CONTENTS

LAMP No.	NAME	FUNCTION
1.0	U150	Runs LAMP Analyses and Models
1.1	U400A	Analyzes Quasi-Continuous Variables
1.2	U400B	Analyzes U and V Wind Components and Wind Speed
1.3	U400D	Analyzes Discontinuous Variables
1.4	U450	Forecasts Sea Level Pressure
1.5	U454	Forecasts Discontinuous Variables
1.6	U452	Forecasts Saturation Deficit
1.7	U453	Forecasts Saturation Deficit
1.8	U120	Converts .HDF Radar Data to TDLPACK
1.9	U455	Forecasts Discontinuous Variables
2.0	ugem.ksh	Processes Data Requests for GEMPAK Display
4.1	CUT	Cuts a Grid
4.2	DENSER	Makes a Grid Denser by Linear Interpolation
4.3	DENSRQ	Makes a Grid Denser by Quadratic Interpolation
4.4	CHKGRD	Checks Grid Characteristics
4.5	CLOS	Finds Two Closest Neighbors to a Station Location
4.6	FLTAG	Sets LTAG to Indicate Use of Data
4.7	NEWXY	Calculates Grid Positions
4.8	SIZEGR	Thins or Thickens a Grid Over the Same Area
4.9	THINER	Thins a Grid by Eliminating Alternate Gridpoints
4.10	SMOTH	Smooths a Grid
4.11	TRNSFR	Transfers Data from One Array to Another
4.12	XMSMSH	Finds Nominal Mesh Length from Actual Mesh Length
4.13	NEWSIZ	Calculates a New Grid Size and Pole Position
4.14	POSGRD	Positions a Grid Within a Larger Grid
4.15	XYCOMP	Computes X and Y positions of Stations
4.16	ITRP	Interpolates into Grid
4.17	MSHXMS	Finds Actual Mesh Length from Nominal Mesh Length
4.18	PAWOTG	TDLPACKs and Writes Grid to Sequential File
4.19	PAWING	TDLPACKs and Writes Grid to MOS-2000 Internal File
4.20	CLDCAT	Specifies Cloud Amount and Height in Height Categories
8.0	U860	Verifies Gridded Forecasts of Continuous Variables
8.1	U870	Verifies Gridded Forecasts of Discontinuous Variables

CONTENTS (ALPHABETIZED)

LAMP No.	NAME	FUNCTION
1.8	U120	Converts .HDF Radar Data to TDLPACK
1.0	U150	Runs LAMP Analyses and Models
1.1	U400A	Analyzes Quasi-Continuous Variables
1.2	U400B	Analyzes U and V Wind Components and Wind Speed
1.3	U400D	Analyzes Discontinuous Variables
1.4	U450	Forecasts Sea Level Pressure
1.6	U452	Forecasts Saturation Deficit
1.7	U453	Forecasts Saturation Deficit
1.5	U454	Forecasts Discontinuous Variables
1.9	U455	Forecasts Discontinuous Variables
2.0	ugem.ksh	Processes Data Requests for GEMPAK Display
4.4	CHKGRD	Checks Grid Characteristics
4.20	CLDCAT	Specifies Cloud Amount and Height in Height Categories
4.5	CLOS	Finds Two Closest Neighbors to a Station Location
4.1	CUT	Cuts a Grid
4.2	DENSER	Makes a Grid Denser by Linear Interpolation
4.3	DENSRQ	Makes a Grid Denser by Quadratic Interpolation
4.6	FLTAG	Sets LTAG to Indicate Use of Data
4.16	ITRP	Interpolates into Grid
4.17	MSHXMS	Finds Actual Mesh Length from Nominal Mesh Length
4.13	NEWSIZ	Calculates a New Grid Size and Pole Position
4.7	NEWXY	Calculates Grid Positions
4.19	PAWING	TDLPACKs and Writes Grid to MOS-2000 Internal File
4.18	PAWOTG	TDLPACKs and Writes Grid to Sequential File
4.14	POSGRD	Positions a Grid Within a Larger Grid
4.8	SIZEGR	Thins or Thickens a Grid Over the Same Area
4.10	SMOTH	Smooths a Grid
4.9	THINER	Thins a Grid by Eliminating Alternate Gridpoints
4.11	TRNSFR	Transfers Data from One Array to Another
4.12	XMSMSH	Finds Nominal Mesh Length from Actual Mesh Length
4.15	XYCOMP	Computes X and Y positions of Stations
8.0	U860	Verifies Gridded Forecasts of Continuous Variables
8.1	U870	Verifies Gridded Forecasts of Discontinuous Variables

U150

RUNS LAMP ANALYSES AND MODELS

Harry R. Glahn
December 1, 2000

PURPOSE: U150 is the driver program for the analyses and models for LAMP. It is designed to run each of the analyses and models in sequence for a desired number of hours/days/months. A slightly different version of U150 will be required for operations, but the two will be extremely similar. U150 is an amalgamation of several stand-alone programs; each of these "programs" is highly self-determining, but depend on the care and feeding of U150. The primary modules, besides the housekeeping ones, are:

U400A - Analyzes continuous variables (e.g., sea level pressure)
U400B - Analyzes wind, u and v components and speed together
U400D - Analyzes discontinuous variables (e.g., visibility)
U450 - The sea level pressure forecast model
U453 - The moisture forecast model (saturation deficit)
U454 - The purely advective forecast model (e.g., visibility)

These are explained in separate writeups. U453 is the preferred version of the saturation deficit model. U452 did not allow the setting of saturation deficit to zero for individual time steps. U452 can also be run, but only U452 or U453 in a single U150 run; however, U452 will not be maintained.

U150 is controlled by a control file 'U150.CN'. Also, each major module (e.g., U400A) has one or more control files (e.g., 'U400ASLP.CN' to analyze sea level pressure). The variables analyzed in a particular run are determined by 'U150.CN', and the specifics of the analyses, for instance, are controlled by control files specific to the variable.

U150 uses a driver DRU150 so that dimensions of variables can be tailored by PARAMETER statements to user need without requiring a separate copy of the main program U150 (actually, subroutine) for every application. U150 is written to run on a 32-bit or a 64-bit word-length machine. This is accomplished by PARAMETER statements in the driver. Input to U150 is both gridpoint and vector data in TDLPACK format. TDLPACK is documented in TDL Office Note TDL ON 00-1, and many of the same features are in U150 and other LAMP programs as exist MOS-2000 programs. In addition, subroutines in MOS-2000 are heavily used by U150. The primary output of U150 is a TDLPACK gridpoint dataset containing grids as specified in the control files. In addition, for checkout and quality control, another gridpoint dataset is written containing subsets of the full grid; these grids can be contoured and contain input to U203, a front end to a display program ugem.ksh. Also, vector records can be written that contain (a) only the data that are accepted by the analysis, and (b) the data that are accepted and not questionable. Some familiarity with other MOS-2000 documents will be necessary for full understanding of this writeup.

CONTROL FILE INPUT: 'U150.CN' (Unit = KFILDI)

Record Type 1 - Format (A4,25I3) Output Control

This record contains unit numbers for the run, and can even control the "default" output file number. KFILDI is the input unit number as specified in DRU150.

