPT).

D1 Storage Task

(WDISK, RDISK, HRZFIL, SAVFIL, FILSLB, SLBFIL)

The purpose of the D1 storage array was to provide a memory structure which could be easily replaced by a peripheral storage device (such as a disk) if the core requirements of RAWINS were too large for the computer. Memory limitations have since become less restrictive. This feature of RAWINS is now obsolete, but is still a fundamental part of the RAWINS code.

The transfer of data to and from the D1 storage array is the primary function of six subroutines. The two bottom-level subroutines are WDISK, which transfers a block of data directly to D1; and its counterpart RDISK, which retrieves a block of data directly from D1. These subroutines are the only two routines which directly access D1. All other routines of the D1 Storage Task ultimately use RDISK or WDISK.

One step up in the hierarchy of D1 Storage Subroutines is the subroutine HRZFIL. HRZFIL transfers one strip of a horizontal slab of data to D1 (using WDISK), or retrieves one strip from D1 (using RDISK). In the original version of RAWINS, slabs of data were divided into strips to reduce the in-core memory requirements. In the current version of RAWINS, one strip holds an entire slab of data.

At the same level as HRZFIL is subroutine SAVFIL. SAVFIL transfers an entire block of upper-air data (either the entire DATAH array or the entire CORRH array) to D1 (through WDISK).

At the top of the hierarchy of D1 storage subroutines are subroutines SLBFIL and FILSLB. SLBFIL transfers an entire slab of data to D1 (through HRZFIL and WDISK). FILSLB performs just the opposite task (through HRZFIL and RDISK).

3. RAWINS Subprograms

The bulk of the RAWINS code is composed of over seventy subprograms. This section discusses all of the important FUNCTION and SUBROUTINE subprograms of RAWINS. Several of the subprograms perform trivial or highly specific tasks and in the interest of space are not discussed in this section.

Each subprogram is discussed according to the following pattern:

Purpose: A brief statement summarizing the main purpose or purposes of the subprogram.

On Entry: Descriptions of the state of arguments and certain common variables on entry to the subprogram. All important input to the subroutine is listed in this category.

On Exit: Descriptions of the state of important arguments and common variables on completion of the subprogram. All important output from the subroutine is listed in this category.

Calls: A list of any subprograms which are called by the subprogram being discussed.

Called by: A list of any routines which call the subprogram.

Comments: A more detailed discussion than was provided in the **Purpose** section. The comments may include a discussion of the methods which the subprogram uses to achieve the purpose as well as more detailed descriptions of the subprogram's code itself.

Subroutine ANALMN(A1, A2, ASTA, XOBS, YOBS, IMAX, JMAX, NSTA, RIS, IFAC, IVRBL, ER, IER, BETA, COR, NS, SUM, SUM2)

Purpose: Use the difference values (calculated in GETERR) between the station observations (either surface or upper-air reports) and the current-guess field to nudge the field toward the observations.

On Entry:

A1(IMAX,JMAX): Current-guess field on the mesoscale grid.

ASTA(IRS): Station data for the level (either surface or upper-air) and variable undergoing analysis.

XOBS(IRS): J indices of the observations.

YOBS(IRS): I indices of the observations.

NSTA: Number of stations.

RIS: Square of the radius of influence of the observations for the current pass, taken in the cross-flow direction.

IFAC: Hollerith flag indicating whether ANALMN modifies a current-guess field (IFAC = 6HALTERS) or creates a new first-guess field (IFAC = 6HANALYS).

IVRBL: Integer specifying the variable type currently being processed.

ER(IRS,2): Difference values (observation − current-guess) at each station. ER is calculated in subroutine GETERR. Array ER has two dimensions so that it can hold the difference values for both u and v at one time.

IER: Storage location (index of the second dimension) in ER. IER = 1 for u, T, p_o, and RH. IER = 2 for v.

BETA: Elongation factor, as described in Section IV.A.1. Calculated in BLEND but not used in subroutine ANALMN.

The arrays COR, NS, SUM, and SUM2 (all dimensioned by IMAX,JMAX) are passed to ANALMN as working space only.

On Exit:

A1(IMAX,JMAX): Improved field.

A2(IMAX,JMAX): Original input field.

Calls: Subroutine ANALMN calls no other subprograms.

Called by: Subroutine SETANA.

Comments: Subroutine ANALMN adjusts the value of the current-guess field at each gridpoint to create an improved field. ANALMN *does not* cycle through all gridpoints, find for each gridpoint the observations which influence it, and calculate the correction value α_{ij} (eqs. 4.3 through 4.6). Such an approach would be CPU-intensive.

ANALMN *does* cycle through all the observations. For each observation, two inner DO-loops cycle through only the gridpoints in the region of the observation to calculate the portion of the correction value due to that observation on each of the gridpoints in the region of influence. The weighting function W_{ijk} for each observation applied to each gridpoint is calculated according to equation (4.2), (4.9), or (4.13), depending on whether the Cressman, Ellipse, or Banana scheme is applied. Within the observation loop and for each gridpoint, running totals are kept of the weight (or weight squared) times the difference value D_k, thus calculating $\sum WD$ or $\sum W^2D$.

When all observations have been processed, the values $\sum W^2$, $\sum W$, and $\sum W^2D$ (or $\sum WD$) have been calculated for each gridpoint. The correction values are then calculated according to (4.3), (4.4), (4.5), or (4.6) and applied to the current-guess field, resulting in an improved field.

Subroutine BARNES(NODATA, DAT, XP, YP, NMAX, SUMTHR, NX, NY, K, LAMBDA, SUMONE, SUMTWO, NOS, GRDATA, DX, IGRID, FLAG)

Purpose: Create a first-guess field at intermediate FDDA times from the surface observations, using a Barnes analysis scheme (Barnes; 1964, 1973). NOTE: The Barnes analysis (subroutine BARNES) is currently bypassed by subroutine SFCBLN.

On Entry:

NODATA:	The total number of observations.
DAT(NMAX):	Array holding surface observations.
XP(NMAX):	x-coordinates of the observation locations.
YP(NMAX):	y-coordinates of the observation locations.
NMAX:	The maximum number of stations allowed. NMAX = IRS.

	NX:	x dimension of horizontal slab; NX = IS2 from the calling routine.
	NY:	y dimension of horizontal slab; NX = IS1 from the calling routine.
	K:	Weighting coefficient; set to 40000 in the calling routine.
	LAMBDA:	Convergence coefficient; set to 0.35 in the calling routine.
	DX:	Grid spacing (m).
	IGRID:	Integer flag. IGRID = 1 for cross points, 0 for dot points.
	FLAG:	Flag indicating bad data; set to 1.0×10^6 in the calling routine.

The arrays SUMONE, SUMTWO, NOS, and GRDATA (all dimensioned by NX, NY) are passed to BARNES as working space only.

On Exit:	SUMTHR(NY,NX):	First-guess field created by BARNES. Although BARNES works with a grid orientation different from that of the horizontal domain, SUMTHR is returned to the calling routine with the correct grid orientation.

Calls: Subroutine INTR (not discussed).

Called by: Subroutine SFCBLN. (However, the call to BARNES is currently bypassed.)

Comments: The Barnes analysis (Barnes; 1964, 1973) was included in RAWINS to create surface first-guess fields for intermediate FDDA times. However, it was found that the Barnes scheme produced low-quality first-guess fields in data-sparse regions. Thus, the Barnes scheme is currently bypassed by subroutine SFCBLN, and first-guess fields at intermediate FDDA times are currently created by either temporal interpolation from standard times (0000 and 1200 UTC) or by using an analyzed field from 3 h earlier. The call to subroutine BARNES is left in the code, in case someone desires to use the Barnes analysis scheme.

BARNES calculates for each gridpoint a number of parameters, based on all observations in the region of the gridpoint. A weight factor, W_{ijk}, is calculated by

$$W_{ijk} = \exp\left(\frac{-d_{ijk}^2}{4k_m}\right), \tag{4.23}$$

where k_m is a function of the grid interval, and d_{ijk} is the distance of observation k from gridpoint (i,j). Then, for all gridpoints which are influenced by at least one observation, a new field G_1 is calculated at each gridpoint by

$$G_{1_{ij}} = \frac{\sum_{k=1}^{n} W_{ijk} O_k}{\sum_{k=1}^{n} W_{ijk}}, \tag{4.24}$$

where n is the number of observations influencing gridpoint (i,j), and O_k is the value of the kth observation.

Similar calculations are then made, based not on the observation value, but on a difference value between the observation and the field value.

$$D_k = O_k - G_{1_k}, \tag{4.25}$$

where G_{1_k} is the value of the field G_1 interpolated (by subroutine INTR, not discussed) to the location of the kth observation. For each gridpoint, a new weight factor W_{ijk} is calculated

$$W_{ijk} = \exp\left(\frac{-d_{ijk}^2}{4k_m\lambda}\right), \tag{4.26}$$

where λ is a convergence coefficient. For all gridpoints influenced by at least one observation, a new field G_2 is then calculated at each gridpoint by

$$G_{2_{ij}} = \frac{\sum_{k=1}^{n} W_{ijk} D_k}{\sum_{k=1}^{n} W_{ijk}} + G_{1_{ij}}. \tag{4.27}$$

Finally, values of the field G_2 are assigned in a similar manner at gridpoints which were not influenced by any nearby observations. The final field, oriented in the same manner as the horizontal slabs, is returned to SFCBLN through array SUMTHR(NY,NX).

Subroutine BLEND(ID1, ID2, ID3, HORZH, IH1, IH2, IH3, SLAB1, SLAB2, SLAB3, IS1, IS2, COR, NS, SUM, SUM2, MDATE, AUTBGW, AUTBGR, NVOL40, IFDA)

Purpose: Perform the objective analyses at pressure levels, for variables u, v, T, and RH. No objective analysis is performed for the geopotential heights. Objective analysis is not performed on surface fields in this subroutine.

On Entry:

ID1,ID2,ID3: Dimensions of DATAH and CORRH arrays. DATAH and CORRH themselves are not passed to BLEND, but portions of DATAH and CORRH are retrieved from the D1 storage array.

MDATE: The date/time currently being processed.

AUTBGW: Logical flag indicating whether an autobogus correction file (output) is to be written.

AUTBGR: Logical flag indicating whether an autobogus correction file (input) is to be read.

NVOL40: Fortran unit number for the autobogus output file.

IFDA: The FDDA number of the date/time currently being processed.

The arguments SLAB1, SLAB2, SLAB3, COR, NS, SUM, and SUM2 (all dimensioned by IS1,IS2); and HORZH(IH1,IH2,IH3) are all passed to the subroutine as working space.

On Exit: The analyzed fields of u, v, T, and RH have been written to D1.

Calls: Subroutines RDISK, FILSLB, OUTPT, SETANA, SMTHER, and SLBFIL.

Called by: Main program.

Comments: Subroutine BLEND performs the objective analyses at all pressure levels for the four variables u, v, T, and RH. The subroutine has two main DO-loops. The outer loop processes the mandatory pressure levels on its first iteration and the nonmandatory levels on its second iteration. The inner loop cycles through all pressure levels in either the mandatory or nonmandatory set of levels.

For each pressure level, observations are retrieved from the D1 storage array and put into the array RSOND(5,IRS). At each station, the weighting scheme (Cressman, Ellipse, or Banana) is determined, based on the user option IWTSCM, the observed winds, and the first-guess wind field (as discussed in Section IV.A.1)

Next, subroutine BLEND begins processing the different variables for the current pressure level. The one-dimensional array ASTA(IRS) is loaded with the observations for the particular variable and level being processed. The x- and y-coordinates of the observations are loaded into arrays XOBS(IRS) and YOBS(IRS). An index for identifying a station as either real or bogus is loaded into the array IFD(IRS).

If one of the wind components is being analyzed, the other component must be accessed as well. Thus, if u is being analyzed, the u field is loaded into ASTA, and the v field is loaded into array ASTAWD(IRS); if v is being analyzed, the v field is loaded into ASTA, and the u field is loaded into array ASTAWD.

If the Banana scheme is used, both wind components must be processed in the same call to SETANA. Thus, ASTA is loaded with the v field, ASTA1 is loaded with the u field, and ASTAWD is loaded with the u field.

The first-guess field is then retrieved from D1, and put into array SLAB1(IS1,IS2). Subroutine OUTPT is called to print out a sample of the first-guess field. If the relative humidity field is being analyzed, both the first-guess field and the observations are converted to the humidity variable B, according to

$$B = \left(1 - \frac{RH}{100\%}\right)^{\frac{1}{2}} \quad , \tag{4.23}$$

where RH is the relative humidity (%). This transformation causes regions of high relative humidity to be better maintained during the objective analysis procedure, without numerical dilution by adjacent drier air.

When all of the data necessary for the objective analysis have been assembled, subroutine SETANA is called to perform the actual objective analysis. The analyzed field is returned in the array SLAB1.

After the objectively analyzed field is returned, some simple manipulations are performed. The temperature and humidity fields can be smoothed (by subroutine SMTHER) to remove short wavelength variability. No smoothing is applied to the wind components. The humidity variable is converted from B back to RH. Subroutine OUTPT is called to print out a sample of the analyzed field. The final analyzed field is written to D1, and BLEND repeats the entire procedure until all four variables (u, v, T, and RH) have been analyzed.

Subroutine BOGPTS(DATAH, CORRH, ID1, ID2, ID3, DATAS, IDS1, IDS2, IDS3, IUORS, MDATE, KBOGR, AUTBR, NBRFIL, RSOND, BUFD, ISTNUM, RANG, IFOUND, BUFUPR)

Purpose: Modify specific observations in existing station reports according to the bogus data in the kbogus file and the autobogus correction file.

On Entry:

DATAH(ID1,ID2,ID3): Array holding the mandatory-level observations.

CORRH(ID1,ID2,ID3): Array holding the nonmandatory-level observational data. These data have been interpolated to the new nonmandatory pressure levels from mandatory- and significant-level observations (subroutine SIGDAT).

DATAS(IDS1,IDS2,IDS3): Array holding the surface observations.

IUORS: Integer flag indicating whether upper-air data (IUORS = 1) or surface data (IUORS = 2) are to be modified.

MDATE: Current time period being processed. This time may be a standard upper-air time or an intermediate FDDA time.

KBOGR: Logical flag, .TRUE. if bogus points from the kbogus file are to be accessed; .FALSE. if the kbogus file need not be accessed.

AUTBR: Logical flag, .TRUE. if bogus points from the autobogus correction file are to be accessed; .FALSE. if the autobogus correction file need not be accessed.

NBRFIL: The file number on unit 10 (the autobogus correction file) which contains the autobogus data for the particular MDATE being processed.

ISTNUM(3,IRS): Array holding station identifiers of the stations included in the dataset.

RANG(3,IRS): Array holding an angle for the conversion of wind from meteorological coordinates to grid coordinates at each station. RANG, defined in subroutine SAVSTN, is the angle between the direction in meteorological coordinates and the direction in map coordinates.

IFOUND(3,IRS): Array holding an integer flag identifying each of the stations in the dataset as either a real station (IFOUND(3,n) = 1) or a bogus station (IFOUND(3,n) = 2).

RSOND(5,IRS), BUFD(5,IRS), and BUFUPR(5,IRB) are passed to BOGPTS as working space only.

On Exit: DATAH, CORRH, and DATAS have been updated to include the bogus observations for mandatory-level, nonmandatory-level, and surface bogus observations, respectively. DATAH and CORRH have been transferred to the D1 storage array.

Calls: Subroutines WDISK and RDISK.

Called by: Main program.

Comments: For each standard upper-air time, BOGPTS is called once for the surface data, and once for the upper-air data. For each intermediate FDDA time, BOGPTS is called only for the surface data, since there are no upper-air data.

This subroutine allows the user to correct specific data in surface and upper-air station reports. The kbogus options are for situations in which the user knows in advance, either from an examination of the dataset or a previous execution of RAWINS, that certain observations are erroneous. The autobogus options are for situations in which a previous submittal of RAWINS (Autobogus 1) has flagged suspect observations for the user's examination. BOGPTS is called only if the kbogus or autobogus (or both) options are activated.

When BOGPTS reads a record from the kbogus file or the autobogus correction file, it checks the date and station number to see if the correction refers to an observation in the current dataset. If the correction does not apply to the current dataset, BOGPTS discards the record and reads the next one, until it finds a record it can use.

When BOGPTS has read a useful correction record, it locates the observation in question in the DATAH, CORRH, or DATAS array, based on station identifier, pressure level, and variable type as read from the correction record. The bogus data are then substituted for the original data in the array. BOGPTS then goes back to read another correction record, until all records referring to the current time period have been processed.