IPINIT - 4 characters, usually a user's initials plus a run number, to append to "U150" to identify a particular segment of output indicated by a suffix IP(J) (see below). The run number allows multiple runs of U150 and writing of uniquely named files, provided the user uses a different run number for each run. For example, with IPINIT = 'HRG2' and IP(2) = 40, the file name for Unit No. 40 = 'U201HRG240'. DO NOT USE A BLANK FOR ONE OF THE CHARACTERS. (CHARACTER*4)

IP(J) - Each value (J=1,25) indicates whether (>0) or not (=0) certain information will be written. When IP( ) > 0, the value indicates the unit number for output. These values should not be the same as any other unit numbers used in U150 except possibly KFILDO (the default output file), although a value of one IP( ) can be the same as the value of another IP( ). If another file unit number is designated the same as an IP( ) value, that IP( ) is set to KFILDO. This is ASCII output, generally for diagnostic purposes. Values have been defined as indicated below for values of J:

(1) = All error diagnostics plus other information not specifically identified with other IP( ) numbers. When IP(1) is read as nonzero, KFILDO, the default output file unit number, will be set to IP(1). When IP(1) is read as zero, KFILDO will be used unchanged, as specified in DRU150 DATA statement = 12. Changing the default unit number allows multiple runs of U150 or other programs within the same directory without overwriting.

(2) = The input dates in IDATE( ). These are the dates as actually read in. When there are errors, print will be to the default output file unit KFILDO as well as to unit IP(2).

(3) = The output dates in IDATE( ). These are the dates extended by date spanning. When there are errors, output will be to the default output file unit KFILDO as well as to unit IP(3).

(4) = The input station list (call letters only) when the station list is not from the directory [KFILD(1) ≠ KFILD(2)]. If there are input errors, the station list will be written to the default output file unit KFILDO as well as to unit IP(4) even when KFILD(1) = KFILD(2).

(5) = The station and station directory information in the order to be dealt with in U150. This can be in the order read or alphabetical. If there are input errors in this list, the station list will be written to the default output file unit KFILDO as well as to unit IP(5).

(6) = The variable IDs as they are being read in. This is good for checkout; for routine operation, IP(7) and/or IP(9), may be better.

(7) = The variable ID list in summary form. If there are errors, the predictor list will be written to the default output file unit KFILDO as well as to unit IP(7).

(8) = The variable ID list of all variables that may be needed by this run of U150.

(9) = The variable ID list in summary form after reordering on the ID, low to high. This differs from the print in IP(7) in that IP(9) does not include the parsed ID's in IDPARS(,), but rather includes the information taken from the variable constant file on unit KFILCP (see below).

(10) = The variable ID's for the first day (Day 1) as read from the archive files. This is just a list of all the packed gridpoint and vector fields on the input files.

(11) = The variable ID's of the archived fields actually needed. This is MSTORE(,), but it is not printed if only one date is processed (see Record Type 5).

(12) = The list(s) of stations comprising each directory record read from a vector input will be output on unit IP(12). For input of hourly data, this creates voluminous output. The print occurs in subroutine FINDST, which also prints a list of stations not found on the input file(s) <u>unless</u> compiled with the /D option.

(13) = The contents of LSTORE(,) after compression after each day (actually cycle) LE LSTPRT. LSTPRT is set in U150 DATA statement to 3.

(14) = Diagnostics when there are no data for a desired date/time on a particular input file will be written to unit IP(14). This might happen for each date/time and a lot of output be created.

(15) = X and Y positions of the stations for the LAMP grid used on this run are written to unit IP(15).

(16) = Statements indicating the writing of data to a sequential file.

(17) = A list of stations, their X and Y positions, data values, and LTAGS will be written to unit IP(17). Stations with missing data are not written. For several variables and hours, this will be voluminous output.

(18) = For the analysis of <u>continuous</u> variables, a listing of stations, their X and Y positions, data values, LTAGS, interpolated values for the <u>unsmoothed</u> analyses, and differences between the data and the analysis values <u>for the whole LAMP area</u> will be written to unit IP(18). Stations with missing data are not listed.

(19) = The same as IP(18) except it applies to <u>smoothed</u> analyses. If analyses are not smoothed, IP(19) is not written.

(20) = For <u>continuous</u> variables, a listing of stations, their X and Y positions, data values, LTAGS, interpolated values for the unsmoothed analysis, and differences between the data and the analysis values will be written to unit IP(20) <u>for only the subsetted area</u>. In addition, the degree of fit (average absolute error) between the

unsmoothed analysis and the observations (provided IP(18) NE 0) and between the smoothed analysis and the observations (provided IP(19) NE 0) will be written to units KFILDO and IP(20) (see Record Type 3 below for description of subsetted area).

(21) = The average degree of fit between the data and the interpolated values from the analysis for the unsmoothed and, if smoothed, the smoothed analysis for each pass. This produces one (or two if smoothed) line(s) per pass for each analysis being done (e.g., temperature from U400A).

(22) = The gridprints for the subsetted area.

(23) = Information concerning opening and closing of sequential files.

Record Type 2 - Format (A72) <u>Run Identification</u>

This record is used to identify the run.

<u>RUNID</u> - 72 characters of information to identify the run. (CHARACTER*72)

Record Type 3 - Format (6(I10/),F10.0/18(I10/),I10) <u>Control Parameters</u>

This record contains a variety of control values for the run. Note that each value is on a separate line. A brief explanation can be put on the same line with the variable to assist the user in knowing which value is for which variable.

<u>KSKIP</u> - When nonzero, KSKIP indicates that the output files KFILIO and KFILQC (see Record Types 8 and 11) are to be moved forward until all data for date KSKIP have been skipped. KSKIP is input as either YYMMDDHH or YYYYMMDDHH and then used as YYYYMMDDHH.

<u>NSKIP</u> - The number of errors that will be tolerated on Day 1 before halting. Day 3 is usually completed before the stop actually occurs so that the user can see more results.

<u>JSTOP</u> - The total number of errors that will be tolerated before the program halts (see Comments section).

<u>INCCYL</u> - The increment in hours between date/times that are put into IDATE( ) (see Record Type 5) as a result of date spanning in subroutine DATPRO. This variable is also used in determining what data are to be saved in the Internal MOS-2000 Storage System. The date/times put into Record Type 5 must be such that all could be arrived at by successively adding INCCYL to the first date/time in IDATE( ).

<u>NEW</u> - Indicates whether (=1) the new ICAO call letters are to be used or whether (=0) the old 3-letter call letters are to be used. The directory used in MOS-2000 contains both. In either case, one is the substitute for the other, and there are up to 4 other substitute stations in the directory (see Comments section).

NALPH - Indicates whether (=1) or not (=0) the call letters will be alphabetized according to the station directory. Since the MOS-2000 directory is alphabetized by the new ICAO call letters, using NEW = 0 and NALPH = 1 doesn't make much sense.

PXMISS - The value to substitute in the output for a secondary missing value of 9997. This allows the 9997 to be maintained if desired, set to zero in the case of the implied value of near zero for forecasts, or even set to some other value. Note that this does not apply to missing values other than 9997.

NXL - The size (in number of gridpoints) of the primary LAMP grid for this run in the NX direction in one-quarter Bedient Units. A one-quarter Bedient grid of 95.25 km (at 60 degrees North) has a nominal mesh length of exactly 80 km, about the resolution at 30 degrees North Latitude.

NYL - The size (in number of gridpoints) of the primary LAMP grid for this run in the NY direction in one-quarter Bedient Units.

NXPL - The north pole position (in number of gridpoints) of the primary LAMP grid for this run in the NX direction in one-quarter Bedient Units, considering the lower left corner to be at (1,1).

NYPL - The north pole position (in number of gridpoints) of the primary LAMP grid for this run in the NY direction in one-quarter Bedient Units.

MESHL - The "nominal" mesh (grid) length in km for the quality control (disposable) output grids for continuous variables.