Once all of the bogus records for a time period have been processed, DATAH and CORRH, if they have been modified, are rewritten to the D1 storage array.

The details of using the kbogus and autobogus options are discussed in Section IV.B.3; examples of the kbogus and autobogus files are given in Appendices E and F.

Subroutine BUDDY(N, OB, GRID, ISTA, X, Y, IMAX, JMAX, IVRBL, PL, BUDWGT, MDATE, KBUD, IUVEL, IVVEL, ITEMP, IRELH, IREFPC)

Purpose: Check observations by comparing the difference value at each observation point with an average of the difference values at nearby observation points (buddy check). Flag as erroneous those observations for which the difference value is not consistent with the difference values of surrounding stations.

On Entry:

N: Number of observations (NSTA from the calling subroutine GETERR).

OB(IRS): Observations of one variable at one level (ASTA from subroutine GETERR).

GRID(IMAX,JMAX): The first-guess field (A2 from GETERR).

ISTA(IRS): Station identifiers (ISTNUM from GETERR).

X(IRS), Y(IRS): x- and y-coordinates of the observations (XOBS and YOBS from GETERR).

IVRBL: Integer identifying the variable being processed.

PL: Pressure level (mb) of the field being processed.

BUDWGT: Local Master Input File option, determines tolerance of check.

MDATE: The particular date/time being processed.

IUVEL, IVVEL, ITEMP, IRELH, and IREFPC are integer values indicating variables u, v, T, RH, and p_o, respectively.

On Exit:

KBUD(IRS): Integer flags identifying each observation as erroneous (KBUD(n) = 1) or good (KBUD(n) = 0), based on the buddy check.

MSTAB(399): Station identifiers of observations flagged as erroneous by subroutine BUDDY. Common/BUD/.

XBUD(399):	x-coordinates of observations flagged by BUDDY. Common/BUD/.
YBUD(399):	y-coordinates of the observations flagged by BUDDY. Common/BUD/.
BLEV(399):	Pressure levels of the observations flagged by BUDDY. Common/BUD/.
MVARB(399):	Variable type of the observations flagged by BUDDY. Common/BUD/.
EDIF(399):	Difference values (observation − first guess) of the observations flagged by BUDDY. Common/BUD/.
BDIF(399):	Weighted average of the difference values of the stations near the observations flagged by BUDDY. Common/BUD/.
NBTOT(399):	Number of stations in the vicinity of the observations flagged by BUDDY. Common/BUD/.
NNTOT:	Total number of observations flagged by BUDDY.

Calls: Subroutine BUDDY calls no other subprograms.

Called by: Subroutine GETERR.

Comments: For each observation, Subroutine BUDDY calculates the difference between the observation and the value of the first-guess field interpolated to the observation location:

$$D_k = O_k - G_k. \tag{4.24}$$

Then, the average of the difference values at the nearest stations (up to eight stations) within a certain radius of the observation is calculated.

$$\overline{D} = \frac{\sum_{i=1}^{n} D_i}{n}. \tag{4.25}$$

If the difference value at the observation point differs from the average of the nearby difference values by more than a certain maximum (*i.e.*, $|\overline{D} - D_k| > \delta$), the observation is flagged as erroneous. The maximum threshold δ depends on the variable being processed, the pressure level, and the BUDWGT parameter set by the user in the Local Master Input File. The information about erroneous observations is stored in Common/BUD/ arrays MSTAB, BLEV, EDIF, BDIF, XBUD, YBUD, and NBTOT for later printout.

Subroutine FILSLB(L, IVRBL, IPLACE, HORZH, IH1, IH2, IH3, SLAB, IS1, IS2)

Purpose: Transfer a horizontal slab of data from D1 to SLAB. This subroutine performs just the opposite task of SLBFIL. FILSLB calls subroutine HRZFIL to perform the actual transfer of data.

On Entry:

- L: Index number of level for the data to be retrieved.
- IVRBL: Variable identifier (integer) of the data to be retrieved.
- IPLACE: Working space index of the third dimension of HORZH. The data are first transferred from D1 to one "level" (third dimension) of HORZH, and finally to SLAB.
- HORZH(IH1,IH2,IH3): Working space.

On Exit:

- SLAB(IS1,IS2): Array holding level L of variable IVRBL, retrieved from D1.

Calls: Subroutine HRZFIL

Called by: Main program, and subroutines BLEND, GETRAW, NEWPLV, OUTAP, OUTFDA, PSFC, and SFCBLN.

Subroutine GETERR(A2, ASTA, XOBS, YOBS, IMAX, JMAX, NSTA, IFAC, IVRBL, IFD, PL, ERRMXW, ERRMXT, ERRMXP, ER, IER, NFI, ISTNUM, ASTAWD, KTYPE, LVL, MDATE, AUTBGW, AUTBGR, NVOL40, RANG, KN)

Purpose: Find the difference between the station observations in ASTA and the current-guess field in A2 at the station locations. Check the data for possible errors with a buddy check (subroutine BUDDY) and with the ERRMX criteria ERRMXW, ERRMXT, and ERRMXP.

On Entry:	A2(IMAX,JMAX):	Current-guess field at gridpoints.
	ASTA(IRS):	Station observations.
	XOBS(IRS):	x-coordinates of the station observations.
	YOBS(IRS):	y-coordinates of the station observations.
	NSTA:	Number of stations.
	IFAC:	Hollerith flag. IFAC = 6HALTERS indicates that a first-guess field exists, and the difference values may be calculated; IFAC = 6HANALYS indicates that no first-guess field exists, and the observations are returned as the difference values.
	IVRBL:	Integer identifying the variable being processed.
	IFD(IRS):	Integer flag identifying each observation n as either real (IFN(n) = 1) or bogus (IFD(n) = 2).
	PL:	Pressure (mb) of level undergoing analysis.
	ERRMXW:	Maximum allowable difference for the wind components.
	ERRMXT:	Maximum allowable difference for the temperature field.
	ERRMXP:	Maximum allowable difference for the pressure field.
	IER:	Storage location (second dimension) for the array ER(IRS,2). IER = 1 for variables u, T, p_o, and RH; IER = 2 for v.
	NFI:	Always 1.
	ISTNUM(3,IRS):	Station identifiers.
	ASTAWD(IRS):	Station observations of one wind component, when the other component is being analyzed.
	KTYPE:	Integer flag indicating data type. 1: Mandatory-level data. 2: Surface data. −1: Nonmandatory-level data.
	LVL:	Level index written to the autobogus correction file (output).
	MDATE:	The particular date/time being processed. MDATE (in the format YYMMDDHH) is broken into four integers representing year, month, day, and hour (YY, MM, DD,

and HH) by subroutine DECOMDAT (not discussed).

AUTBGW: Logical flag, .TRUE. if an autobogus correction file is to be written (*i.e.*, RWSUBM = 1, the first of two autobogus submittals).

AUTBGR: Logical flag, .TRUE. if an autobogus correction file is to be read (*i.e.*, RWSUBM = 2, the second of two autobogus submittals).

NVOL40: Unit number of the autobogus correction file (output).

RANG(3,IRS): Array holding an angle used in converting observed winds from meteorological coordinates to map coordinates.

KN: Record number in the autobogus correction file (input) for the observations flagged as suspect for a given date, level, and variable type by a previous execution of RAWINS.

KSCAN: The current scan number (for the Cressman multiscan procedure) in the successive calls to GETERR from SETANA. Common/SCA/.

On Exit: ER(IRS,2): Difference values between observations and current-guess field. The first storage location (second dimension) holds difference values for u, T, p_o, or RH. The second storage location (second dimension) holds the difference values for v when the first holds the difference values for u.

The arrays MSTAE, XERM, YERM, ELEVM, MVARE, ERRMXE, and EDIFM (all dimensioned by 399) and the variable MMTOT are returned to the main program through Common/MXER/. These arrays hold the station identifier, location, pressure level, variable identifier, maximum-error criterion, and the difference value for each observation that was flagged as erroneous by GETERR.

Calls: Subroutines BUDDY and DECOMDAT.

Called by: Subroutine SETANA.

Comments: Subroutine GETERR first calls subroutine BUDDY to flag any stations at which the difference values are not consistent with those at surrounding stations. Although the observations are checked with the buddy check, the additional checks performed in GETERR (ERRMX checks) are important in regions of low-observation density where the buddy check is less effective.

A DO-loop cycles through all observations to check those observations that were not flagged as suspect by BUDDY.

For each observation, the first-guess field is interpolated by a simple bilinear interpolation method to the location of the observation. The difference value D is calculated by subtracting the interpolated value of the analyzed field from the observed value. This difference value D is compared to the appropriate ERRMX criterion, and if $|D|$ exceeds the maximum error allowed, the observation is flagged as suspect, to be removed from the analysis.

The ERRMX criteria are slightly different for each field. They depend on the variable, the time, the bottom surface (land or water), the level, and the parameters ERRMXW, ERRMXT, and ERRMXP. The ERRMX values are all adjusted according to Tables 4.4, 4.5, and 4.6.

Wind observations are accepted as correct if the difference value is less than either the modified ERRMXW or 0.15 times the value of the first-guess field at the station location. Pressure and temperature observations are accepted if the difference values are less than the modified ERRMXP and ERRMXT parameters.

If the ERRMX criteria indicate a probable error, a message is printed showing the variable type, gridpoint location of the station, the analyzed field value, and the observation value. The difference value and the observation are both changed to the flag value of 1.0×10^{32} to ensure that the observation is removed from further consideration.

Information concerning the observation, error magnitude, and error type is saved in arrays for later printout and accounting. If the autobogus file is being constructed, one record for that file is created. Processing continues with the next observation, until all observations of the variable at the level and for the time period have been processed. The difference values are returned to subroutine SETANA in the array ER.

If RAWINS is to create a new analysis from the observations without using a first-guess field (IFAC = 6HANALYS), then on the first call to GETERR for each field, the observed values themselves are returned instead of the difference values. Subsequent calls the GETERR from SETANA calculate difference values.

Subroutine GETRAW(DATAH, CORRH, ID1, ID2, ID3, DATAS, IDS1, IDS2, IDS3, MDATE, MHR, IFIL, FDSF, IFDA, HORZH, IH1, IH2, IH3, SLAB1, IS1, IS2)

Purpose: Access the raw observations. Data files are read and unpacked, and the desired data are put into arrays DATAH, CORRH, and DATAS. Bogus station reports are accessed as well, and incorporated into the observational dataset.

On Entry:

- MDATE: The current date/time being processed (YYMMDDHH). MDATE may be either a standard upper-air time or an intermediate FDDA time.
- MHR: The hour of the current time being processed (HH from MDATE).
- IFIL: The number of the current date/time sequence being processed.
- FDSF: Logical flag, .TRUE. if FDDA processing is to be performed.
- IFDA: The number of the FDDA time period being processed for a particular 12 h period.

Arrays HORZH(IH1,IH2,IH3) and SLAB1(IS1,IS2) are both passed to GETRAW as working space only.

On Exit:

- ISTNUM(3,IRS): Array holding station identifiers. Common/C/.
- IFOUND(3,IRS): Integer flag indicating whether the station report n is real (IFOUND(3,n) = 1) or bogus (IFOUND(3,n) = 0. Common/C/.
- RANG(3,IRS): Angle at each station for the conversion of winds from meteorological coordinates to map coordinates. Defined in subroutine SAVSTN.

Upper-air data (real and bogus) have been put into arrays DATAH (for mandatory-level data) and CORRH (for significant-level data). DATAH and CORRH (both dimensioned by ID1, ID2, ID3) have been transferred to storage array D1.

Surface data (real and bogus) have been put into array DATAS (IDS1, IDS2, IDS3) DATAS is not transferred to D1.

Calls: Subroutines GETUNIOB, FILSLB, RDADP, PROUPR, PROSFC, PROBGU, PROBGS, and SAVFIL.

Called by: Main program.

Comments: If real-time observations have been requested, GETRAW first calls subroutine GETUNIOB to retrieve the observations and write them to temporary files NVOLS6 and NVOLUP (for surface and upper-air observations, respectively).

For processing historical observations, GETRAW starts with the upper-air files (beginning with FORTRAN unit 15). GETRAW calls subroutine RDADP to read one station report from the archived data files. Key identification variables are returned to GETRAW through Common/B/. For each report read, GETRAW performs a number of checks on the identification variables. If the report, station identifier, station location, and report type do not match the needed parameters, then the record is discarded and the next report is read. When GETRAW finds a station report which passes the tests and should thus be included in the dataset for the objective analyses, GETRAW calls PROUPR to unpack that one upper-air report and put the data into arrays DATAH and CORRH. GETRAW then returns to the beginning of the cycle to read another upper-air report and repeat the process until all upper-air reports from the current time period have been processed.

After the upper-air reports have been processed, GETRAW calls PROBGU to process the bogus upper-air reports. When PROBGU has finished, DATAH holds all mandatory-level data (real and bogus) and CORRH holds all significant-level data (real and bogus). GETRAW calls SAVFIL to store DATAH and CORRH to D1.

Once all upper-air processing has been completed, GETRAW processes the surface data. GETRAW calls RDADP to read a surface report, and makes the same checks that it did for the upper-air reports. If the station report is to be used, GETRAW calls PROSFC to process that one surface report and put the data into array DATAS. Another report is then read, and the procedure is repeated until all surface reports have been accessed. GETRAW calls PROBGS to process all bogus surface observations. When PROBGS has completed, all surface station reports (real and bogus) have been put into the array DATAS.

The locations of all surface-level data (from both upper-air and surface reports) are then compared. Surface and upper-air stations which share the same locations are counted, and the total number of redundant surface reports is printed. All surface observations, including redundant observations, are considered in the objective analysis.

GETRAW sets the radius of influence for surface observations (RINSFC) equal to the radius of influence for upper-air observations (RIN).

Subroutine GETUNIOB(NVOLS6, NVOLUP)

Purpose: Retrieve the real-time observations (one time period) from Unidata files, write the data to temporary files.

On Entry: NVOLS6: FORTRAN unit number to which the surface observations are written.

NVOLUP: FORTRAN unit number to which the upper-air observations are written.

On Exit: Surface and upper-air observations are written to FORTRAN units NVOLS6 and NVOLUP.

Calls: Subroutines SFSCVT and SFUAOB.

Called by: Subroutine GETRAW.

Comments: Subroutine GETUNIOB accesses from the Unidata files the real-time observations to be used in RAWINS. GETUNIOB first calls subroutine SFSCVT, which retrieves the surface observations. GETUNIOB checks the data for gross errors, and writes the sea-level pressure, surface temperature, dewpoint depression, wind direction, wind speed, and observation time to unit NVOLS6 for each surface report.

After the surface reports have been accessed and written, GETUNIOB calls subroutine SFUAOB, which retrieves the upper-air observations. GETUNIOB then loops through each of the rawinsonde reports. For each report, GETUNIOB cycles through all mandatory and significant temperature levels, and then through all wind levels. Each observation is checked for gross errors. Finally, the observations are written out to unit NVOLUP.

Subroutine HEDRIN(I1, I2, I3, I4, I5, I6, I7, I8, I9, J1, J2, J3, J4, R1, R2, R3, R4, R5, R6, L1, L2, L3)

Purpose: Retrieve the Master Input File variables from the MIF, MRF, and MLF arrays, which have been read from the DATAGRID header record.

On Entry: MIF(30): Array holding integer values. MIF was read from the DATAGRID output. Common/HEDMIF/.

MRF(10): Array holding real values. MRF was read from the DATAGRID output. Common/HEDMIF/.

MLF(10): Array holding logical values. MLF was read from the DATAGRID output. Common/HEDMIF/.

On Exit: The integer values I1 through J4 are initialized to values from the MIF array.

The floating-point values R1 through R6 are initialized to values from the MRF array.

The logical values L1 through L3 are initialized to values from the MLF array.

Calls: Subroutine HEDRIN calls no other subprograms.

Called by: Main program.

Comments: The Common Master Input File and the DATAGRID Local Master Input File both contain information which must be accessible to RAWINS. Both files were read by program DATAGRID, and the variables in them were placed in three arrays. These three arrays were included in the header record written out to the DATAGRID output file along with the actual first-guess fields. When RAWINS reads the header record from the DATAGRID output, the three arrays are read in as arrays MIF, MRF, and MLF. Subroutine HEDRIN extracts the data stored in these arrays.

Extracted from the MIF array are the variables IFILES, IMAX, JMAX, IMAXN, JMAXN, ICNS, JCNS, NSTTYP, INY, JNX, KSIGT, ISEQ, and IFCST.

Extracted from the MRF array are the variables DS, PHIC, XLONC, DSN, AEXP, and PTOP.

Extracted from the MLF array are the variables IFNEST, IEXP, and IMSG.

Subroutine HEDROT

Purpose: Add variables to the MIF, MRF, and MLF arrays, which are later written out in the header records. Initialize variables NTOTLV, NVERT, and LEVEL1. Initialize arrays GLVL(50), LVL(40), and IHEDPR(40).

On Entry: I1, ..., I7: Integer values to be put into the array MIF(30).

R1, ..., R3: real values to be put into the array MRF(10).

L1, ..., L8: Logical values to be put into the array MLF(10).

	GNLVL(50):	Array holding the pressures at the nonmandatory levels, as set by the user in the Common Master Input File, and read from the DATAGRID output.
	NNEWPL:	Number of nonmandatory levels to be created by RAWINS, as read from the DATAGRID output header.
	NLVDAT:	Number of mandatory levels in the data read from DATAGRID. Common/HEDMIF/.
	LPRDAT(20):	Array holding pressures at the mandatory levels, as read from the DATAGRID output header. Common/HEDMIF/.
On Exit:	GLVL(50):	Array holding pressure levels. GLVL holds pressure level 1001 (which denotes the surface level), followed by the mandatory levels as read from DATAGRID.
	NVERT:	Number of mandatory levels, plus 1 (NLVDAT + 1). The one additional index is for pressure level 1001.
	LEVEL1:	Set to 2, the index of the first pressure level.
	IHEDPR(40):	Array holding pressure levels. Pressure levels in IHEDPR are 1001, followed by the mandatory levels. For example, IHEDPR might hold (1001, 1000, 850, 700, 500, 400, 300, 250, 200, 150, 100, 900, 800, 750, 650, 600, 550, 450, 350). Common/HEDMIF/.
	LVL(40):	An array containing the indices for the pressures in the IHEDPR array, in order of decreasing pressure. For example, for the sample pressures in IHEDPR, the array LVL would hold (1, 2, 12, 3, 13, 14, 4, 15, 16, 17, 5, 18, 6, 19, 7, 8, 9, 10, 11). Common/HEDMIF/.
	MIF(30):	Array holding integer values to be written to the header record. Common/HEDMIF/.
	MRF(10):	Array holding real values to be written to the header record. Common/HEDMIF/.
	MLF(30):	Array holding logical values to be written to the header record. Common/HEDMIF/.

Calls: Subroutine HEDROT calls no other subprograms.

Called by: Subroutine SETUP.

Comments: Array MIF is loaded with the integer variables IFAC, IWTSCM, IWIND, NSTATN, KSTRPH, IPROJ, and RWSUBM.

Array MRF is loaded with the real variables ERRMXT, ERRMXW, and ERRMXP.

Array MLF is loaded with logical variables NOBLND, IWT, SMOOTH, ALRAWS, IPOT, ISFC, AUTBGR(1), AND FDDASF(1).

Subroutine HRZFIL(JSH, LEV, IVRBL, IPLACE, HORZH, IH1, IH2, IH3, UPDATE)

Purpose: Transfer horizontal strips (slabs) of data to storage array D1 from HORZH, or to HORZH from D1.

On Entry:

On Entry:	JSH:	Horizontal strip index number, should always be equal to 1. In the current version of RAWINS, a horizontal slab (dimensioned IMAX, JMAX+1) is represented by one strip.
	LEV:	Level index identifying the vertical level of the slab to be transferred.
	IVRBL:	Integer identifying variable type of the slab to be transferred.
	IPLACE:	Integer (1 through 5) indicating which one of the slabs held in HORZH (third dimension) is to be transferred.
	HORZH(IH1,IH2,IH3):	Array holding data to be transferred to D1, if UPDATE is .TRUE.
	UPDATE:	Logical flag, .TRUE. if data are to be transferred from HORZH to D1, .FALSE. if data are to be transferred from D1 to HORZH.
On Exit:	HORZH:	Array holding data retrieved from D1, if UPDATE is .FALSE.

Calls: Subroutines RDISK and WDISK.

Called by: Subroutines FILSLB, NEWPLV, PSFC and SLBFIL.

Comments: HORZH may hold up to IH3 = 5 different horizontal strips of data. Only one strip, identified by IPLACE, may be transferred to or from D1 in any particular call to HRZFIL.

If UPDATE is .TRUE., data in HORZH are transferred to D1. The value NAME is calculated, a unique integer identifying each variable at each level. NAME is a function of the level index LEV and the variable identifier IVRBL. Subroutine WDISK is called to perform the actual transferral of data to D1.

If update is .FALSE., a block of data in D1 is to be transferred to HORZH. NAME, the unique variable identifying the block of data, is calculated, and subroutine RDISK is called to perform the actual transferral of data to HORZH.

Subroutine INACCT(ID1, ID3, ISTATN, SSTA, IDF, NSTA, ALRAWS, DATBOG, KLAT, KLON, KELEV, JSTATN, IDS3)

Purpose: Perform accounting for station counters and other identifiers.

On Entry:

- ID1: Maximum number of variables. ID1 = 5.
- ID3: Maximum number of upper-air reports which RAWINS may process. ID3 = IRB.
- ISTATN: Number of stations (surface and upper-air, real and bogus) currently saved.
- SSTA: Station identifier of the current station report being processed.
- IDF: Current data file number. Should always be equal to 1.
- NSTA: Position in the data file of the current station report.
- ALRAWS: Logical, should always be .TRUE.
- DATBOG: Integer flag indicating real (DATBOG = 1) or bogus (DATBOG = 2) data.
- KLAT: Station latitude.
- KLON: Station longitude.
- KELEV: Station elevation.
- JSTATN: Number of surface stations currently saved.
- IDS3: Total number of surface stations which may be processed.

On Exit: ISTATN has been incremented.

JSTATN has been incremented, if the station report was from a surface station.

The arrays ISTNUM(3, IRS), IKLAT(2, IRS), IKLON(2, IRS), IKELEV(2, IRS), and IFOUND(3, IRS), are updated with the current station's station identifier, latitude, longitude, elevation, and type (real or bogus), respectively.

Calls: Subroutine INACCT calls no other subprograms.

Called by: Subroutines PROBGU, PROBGS, PROUPR, and PROSFC.

Subroutine INHRZ1(HORZH, IH1, IH2, IH3, SLAB1, SLAB2, IS1, IS2, LAND, SNOW, MDATE)

Purpose: Read the terrestrial fields and the surface and upper-air first-guess fields for one time period from the DATAGRID output volume (unit 4).

On Entry: MDATE: Time period currently being processed. MDATE refers only to standard times (not intermediate FDDA times) for this subroutine.

HORZH(IH1,IH2,IH3), SLAB1(IS1,IS2), and SLAB2(IS1,IS2) are passed to the subroutine as working space.

On Exit: LAND(IS1,IS2): Array holding land-use characteristics.

SNOW(IS1,IS2): Array holding snow-cover field.

Calls: Subroutines RTAPE and SLBFIL.

Called by: Main program.

Comments: Subroutine INHRZ1 accesses all surface and mandatory-level first-guess fields created by DATAGRID. Data input may also be forecast grids from previous runs of the forecast model. As each horizontal level of data is read, it is transferred to the storage array D1 by subroutine SLBFIL. Land-use and snow-cover fields are read directly into the permanent arrays LAND and SNOW, and are not transferred to D1.

For all fields except land-use and snow-cover, INHRZ1 calls RTAPE to read the field from unit 4 into a working array (either SLAB1 or SLAB2). INHRZ1 then calls subroutine SLBFIL to transfer the field to D1. If the user has selected (by the Local Master Input File option IWIND) to use surface geostrophic winds as the first-guess surface wind field, the winds calculated in the DATAGRID program are transferred to D1. If the user selected 1000 mb winds as the first-guess surface winds, the 1000 mb winds instead of the surface geostrophic winds are transferred to D1.

Relative humidity fields at pressure levels up to 300 mb are read and transferred to D1. INHRZ1 creates additional levels above 300 mb (if any such levels are required) by setting the relative humidity to 10%. These additional levels are also transferred to D1. The 1000 mb relative humidity field is also stored as the surface field.

Subroutine INHRZ2(HORZH, IH1, IH2, IH3, MDATE, IFILES, IFDATL, IS1, IS2, SLABA1, SLABA2, SLABA3, SLABA4, SLABA5, SLABB1, SLABB2, SLABB3, SLABB4, SLABB5, SCR)

Purpose: Read the input surface analyses (output from DATAGRID for standard upper-air times) and interpolate temporally to create surface first-guess fields of u, v, T, RH, and p_o at intermediate FDDA times.

On Entry:

MDATE: The current date/time being processed.

IFILES: The total number of time periods to process; from the Common Master Input File.

IFDATL: The number of times to process during each 12 h time period, including all intermediate FDDA times as well as the standard upper-air time.

All of the other arguments are arrays (with dimensions IH1,IH2,IH3 and IS1,IS2) passed to INHRZ2 for working space.

On Exit: All FDDA first-guess fields, interpolated temporally from surface fields at standard upper-air times, have been written to unit 45.

Calls: Subroutines SKIPF and SKIPR (not discussed).

Called by: Main program.

Comments: Subroutine INHRZ2 interpolates from 0000 and 1200 UTC surface first-guess fields to create surface first-guess fields at intermediate FDDA times (*i.e.*, 0300, 0600, 0900, 1500, 1800, and 2100 UTC). Thus, INHRZ2 is called only if the FDDA option is activated, and only if the LAGTEM option is not activated (see Local Master Input File, Table 4.1).

INHRZ2 first reads in surface first-guess fields at the initial time. Surface first-guess fields of p_o, T, u, v, and RH are read into arrays SLABA1 through SLABA5. All other surface fields and all upper-air fields are read into the scratch array SCR and discarded.

INHRZ2 then enters a DO-loop, cycling through the remainder of the standard upper-air times. Surface first-guess fields at the next standard upper-air time (12 h later than those fields in the SLABA arrays) are read into arrays SLABB1 through SLABB5. Temporal interpolation is performed to generate first-guess fields at each of the intermediate times. As the fields are generated, they are written to unit 45 for temporary storage. When fields at each of the intermediate times have been generated and stored, INHRZ2 transfers the fields in the SLABB arrays to the SLABA arrays. The DO-loop then processes the next 12 h period, until all FDDA first-guess fields have been generated.

Subroutines SKIPF and SKIPR (not discussed) skip a specified number of files and records, respectively, in the DATAGRID output file.

Function **LOGCMP**(B, I, C)

Purpose: Compare a specific portion of the packed record to a specified character word. LOGCMP is called to check for end-of-file, end-of-record, and end-of-report marks in NMC data records, and the string 'WASHINGT', which identifies a data record.

On Entry:
- B: The packed record; NB(820) from the calling routine.
- I: Word pointer to locate a character string within B.
- C: Character word used for logical comparison to the bit string extracted from B. LOGCMP is called with C equal to '8HWASHINGT', '8HENDOF FI', '8HEND RECO', or '8HEND REPO'.

On Exit: LOGCMP equals 0 or 1: 1 indicates that the string C matched the bit string extracted from B; 0 indicates that the string C did not match.

Calls: NCAR Subroutines GBYTES and SBYTES.

Called by: Subroutines MANADP, RDADP, SFCADP, SIGADP, TRPADP, and WPPADP.

Comments: LOGCMP calls GBYTES, which skips a specified number of characters in the bit string B, and unpacks the next eight characters into the holding array

IH. Then, LOGCMP calls SBYTES to transfer the same characters from IH to the word A. A logical IF statement is then used to compare the character string in the word A with the specified string C. LOGCMP returns 0 if the comparison is false, and 1 if the comparison is true.

Subroutine MANADP(P, Z, T, H, D, F, Q, NLV, MAXLEV)

Purpose: Extract the mandatory-level data from a packed upper-air report in buffer NB. MANADP is called only for historical data.

On Entry:

- NB(820): Buffer holding packed upper-air data. Common/ADP/.
- NPT: Index in array NB at which MANADP starts reading. Common/ADP/. NPT keeps track of how much of the NB array has been unpacked.
- LTH: Length of the record read in RDADP, in 10-character units. Common/ADP/.
- MAXLEV: The maximum number of levels which RAWINS can process.

On Exit:

- P: One dimensional array holding pressures (mb) at mandatory levels.
- Z: One dimensional array holding geopotential heights (m) at mandatory levels.
- T: One dimensional array holding temperatures (°C) at mandatory levels.
- H: One dimensional array holding dewpoint depression (°C) at mandatory levels.
- D: One dimensional array holding wind direction (degrees) at mandatory levels.
- F: One dimensional array holding wind speed (kts) at mandatory levels.
- Q: One dimensional array holding error characters from the mandatory-level report.
- NLV: Number of mandatory levels of data, up to MAXLEV, read from the data file.

Calls: NCAR Subroutines GBYTES, and SBYTES. Functions LOGCMP and NTENS.

Called by: Subroutine PROUPR.

Comments: Subroutine MANADP extracts pressure, geopotential height, temperature, dewpoint depression, wind speed, and wind direction.

MANADP calls NCAR subroutine GBYTES and RAWINS function NTENS to unpack from the data buffer NB the identifier for the data type into variable NCAT. If the flag identifying data type does not indicate mandatory-level data (*i.e.* if NCAT $\neq 1$), then MANADP repeats the calls to GBYTES and NTENS to search the data buffer for the correct identifier (NCAT $= 1$). If after searching through the buffer, MANADP has not found the correct identifier, it will return to the calling routine (subroutine PROUPR) with no mandatory-level data (NLV $= 0$).

If the flag identifying mandatory-level data (NCAT $= 1$) is found, MANADP calls NTENS to extract the number of mandatory levels in the report (NLV). A DO-loop then cycles through the mandatory levels, calling GBYTES, SBYTES, and NTENS to extract the data into the one-dimensional arrays P, Z, T, H, D, F, and Q.

Subroutine NEWPLV(LPR, IVRBL, HORZH, IH1, IH2, IH3, SLAB1, IS1, IS2)

Purpose: Generate a first-guess field of geopotential height, u, v, T, or RH at a new nonmandatory pressure level.

On Entry:

GNLVL: Array holding nonmandatory pressure levels specified by the user. Common/A/.

LPR: Index of new nonmandatory pressure level in array GNLVL.

IVRBL: Integer indicating variable type.

Arguments HORZH(IH1,IH2,IH3) and SLAB1(IS1,IS2) are passed to NEWPLV only as working space.

Calls: Subroutines HRZFIL, FILSLB, and SLBFIL.

Called by: Main program.

Comments: Subroutine NEWPLV is called for each variable at each nonmandatory pressure level. Each call to NEWPLV creates one new first-guess field by vertical interpolation from the bracketing data at the surrounding pressure levels. A simple linear interpolation is used for the height, u, v, and RH fields; T is interpolated linearly in $ln(p)$. The new first-guess is saved to the storage array D1.

If the new pressure level is above the 850 mb level, the vertical interpolation is performed from the mandatory levels immediately above and below the new pressure level.

If the new pressure level is below the 850 mb level, NEWPLV treats two cases differently. If the new pressure level is above both the surface and the 1000 mb level, the interpolation is performed from the 850 mb and surface fields. If the new pressure level is between the 1000 mb level and the surface, the interpolation is performed from the 1000 mb and surface fields.

In some situations, the new pressure level may be below the 1000 mb level while locally high terrain causes the surface pressure to be less than 1000 mb. In this instance, both known levels are above the new nonmandatory level and the resulting extrapolation is not appropriate. However, because the questionable points on the nonmandatory level are below the ground surface, there is no contamination of atmospheric regions.

Geopotential heights at nonmandatory levels below the 850 mb level are recalculated using the hypsometric equation with the 1000 or 850 mb temperature and the 1000 or 850 mb geopotential height, according to

$$\phi_n = \phi_\ell - \frac{R_d}{g}\left(\frac{T_\ell + T_n}{2}\right)\ln\left(\frac{p_n}{p_\ell}\right), \tag{4.26}$$

where the subscript n indicates values at the new nonmandatory level and the subscript ℓ indicates values at the 1000 or 850 mb level, whichever is immediately above the new level n.

Function **NTENS**(KC, KN)

Purpose: Returns the integer value of a character string consisting of digits and signs. The character string has been read from the packed data record.

On Entry: KC: Incoming one-dimensional character-holding buffer array.