MESHD - The same as MESHL except for discontinuous variables, That is, sea level pressure may be output at MESHL = 80 km, while visibility might be output at MESHD = 10 km. If these are to be overlain for viewing, then MESHL and MESHD must be the same.

NXGMIN, NXGMAX, NYGMIN, NYGMAX - Define a subsetted area in terms of the NXL by NYL LAMP grid to be gridprinted and written for viewing to Unit No. IP(22) and to be written to a TDLPACK record on Unit No. KFILOG (see the Data Output Section, Subsection B). The LAMP grid may be quite large and it may be desired for checkout or quality control to output only a portion of the grid. The same area will be output at the MESHL resolution no matter what resolution is actually used (as specified in specific .CN control files). This means the output for different variables can be overlain for viewing. This facility does not impact the actual output to Unit No. KFILIO, but may require interpolation or thinning to get the areas to match. This same area can also be overlain with discontinuous fields, but only when MESHL = MESHD. NXGMIN = 0 implies no subsetting desired. If these values are internally inconsistent or inconsistent with NXL or NYL, KFILOG is set = 0, and there will be no subsetted output.

See Section "Data Output, Section B" for the identifiers.

IU400A - 1 when U400A is to be accessed; zero otherwise.

IU400B - 1 when U400B is to be accessed; zero otherwise.

IU400D - 1 when U400D is to be accessed; zero otherwise.

IU450 - 1 when U450 is to be accessed; zero otherwise.

IU451 - Not used.

IU452 - 1 when U452 is to be accessed; zero otherwise. U452 is normally not used.

IU453 - 1 when U453 is to be accessed; zero otherwise.

IU454 - 1 when U454 is to be accessed; zero otherwise.

INCDD - Increment to add to LAMPNO (= 5) when writing forecasts from U450, U452, U453, and U454 to both archive and disposable files (see Record Types 8 and 9).

Record Type 4 - Format (I3,4XA60) Date List File

This record (plus the terminator record) identifies the data set from which the date list is read. Records are read until the terminator KFILDT( ) = 99 is reached. Maximum number of records, sans the terminator record, = 1. This Record Type 4 is read by subroutine RDSNAM.

KFILDT - Unit number for the file containing the input date list.

NAMDT - Name of file where this date list resides. When KFILDT = KFILDI, NAMDT is not used and can be read as "DEFAULT". (CHARACTER*60)

Record Type 5 - Format (7I10) Date List

This group of records determines the date/times for which output is wanted. If KFILDT (read in Record Type 4) ≠ KFILDI, this Record Type 5 is omitted.

IDATE(J) - Initial date list, which may contain negative values indicating date spans. When a negative occurs, all dates between this value and the previous date are filled in at the increment of hours specified in INCCYL. Note that any negative date must be immediately preceded by a positive date. This input date list is modified in subroutine DATPRO to contain the complete date list with the dates in the spans filled in (J=1,NDATES). Dates are input as YYMMDDHH and modified to YYYYMMDDHH. This list is read by subroutine RDI which eliminates any zeros found in the input. Terminator is 99999999. Maximum number of dates, sans terminator, is ND8. The date/times put into Record Type 5 must be such that all could be arrived at by successively adding INCCYL (read in Record Type 1) to the first date/time in IDATE( ).

Record Type 6 - Format (2I3,1XA60) Gridpoint and Vector Input Data Files

This group of records identifies the data sets from which the gridpoint and/or vector data are read. Records are read until the terminator KFILIN( ) = 99 is reached. Maximum number of records, sans the terminator record, = ND6. This Record Type 6 is read by subroutine RDSNAM. Upon completion of reading this record type, J=1,NUMIN. If all input data are from a constant file, only the terminator is necessary.

KFILIN(J) - Unit number for the input gridpoint or vector file. KFILIN( ) must be < 80 for gridpoint data and ≥ 80 for vector data. Also note that values 97, 99, and 44 through 49 are reserved for other uses.

MODNUM(J) - Source of data that are on this file (e.g., 5 = LAMP, 8 = aviation model). For gridpoint data, this number is used to match with the "DD" in the variable ID. Normally, this value is zero for vector data (see Sequential Vector Data Section for more details).

NAMIN(J) - Name of file where this input resides. (CHARACTER*60)

Note that when data sets are to be used in sequence, they should be read in the proper order, be in sequence, and have the same unit number. That is, if 2 years of data are to be used and one year is on one data set and the other year on another, then the first should immediately precede the second in the list and both should have the same unit number. Vector data can either precede, follow, or be mixed with gridpoint data sets in the list read; the only restriction is that those with the same unit number are be proper sequence.

Record Type 7 - Format (I3,4XA60) Random Access Files

This group of records identifies the random access data sets from which constant data are read. Records are read until the terminator KFILRA( ) = 99 is reached. Maximum number of records, sans the terminator record, = 6. This Record Type 7 is read by subroutine RDSNAM. Upon completion of reading this record type, J=1,NUMRA. If no data are needed from a constant file, only the terminator is necessary.

KFILRA(J) - Unit number for the random access constant file (J=1,NUMRA). Unit numbers must be in the range 44 to 49; see "Restrictions" for more information.

RACESS(J) - Name of file corresponding to KFILRA(J) (J=1,NUMRA). (CHARACTER*60)

Record Type 8 - Format (I3,4XA60) Gridpoint Output File

This record (plus the terminator record) identifies the archive output file. Records are read until the terminator KFILIO( ) = 99 is reached. Maximum number of records, sans the terminator record, = 1. This Record Type 8 is read by subroutine RDSNAM. This file will be opened as 'OLD'. If it does not exist, it is opened as 'NEW'. If packed data are not to be

saved, only the terminator is necessary here; in that case, a file is not opened and data are not written.

KFILIO - Unit number for the output file.

OUTNAM - Name of file where this output is to reside. (CHARACTER*60)

Record Type 9 - Format (I3,4XA60) **Disposable Gridpoint Output File**

This record (plus the terminator record) identifies the disposable grid output file. Records are read until the terminator KFILOG( ) = 99 is reached. Maximum number of records, sans the terminator record, = 1. This Record Type 9 is read by subroutine RDSNAM. This file will be opened as 'NEW'. If packed data are not to be saved, only the terminator is necessary here; in that case, a file is not opened and data are not written.

KFILOG - Unit number for the output file.

OUTDIS - Name of file where this output is to reside. (CHARACTER*60)

Record Type 10 - Format (I3,4XA60) **Disposable Vector Output File**

This record (plus the terminator record) identifies the disposable vector output file. Records are read until the terminator KFILOV( ) = 99 is reached. Maximum number of records, sans the terminator record, = 1. This Record Type 10 is read by subroutine RDSNAM. This file will be opened as 'NEW'. If packed data are not to be saved, only the terminator is necessary here; in that case, a file is not opened and data are not written.

KFILOV - Unit number for the output file.

OUTVEC - Name of file where this output is to reside. (CHARACTER*60)

Record Type 11 - Format (I3,4XA60) **Quality Controlled Observation Output File**

This record (plus the terminator record) identifies the quality controlled observation output file. Records are read until the terminator KFILQC( ) = 99 is reached. Maximum number of records, sans the terminator record, = 1. This Record Type 11 is read by subroutine RDSNAM. This file will be opened as 'NEW'. If quality controlled data are not to be saved, only the terminator is necessary here; in that case, a file is not opened and data are not written.

KFILQC - Unit number for the output file.

OUTQCV - Name of file where this output is to reside. (CHARACTER*60)

Record Type 12 - Format (I3,4XA60) **Station and Location Files**

This pair of records (plus the terminator record) identifies the file(s) from which the station (or location) information is obtained. Records are read until the terminator KFILD( ) = 99 is reached. Maximum number of records, sans the terminator record, = 2. This Record Type 12 is read by subroutine RDSNAM. Upon completion of reading this record type, J=1,2.