KN: Number of characters in the buffer KC to be transformed into a single integer.

On Exit: NTENS: Integer value represented by the KN digits in the buffer array KC.

Calls: Function NTENS calls no other subprograms.

Called by: Subroutines MANADP, RDADP, SFCADP, TRPADP, and WPPADP.

Comments: Function NTENS receives a character string in buffer KC, and the number of words to extract, KN. Each word in array KC should represent a digit or a sign character, unpacked from the data record read by subroutine RDADP. The first word address assigned to KC is generally a word location within the character string holding array IC in the calling subprogram.

Suppose, for example, that in the calling routine the array IC contains the character values IC(17) = '−', IC(18) = '6', IC(19) = '3', and IC(20) = '9'. The function call NTENS(IC(17),4) would then return the integer value −639.

Subroutine OUTAP(HORZH, IH1, IH2, IH3, SLAB1, SLAB2, IS1, IS2, LAND, SNOW, SLAB4, SFCPR, LANDC, MAXIC, MAXJC, MDATE, NSELIM, NBOGUS, KBOGUS)

Purpose: Write the final analyses to the main output file, unit 2.

On Entry:

LAND(IS1,IS2):	The land-use field on the expanded grid.
SNOW(IS1,IS2):	The snow-cover field.
LANDC(MAXIC,MAXJC):	The land-use field at cross points (unexpanded grid).
SFCPR(IS1,IS2):	Surface pressure field, as calculated in subroutine PSFC.
MDATE:	The time period being processed.
NSELIM(18):	Logical flags indicating whether station reports have been deleted.
NBOGUS:(18):	Logical flags indicating whether bogus station reports were included.
KBOGUS:(18):	Logical flags indicating whether bogus corrections were made to the observed data.

Arguments HORZH(IH1,IH2,IH3), SLAB1(IS1,IS2), SLAB2(IS1,IS2), and SLAB4(MAXIC,MAXJC) are passed to OUTAP as working space only.

Calls: Subroutines FILSLB, SLBFIL, and WTAPE.

Called by: Main program.

Comments: The output file is described in greater detail in Section IV.B.1.b. The header record is written out first, followed by the data records. The information written to the header record is supplied to OUTAP through Common/HEDMIF/, Common/A/, Common/D/, and the argument list. For most of the data records, OUTAP first calls FILSLB to retrieve the data from storage array D1 and then calls WTAPE to write the data to FORTRAN unit 2. Land-use and snow-cover fields, however, are passed to OUTAP through the arguments LAND, LANDC, and SNOW.

Before writing the temperature fields to the output file, OUTAP removes any superadiabatic layers at the surface and at pressure levels below 500 mb. To do this, OUTAP begins by comparing the potential temperature of the 500 mb level to that of the level immediately below. If the potential temperature at the lower level exceeds that at the upper level, the temperature at the lower level is adjusted such that the potential temperature is 0.1°C less than the potential temperature at the upper level. OUTAP then compares the adjusted potential temperature at the lower level to that at the level immediately below in a similar manner, until the surface is reached.

Subroutine OUTAP is called only for standard upper-air times. FDDA times are written out by subroutine OUTFDA.

Subroutine OUTFDA(HORZH, IH1, IH2, IH3, SLAB1, SLAB2, SLAB3, PDIF12, IS1, IS2, MDATE, ICHK, LCHK, NVOL39, NGRID, SLAB4, MAXIC, MAXJC, NBOGUS, KBOGUS)

Purpose: Create an output volume (unit 39) of the analyzed surface fields at FDDA times (both standard and intermediate times).

On Entry:

PDIF12(IS1,IS2): Difference between sea-level pressure and surface pressure at 0000 or 1200 UTC.

MDATE: The time period currently being processed.

NVOL39: The unit number of the FDDA volume being created.

NGRID: Integer flag indicating whether coarse-grid (NGRID = 1) or fine-grid (NGRID = 2) analyses have been created. In the current version of RAWINS, NGRID = 1.

NBOGUS(18): Integer flags indicating whether bogus station reports have been included for a particular time period.

KBOGUS(18): Integer flags indicating whether bogus corrections to station reports have been made for a particular time period.

Arrays HORZH(IH1,IH2,IH3); SLAB1, SLAB2, SLAB3(IS1,IS2); and SLAB4(MAXIC,MAXJC) are all passed to OUTFDA as working space only.

On Exit: ICHK(LCHK): Integer flag indicating the density of observations. ICHK = 0 if fewer than 25 grid boxes contained observations. ICHK = 1 if 25 or more grid boxes contained observations.

Calls: Subroutines WTAPE and FILSLB.

Called by: Main program.

Comments: The FDDA output file is described in greater detail in Section IV.B.2.b. The header record is written out first, followed by the data records. For each data record, OUTFDA first calls FILSLB to retrieve the data from D1, and then calls WTAPE to write the data to FORTRAN unit 39.

Subroutine OUTPT(FLD, IYY, IA, IB, INY, JXX, JA, JB, JNX, KSIGT, NAME, ILB)

Purpose: Print out a sample of array FLD.

On Entry: FLD(JXX,IYY): Array of data (one level, one variable) to be printed out.

IA, IB: Initial and final points to print out in the I (y) direction.

JA, JB: Initial and final points to print out in the J (x) direction.

INY, JNX: Intervals at which data are to be printed in the I and J directions.

KSIGT: The number of significant digits to print out for each value.

NAME: Hollerith name for array FLD.

ILB: Integer to check the length of the label to be printed above the output field.

Calls: Subroutine VTRAN (not discussed).

Called by: Subroutines BLEND, PSFC, SFCBLN, and WTAPE.

Comments: Subroutine OUTPT first calls subroutine VTRAN, which transposes the array FLD so that the field will be printed out in the proper orientation. OUTPT then calculates the scaling factor for the field, based on the maximum absolute value in the field and the number of significant digits to be printed out for each value. The field is automatically scaled so that the largest absolute value will have KSIGT digits. Values more than KSIGT orders of magnitude less than the largest absolute value will be printed as positive or negative zero. Captions for the field are printed, followed by the field itself. As the scaled data values are printed, labels showing the I and J indices are printed at the boundaries of the field. Finally, the array is transposed (subroutine VTRAN) back to its original state.

Subroutine PLOTAB(F1, F2, XLOND, DUM, DUMA, DUMB, SLAB, SLP, PCHOOSE, ISTART, IEND, JSTART, JEND, IFILES, NVOLDUM, IPLTCNT)

Purpose: Create plots of the analyses with suspect data (ignored during the objective analysis) superimposed (*e.g.* Fig. 4.3) for the Autobogus 1 case. This routine calls numerous NCAR-Graphics subroutines, as well as several RAWINS subroutines. Most of the RAWINS subroutines called by PLOTAB deal specifically with generating the plots, and are not discussed in this document.

Calls: Numerous NCAR-Graphics subroutines.

RAWINS subroutines DECOMDAT, PUTSLAB, PAD, PLGRID, and SKIPF.

Called by: Main program.

Subroutine PLSOND(IDENT, NNNNN, P4, TT, TD, NWWWW, PW, WD, WS, LAT, LONG, YEAR, MONTH, DAY, HOUR)

Purpose: Plot soundings in T, p^κ coordinates (*e.g.* Fig.4.2).

On Entry:

IDENT:	Five-digit upper-air station identification number.
NNNNN:	Number of temperature and dewpoint data levels.
P4(100):	Array holding pressures (mb) at temperature levels.
TT(100):	Array holding temperatures (°C)
TD(100):	Array holding dewpoints (°C)
NWWWW:	Number of wind data levels.
PW(100):	Array holding pressures (mb) at wind levels.
WD(100):	Array holding wind directions (degrees).
WS(100):	Array holding wind speeds (m s^-1).
LAT:	Latitude of the upper-air station.
LONG:	Longitude of the upper-air station.
YEAR:	Year of the observation.
MONTH:	Month of the observation.
DAY:	Day of the observation.
HOUR:	Hour of the observation (UTC).

Calls: Subroutine PLSOND calls many NCAR-Graphics subroutines.

Called by: Subroutine SIGDAT.

Comments: PLSOND uses standard NCAR-Graphics subroutines. Temperature, dewpoint, and winds are plotted for all mandatory and significant levels on a Stüve diagram. One frame is created for each upper-air report within the map domain, if the logical flag IPLOT (Local Master Input File) is .TRUE.

The data plotted have been checked for vertical consistency, but have not been compared to the first-guess fields or to surrounding data. The upper-air report may be changed by objective checks later in the program.

Subroutine PROBGS(DATAH, CORRH, ID1, ID2, ID3, ITOSS, IMAND, ISIGT, ISIGW, ISFCD, KTOPPR, KMANPR, KSIGPR, KWINDP, KWINDZ, KREMEM, DATAS, IDS1, IDS2, IDS3, SLAB1, IS1, IS2, MDATE, ISI)

Purpose: Access all bogus surface reports for a time period and put the data into DATAS.

On Entry:

DATAS(IDS1,IDS2,IDS3): Array containing all the surface-station data processed thus far. On entry, DATAS contains no bogus data.

ISFCD: Integer identifying surface data. ISFCD = 4.

MDATE: Date/time currently being processed.

DATAH and CORRH(ID1,ID2,ID3), and SLAB1(IS1,IS2) are passed to PROBGS as working space only.

Arguments KTOPPR, KMANPR, KSIGPR, KWINDP, KWINDZ, KREMEM, and ISI are all integer flags used in processing upper-air data.

On Exit: DATAS has been updated to include all bogus surface reports.

Calls: Subroutines INACCT and SAVSTN.

Called by: Subroutine GETRAW.

Comments: Unlike PROSFC, which is called once for each surface-station report, PROBGS is called once to process all bogus surface reports for a particular time period.

PROBGS has a DO-loop cycling through all the bogus surface data in one time period. PROBGS first reads bogus header information from the nbogus file, unit 13. If the date/time of the bogus data matches the date/time currently being processed, the bogus data are read. If the date/time does not match, the report is skipped and another header record is read. Once PROBGS has found bogus data which should be included in the dataset, it calls INACCT to update accounting arrays for data storage and retrieval. SAVSTN is then called to put the bogus data into DATAS. Processing continues until all bogus surface reports from one time period have been read.

Subroutine PROBGU(DATAH, CORRH, ID1, ID2, ID3, ITOSS, IMAND, ISIGT, ISIGW, ISFCD, KTOPPR, KMANPR, KSIGPR, KWINDP, KWINDZ, KREMEM, DATAS, IDS1, IDS2, IDS3, SLAB1, IS1, IS2, MDATE, ISI)

Purpose:	Access all bogus upper-air reports for one time period and put the data into DATAH (for mandatory-level data) and CORRH (for significant-level data).	
On Entry:	DATAH(ID1,ID2,ID3):	Array holding all the mandatory-level data processed thus far. On Entry, DATAH contains no bogus data.
	CORRH(ID1,ID2,ID3):	Array holding all the significant-level data processed thus far. On Entry, CORRH contains no bogus data.
	IMAND:	Integer flag indicating mandatory-level data. IMAND = 1.
	ISIGT:	Integer flag indicating significant-level temperature and humidity data. ISIGT = 2.
	ISIGW:	Integer flag indicating significant-level wind data. ISIGW = 3.
	ISFCD:	Integer flag identifying surface data. ISFCD = 4.
	DATAS(IDS1,IDS2,IDS3)	is used for processing surface data.
On Exit:	ITOSS:	Integer flag, ITOSS = 1 if station is outside the domain.
	KTOPPR:	Integer flag, KTOPPR = 1 if the data from the bogus report extend to PTOP.
	KMANPR:	Integer flag, KMANPR = 1 if mandatory-level data are included in the bogus report.
	KSIGPR:	Same as KMANPR, except for significant-level temperature and humidity data.
	KWINDP:	Same as KMANPR, except for wind data at significant pressure levels.
	KWINDZ:	Same as KMANPR, except for wind data at heights.
	KREMEM:	Integer print flag for mandatory levels.
	DATAH and CORRH have been updated to include the upper-air bogus data.	
Calls:	Subroutines INACCT, SAVFIL, and SAVSTN.	
Called by:	Subroutine GETRAW.	

Comments: Unlike subroutine PROUPR, which is called once for each upper-air report, subroutine PROBGU is called once to process all bogus upper-air reports for a particular time period.

PROBGU has one loop which cycles over all bogus upper-air reports in the time period. A second loop processes a report's mandatory-level data on its first iteration, and the same report's significant-level data on its second iteration.

For each bogus report, PROBGU reads a header record from the nbogus file, unit 13. If the date/time of the bogus report does not match the date/time currently being processed, the bogus report is skipped and another header record is read. If the date matches, but the order of the data is inconsistent, the bogus report is skipped and another header record is read. When PROBGU has found a bogus report which should be included in the dataset, PROBGU reads the data and calls INACCT to update accounting arrays for data storage and retrieval. SAVSTN is then called to store the mandatory-level bogus data to the DATAH array, and significant-level bogus data to CORRH. Processing continues until all mandatory- and significant-level bogus data have been included in the DATAH and CORRH arrays.

Subroutine PROSFC(DATAH, CORRH, ID1, ID2, ID3, ITOSS, IMAND, ISIGT, ISIGW, ISFCD, KTOPPR, KMANPR, KSIGPR, KWINDP, KWINDZ, KREMEM, DATAS, IDS1, IDS2, IDS3, SLAB1, IS1, IS2, MAXLEV, ISI)

Purpose: Put observations from one surface station into DATAS.

On Entry: DATAS(IDS1,IDS2,IDS3): Array holding all the surface station data processed thus far. On entry, DATAS does not include data from the station currently being processed.

ISFCD: Integer flag indicating surface data. ISFCD = 4.

ISI: Integer flag indicating whether the DATAS array has been cleared.

DATAH and CORRH, both dimensioned (ID1,ID2,ID3), are used for upper-air processing.

SLAB1(IS1,IS2) is passed to PROSFC only as working space.

IMAND, ISIGT, ISIGW, KTOPPR, KMANPR, KSIGPR, KWINDP, KWINDZ, KREMEM, and MAXLEV are all integers used in the processing of upper-air data.

On Exit: DATAS has been updated to include the data from the particular surface station report currently being processed.

Calls: Subroutines INACCT, UNISF, SFCADP, and SAVSTN.

Called by: Subroutine GETRAW.

Comments: PROSFC is called when GETRAW has read and checked the header record of a surface-station report and has determined that the station's data should be included in the set of observations. PROSFC is called once for each surface-station report.

First, PROSFC calls INACCT to update accounting arrays for data storage and retrieval. Then, either UNISF (for real-time data) or SFCADP (for historical data) is called to retrieve the station's data. Finally, SAVSTN is called to transfer the station's data to DATAS.

Subroutine PROUPR(IPLANE, DATAH, CORRH, ID1, ID2, ID3, ITOSS, IMAND, ISIGT, ISIGW, ISFCD, KTOPPR, KMANPR, KSIGPR, KWINDP, KWINDZ, KREMEM, DATAS, IDS1, IDS2, IDS3, SLAB1, IS1, IS2, MAXLEV, ISI)

Purpose: Process one upper-air report; put mandatory-level data into DATAH; put significant-level data into CORRH.

On Entry:

IPLANE:	The number of aircraft reports accessed thus far. Should be 0.
DATAH(ID1,ID2,ID3):	Array holding all the mandatory-level data processed thus far.
CORRH(ID1,ID2,ID3):	Array holding all the significant-level data processed thus far.
IMAND:	Integer identifying mandatory-level data. IMAND = 1.
ISIGT:	Integer identifying significant-level temperature and humidity data. ISIGT = 2.

	ISIGW:	Integer identifying wind data at heights. ISIGW = 3.
	ISFCD:	Integer identifying surface data. ISFCD = 4.
	MAXLEV:	Maximum number of levels that can be processed by RAWINS.
	ISI:	Integer flag indicating whether the DATAS array has been initialized.
	From Common/B/:	IREC, IRTYP, INSTYP, IYR, IMO, IDY, IHR, TIME, SSTA, YLAT, YLON, ELEV. (See subroutine RDADP.)
		The argument DATAS(IDS1,IDS2,IDS3) is used for the processing of surface data.
		SLAB1(IS1,IS2) is passed to PROUPR only for working space.
On Exit:	DATAH(ID1,ID2,ID3):	Array holding all the mandatory-level data processed thus far, including data from the current upper-air report.
	CORRH(ID1,ID2,ID3):	Array holding all the significant-level data processed thus far, including data from the current report.
	ITOSS:	Integer flag; ITOSS = 1 if the station is outside the domain.
	KTOPPR:	Integer flag. KTOPPR = 1 if the data from this report extend to PTOP; KTOPPR = 0 if the data do not extend to PTOP.
	KMANPR:	Integer flag. KMANPR = 1 if mandatory level data have been found for this report; KMANPR = 0 if mandatory level data have not been found.
	KSIGPR:	Same as KMANPR, except for significant-level data.
	KWINDP:	Same as KMANPR, except for pressure-level wind data.
	KWINDZ:	Same as KMANPR, except for wind data at heights.

KREMEM: Integer print flag for mandatory levels.

Calls: Subroutines INACCT, UNIUP, MANADP, SIGADP, SAVSTN, SAVFIL, WPPADP, and WZZADP.

Called by: Subroutine GETRAW.

Comments: Subroutine PROUPR is called when GETRAW has found an upper-air report which should be included in the observational dataset.

For upper-air reports, PROUPR calls subroutine INACCT to update accounting arrays for data storage and retrieval to and from array D1. Next, the actual meteorological data is retrieved by subroutine UNIUP for real-time input; or subroutines MANADP, SIGADP, WPPADP, and WZZADP for historical input. The heights of the wind data are converted to pressure levels using logarithmic interpolation from bracketing pressure levels. All data are stored in the DATAH (for mandatory-level data) or CORRH (for significant-level data) arrays.

Subroutine PSFC(HORZH, IH1, IH2, IH3, SLAB1, SLAB2, SFCPR, IS1, IS2, IP)

Purpose: Calculate surface pressure at dot and cross points from sea-level pressure; terrain height; 850 mb height, and 850, 700, and 500 mb temperatures.

On Entry: IP: Integer flag; IP = 1.

Arrays HORZH(IH1,IH2,IH3), SLAB1(IS1,IS2), and SLAB2(IS1,IS2) are all passed to PSFC as working space only.

On Exit: SFCPR(IS1,IS2): Array holding surface-pressure field (at cross points) as calculated by this subroutine. SFCPR has also been transferred to storage array D1.

Calls: Subroutines FILSLB, HRZFIL, SLBFIL, DOTS, and OUTPT.

Called by: Main program.

Comments: First, PSFC calls subroutine FILSLB to retrieve the sea-level pressure field from storage array D1. PSFC calls HRZFIL five times to put the terrain heights; the 850, 700, and 500 mb temperatures; and the 850 mb geopotential heights into the 5 levels of 3-d array HORZH (IH1,IH2,IH3).

Subroutine PSFC calculates surface pressure p_s by using the hypsometric equation with a fictitious mean temperature $\overline{T}$ between the surface and sea level:

$$p_s = p_o \exp\left[\frac{-gz_s}{R_d\overline{T}}\right], \tag{4.27}$$

where p_o is the sea-level pressure, g is the acceleration due to gravity, z_s is the terrain height, and R_d is the gas constant of dry air. The fields p_o and z_s have been retrieved directly from D1, but $\overline{T}$ must be derived.

The fictitious mean temperature $\overline{T}$ is defined as:

$$\overline{T} = \frac{T_o' + T_s'}{2}, \tag{4.28}$$

where T_o' is a fictitious sea-level temperature, and T_s' is an approximate surface temperature without serious diurnal effects. The actual surface temperature is inappropriate for this calculation because of large diurnal fluctuations in temperature near the ground, so a representative surface temperature T_s' must be derived.

To obtain the fictitious surface temperature T_s', the chain rule and the equation of state are applied to the hydrostatic equation so that the atmospheric lapse rate, $\gamma = -\frac{\partial T}{\partial z}$, can be written as

$$\gamma = \frac{pg}{R_dT}\frac{\partial T}{\partial p} \quad = \quad \frac{g}{R_d}\frac{\partial \ln T}{\partial \ln p}. \tag{4.29}$$

Next, a preliminary estimate of surface pressure, p_s', is obtained from the terrain height z_s and the 850 mb height, z_{850}, by assuming that pressure is logarithmically related to height:

$$p_s' = p_o\left(\frac{p_o}{p_{850}}\right)^{-\frac{z_s}{z_{850}}}. \tag{4.30}$$

While this method is not very precise, it is not necessary that p_s' be very accurate because it is only used to estimate a pressure level ℓ that is above the region of substantial diurnal temperature fluctuations. Estimated values of diurnally independent sea-level temperature, T_o', and surface temperature, T_s' can then be found by downward integration from the reference level ℓ. A preliminary fictitious sea-level temperature T_o can be calculated from

$$T_o = T_\ell\left(\frac{p_o}{p_\ell}\right)^{\gamma_s\frac{R}{g}}, \tag{4.31}$$

where the standard atmospheric lapse rate $\gamma_s = 6.5 \times 10^{-3}$K m^{-1}; the pressure p_ℓ is defined as

$$p_\ell = \begin{cases} 850 \text{ mb} & \text{for } p_s' \geq 950 \text{ mb}, \\ p_s - 100 \text{ mb} & \text{for } p_s' < 950 \text{ mb}; \end{cases} \tag{4.32}$$

and the temperature at level ℓ is given by

$$T_\ell = \begin{cases} T_{850} & \text{for } p_s' \geq 950 \text{ mb}, \\ T_{700}\left(\frac{p_\ell}{p_{700}}\right)^{\gamma_{78}\frac{R_d}{g}} & \text{for } 950 > p_s' \geq 850 \text{ mb}, \\ T_{500}\left(\frac{p_\ell}{p_{500}}\right)^{\gamma_{57}\frac{R_d}{g}} & \text{for } 850 \text{ mb} > p_s'. \end{cases} \tag{4.33}$$

The values of the lapse rates, γ_{78} and γ_{57} are found by integrating (4.29) between 700 and 850 mb, and between 500 and 700 mb, respectively. This integration gives the result

$$\gamma_{78} = \frac{g}{R_d}\left[\frac{\ln(T_{850}/T_{700})}{\ln(p_{850}/p_{700})}\right]$$

$$\gamma_{57} = \frac{g}{R_d}\left[\frac{\ln(T_{700}/T_{500})}{\ln(p_{700}/p_{500})}\right]. \tag{4.34}$$

From the value T_o in (4.31), an approximate surface temperature without serious diurnal effects can be found according to

$$T_s' = T_o - \gamma_s z_s. \tag{4.35}$$

To obtain the fictitious sea-level temperature T_o', for use in equation (4.28), subroutine PSFC uses the following method. By choosing a critical temperature of $T_c = 290.66$ K (17.5°C), the value T_o' is defined as

$$T_o' = \begin{cases} T_o & \text{for } T_s' \leq T_c,\ T_o < T_c; \\ T_c & \text{for } T_s' \leq T_c,\ T_o \geq T_c; \\ T_c - 0.005\,(T_s' - T_c)^2 & \text{for } T_s' > T_c,\ T_o \geq T_c. \end{cases} \tag{4.36}$$

The surface pressure may then be calculated from (4.27).

Once the surface pressure field at cross points has been calculated and put into array SFCPR, subroutine PSFC calls subroutine SLBFIL to transfer the surface-pressure field to the storage array D1. Then, subroutine DOTS (not discussed) is called to interpolate the surface pressure from cross points to dot points, and the field is transferred to the storage array D1. Finally, PSFC calls OUTPT to print out a sample of the surface-pressure field at cross points.

Subroutine RDADP(IUN, IST, YLONH, NYR, NMO, NDY, NHR)

Purpose: Read a packed station report from unit IUN, and return information including the time, location, and type of report. RDADP is called to retrieve both real-time data and historical data, but different procedures are followed for each case.

On Entry:

- IUN: Unit number of file to read. IUN may refer to a file of real-time data preprocessed by subroutine GETUNIOB, or it may refer to a file of historical data retrieved from the NCAR MSS.
- IST: Integer flag indicating whether more data is to be read from the current record (IST = 0) or an end-of-file mark has been encountered (IST = 1).
- UNIOBS: Logical flag indicating whether real-time observations from Unidata (UNIOBS = .TRUE.) or historical observations archived on the NCAR MSS (UNIOBS = .FALSE.) are to be accessed. Common/B/.

On Exit:

- IST: Integer flag indicating whether an end-of-file mark has been encountered (IST = 1) or not (IST = 0).
- YLONH: Longitude of the station, as originally read from the data file, in the range of 0° to 360° W.
- NYR: The year of the report.
- NMO: The month of the report.
- NDY: The day of the report.
- NHR: The hour of the report.

NB(820): The packed data record (for historical data), returned through Common/B/.

Additionally, a number of identification variables are returned through Common/B/.

Calls: NCAR Subroutines IOWAIT, GBYTES, SBYTES, RDTAPE.

RAWINS Functions LOGCMP and NTENS.

Called by: Subroutine GETRAW.

Comments: If Unidata observations are used (UNIOBS = .TRUE.), the data have been read and preprocessed by subroutine GETUNIOB and written to units 1 (for surface reports) and 2 (for upper-air reports). In this case, RDADP reads directly the station identifier, latitude, longitude, elevation, and the date/time of the report. For the upper-air observations, there are two additional variables LTHERM and LWIND, identifying the number of temperature and wind levels in the report. All of the variables read from the data file are returned to the calling routine (subroutine GETRAW) by Common/B/.

If historical observations are used (UNIOBS = .FALSE.), then the procedure is more complicated. Subroutine RDADP first checks to see whether a new header record must be read, or if more data can be expected in the current record. If a new header record must be read, RDADP calls NCAR subroutines RDTAPE and IOWAIT to read the record. The record is checked for end-of-record and end-of-file marks (subroutine LOGCMP). The record is also checked for the character string "WASHINGT" in the third word, which will identify the record as a header record. NCAR subroutine GBYTES and RAWINS function NTENS then extract from the packed record the date/time of the data records to follow.

After RDADP has unpacked the date/time variables from the header record, the subroutine repeats the calls to RDTAPE and IOWAIT to obtain a packed data record. NCAR subroutines GBYTES and SBYTES and RAWINS function NTENS extract variables containing station identifier, latitude, longitude, elevation, observation time, report type, and instrument type. The longitude extracted from the file is converted from the range of 0° to 360° W to the range of −180° to 180° E. All of the variables are returned to the calling routine (subroutine GETRAW) in Common/B/.

Each time RDADP is called, new data records will be read to obtain another station report. If another header record is encountered, RDADP simply treats it as described earlier and proceeds to the next data record.

Subroutine RDISK(NAME, HOLD, LEN, IFWA, NWDS, CHECK)

Purpose:	Transfer a block of data from the D1 storage array to the one-dimensional array HOLD. The block of data may correspond to either one horizontal slab of surface or upper-air data (*i.e.*, an analysis) or one level of observational data. Subroutine RDISK performs just the opposite task of subroutine WDISK.	
On Entry:	NAME:	Integer value identifying the block of data to be retrieved from D1.
	LEN:	Length of array HOLD.
	IFWA:	Address in HOLD of the first word of data to be retrieved from D1, minus 1.
	NWDS:	The number of words to be retrieved from D1.
	CHECK:	Logical flag; RDISK prints a brief message indicating the value of NAME, if CHECK = .TRUE.
On Exit:	HOLD(LEN):	One-dimensional array holding data for a two-dimensional field.
Calls:	Subroutine RDISK calls no other subprograms.	
Called by:	Subroutines BLEND, BOGPTS, HRZFIL, SEAPRS, and SFCBLN.	

Comments: When RDISK retrieves observational data, argument HOLD corresponds to array BUFUPR(5,IRB). Since the entire BUFUPR array is to be filled, arguments LEN and NWDS both have the value 5 $\times$ IRB, and argument IFWA has a value of 0.

When RDISK is retrieving a horizontal slab of data, argument HOLD corresponds to array HORZH(IH1,IH2,IH3). Thus, argument LEN has the value IH1 $\times$ IH2 $\times$ IH3. Although HORZH can hold up to five slabs, only one slab may be transferred to HORZH at one time. NWDS, then, represents the size of one slab (NWDS = IH1 $\times$ IH2). Argument IFWA defines which slab (third dimension of HORZH) is to be filled.

Subroutine RTAPE(L, IVRBL, SLAB, IS1, IS2, NVOL, LP)

Purpose: Read a horizontal slab (one variable at one level) of input data (first-guess fields) from unit NVOL to the array SLAB.

On Entry:	L:	Level assigned to the slab of data. L is assigned in the calling subroutine; its value is used to label samples of the data printed out.
	IVRBL:	Integer representing the variable name assigned to the slab of data read from unit NVOL.
	NVOL:	Unit number from which data are to be read.
	LP:	Pressure-level index of the pressure from array IPRES(16). LP is used to label samples of the data printed out.

On Exit: SLAB(IS1,IS2) is filled with the data from unit NVOL, and is assigned the variable name IVRBL and level L.

Calls: Subroutine OUTPT.

Called by: Subroutine INHRZ1.

Comments: If the print option DRAWI is activated (in subroutine SETUP), a sample of the field is printed out (by subroutine OUTPT). If DRAWI is not activated, only the first element of the array SLAB will be printed out, along with a brief message identifying the data.

Subroutine SAVFIL(FILEH, ID1, ID2, ID3, ISTATN, ITYP, IDS3)

Purpose: Transfer the array FILEH containing upper-air data to the storage array D1.

On Entry:	FILEH(ID1,ID2,ID3):	Array containing upper-air data to be stored. SAVFIL is called with argument FILEH corresponding to DATAH (for mandatory-level data) or CORRH (for significant-level data).
	ISTATN:	The total number of stations processed thus far.
	ITYP:	Type of upper-air data. 1: mandatory-level data; 3: significant-level data.
	IDS3:	Unused in SAVFIL.

On Exit: The data in FILEH have been transferred to D1.

Calls: Subroutine WDISK.

Called by: Subroutines GETRAW, PROBGU, and PROUPR.

Comments: In SAVFIL, each vertical level in the FILEH array is transferred individually to D1. For each vertical level, the data are first transferred from FILEH (ID1,ID2,ID3) to BUFPR(ID1,ID2). A unique integer name (a function of level) is calculated for each vertical level, to serve as an identifier for later retrieval of the data. WDISK is then called to transfer that level of data from BUFPR to D1.

Subroutine SAVSTN(FILEH, ID1, ID2, ID3, ITOSS, ITYP, KTOPPR, KMANPR, KSIGPR, KWINDP, KWINDZ, KREMEM, FILES, IDS1, IDS2, IDS3, SLAB1, IS1, IS2, ISI)

Purpose: Transfer observations from arrays holding the observations from one station report to the arrays DATAH or CORRH (corresponding to argument FILEH for upper-air data), or DATAS (corresponding to argument FILES for surface data). SAVSTN performs miscellaneous transformations to the data compatible with other conventions within RAWINS.

On Entry:

FILEH(ID1,ID2,ID3): Array containing upper-air data. SAVSTN is called with FILEH corresponding to either DATAH (for mandatory-level data) or CORRH (for significant-level data).

ID1: First index of upper-air data array, represents the number of variables accessed. ID1 = 5. The index levels are used as follows:
1: u (m s^{-1})
2: v (m s^{-1})
3: T (°C)
4: RH (%)
5: height (m)

ID2: Second index of upper-air data array, represents the maximum number of levels (mandatory or significant) for any particular upper-air report. ID2 = 55.

ID3: Third index of upper-air data array, represents the maximum number of upper-air stations that may be processed. ID3 = IRB, set by a parameter statement.

ITYP: Integer index, identifying the type of data to be processed:

1: mandatory-level data.
2: significant-level T and RH.
3: significant-level wind data.
4: surface-station data.

	FILES(IDS1,IDS2,IDS3):	Array containing surface data. SAVSTN is called with FILES corresponding to array DATAS.
	IDS1:	First index of surface data array, represents the number of variables accessed. IDS1 = 5.
	IDS2:	Second index of surface data array; IDS2 = 2 for 2 different categories of data held in FILES. The first category is the latitude, longitude, and elevation data for each station; The second is the meteorological data.
	IDS3:	Third index of surface data array; IDS3 = IRS, the maximum number of surface and upper-air reports in FILES.
	IFOUND(3,IRS):	Integer flag identifying data as real (IFOUND = 1) or bogus (IFOUND = 2). Common/C/.
On Exit:	ITOSS:	Integer flag; ITOSS = 1 if station is outside domain.
	KTOPPR:	Integer flag; KTOPPR = 1 if station data extend to PTOP.
	KMANPR:	Integer flag; KMANPR = 1 if mandatory-level data were found for this station.
	KSIGPR:	Integer flag; KSIGPR = 1 if significant-level temperature and humidity data were found for this station.
	KWINDP:	Integer flag; KWINDP = 1 if pressure-level wind data were found for this station.
	KWINDZ:	Integer flag; KWINDZ = 1 if wind data at heights were found for this station.
	KREMEM:	Integer print flag for mandatory levels.