KFILD(J) - Unit number for the file containing the station call letters which are to be used in the analyses (J=1) and the station directory which holds the latitudes, longitudes, WBAN numbers, elevations, and names for each possible station (J=2). KFILD(1) can be the default input file number, KFILDI, in which case DIRNAM(1) is not used. Also, KFILD(1) can equal KFILD(2), in which case the call letters list will consist of all stations on the directory file; note that both a list of stations and a directory cannot exist on unit KFILD(1) = KFILD(2).

DIRNAM(J) - Name of file matching KFILD(J). When KFILD( ) = KFILDI, DIRNAM( ) is not used and can be read as "DEFAULT". (CHARACTER*60)

Record Type 13 - Format (7(A8,1X)) Station List

This group of records identifies the stations for which data are to be used. If KFILD(1) ≠ KFILDI (the default input file) or if KFILD(1) = KFILD(2), this group is omitted, and the information is taken from another source.

CCALL(K) - Call letters (or other 8-character location designator) of stations (or locations) for which data are to be used in the analyses (K=1,NSTA). This list is read within subroutine RDSTAD or RDSTAL by subroutine RDC, which eliminates any blanks found in the input. Duplicate stations in the list are kept, but a diagnostic is furnished on unit IP(5). For NALPH = 1, the stations are placed in alphabetical order providing the directory is in alphabetical order; stations not in the directory are put at the end of the list. The call letters should (normally) be left justified, and if a full 8 characters are not present, CCALL( ) will be blank (b) filled on the right, e.g., 'OKCbbbbb' could be 'OKC' or 'OKCb'. Terminator is '99999999'. Maximum number of stations, sans terminator, is ND1. (CHARACTER*8)

Record Type 14 - Format (I3,4XA60) Variable File

This record (plus the terminator record) identifies the file from which the variable ID's are to be taken. Records are read until the terminator KFILP = 99 is reached. Maximum number of records, sans the terminator record, = 1. This Record Type 14 is read by subroutine RDSNAM.

KFILP - Unit number for the variable ID's.

PRENAM - Name of file corresponding to KFILP. When KFILP = KFILDI, PRENAM is not used and can be read as "DEFAULT". (CHARACTER*60)

Record Type 15 - Format (I3,4XA60) Variable Constants File

This record (plus the terminator record) identifies the file from which the variable constants are to be taken. (Variable constants include plain language description and gridprinting information.) Records are read until the terminator KFILCP = 99 is reached. Maximum number of records, sans the terminator record, = 1. This Record Type 15 is read by subroutine RDSNAM.

KFILCP - Unit number for the constants.

CONNAM - Name of file corresponding to KFILCP. (CHARACTER*60)

Record Type 16 - Format (I9,1XI9,1XI9,1XI3,1XA1,1XI4,1XI3,4X3I2) Variable List

This group of records contains the variable ID's. When KFILP ≠ KFILDI, this group is omitted, and the variable list is taken from another source. Records are read until the terminator ID(1,) = 999999 is reached. Maximum number of records, sans the terminator record, = ND4. This Record Type 16 is read by subroutine RDLVRB. Upon completion of reading this record type, N=1,NPRED.

ID(J,N) - The first 3 (J=1,3) words of the variable ID plus the last 3 digits of the 4th word, followed in order by the components of a threshold value consisting of (1) sign (either minus, or plus or blank for plus, read as A1), (2) 4 digits to follow a decimal point, and (3) 3 digits representing the power of 10 by which to multiply the decimal value just read. For easy reading (only) (1) and (2) above can be separated by a decimal point and (2) and (3) separated by an "E". From these values, the 4th ID word (J=4) is composed.

JP(J,N) - For each variable N, JP(1,N) indicates whether that variable can have a gridprint on Unit No. IP(22), and JP(2,N) indicates whether that variable can have a TDLPACK grid output to unit No. KFILOG. JP(3,N) indicates whether the disposable vector records will be written to Unit KFILOV (see the Data Output section). This is an override feature for the parameters for gridprinting and TDLPACKing in each variable's control file (e.g., 'U400ASLP.CN'.)

CONTROL FILE INPUT: (Name read from 'U150.CN') (Unit = KFILDT)

Record Type 1 - Format (7I10) Date List

When the dates are not provided in file 'U150.CN', this group of records determines the date/times for which data are to be input and processed. If KFILDT (read in Record Type 4) = KFILDI (the input unit number as specified in DRU150), this file is omitted.

IDATE(J) - Initial date list, which may contain negative values indicating date spans. When a negative occurs, all dates between this value and the previous positive date are filled in at the increment of hours specified in INCCYL. This input date list is modified in subroutine DATPRO to contain the complete date list with the dates in the spans filled in (J=1,NDATES). Dates are input as YYMMDDHH and modified to YYYYMMDDHH. This list is read by subroutine RDI which eliminates any zeros found in the input. Terminator is 99999999. The date/times put into Record Type 5 must be such that all could be arrived at by successively adding INCCYL (read in Record Type 1) to the first date/time in IDATE( ). Maximum number of dates, sans terminator, is ND8.

CONTROL FILE INPUT: (Name read from 'U150.CN') (Unit = KFILD(1))

Record Type 1 - Format (7(A8,1X)) Station List

When the station list is not provided on file 'U150.CN', this group of records identifies the stations (or locations) to be used in the analyses. It is not needed when KFILD(1) = KFILDI; in this case, the call letters are read from KFILDI. It is also not needed when KFILD(1) = KFILD(2); for this case, both the station list and the associated information are read according to the format in the following control file description for Unit No. KFILD(2).

CCALL(K) - Call letters (or other 8-character location designator) of stations (or locations) which are to be used (K=1,NSTA). This list is read with subroutine RDC, which eliminates any blanks found in the input. Terminator is '99999999'. Maximum number of stations, sans terminator, is ND1. (See Record Type 9, Control File 'U150.CN', unit KFILDI for additional information.) (CHARACTER*8)

CONTROL FILE INPUT: (Name read from 'U150.CN') (Unit = KFILD(2))

Record Type 1 - Format Station Locations
(A8,1XA8,1XA17,4XA2,8XI5,1XA1,F7.4,1XA1,F8.4,9XA8,3(1XA8),12XI5)

This group of records provides information about the stations (or locations) which are to be used. It is needed unless KFILD(1) = KFILD(2) = KFILDI; for this case, both the station list and the associated information are read from the input control file KFILDI. When KFILD(1) = KFILD(2) ≠ KFILDI, this control file is the same as the one above and both the station list and the associated information are taken from this file. If both files are needed (KFILD(1) ≠ KFILD(2) ≠ KFILDI), the call letters here are matched with those in the station list and the appropriate information extracted. When this station directory is alphabetical and NALPH = 1 (see Record Type 3), the final list of stations will be alphabetical no matter the order read in. However, the directory used in MOS-2000 is alphabetized by the new ICAO call letters, which would eliminate the possibility of alphabetizing if the old 3-letter call letters were used.) The number of stations in this directory is not limited, except when it also constitutes the list to be kept, in which case the number of entries is limited by ND1-1. No terminator is used. Most of this information is used only within subroutine RDSTAD or RDSTAL to give information about the stations. Either the new ICAO or old 3-letter call letters can be used according to the value of NEW (see Record Type 3).