On completion of SAVSTN, the data have been transferred to FILEH or FILES, which at this point contain all the upper-air data (FILEH) and surface data (FILES) processed thus far.

Calls: Subroutine SAVSTN calls no other subprograms.

Called by: Subroutines PROUPR, PROSFC, PROBGU, and PROBGS.

Comments: Values of temperature, geopotential height, and pressure do not require conversion and are transferred directly into the FILEH array. Gross error checks are made so that heights above 40000 m, pressures above 1100 mb, and temperatures above 100°C will not be stored. More precise checks are made during the objective analysis.

Dewpoint depressions are converted to relative humidity (%) according to:

$$RH = 100\% \times \exp\left[\frac{m_v}{m_a}\frac{L}{R_d}\left(\frac{1}{T} - \frac{1}{T_d}\right)\right], \tag{4.37}$$

where $\frac{m_a}{m_v} = 1.61$ is the ratio of the molecular weights of air to water vapor, $L = 2.5 \times 10^6$ J kg^{-1} is the latent heat of condensation, R_d is the gas constant for dry air (287 J kg^{-1} K^{-1}), T is the temperature (K), and T_d is the dewpoint (K). Dewpoint depression $T - T_d$ must be less than 100 K. Relative humidities greater than 99.9% are set to 99.9%. Relative humidities less than 10% are set to 10%.

Winds are converted from speed and direction to u and v components in the coordinate system of the mesoscale map. An intermediate step converts winds from speed and direction to u and v components in the natural earth coordinates (u_e, v_e):

$$u_e = -V_e \sin A_e \qquad \text{and} \qquad v_e = -V_e \cos A_e, \tag{4.38}$$

where V_e is the reported wind speed (from array F, Common/C/), and A_e is the reported wind angle (from array D, Common/C/).

Conversion to map coordinates is then accomplished by

$$u = r_k \left(v_e \sin\alpha_m + u_e \cos\alpha_m\right)$$

and

$$v = r_k \left(v_e \cos\alpha_m - u_e \sin\alpha_m\right), \tag{4.39}$$

where $r_k = 0.5144$ is the conversion factor for m s^{-1} to knots (1 kt = 0.5144 m s^{-1}), and α_m is given by

$$\alpha_m = n\left(\lambda_c - \lambda_o\right) \tag{4.40}$$

where n is the cone constant for the projection, λ_c is the central longitude of the grid, and λ_o is the longitude of the observation. α_m is the angle between the wind direction in earth coordinates and the direction in the mesoscale grid coordinates.

Since significant-level wind data are processed after significant-level temperature and moisture, pressure levels of the significant-level temperature and wind reports are scanned so that the significant level winds may be inserted into the proper location within FILEH. Data at higher levels are shifted to greater indices to make room. If the number of significant levels becomes too large, data will be lost at the highest levels and a warning message to that effect is printed.

Once the meteorological data have been transformed and stored in the holding array, the latitude and longitude of the station must be converted to gridpoint values for both the dot and cross point meshes. The dot point grid locations for station S, in terms of the grid indices I and J, is calculated for the Mercator projection as

$$\mathrm{I}_S = \frac{\left[-C_2 \ln\left(\frac{\cos\phi_o}{1+\sin\phi_o}\right) + y_D\right]}{\Delta s} + 1$$

and

$$(4.41)$$

$$\mathrm{J}_S = \frac{C_2\left(\lambda_o - \lambda_c\right) + x_D}{\Delta s} + 1,$$

where ϕ_o is the latitude of the observation, Δs is the grid spacing, and x_D and y_D are the distances from the lower left corner of the grid to the center in the x and y directions, respectively. The term C_2 is given by

$$C_2 = R_e \cos\phi_c, \tag{4.42}$$

where $R_e = 6370$ km is the earth's radius and $\phi_c = 0.0$ is the latitude at which the Mercator projection is exact.

For Lambert conformal and polar stereographic projections, the equations are somewhat different. The positions of the dot points in terms of the grid indices are given by

$$\mathrm{I}_S = \frac{C_1 \cos\lambda_1 + y_D}{\Delta s} + 1$$

and

$$\mathrm{J}_S = \frac{-C_1 \sin\lambda_1 + x_D}{\Delta s} + 1, \tag{4.43}$$

where C_1 for a Lambert conformal projection is given by

$$C_1 = \frac{-R_e}{n} \sin\left(\frac{\pi}{2} - \phi_c\right) \left[\frac{\frac{\tan\left(\frac{\pi}{2} - \phi_o\right)}{2}}{\frac{\tan\left(\frac{\pi}{2} - \phi_c\right)}{2}}\right]^n, \tag{4.44}$$

where ϕ_c is the latitude for which the projections are exact, and $n = 0.716$. The angle λ_1 is defined as

$$\lambda_1 = n\left(\lambda_o - \lambda_c\right). \tag{4.45}$$

For the polar stereographic projection the equations (4.43), and (4.45) are valid, but $n = 1.0$ and

$$C_1 = -R_e \sin\left(\frac{\pi}{2} - \phi_o\right) \left[\frac{1 + \cos\left(\frac{\pi}{2} - \phi_c\right)}{1 + \cos\left(\frac{\pi}{2} - \phi_o\right)}\right]. \tag{4.46}$$

The cross point locations for the station are obtained by making a one-half grid increment correction to the dot point locations J_S and I_S. SAVSTN uses slightly modified forms of (4.41) through (4.46) to calculate positions in the southern hemisphere.

Once the position of the station in the grid has been determined, a check is made to determine if the station is within the domain of the mesoscale grid. If the station is outside the domain, the station counter, ISTATN or JSTATN, is decremented, and the data are removed from FILEH or FILES. If the station is within the mesoscale domain, the terrain elevation is added to the data stored in FILEH and the station calculations are complete. The data for the station are printed out. Any missing data are represented by asterisks in the output.

Subroutine SEAPRS(LENBUF, N, NF, NL)

Purpose: Convert surface pressure (from upper-air reports) to sea-level pressure.

On Entry:

LENBUF: Number of data for one level of upper-air observations. LENBUF = ID1 × ID3; *i.e.*, 5 variables × IRB stations.

N: File number; N = 1.

NF: Sequential number of the first station in the file; NF = 1.

NL: Sequential number of the last station in the file.

BUFD(5,IRS): Station elevations and locations. Common/C/.

RSOND(5,IRS): Surface observations from surface and upper-air stations. Common/C/. RSOND holds surface pressure for upper-air stations; sea-level pressure for surface stations, and sea-level pressure for bogus stations.

On Exit:

RSOND(5,IRS): Surface observations from surface and upper-air stations, with sea-level pressures replacing the original surface pressures.

Calls: Subroutines RDISK and WDISK.

Called by: Subroutine SFCBLN.

Comments: Subroutine SEAPRS converts surface pressures to sea-level pressures by a procedure similar to that used in subroutine PSFC: applying the hypsometric equation with a fictitious mean temperature between sea level and the surface.

SEAPRS first copies the surface-level rawinsonde data from array RSOND into array BUFSFP(5,IRB). The 850, 700, and 500 mb data are retrieved from D1 and copied to arrays PSOND1(5,IRS), PSOND2(5,IRS), and PSOND3(5,IRS), respectively.

The sea level pressure p_o is calculated by

$$p_o = p_s \exp\left(\frac{g z_s}{R_d \overline{T}}\right), \qquad (4.47)$$

where, as in subroutine PSFC, p_s is the surface pressure, g is the acceleration due to gravity, z_s is the terrain height, R_d is the ideal gas constant for dry air, and $\overline{T}$ is the fictitious mean temperature between sea level and the surface. For bogus reports, which supply sea-level pressure p_o, the surface pressure p_s is calculated by

$$p_s = p_o \exp\left(-\frac{g z_s}{R_d \overline{T}}\right). \tag{4.48}$$

$\overline{T}$ is defined, as in subroutine PSFC, by

$$\overline{T} = \frac{T_s' + T_o'}{2}, \tag{4.49}$$

where T_s' is the fictitious surface temperature without diurnal effects, and T_o' is the fictitious sea level temperature.

To find the fictitious surface temperature T_s', the actual surface pressure p_s is substituted for the approximate surface pressure p_s' in equations (4.30) through (4.33) of subroutine PSFC. Those equations give the reference-level temperature T_ℓ and pressure p_ℓ at a given station. T_s' is then calculated as

$$T_s' = T_\ell \left(\frac{p_s}{p_\ell}\right)^{\gamma_s \frac{R}{g}}. \tag{4.50}$$

To find the fictitious sea level temperature T_o', an approximate sea-level temperature is first estimated as

$$T_o = T_s' + \gamma_s z_s. \tag{4.51}$$

Then, equation (4.36) of subroutine PSFC are applied to find T_o'. Sea-level pressure may then be calculated by (4.47).

If the station's surface pressure or any of the key temperature observations are missing, SEAPRS cannot determine the sea-level pressure and the dummy value 1.0×10^{34} will be used.

The values of p_o calculated in SEAPRS are returned (to SFCBLN) in array RSOND, in place of the original surface pressures. The original surface pressures and any new surface pressures calculated from the bogus sea-level pressures are put into array BUFSFP and transferred to the storage array D1 by subroutine WDISK.

Subroutine SETANA(A1, A2, A3, ASTA, ASTA1, XOBS, YOBS, IMAX, JMAX, NSTA, DS, RIN, IFAC, IVRBL, IFD, ERRMXW, ERRMXT, ERRMXP, IWTSCM, BETA, PL, NFI, ISTNUM, COR, NS, SUM, SUM2, ASTAWD, KTYPE, LVL, MDATE, AUTBGW, AUTBGR, NVOL40, RANG, IFDA, LAGTEM)

Purpose: Perform the objective analysis for one variable at one level. Subroutine SETANA is a driver routine which manages subroutines GETERR and ANALMN.

On Entry:

ASTA (IRS): Observations of the variable being processed at the level being analyzed. ASTA holds observations of u, v, p_o, T, or RH. When wind is being analyzed with the Banana scheme, ASTA holds v and ASTA1 holds u.

ASTA1(IRS): Used only when the wind is analyzed with the Banana scheme. ASTA1 holds u and ASTA holds v.

ASTAWD(IRS): When the wind field is analyzed, ASTAWD holds the complementary component to the component held in ASTA. If ASTA holds u, ASTAWD holds v. If ASTA holds v, then ASTAWD holds u.

A1(IMAX,JMAX): Current-guess field. If the wind is analyzed with the Banana scheme, A1 holds the current guess v, and A3 holds the current-guess u.

A3(IMAX,JMAX): Used only when the wind is analyzed with the Banana scheme. In that case, A3 holds the current-guess u, and A1 holds the current-guess v.

XOBS(IRS): x-coordinates of observations.

YOBS(IRS): y-coordinates of observations.

NSTA: Number of stations in the observational dataset.

DS: Mesoscale grid spacing (m).

RIN: Radius of influence (in grid increments).

IFAC: Hollerith flag: IFAC = 6HANALYS if a new first-guess field is to be created; IFAC = 6HALTERS if an existing first-guess field is to be improved.

IVRBL: Integer flag identifying the variable being processed.

IFD(IRS): Array of integer flags identifying the stations as real

(IFD(n) = 1) or bogus (IFD(n) = 0).

ERRMXW: Maximum allowable difference between the first-guess wind components and the observations.

ERRMXT: Maximum allowable difference between the first-guess temperature field and the observations.

ERRMXP: Maximum allowable difference between the first-guess surface pressure field and the observations.

IWTSCM: Hollerith flag identifying the weighting scheme to be used:
- 6HCRESMN: Circular region of influence;
- 6HELLIPS: Elliptical region of influence;
- 6HBANANA: Banana-shaped region of influence.

BETA: Elongation factor for ellipses and bananas.

NFI: File number; should always be 1.

ISTNUM(3,IRS): Array holding station identifiers.

LVL: Vertical level index. (1 = Surface; nonmandatory levels have higher indices than mandatory levels.)

MDATE: Current date/time being processed.

AUTBGW: Logical flag, AUTBGW = .TRUE. when the autobogus correction file is to be written (*i.e.* the Autobogus 1 submittal).

AUTBGR: Logical flag, AUTBGR = .FALSE. when the autobogus correction file is to be read (*i.e.* the Autobogus 2 submittal).

NVOL40: Unit number of the autobogus correction file (output).

RANG(3,IRS): Array holding angles for conversion of observed winds (in meteorological coordinates) to map coordinates.

IFDA: The FDDA number of the current date/time.

LAGTEM: Logical flag, indicating whether FDDA first-guess fields are from a 3 h lag time from the previous analyzed field (LAGTEM = .TRUE.) or temporal interpolation from standard times (LAGTEM = .FALSE.).

On Exit: A1(IMAX,JMAX): The improved field.

A3(IMAX,JMAX): Used only when the wind field is being analyzed by the Banana scheme. In that case, A1 holds the improved v field, and A3 holds the improved u field.

Calls: Subroutines GETERR and ANALMN.

Called by: Subroutines BLEND and SFCBLN.

Comments: Subroutine SETANA receives a first-guess field and a set of observations for a particular variable and level, and drives the objective analysis process. SETANA first sets the number of passes to be made to nudge the analysis towards the observations. For u, v, T, and p_o, the number of passes, NSCAN, is four. For RH, NSCAN is set to five. If IFAC = 6HANALYS, one initial pass is made to create a first-guess field. NSCAN is then set to five for u, v, T, and p_o; and six for RH. a DO-loop then performs the successive scans while the critical radius of influence around the observations is decreased on each pass (the radius for a pass is decreased to 70% of the radius of the previous pass).

Each iteration of the DO-loop calls subroutines GETERR and ANALMN. GETERR calculates the difference value between the observations and the current guess at each station. ANALMN uses the difference values to calculate correction values at each gridpoint, and applies those correction values to improve the current guess.

When winds are analyzed by the Banana scheme, both wind components are processed in one call to SETANA. Each iteration of the DO-loop calls both GETERR and ANALMN twice, once for each of the wind components.

Subroutine SETUP(ID1, ID2, ID3, IH1, IH2, IH3, IS1, IS2, IFILE, IFDA, RWSUBM)

Purpose: Initialize a number of miscellaneous variables and arrays; print out numerous parameters to inform the user what options have been selected; make numerous checks to insure that the dimensioning, equivalencing, Master Input, and PARAMETER declarations are all compatible.

On Entry: ID1,ID2,ID3: Dimensions of DATAH and CORRH arrays, which hold the observations.

IH1,IH2,IH3: Dimensions of the HORZH working array.

IS1,IS2: Dimensions of the many horizontal slab arrays which hold data on the mesoscale grid.

	IFILE:	The number of the current date/time period being processed. SETUP is called only when IFILE = 1.
	IFDA:	The number of the FDDA time period being processed. SETUP is called only when IFDA = 1.
	RWSUBM:	Integer flag; RWSUBM = 1 for an Autobogus 1 submittal; RWSUBM = 2 for an Autobogus 2 submittal; RWSUBM = 0 otherwise.
On Exit:	IMAX:	Recalculated to be the I (y) dimension of the expanded grid if the expanded grid is selected. Common/A/.
	JMAX:	Recalculated to be the J (x) dimension of the expanded grid if the expanded grid is selected. Common/A/.

Calls: Subroutine HEDROT.

Called by: Main program.

Comments: Although the main program places subroutine SETUP inside the two main loop structures, SETUP is called only once during the program's execution.

Subroutine SETUP calls subroutine HEDROT, which creates several arrays defining pressure levels. HEDROT creates the arrays GLVL, LVL, and IHEDPR; and initializes related variables LEVEL1, NVERT, and NTOTLV. See the discussion of subroutine HEDROT for further details.

If the expanded grid is selected (Common Master Input File option IEXP), SETUP calculates the dimensions of the expanded grid. IMAX and JMAX are changed from the unexpanded dimensions to the expanded dimensions according to

$$\text{IMAX} = I_{nex} \times 2 \times \text{Int}\left[\left(\frac{AEXP}{\Delta s} + 0.001\right) + 1\right]$$

and

(4.52)

$$\text{JMAX} = J_{nex} \times 2 \times \text{Int}\left[\left(\frac{AEXP}{\Delta s} + 0.001\right) + 1\right],$$

where IMAX and JMAX are the expanded grid dimensions (IMX and JMX in the PARAMETER statements), I_{nex} and J_{nex} are the unexpanded grid dimensions (IMXC and JMXC in the PARAMETER statements), $AEXP$ is the approximate distance (km) by which all boundaries are expanded, and Δs is the grid spacing (km).

Subroutine SETUP defines an initial latitude-longitude search window for observations. This window extends 2.5° beyond the easternmost, northernmost, westernmost, and southernmost points of the expanded grid. SETUP returns the latitude and longitude limits of the search area in the Common/B/ variables XLATN, XLATS, XLONE, and XLONW.

A radius of influence RIN (Common/A/), to be used in the objective analyses, is defined in subroutine SETUP. RIN (in gridpoint units) is defined as

$$\mathrm{RIN} = 325 \times \frac{1.6}{\Delta s} \tag{4.53}$$

where 325 is the assumed mean station separation in km. This constant can only be changed by modification of the source code in SETUP.

Subroutine SETUPS(ID1, ID2, ID3, IFILE, IFDA)

Purpose: Reset some memory variables and arrays for data processing during each time period; calculate certain map-location parameters; read station numbers of any specific station reports to be deleted from the observational data set; check variables for consistency with other definitions.

On Entry:

ID1,ID2,ID3:	Dimensions of the DATAH and CORRH arrays.
IFILE:	Number of the current date/time being processed.
IFDA:	Number of the current FDDA date/time being processed.

On Exit:

ISELIM(40):	Station identifiers of up to 40 stations which should be eliminated from the dataset for the current time.

Calls: Subroutine SETUPS calls no other subprograms.

Called by: Main program.

Comments: Subroutine SETUPS is called once for each standard and intermediate time.

SETUPS reads from the nbogus file, unit 13, the station identifiers of any specific stations which are to be deleted from the observational data set. For details on the nbogus file format, see Section IV.B.3.a and Appendix D.

Subroutine SETUPS defines map-location parameters XCORD, YCORD, XCORC, YCORC (Common/B/).

Subroutine SFCADP(INN, SFCDD, SFCFF, IVV, IWW, IW, SLP, SFCP, SFCT, IN, IC1, IH, IC2, IC3, SFCTD, IPCHAR, ABDPDT, R6, R24, SNODEP, TSEA, Q, KLV, LLV)

Purpose: Unpack one surface-station report (land or sea) held in buffer NB.

On Entry:

- NB(820): Buffer holding packed surface station report. Common/ADP/.
- NPT: Index in array NB at which SFCADP starts reading. NPT keeps track of how much of the NB array has been unpacked. Common/ADP/.
- LTH: Length of the record read in RDADP, in 10-character units. Common/ADP/.

On Exit:

- SFCDD: Surface wind direction (degrees) from the station report.
- SFCFF: Surface wind speed (kts) from the station report.
- SLP: Sea-level pressure (mb) from the station report.
- SFCP: Surface pressure (mb) from the station report.
- SFCT: Surface temperature (°C) from the station report.
- SFCTD: Surface dewpoint depression (°C) from the station report.
- Q: Error characters from the station report. Q is otherwise unused in the program.
- KLV: Number of levels of surface data. KLV equals either 0 or 1.
- LLV: Number of levels of data from surface ship report. LLV equals either 0 or 1.

SFCADP returns many other surface data which can be printed but are otherwise not used by RAWINS. These data are returned in arguments INN, IVV, IWW, IW, IN, IC1, IH, IC2, IC3, IPCHAR, ABDPDT, R6, R24, SNODEP, and TSEA.

Calls: RAWINS Functions LOGCMP and NTENS.

NCAR Subroutines GBYTES, and SBYTES.

Called by: Subroutine PROSFC.

Subroutine SFCBLN(ID1, ID2, ID3, HORZH, IH1, IH2, IH3, SLAB1, SLAB2, SLAB3, IS1, IS2, COR, NS, SUM, SUM2, DATAS, IDS1, IDS2, IDS3, IFDA, MDATE, AUTBGW, AUTBGR, NVOL40)

Purpose: Perform the objective analyses of the surface fields of p_o, u, v, T, and RH.

On Entry:

ID1,ID2,ID3:	Dimensions of the CORRH array. Although the CORRH array is not used in SFCBLN, portions of the CORRH array are retrieved from the D1 storage array.
DATAS(IDS1,IDS2,IDS3):	Array holding surface observations.
IFDA:	Sequence number of the FDDA time.
MDATE:	Current date/time being processed.
AUTBGW:	Logical flag; AUTBGW = .TRUE. if the autobogus correction file is to be written.
AUTBGR:	Logical flag; AUTBGR = .TRUE. if the autobogus correction file is to be read.
NVOL40:	Unit number of the autobogus correction file.

Argument HORZH(IH1,IH2,IH3), and arguments SLAB1, SLAB2, SLAB3, COR, NS, SUM, and SUM2 (all dimensioned IS1,IS2) are passed to SFCBLN as working space only.

Calls: Subroutines RDISK, SEAPRS, FILSLB, BARNES, OUTPT, SETANA, SMTHER, and SLBFIL.

Called by: Main program.

Comments: This subroutine performs the same tasks for the surface fields that subroutine BLEND performs for the upper-air fields. Surface values of p_o, u, v, T, and RH are accessed from both the significant-level data stored in D1 (only surface data are retrieved from upper-air dataset) and the surface-station data stored in DATAS. The surface data are then available to produce objectively analyzed surface fields for each variable.

First, the elevations and the x and y locations of the upper-air stations are retrieved (using subroutine RDISK) from D1, and put into the BUFD(5,IRS) array. Next, the surface data from the upper-air stations are retrieved from D1 and put into the RSOND(5,IRS) array. SFCBLN then calls subroutine SEAPRS to convert the surface pressures reported in the rawinsonde reports to sea-level pressure (sea-level pressure, not surface pressure, is analyzed). Then, the data from the surface stations are added to arrays BUFD (for location and elevation information) and RSOND (for the meteorological data). Arrays BUFD and RSOND then hold all the surface data from both the surface and upper-air stations.

Once all of the surface data are available, SFCBLN begins to process each variable in succession. For each variable, the x and y locations of the stations are copied into the one-dimensional arrays XOBS(IRS) and YOBS(IRS). The meteorological data are copied to the one-dimensional array ASTA(IRS). When the wind components are processed, both components must be available; one component is copied into ASTA, and the other component is copied to a separate one-dimensional array ASTAWD(IRS).

SFCBLN then calls subroutine FILSLB to retrieve the first-guess surface field from D1 and place it in SLAB1. Subroutine OUTPT is then called to print a sample of the first-guess field. Subroutine SFCBLN then calls subroutine SETANA to perform the objective analysis.

Upon completion of the objective analysis of one variable, a sample of the final analyzed surface field is printed, and the entire field is written to D1 by subroutine SLBFIL. The processing is then repeated for the next variable until all five fields have been processed.

For the analysis of relative humidity, the humidity variable is transformed as in subroutine BLEND. After the analysis, values of relative humidity less than 10% are set to 10%.

SFCBLN includes code for creating surface first-guess fields at intermediate FDDA times by a Barnes analysis (subroutine BARNES). The provision is currently bypassed because it was found that the Barnes analysis tended to produce low-quality first-guess fields in data-sparse regions. Subroutine SFCBLN must be modified if the Barnes analysis is to be used.

Subroutine SFMERG(NREP, NM, NS, GM, GS, GMZ, GMT, GST, GMD, GSD, GMS, GMW)

Purpose: Merge separate files of significant- and mandatory-level observations (real-time observations only) into single files organized by pressure.

On Entry:

NREP:	Number of reporting stations.
NM(MAXREP):	A list of the number of mandatory levels for each report.
NS(MAXREP):	A list of the number of significant levels for each report.
GM(NUMANT,MAXREP):	The mandatory pressure levels for each report.
GS(NUMSIG,MAXREP):	The significant pressure levels for each report.
GMZ(NUMANT,MAXREP):	The geopotential heights at mandatory levels.
GMT(NUMANT,MAXREP):	The temperatures at mandatory levels.
GST(NUMSIG,MAXREP):	The temperatures at significant levels.
GMD(NUMANT,MAXREP):	The dewpoints at mandatory levels.
GSD(NUMSIG,MAXREP):	The dewpoints at significant levels.
GMS(NUMANT,MAXREP):	The wind speed at mandatory levels.
GMW(NUMANT,MAXREP):	The wind direction at mandatory levels.

On Exit:

PU(MAXLEV,MAXREP):	Pressure observations. Common /DATLIS/.
ZU(MAXLEV,MAXREP):	Height observations. Common /DATLIS/.
TU(MAXLEV,MAXREP):	Temperature observations. Common /DATLIS/.
TDU(MAXLEV,MAXREP):	Dewpoint observations. Common /DATLIS/.
SPDU(MAXLEV,MAXREP):	Wind-speed observations. Common /DATLIS/.
DIRU(MAXLEV,MAXREP):	Wind-direction observations. Common /DATLIS/.

	SRC(MAXLEV,MAXREP):	Identifies the observation as coming from a significant level or a mandatory level. Common /DATLIS/.
	PNLV(MAXREP):	Number of pressure levels for each report. Common /DATLIS/.

Calls: Subroutine SFMERG calls no other subprograms.

Called by: Subroutine SFUAOB.

Comments: Subroutine SFMERG is called for the preprocessing of real-time observations only. The observations accessed by SFMERG are eventually written to a temporary file by subroutine GETUNIOB.

Subroutine SFSCVT(NREP)

Purpose: Read all surface data reports for one time period from the real-time files.

On Exit:	NREP:	The number of reports stored in each of the arrays of common blocks SCVTCH and SCVTNM.
	STID(MAXSFC):	A list of reporting station identifiers. Common /SCVTCH/.
	TIME(MAXSFC):	The times of the reports. Common /SCVTNM/.
	LAT(MAXSFC):	The station latitudes (°N). Common /SCVTNM/.
	LON(MAXSFC):	The station longitudes (°E). Common /SCVTNM/.
	ELEV(MAXSFC):	The station elevations (m). Common /SCVTNM/.
	TEMP(MAXSFC):	The reported temperatures (°C). Common /SCVTNM/.
	TD(MAXSFC):	The reported dewpoints (°C). Common /SCVTNM/.
	SLP(MAXSFC):	The reported sea-level pressures (mb). Common /SCVTNM/.
	SPEED(MAXSFC):	The reported wind speed (m s^{-1}). Common /SCVTNM/.
	WDIR(MAXSFC):	The reported wind direction (degrees). Common /SCVTNM/.

Calls: NetCDF subroutines NCDINQ, NCVGTC, NCVGT, and NCCLOS.

NetCDF functions NCOPN, NCDID, and NCVID.

Called by: Subroutine GETUNIOB.

Comments: Subroutine SFSCVT retrieves all surface data reports from the netCDF files. Repeated calls to function NCVID (not discussed) and NCVGT (not discussed) load the observations into the arrays LAT, LON, ELEV, TEMP, TD, SLP, SPEED, and WDIR, which are returned to the calling routine (GETUNIOB) through Common /SCVTNM/.

Only one time period of real-time observations may be accessed.

Subroutine SFUAOB(NREP)

Purpose: Read all real-time upper-air observations for one time period.

On Exit:

NREP:	Number of reporting stations.
STIDU(MAXREP):	List of station identifiers. Common /STAREP/.
LATU(MAXREP):	Latitudes (°N) of the stations in STIDU. Common /STAINF/.
LONU(MAXREP):	Longitudes (°E) of the stations in STIDU. Common /STAINF/.
ELVU(MAXREP):	Elevations (m) of the stations in STIDU. Common /STAINF/.
PU(MAXLEV,MAXREP):	Pressure observations (mb). Common /DATLIS/.
ZU(MAXLEV,MAXREP):	Height observations (m). Common /DATLIS/.
TU(MAXLEV,MAXREP):	Temperature observations (°C). Common /DATLIS/.
TDU(MAXLEV,MAXREP):	Dewpoint observations (°C). Common /DATLIS/.

SPDU(MAXLEV,MAXREP):	Wind-speed observations (m s^-1). Common /DATLIS/.
DIRU(MAXLEV,MAXREP):	Wind-direction observations (degrees). Common /DATLIS/.
SRC(MAXLEV,MAXREP):	Integer identifying observations as coming from mandatory levels (SRC = 3) or significant levels (SRC = 4). Common /DATLIS/.
PNLV(MAXREP):	Number of pressure levels for each report. Common /DATLIS/.
ZW(MAXLEV,MAXREP):	Significant height observations. Common /DATSIG/.
DIRW(MAXLEV,MAXREP):	Significant-level wind-direction observations. Common /DATSIG/.
SPDW(MAXLEV,MAXREP):	Significant-level wind-speed observations. Common /DATSIG/.
SNLV(MAXREP):	Number of significant levels for each report. Common /DATSIG/.

Calls: Subroutines SFMERG and SFCLUP.

NetCDF subroutines NCDINQ, NCVGT, NCVGTC, and NCCLOS.

NetCDF functions NCOPN, NCDID, and NCVID.

Called by: Subroutine GETUNIOB.

Comments: Subroutine SFUAOB is called to retrieve all upper-air observations from the netCDF files. Repeated calls to function NCVID and subroutine NCVGT (not discussed) load the observations at mandatory and significant pressure levels into several arrays. SFUAOB then calls subroutine SFMERG, which merges the data from each report into the arrays of common /DATLIS/. Several more calls to function NCVID and subroutine NCVGT load observations reported at heights into the arrays of common /DATSIG/. SFUAOB then calls subroutine SFCLUP (not discussed), which removes levels for which geopotential height was not reported.

When the subroutine is completed, it returns the observations to the calling routine (GETUNIOB) in the arrays of common blocks DATLIS and DATSIG.

Subroutine SIGADP(SP, ST, SH, SQ, NSLV, MAXLEV)

Purpose:	Extract the significant-level values of pressure, temperature, and dewpoint depression from a packed upper-air report.	
On Entry:	NB(820):	Buffer holding a packed upper-air report. Common/ADP/.
	NPT:	Index in array NB at which SIGADP starts reading. NPT keeps track of how much of the NB array has been unpacked. Common/ADP/.
	LTH:	Length of the record read by RDADP, in 10-character units. Common/ADP/.
	MAXLEV:	The maximum number of levels which RAWINS can process.
On Exit:	SP:	One dimensional array holding pressures (mb) at significant levels.
	ST:	One dimensional array holding temperatures (°C) at significant pressure levels.
	SH:	One dimensional array holding dewpoint depression (°C) at significant pressure levels.
	SQ:	One dimensional array holding error characters of significant-level data. SQ is not otherwise used in RAWINS.
	NSLV:	Number of levels (up to MAXLEV) of significant-level temperature and humidity data read from the data file.

Calls: NCAR Subroutines GBYTES and SBYTES.

RAWINS Functions LOGCMP and NTENS.

Called by: Subroutine PROUPR.

Comments: Subroutine SIGADP extracts pressure, temperature, and dewpoint depression for each significant level. The procedure used in SIGADP is similar to that used in subroutine MANADP.

SIGADP calls NCAR subroutine GBYTES and RAWINS function NTENS to unpack the identifier for the data type into variable NCAT. If the flag identifying data type does not indicate significant-level data (*i.e.* if NCAT $\neq 2$), SIGADP repeats the calls to GBYTES and NTENS to search the data buffer for the correct identifier (NCAT $= 2$). If after searching through the buffer, SIGADP has not found the correct identifier, it will return to the calling routine (subroutine PROUPR) with no significant-level data (NSLV = 0).

If the flag identifying significant-level data (NCAT = 2) is found, SIGADP calls NTENS to extract the number of significant levels in the report (NSLV). A DO-loop then cycles through the significant levels, calling GBYTES, SBYTES, and NTENS to extract the data into the one-dimensional arrays SP, ST, SH, and SQ.

Subroutine SIGDAT(DATAH, CORRH, ID1, ID2, ID3, IPLT, LFILL, NSORT, DATAS, IDS1, IDS2, IDS3)

Purpose: Interpolate rawinsonde data to the nonmandatory pressure levels requested by the user. SIGDAT also calls subroutines SORT, which objectively checks each upper-air report for errors, and PLSOND, which generates sounding plots.

On Entry:

IPLT: Logical flag; IPLT = .TRUE. indicates that sounding plots will be generated.

LFILL: Sequential number of the date/time being processed. LFILL corresponds to variable IFILE in the main program.

DATAS(IDS1,IDS2,IDS3): Array holding surface observations.