CCALLD(K,J) - Call letters (or other character location designator) of stations (or locations) (J=1). As stated above, these call letters are matched with those in the station list unless KFILD(1) = KFILD(2), in which case these call letters comprise the station list and at the completion of reading this file, K=1,NSTA. When NEW = 1, CCALLD(K,1) is read from the first field (A8) and CCALLD(K,2) is read from the second field (A4). When NEW ≠ 1, CCALLD(K,1) is read from the second field and CCALLD(K,2) is read from the first field.

NAME(K) - 20-character name of station. This is used for visual identification of the station in certain output. Format is A17,1XA2; this provides for a 17-character name, a blank, and a 2-character state abbreviation. Note that the last three characters in the "name" field in the directory are not used. (CHARACTER*20)

NELEV(K) - Elevation of the station. Format is I5.

SIGNLA - Sign of the latitude of the station, read as either "S" or "N". When read as "S", the latitude is set negative indicating South latitude. Format is A1.

XLAT - Latitude in degrees. Format is F7.4.

SIGNLO - Sign of the longitude of the station, read as either "E" or "W". When read as "E", the longitude is modified to make all longitudes West. That is, longitude will range from 0 through 360 and be in degrees West over the United States. Format is A1.

LONDD - Longitude in degrees west. Format is F8.4.

CCALLD(K,J) - Call letters of substitute stations (or locations) (J=3,6) (K=1,NSTA).

IWBAN(K) - The WBAN number of the station. Format is I5.

The number of stations in this directory is not limited, except when it also constitutes the list to be kept, in which case the number of entries is limited by ND1-1. No terminator is used.

CONTROL FILE INPUT: (Name read from 'U150.CN') (Unit = KFILP)

Record Type 1 - Format (I9,1XI9,1XI9,1XI3,1XA1,1XI4,1XI3,4X3I2) Variable List

When the variables to use are not in file 'U150.CN', this group of records contains the variable ID's. When KFILP = KFILDI, this file is omitted, and the variable list is taken from input file KFILDI. Records are read until the terminator ID(1,) = 999999 is reached. Maximum number of records, sans the terminator record, = ND4. This Record Type 1 is read by subroutine RDLVRB. Upon completion of reading this record type, N=1,NPRED. See Record Type 16, file 'U150.CN', unit KFILDI, for other details; the format is exactly the same.

CONTROL FILE INPUT: (Name read from 'U150.CN') (Unit = KFILCP)

Record Type 1 - Format
(3(I9,1X),I3,2XA32,I3,F11.4,F10.4,F9.2,F8.2,2XA12) Variable Constants

This group of records contains information about the variables, as defined by ID(J,) (read previously) and IDTEMP(J). This file should be universally useable by all LAMP-2000 users and is expected to be a separate file. Note that the format matches that for a file input to MOS-2000 programs such as U600, U660, and U850.

IDTEMP(1) - First word of variable ID, either with or without the "B" and "DD". This is matched with all variables read for this run, both with and without the "B" and "DD". When there is a match, the constant information with IDTEMP(1) is stored as indicated below.

IDTEMP(J) - These 3 words (J=1,3) are currently not used, but are meant to correspond to the ID words 2-4.

PLAINT - When IDTEMP(1) matches an ID(1,N), PLAINT is stored in PLAIN(N). These 32 characters are used for visual identification of variables in certain output. Although 32 characters are allowed, the first 5 are reserved for a height indicator (e.g., 1000-), and those after character 23 may be overwritten for vertical or time processing. This generally leaves 18 characters besides height, smoothing, and other processing indicators. (CHARACTER*32)

ISCALT - When IDTEMP(1) matches an ID(1,N), ISCALT is stored in ISCALD(N). This variable is the scaling constant (power of 10) to use when packing the interpolated data.

SMULTT - When IDTEMP(1) matches an ID(1,N), SMULTT is stored in SMULT(N). This variable is the multiplicative factor (power of 10) to use when gridprinting the gridpoint data.

SADDT - When IDTEMP(1) matches an ID(1,N), SADDT is stored in SADD(N). This variable is the additive factor to use when gridprinting the gridpoint data.

CONTT - When IDTEMP(1) matches an ID(1,N), CONTT is stored in CINT(N). This variable is the contour interval to use when gridprinting the data. It applies to the units in which the data exist on the packed input tapes, not after manipulation by SMULTT and SADDT.

ORIGNT - When IDTEMP(1) matches an ID(1,N), ORIGNT is stored in ORIGIN(N). This variable is the contour origin to use when gridprinting the data It applies to the units in which the data exist on the packed input tapes, not after manipulation by SMULTT and SADDT.

UNITST - When IDTEMP(1) matches an ID(1,N), UNITST is stored in UNITST(N). These characters define the units of the data after application of SMULTT and SADDT and are used in the visual inspection of the data in gridprinted maps. (CHARACTER*12)

Even when a match is found, the rest of the ID's in ID(1,) are checked because there might be more than one match.

Other processing occurs with the reading of these records in RDPRED, much of it associated with the plain language description. See Chapter 4, Variable Identification, of "MOS-2000" TDL Office Note 00-1 for details.

DATA INPUT:

All archived model and vector data input to U150 will be in the TDLPACK format read from sequential files, the vector data being accompanied by one or more directory records. The unit numbers and file names are provided in KFILIN(J) and NAMIN(J) (J=1,NUMIN), respectively. Constant data (if any) are provided in the random access MOS-2000 External File System; these data are also in TDLPACK format, except for the call letters (directory) record, on Units 45 through 49; gridpoint data can be provided on Unit No. 44, and has no directory record. Reading is done with standard FORTRAN binary reads, and unpacking is done with subroutine UNPACK and its associated subroutine UNPKBG.

A. SEQUENTIAL GRIDPOINT DATA

One or more sources of gridpoint data are accommodated, the dataset names and unit numbers having been provided to NAMIN(J) and KFILIN(J), respectively, from the control file 'U150.CN', Record Type 6. Each file is closed when an EOF is reached.

The definition of the grid to which the data in each record pertain is contained in that same record. The interpolation routines use this information, and precompute station locations (convert latitude/longitude to I/J) so that this computation doesn't have to be done for each record. However, a check is always made to assure that the correct grid information is used. In fact, not only can the data sources have different grid definitions, the definition within a particular source can change once or many times. It might be normal for the grid definition to change during the archive period. Be aware, though, that each time the grid definition changes when reading records that are to be used, extra computation must be done. Normally, ND11 (see "Setting up the Driver DRU150") will be $\geq$ the number of input grid sources (unit numbers).

For gridpoint data, the unit number KFILIN( ) must be < 80, and the model number MODNUM( ) must match the model number DD in the ID's for which data are to be taken from that unit. Unit numbers 44 through 49 are reserved for the MOS-2000 External Random Access System. The RR is operative for the lookback feature.

B. SEQUENTIAL VECTOR DATA

One or more sources of vector data are accommodated, the dataset names and unit numbers having been provided to NAMIN(J) and KFILIN(J), respectively, from the control file 'U150.CN', Record Type 6. Each file is closed when an EOF is reached.

Each source (file) must have a directory record at the beginning, and one or more other directory records can occur in the same file following a trailer record (see MOS-2000 Software Documentation, Chapter 6). Each file, even read on the same unit number, must have a directory record at the beginning. The occurrence of a new directory record causes some additional computation.

For vector data, the unit number KFILIN( ) must be ≥ 80 (Nos. 97 and 99 are reserved for other uses) and the model number MODNUM( ) (see Record Type 6) must be zero. The RR is (is not) operative when the DD in the variable ID is (is not) zero. In this way, hourly data, which will have DD = 0, will have RR operative.