DATAH and CORRH (both dimensioned ID1,ID2,ID3) are passed to SIGDAT as working space only.

On Exit:

NSORT(18): Integer flag, NSORT = 1 indicates that SORT had trouble correcting a report.

The corrected upper-air data, composited from significant and mandatory-level data, are put into DATAH (for mandatory levels) and CORRH (for new nonmandatory levels), which are both transferred to the storage array D1.

Calls: Subroutines RDISK, SORT, PLSOND, and WDISK.

Called by: Main program.

Comments: SIGDAT calls RDISK to retrieve mandatory- and significant-level data at all levels. The mandatory-level data are put into DATAH, and the significant-level data into CORRH. Unlike the mandatory level data, a given “level” (second dimension) in the significant data array CORRH does not correspond to a specific pressure level. Each station’s significant levels are determined independently.

Each upper-air report is reconstructed in the WKSOND array by combining the separate mandatory- and significant-level data in order by pressure. The reports are reconstructed for three reasons:

1) So that the observed data can be interpolated to new nonmandatory levels;

2) So that the report can be objectively checked in the vertical;

3) So that the sounding can be plotted.

Subroutine SORT is called to objectively check the upper-air report, and correct or remove any suspect data. The corrected data are written out to FORTRAN unit 11. SIGDAT then calls subroutine PLSOND to generate a Stüve diagram of the composited, corrected sounding. If subroutine SORT removed any suspect data, those data are replaced, if possible, with data vertically interpolated from bracketing levels. The corrected mandatory-level data are put into the DATAH array. Both mandatory- and corrected significant-level data are vertically interpolated to the new nonmandatory levels. The data at the new levels are put into the CORRH array. The procedure is then repeated for the next upper-air report, until all reports have been composited, corrected and interpolated.

Once all upper-air reports have been processed, DATAH and CORRH are written to the D1 storage array for later retrieval.

Subroutine SLBFIL(L, IVRBL, IPLACE, HORZH, IH1, IH2, IH3, SLAB, IS1, IS2)

Purpose: Transfer an entire horizontal slab of data, in SLAB, to storage array D1. This subroutine performs the opposite task of subroutine FILSLB.

On Entry:

L:	Vertical index number for the slab of data. Used for calculating the unique integer which will identify the slab.
IVRBL:	Integer specifying the variable type. Also used for calculating the unique identifying integer.
IPLACE:	Working space index for the third dimension of HORZH. The slab is first transferred to one strip (third dimension) in HORZH, and then to D1.

HORZH(IH1,IH2,IH3): Passed to SLBFIL as working space only.

SLAB(IS1,IS2): Array of data (one horizontal slab) to transfer to D1.

Calls: Subroutine HRZFIL.

Called by: Subroutine BLEND, INHRZ1, NEWPLV, OUTAP, PSFC, and SFCBLN.

Comments: Subroutine SLBFIL first transfers the entire slab of data to one strip of HORZH. SLBFIL then calls subroutine HRZFIL to actually transfer the data from HORZH to D1.

Subroutine SMTHER(SLAB, IS1, IS2)

Purpose: Spatially smooth a two-dimensional field of data to dampen short-wavelength components.

On Entry: SLAB(IS1,IS2): Data field to be smoothed.

On Exit: SLAB(IS1,IS2): Smoothed data field.

Calls: Subroutine SMTHER calls no other subprograms.

Called by: Subroutines BLEND and SFCBLN.

Comments: If the Local Master Input File option SMOOTH is set to .TRUE., subroutine SMTHER is called to smooth the final analyzed fields of T, RH, and p_o.

The smoothing algorithm consists of successive applications of (4.54) and (4.55) to a variable, α, and results in the smoothed variable α^*. For the J direction,

$$\hat{\alpha}_{i,j} = \alpha_{i,j} + 0.50\left[\frac{1}{2}\left(\alpha_{i,j+1} + \alpha_{i,j-1}\right) - \alpha_{i,j}\right], \tag{4.54}$$

and

$$\alpha^*_{i,j} = \hat{\alpha}_{i,j} - 0.52\left[\frac{1}{2}\left(\hat{\alpha}_{i,j+1} + \hat{\alpha}_{i,j-1}\right) - \hat{\alpha}_{i,j}\right]. \tag{4.55}$$

The algorithm is then applied in a similar manner in the I direction. The algorithm may be applied repeatedly to a particular field to obtain the desired degree of smoothing by embedding the call to SMTHER in a DO-loop.

Subroutine SORT(PP, TT, TD, PH, NVV, PWH, WH, WD, WS, NQQ, ELEV, NVHD, NQHD, IBUG)

Purpose: Objectively check one upper-air report for errors. Correct or remove suspect data.

On Entry:

PP(100):	Pressure levels (mb) for temperature and humidity data.
TT(100):	Temperatures (°C).
TD(100):	Dewpoints (°C).
PH(100):	Geopotential heights (m) for temperature and humidity data.
NVV:	Number of levels of temperature and humidity data.
PWH(100):	Pressure levels (mb) for wind data.
WH(100):	Geopotential heights (m) for wind data.
WD(100):	Wind directions (degrees).
WS(100):	Wind speeds (kts).
NQQ:	Number of levels of wind data.
ELEV:	Elevation of the station (m).
IBUG:	Integer flag; set it to 1 to print out errors found by the subroutine.

On Exit:

NVHD:	Number of temperature and humidity levels after deletions.
NQHD:	Number of wind levels after deletions.

Also, SORT returns the corrected arrays of upper-air data.

Calls: Subroutines ELIM, M12N12, and VAPRES (not discussed).

Called by: Subroutine SIGDAT.

Comments: Subroutine SORT is called for each upper-air report processed. SORT detects and attempts to correct several possible errors in the upper-air report.

Duplicate pressure-level reports are removed. If levels are in the wrong order, the data are reordered. The temperature profile is checked for spikes, and the suspect data are removed. The geopotential heights are then recalculated based on the corrected temperature profile. The wind profile is checked for spikes. If suspect data are found, an attempt is made to correct the errors by altering the most significant digit in the wind direction or speed. If they cannot be corrected, the data are deleted. An attempt to replace the deleted value for wind and temperature profiles with a vertically interpolated value is made in the calling routine SIGDAT after the return from SORT.

Subroutine SORT uses subroutine ELIM (not discussed) to eliminate a suspect data point, and subroutine M12N12 (not discussed) to find the observations immediately above and below a given point.

Subroutine UNISF(INN, SFCDD, SFCFF, IVV, IWW, IW, SLP, SFCP, SFCT, IN, IC1, IH1, IC2, IC3, SFCTD, IPCHAR, ABDPDT, R6, R24, SNODEP, TSEAS, IH, KLV, LLV, NVOLS6)

Purpose: Retrieve one station report of real-time surface observations from FORTRAN unit NVOLS6, to which they were stored by subroutine GETUNIOB.

On Entry: NVOLS6: FORTRAN unit number of the file which holds the real-time surface data.

On Exit: SLP: Sea-level pressure (mb) from the station report.

SFCT: Surface temperature (°C) from the station report.

SFCTD: Surface dewpoint depression (°C) from the station report.

SFCDD: Surface wind direction (degrees) from the station report.

SFCFF: Surface wind speed (kts) from the station report.

STIME: Time of the surface station report.

KLV and LLV are set to 1. All other arguments (except NVOLS6) are set to either 99999 or 9999999., as they are not needed by the program.

Calls: Subroutine UNISF calls no other subprograms.

Called by: Subroutine PROSFC.

Comments: Subroutine UNISF reads a record containing surface data (preprocessed by subroutine GETUNIOB) from NVOLS6.

Subroutine UNIUP(NVOLUP, P, Z, T, H, D, F, PP, DP, FP, LTHERM, LWND, LJTYP, LLL)

Purpose: Read one category of real-time upper-air data from unit NVOLUP, where the data have been stored by subroutine GETUNIOB.

On Entry:

LJTYP: Integer flag indicating the category of data to be accessed:
1: Mandatory-level data.
2: Significant-level T and RH .
3: Significant-level wind data.
4: Wind data reported at heights.

NVOLUP: FORTRAN unit number of the file which holds the real-time observations.

LTHERM: Number of significant levels of temperature and humidity data.

LWIND: Number of levels of wind data.

On Exit:

P(75): Array holding pressures (mb) at mandatory and significant levels.

Z(75): Array holding heights (m) for mandatory-level data and wind data reported at heights.

T(75): Array holding temperatures (°C) at mandatory and significant levels.

H(75): Array holding dewpoint depression (°C) at mandatory and significant levels.

D(75): Array holding wind direction (degrees) for winds at mandatory levels and at heights.

F(75): Array holding wind speed (kts) for winds at mandatory levels and at heights.

PP(60): Array holding pressures (mb) for significant-level wind data (at pressure levels).

DP(60): Array holding wind direction (degrees) for significant-level wind data (at pressure levels).

FP(60):	Array holding wind speed (kts) for significant-level wind data (at pressure levels).

Calls: Subroutine UNIUP calls no other subprograms.

Called by: Subroutine PROUPR.

Comments: UNIUP reads the data from FORTRAN unit NVOLUP, where it was written by subroutine GETUNIOB. UNIUP accesses real-time upper-air data; it performs the task that subroutines MANADP, SIGADP, WZZADP, and WPPADP, perform for the historical data.

UNIUP executes a READ statement for each of the categories of upper-air data indicated by the integer flag LJTYPE.

Subroutine WDISK(NAME, HOLD, LEN, IFWA, NWDS, CHECK)

Purpose: Transfer a block of data from the HOLD array to the storage array D1. HOLD holds one two-dimensional slab of data in a one-dimensional array. This subroutine performs the opposite of subroutine RDISK. WDISK is the only subroutine which transfers data directly to D1.

On Entry:

NAME:	Unique integer which identifies the data to be transferred to D1. NAME is calculated in the calling routine. In subroutine WDISK, NAME is stored in NBFW1 (common/MEMCOR), for later reference.
HOLD(LEN):	One-dimensional array holding the data. LEN, the dimension of HOLD, is calculated in the calling subroutine.
IFWA:	Address in HOLD of the first word of data to be written to D1, minus 1.
NWDS:	The number of words to be written from HOLD to D1.
CHECK:	Logical flag; if CHECK = .TRUE., WDISK prints out the value of the variable NAME.

Calls: Subroutine WDISK calls no other subprograms.

Called by: Subroutines BOGPTS, HRZFIL, SAVFIL, SEAPRS, and SIGDAT.

Comments: Subroutine WDISK is called to write data from a single vertical level to the storage array D1. The data may be either an analysis on the mesoscale grid or one level of observations at the station locations.

The location of the data within D1 is retained by storing the first-word address of the sub-region of D1 in the nth element of array IBFW1 (common/MEMCOR/), where n is the nth block of data stored in D1. WDISK also stores the unique integer identification number, NAME, in the nth element of a parallel array NBFW1 (common/MEMCOR/). When the data from a particular variable and level are to be retrieved, subroutine RDISK scans the array NBFW1 for the unique integer NAME. The first-word address of the data block is then retrieved from the parallel location in array IBFW1.

WDISK first scans the array NBFW1 for the value NAME. If NAME has been used previously, WDISK will find NAME in NBFW1, and the first-word address of the data in IBFW1. WDISK then overwrites the old data. Since the length for each data block will not change during program execution, there is no danger of overwriting data in another portion of D1.

If NAME has not been used previously, WDISK will not find NAME in NBFW1, and will calculate a new first-word address in D1 for the new block of data. WDISK sets the first-word address to one location beyond the last word previously written to D1. The data are then transferred directly from HOLD to D1. If the storage capacity of D1 has been exceeded, WDISK prints out a warning message to that effect and stops the execution of the program.

Subroutine WPPADP(P, D, F, Q, NLV, MAXLEV)

Purpose: Extract wind direction and wind speed from a packed upper-air report. WPPADP is called twice for each report: once for winds at pressure levels, and once for winds at heights.

On Entry:

NB(820):	Buffer holding packed upper-air report. Common/ADP/.
NPT:	Index in array NB at which WPPADP starts reading. NPT keeps track of how much of the NB array has been unpacked. Common/ADP/.
LTH:	Length of the record read by RDADP, in 10-character units. Common/ADP/.
MAXLEV:	The maximum number of levels which RAWINS can process.

	IEFLAG:	Integer flag indicating whether wind data at heights (IEFLAG = 1) or pressure levels (IEFLAG = 0) are to be retrieved. Common/MULENT/.
On Exit:	P:	One-dimensional array holding pressures (mb) for pressure-level wind data, and heights (m) for wind data at heights.
	D:	One-dimensional array holding wind direction (degrees) at pressure levels or heights.
	F:	One-dimensional array holding wind speed (kts) at pressure levels or heights.
	Q:	One-dimensional array holding error characters for wind data. Q is otherwise not used in the program.
	NLV:	Number of levels of data, up to MAXLEV, read from the upper-air report.

Calls: Functions LOGCMP and NTENS.

NCAR Subroutines GBYTES, and SBYTES.

Called by: Subroutines PROUPR (for pressure levels) and WZZADP (for heights).

Comments: Subroutine WPPADP extracts wind data, which exist in the station reports associated with either a pressure level or a height level. Each call to WPPADP will acquire winds at only one of the two types of levels. The type of level is specified by the integer flag IEFLAG, stored in Common/MULENT/. The procedure used in WPPADP for extracting the data is similar to that used in MANADP and SIGADP.

WPPADP calls NCAR subroutine GBYTES and RAWINS function NTENS to unpack the identifier for the data type into variable NCAT. If the flag NCAT, identifying data type, does not indicate wind data at either pressures (NCAT = 3) or heights (NCAT = 4), then WPPADP repeats the calls to GBYTES and NTENS to search for the correct identifier. If after searching through the buffer, WPPADP has not found the correct identifier, it will return to the calling routine with no wind data (NLV = 0).

If the flag identifying the correct type of data is found, WPPADP calls NTENS to extract the number of wind levels in the report (NLV). A DO-loop then cycles through the levels, calling GBYTES, SBYTES, and NTENS to extract the data into the one-dimensional arrays P, D, F, and Q.

Subroutine WTAPE(L, IVRBL, SLAB, IS1, IS2, SLAB4, MAXIC, MAXJC, NVOL, LP, KP)

Purpose: Write a horizontal slab of data to unit NVOL.

On Entry:

L:	Level index of the slab of data. L is set in the calling routine; it is used to label samples of the data printed out.
IVRBL:	Integer identifying the variable type.
SLAB(IS1,IS2):	Horizontal slab of data to be written.
SLAB4(MAXIC,MAXJC):	Working space, used to hold data on the unexpanded grid.
NVOL:	Unit number to which the slab is written.
LP:	Index number of the pressure level in array IPRES. LP is used to label printed samples of the data.
KP:	Used to label printed samples of the data.

Calls: Subroutine OUTPT.

Called by: Subroutines OUTAP and OUTFDA.

Comments: WTAPE is called in the process of creating final output files, and writes out horizontal slabs of data which have been objectively analyzed on the mesoscale grid. If the data are on the expanded grid (*i.e.*, SLAB1(IS1,IS2)), the array is reduced to the unexpanded grid (*i.e.*, SLAB4(MAXIC,MAXJC)), before being written.

If the print option DRAWO is activated (in subroutine SETUP), a sample of the field is printed out (by subroutine OUTPT). If DRAWO is not activated, only the first element of the array SLAB will be printed, along with a brief message identifying the data.

Subroutine WZZADP(P, D, F, Q, NLV, MAXLEV)

Purpose: Extract the values of wind direction and wind speed (reported at heights) from a packed upper-air report.

On Entry:	MAXLEV:	The maximum number of levels which RAWINS can process.
	IEFLAG:	Integer flag, indicating that the call to WPPADP is meant to extract wind data at height levels (IEFLAG = 1). Common/MULENT/.
On Exit:	P:	One-dimensional array holding heights (m) for wind data.
	D:	One-dimensional array holding wind direction (degrees) at heights.
	F:	One-dimensional array holding wind speed (knots) at heights.
	Q:	One-dimensional array holding error characters for wind data. Q is otherwise unused in RAWINS.
	NLV:	Number of levels of data, up to MAXLEV, read from the upper-air report.

Calls: Subroutine WPPADP.

Called by: Subroutines PROUPR.

Comments: WZZADP simply sets the flag IEFLAG to 1 (indicating that WPPADP is to extract wind data at heights) and calls WPPADP. WPPADP returns arrays P, D, F, and Q. After the call to WPPADP, WZZADP resets IEFLAG back to 0.