C. RANDOM ACCESS VECTOR DATA

Up to 5 sources of station-oriented random access data and one source of gridpoint data are accommodated, with the restriction the total is ≤ 5. These data exist in the MOS-2000 External Random Access File System. These data are accessed with prescribed unit numbers, depending on the type of data; see "Restrictions" for more information.

In addition, station latitude and longitude are available from Unit Number KFILD(2) and are stored permanently for use in STALAT( ) and STALON( ), respectively.

DATA OUTPUT

There are five forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). All gridpoint data written will have DD = LAMPNO defined in a Data Statement to be 5. See the individual modules (e.g., U400A) for additional details.

A. Gridpoint ASCII FOR VIEWING OR PRINTING

Gridprinted maps (zebra charts) over the subset area (see Record Type 3 of control file U150.CN) of each variable for which JP(1,) = 1 will be written to unit No. IP(22), provided IP(22) is not zero and provided the appropriate values of NPRT( ) and JPRT( ) indicate gridprinting (these values are in the individual analysis control files). These data are written by individual modules (e.g., U400A).

B. TDLPACK GRIDS FOR CHECKOUT OR QUALITY CONTROL

TDLPACK grids over the subset area (see Record Type 3) of each variable for which JP(2,) = 1 will be written to the dataset whose name has been provided to **OUTDIS** and unit number to **KFILOG** from the control file 'U150.CN', Record Type 9, provided KFILOG is not zero and provided the appropriate values of NTDL(J) and JTDL(J) indicate TDLPACKing (these values are in the individual analysis control files). These "disposable" grids are written by individual modules (e.g., U400A) and pertain to each analysis pass J.

The IDs for these TDLPACK disposable grids are:

ID(1) = CCCFFF0DD, the same as the analysis being computed.
ID(2) = pass number*10000 for the analyses and 990000 for the first guess (the LLLL portion) + the level [UUUU = IDPARS(7)].
ID(3) = 0.
ID(4) = 000 (unsmoothed) or 010 (smoothed). The smoothing is indicated the same as in other MOS-2000 IDs (S = 1 for 5-point smoothing). (However, note that the 5-point

smoothing in U150 is not necessarily the same as the 5-point smoothing in U201.)

For the discontinuous fields in U400D, the final analysis is also written with ID(2) = 0 and ID(4) = 0 to facilitate use by U203 and gempack.

The DD in ID(1) is normally LAMPNO = 5; however for forecasts, DD = LAMPNO + INCDD. This is so results from different runs of, say, U454 can be compared in the verification programs. This is a way of distinguishing them. It is suggested INCDD, when not 0, be 10, 20, 30, etc.

See U450, U452, U453, and U454 for model specific output.

C. TDLPACK VECTORS FOR CHECKOUT OR QUALITY CONTROL

TDLPACK vectors of observations for each variable for which JP(3,) = 1 will be written to the dataset whose name has been provided to **OUTVEC** and unit number to **KFILOV** from the control file 'U150.CN', Record Type 10, provided KFILOV is not zero. Each variable so designated will have two records written per pass--(a) a record with only tossed observations not set to 9999, and (b) a record with only questionable observations not set to 9999. "Questionable" is defined by a threshold by pass which is a fraction of the ER1 read in the individual module .CN files. An observation called "tossed" is not also called "questionable." These data are written by individual modules BCD and BCDW in U400A and U400B, respectively. These records are not applicable to discontinuous variables dealt with by U400D.

The IDs for these TDLPAK disposable vectors are:

ID(1) = CCCFFF000, same as the observation (s) being analyzed.
ID(2) = pass number*10000 for the "tossed" data (the LLLL portion).
= pass number*10000 + 100000 for the "questionable" data (the LLLL portion).
ID(3) = 0.
ID(4) = 0.

D. SEQUENTIAL TDLPACK GRIDPOINT ARCHIVE

TDLPACK grids are written to the dataset whose name has been provided to **OUTNAM** and unit number to **KFILIO** from the control file 'U150.CN', Record Type 8, provided KFILIO is not zero. These data are written by U150, not by the individual modules. No data will be output for a particular date unless all are available--that way, a partial archive day will not occur. When KFILIO ≠ 0, the final output (last pass of analysis) will be written for each variable. The mesh length of the archived grids is determined within the individual modules, U400A, etc.

The IDs for these TDLPAK archive grids are:

ID(1) = CCCFFF0DD, same as the analysis being computed.
ID(2) = 0.
ID(3) = 0.
ID(4) = 0 (note that this is zero whether or not the grid was smoothed; that is, this is the final analysis, no matter how it was arrived at.

The DD in ID(1) is normally LAMPNO = 5; however for forecasts, DD = LAMPNO + INCDD. This is so results from different runs of, say, U454 can be compared in the verification programs. This is a way of distinguishing them. It is suggested INCDD, when not 0, be 10, 20, 30, etc.

E. SEQUENTIAL TDLPACK VECTOR ARCHIVE

TDLPACK vectors are written to the dataset whose name has been provided to **OUTQCV** and unit number to **KFILQC** from the control file 'U150.CN', Record Type 11, provided KFILQC is not zero. These data are written by the individual modules, and therefore if a run aborts, some data may be available but not all. When KFILQC ≠ 0, one record will be written for each case (date/time) for each continuous variable containing the "accepted" stations, values for the tossed stations being set to 9999. For a vector data record to be written, the ID must have been read in the 'U150.CN' file, Record Type 16. Normally, for wind analyses, the ID for u and v components and speed will all be there (although the analysis would be done if only one or two of them were there). Although not being analyzed directly, the wind direction will be written as a quality controlled vector if the wind direction ID (004200005) is included in the 'U150.CN' file. These quality controlled obs are written only when error checking is done on the last pass, which would be normal.

The IDs for these TDLPAK archive vectors are:

ID(1) = C1CFFF000, the same as the variable(s) being analyzed, except the middle digit of CCC (which is always zero) is replaced by a "1" signifying quality controlled observations.
ID(2) = 0.
ID(3) = 0.
ID(4) = 0.

EXAMPLE CONTROL FILE: 'U150.CN'

An example exists as file 'U150.CN' in directory 'users24/glahn/lamp/dru150' on ice. The easiest way to set up a run is to take an existing control file and modify it; DO NOT START FROM SCRATCH.

OUTPUT:

Output that can be printed can be put onto one or more of several files as described above under control file 'U150.CN', Record Type 1 and the definition of KFILDO in the driver DRU150. All errors will be to the

default output file (KFILDO as possibly modified by IP(1)) as well as possibly to other files as defined by IP( ). Every effort has been made to notify the user of problems and potential problems and to proceed under user control without jeopardizing data.

RESTRICTIONS

Most restrictions are only associated with variables in PARAMETER statements in the driver which control array sizes. In some places, machine word length is a factor, and 32-bit and 64-bit machines have been provided for by the use of PARAMETER statements, and the setting of L3264B to either 32 or 64. Only north polar stereographic gridpoint data are accommodated. Formats and other guidelines in MOS-2000 documents are followed.

The grid size NXL by NYL and pole position NXPL, NYPL must be such that NXL-1 and NYL-1 are evenly divisible by 4 and NXPL-1 and NYPL-1 are evenly divisible by 2. Otherwise, inconsistencies may occur in changing from one grid mesh length to another.

HP workstations accommodate optionally compiled statements with a "D" in Column 1. The mainframe IBM treats them as comments.

The TDLPACK format for gridpoint data provides for both a primary and a secondary missing data indicator. If a primary value is found during unpacking of gridpoint (not vector) data, this is flagged at U150 completion by a statement on the output KFILDO file. However, the computations on grids do not provide for missing values, and the packed output records will not indicate that missing values may be present. It is not expected that archived MOS-2000 gridpoint data will have missing values. If missing values become possible at some time, then U150 will have to be modified to handle the situation. An exception to this is in the analysis of discontinuous variables (e.g., ceiling height) in U400D and forecasts in U454.

Arrays FD1( ), FD2( ), ... FD7( ) have been provided for general work arrays.

The unit numbers for the random access files must be in the range 44 through 49 (although U150 will not write to a random access file); those unit numbers are used for the following purposes related to values of CCC in ID(1):

Unit No.	CCC Range	Use
44	400-499	Gridpoint data
45	400-499	"True" constants (rel. freq., means, etc.)
46	500-599	1-d and 2-d constants, probably for U201
47	800-899	Thresholds for best category forecasts, etc.
48	200-299	Forecasts read only
49	200-299	Forecasts read/write

A unit number of zero should never be read in the .CN control file. If a file is not needed, just omit it from the control file; this will be interpreted in U150 and called routines as 0. The routine RDSDAM does not recognize a zero unit number.

COMMENTS

It is expected that U150 will be run on workstations (e.g., the HP GDPs). There seems to be no reason to put U150 on the IBM mainframe. If that becomes necessary, some adjustments might have to be made.

Each source of vector data has a directory record associated with it which pertains to the vectors following it up until another directory record is encountered. The directory indicates, in terms of station identifiers, where the datum in each record is to be found for a particular station's identifier (usually call letters). However, because station identifiers sometimes are changed and it is desired to mix data from the same location regardless of the particular identifier, provision is made for up to 5 substitute stations. When the new ICAO identifiers are being used (NEW = 1), the first substitute station is the old call letters taken from the second field in the station directory. When the old call letters are being used (NEW ≠ 1), the first substitute station is the ICAO identifiers from the first field in the directory. The directory also contains up to 4 other stations (see the station directory documentation) that can be substituted for the station identifiers being used. Subroutine RDDIR gives additional details.

Most errors are output for printing starting with **** to the file with number designated by IP( ) (see Record Type 1) and a count is kept for matching with NSKIP and JSTOP (see Record Type 3). Missing data will usually not be counted as an error, but a diagnostic is provided. It is not always obvious what is an error as opposed to something that might be expected to happen occasionally; therefore, the count can't be considered as absolute. At the end of Day 1 and at the end of the run, the number of "errors" and the number of missing variables (data records) are printed.

Some information is provided for printout on each run that may seem repetitive. However, it is believed that the user should monitor this information and investigate seeming abnormalities.

When no variable in a run has a particular model number, the file containing that model will not be needed; therefore, the file is closed. Closing the file keeps the data from being read when not needed.

It is not necessary that each variable be present for Day 1 for that variable to be used on subsequent days.

A count is kept of each time a missing value indicator is found when unpacking grids (not vector data). This is generally unexpected, and at the end of a run, the value is printed if non-zero. However, the radar data and analyses and forecasts of discontinuous fields will contain missing values.

A "missing" value of 9997 encountered in input vector data is set to PXMISS (see Record Type 3). This allows 9997 to remain intact (PXMISS = 9997), set to zero (PXMISS = 0), set to "normal" missing (PXMISS = 9999), or even some other value.

Although the primary purpose of U150 is to produce packed gridpoint data, writing of such data is not done if KFILIO = 0 (see Record Type 8). That is, no output unit number and file name have been provided. A secondary purpose of U150 is to produce packed quality controlled observational

(vector) data, writing of such data is not done if KFILQC = 0 (see Record Type 11). That is, no output unit number and file name have been provided. This feature can be used for checkout runs.

The modules of U150 (e.g., U400A) operate on grids of variable dimension. The grid length is specified at each stage of the process. For instance, the first guess and first pass of the sea level pressure analysis may be at 80 km.--the mesh length of the NCEP grid available for a first guess. Subsequent passes of the analysis may use other mesh lengths. U150 interpolates either linearly or quadratically to obtain a denser grid, or will use alternate grid points to thin a gird. Note, however, that interpolation does not work well for discontinuous fields and must be avoided. This means, for instance, that if U454 uses 10 km for a mesh for a particular variable (it can vary by variable), the analysis in U400D should be that also. It is important to note that the value of MESH is the mesh length of the active grid at that point and that XP( ) and YP( ) contain the IX and JY positions of the stations <u>relative to that grid</u>. That is, MESH and XP( ) and YP( ) follow the grid. Or you could think of the grid and XP( ) and YP( ) following MESH--same difference. Subroutines exist for these mesh length transformations which always have to be done in steps of a factor of two.

It is assumed that the grid that is to furnish a first guess (e.g., an NCEP AVN model or previous LAMP forecast) is large enough to encompass the primary LAMP grid (see NXL and NYL in Record Type 3). The primary LAMP grid is defined for this run by a lower left latitude and longitude and a "basic" mesh length equal to a quarter Bedient. This has, then, a defined NX by NY extent and pole position. The first step in obtaining a first guess (assuming it is from a grid and not an average of observations or a constant) is to position the primary LAMP grid within the donor gird at the donor gird mesh length (subroutine POSGRD is used). The next step is to make the guess the desired mesh length (subroutine SIZEGR is used).

Five forms of output (besides diagnostics, etc.) are provided for (see the section Data Output). The primary output is for gridpoint archival and is written to unit number KFILIO (providing it is not zero). At the successful completion of all modules, the final analysis and forecast grids are TDLPACKed and written. These will all be over the primary LAMP grid but at whatever mesh length was computed (e.g., the last pass of sea level pressure analysis may have been at MESH = 40 km; if so, that is what is written). Since the grids are self documenting in TDLPACK, reading software can handle variable sized grids.

The quality control of continuous variables, with U400A and U400B, is quite good, and an archive of these quality controlled observations is possible on Unit No. KFILQC. This archive can be used in development, verification, etc.

Because the LAMP grid may be quite large, it is inconvenient to try to look at all of it at one time. Therefore, a subset of the LAMP area is provided as either a gridprint (zebra chart) on unit number IP(22) or as a TDLPACK record on unit number KFILOG. For instance, the subset (see NXGMIN, etc., in 'U150.CN', Record Type 3) can be of such a size that the gridprint will exactly fill one page of print. The TDLPACK output is expected to be used in GEMPACK-related software, U203, although the TDLPACK output on unit number KFILIO can also be used for that purpose. A feature of the subset is that every grid of "continuous" variables will be

on a mesh length of MESHL, and therefore will overlay with grids of other variables. For this to happen, some grids may be thickened or thinned from their original mesh length. In a similar manner, every grid of "non-continuous" variables will be on a mesh length of MESHD, and will therefore overlay with grids of other such variables. When MESHD = MESHL, then all grids will overlay. Note that thickening of discontinuous grids cannot be done; they must be created at, or less, the mesh length to be displayed.

Consistency of IU400A, IU400B, IU400D, IU450, IU453, IU454, and the variables in the list in Record Type 16 is expected. For instance, U454 cannot advect with the LAMP 1000-mb wind unless U450 has been run, and that requires U400A. U400A uses results of U400B if available, but will run without the quality controlled vector winds generated by U400B.

CHARACTERISTICS OF THE U150 LOOKBACK FEATURE

U150 saves data from previous cycles (date/times previous to NDATE, the date/time being processed) when necessary. Such data are denoted by RR > 0 in ID(3). The subroutine AUGIDS is used to: a) provide a list of all variables U150 will need and b) to provide individual .CN file names so that IBACKN and IBACKL can be read from each to determine how far back the "lookback" feature needs to operate and how long the data need to be saved. AUGIDS can have IDs added to it as necessary.

SETTING UP THE DRIVER DRU150

The preparation of the driver for a particular U150 run is relatively painless; it consists of using a template driver and modifying as necessary certain PARAMETER statements. These statements set values of:

L3264B - Set to 32 for a 32-bit machine (e.g., the HP's) or 64 for a 64-bit machine (e.g., the CRAY).

NBLOCK - Set to 6400/L3264B.

MAXSTA - Maximum number of stations (or points) that can be dealt with (i.e., used in the analysis). Note that this does not include the number of stations in the directory (read on Unit No. KFILD(2)) unless, of course, the station directory is to be used as the station list. ND1 is set as the maximum of NBLOCK and MAXSTA.

ND2 - ND2*ND3 is the maximum size of the grid that can be dealt with. ND2 and ND3 are set separately to highlight the possible dimension of the grids. However, in the called routines, the size is only limited by the product, not each dimension individually.

ND3 - See ND2.

ND4 - The maximum number of variables in the variable list.

ND5 - Dimension of IPACK( ), IWORK( ), and DATA( ). ND2X3 and ND5 should be set in the driver DRU150:

```
ND2X3 = MAX(ND1,ND2*ND3)
ND5=ND2x3
```

ND6 - Maximum number of all sequential file input sources (e.g., models) that can be dealt with. If data from a model is on two files, then this would be counted as two, not one, etc.

ND7 - The size of ISO( ), IS1( ), IS2( ), and IS4( ). This would normally be 54.

ND8 - The maximum number of date/times that can be used. This is the "extended" date list, not just the values read in.

ND9 - The maximum number of fields (variables) stored in the MOS-2000 Internal Storage System. Since all fields are stored for Day 1, ND9 must be large enough to hold all records of the date/time of Day 1 on all input files.

ND10 - The number of words of storage provided in the variable CORE( ) for the MOS-2000 Internal Storage System. When this is filled, a scratch disk file is used. Too small a number will result in more disk accesses than necessary (although caching may alleviate that); too large a number will result in wasted memory and possible excess paging.

ND11 - The maximum number of grid combinations that can be dealt with. For instance, if the NGM and AVN data are being used, and the grids are different, then ND11 would be $\geq$ 2.

Do not change the computation for the variable L3264W, and probably not for NBLOCK. NBLOCK is the block size in words for disk records for the MOS-2000 Internal Storage System. (Experimentation may determine that a larger value is desirable.) Note that ND1 must be GE NBLOCK.

The user can see from the template what effect each of these values has on storage from where it occurs in the DIMENSION statements. Some have relatively little effect (e.g., ND7), while others have considerable effect (e.g., ND2*ND3). Every effort has been made so that the variables will not be overflowed if values too small are used; however, some danger may still exist.

The "startup" control file 'U150.CN' is opened in DRU150, so that this aspect of U150 can be rather easily changed without recompiling or having separate version of the main (sub)routine U150.

NONSYSTEM ROUTINES USED

Use the load line in file 'home24/glahn/lamp/dru150', dataset 'u150.com' on ice.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: u150lib. The driver is in home24/glahn/lamp/dru150 on ice.

U400A

ANALYZES QUASI-CONTINUOUS VARIABLES

Harry R. Glahn
December 1, 2000

PURPOSE: To produce objective analyses of quasi-continuous variables, such as observed surface temperature, sea level pressure, or forecast probability of precipitation, on a variable-sized computational grid. Analyses are performed with the Bergthorssen-Cressman-Doos successive approximation techniques with a first guess field provided from an NCEP model, a LAMP forecast from a previous hour, a constant field, or the average of all data values to be used in the analysis. Data to be analyzed are error checked as specified in a control file.

U400A is called from U150 and is controlled by a set of control files 'U400AXXX.CN', where XXX pertains to a specific variable (e.g., XXX = SLP for sea level pressure). 'U400AXXX.CN' is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below (e.g., IP22).

CONTROL FILE INPUT: 'U400AXXX.CN' (Unit = KFILDI)

Record Type 1 - Format (8I4,F8.0,1X,A16,I4,2F4.0,F6.0,I4) Overall Control

This record contains the number of analysis passes to perform and other information which is not pass specific.

NPASS - Number of analysis passes to make, maximum of 6. However, one or more of these can be omitted (e.g., the first) by setting a corresponding NTYPE(,) = 0 (see below).

IFSTGS - Controls gridprinting to unit number IP22 or TDLPACKing and writing of the first guess to unit number KFILOG of the subsetted area:
0 = first guess is neither gridprinted nor TDLPACKed,

1 = the first guess is to be gridprinted,
2 = first guess is to be TDLPACKed and written,
3 = first guess is to be both gridprinted and TDLPACKed and written.

IGUESS(J) - The type of first guess to use in priority order (J=1,4).
1 = A constant (see GUESS),
2 = An NCEP model forecast,
3 = A LAMP forecast from a previous hour,
4 = The average of all data values to be used in the analysis.
For example, if IGUESS(1) = 2, then the first priority for first guess is an NCEP forecast. If IGUESS(2) = 1, the second priority is the constant GUESS (see below). If one source is not available, the next priority is tried.

IBACKN - The maximum number of cycles to go back from the "current" cycle for an NCEP forecast to be used as a first guess at 6-h intervals. For instance, if IBACKN = 2 and the current run of the NCEP forecast model is not available, U400A will look for 2 previous runs at 6-hr intervals. By setting IBACKN = 2, then either the 6-h or 12-h previous run could be used in that priority order. When a previous run is used, the projection(s) are adjusted accordingly. The "current run" of the NCEP forecast model is defined to be the cycle of the LAMP analysis, mod 6. That is, if the LAMP analysis is for 05Z, the current NCEP model run is defined as 00Z. No assumption is made as to when the model actually runs. If only the current run is desired (no lookback), set IBACKN = 0. If IBACKN < 0, an NCEP forecast is not used even when IGUESS( ) = 2.

IBACKL - The maximum number of cycles to go back for a LAMP forecast to be used as a first guess at 1-h intervals. For instance, if IBACKL = 2, and the 1-h forecast from the previous run of the LAMP forecast model is not available, U400A will look for 1 more previous run. When a previous run of LAMP is used, the projection(s) are adjusted accordingly (e.g., a 2-h forecast is used from a LAMP run 2 hours earlier). Note that IBACKN refers to the number of runs to look for back from the "current" run; IBACKL refers to the total number of runs to look back. If IBACKL ≤ 0, a LAMP forecast is not used even when IGUESS( ) = 3.

GUESS - The constant value to use as a first guess when IGUESS( ) = 1.

TITLE - A 16-character title for this variable, the variable being defined by the 4-word ID read by U150. (CHARACTER*16)

NSMTYP - Type of smoothing.
1 = Normal 5-point.
2 = Same as 1, except no change is made unless one or more of the points to contribute to the new value has been changed.
3 = 9-point smoothing used on last pass only, any other pass defaults to NSMTYP = 2.
4 = For passes GE 4 for sea level pressure (SLP) only, same as 2 except a point is not changed if it is lower than both points above and below it or if it lower than both side points. Diagonals are also considered. For other passes and other variables, defaults to 2. (Also see Comments.)

WNDTHR - Threshold in knots for wind speed for wind to be used in the sea level pressure analysis. That is, the wind will not be used unless the speed is > WNDTHR. Used only for SLP analysis.

WNDTRN - Degrees to turn the surface wind toward lower pressure before applying geostrophic correction. Used only for SLP analysis.

WNDGRD - Parameter for converting wind speed to sea level pressure gradients. Used only for SLP analysis.

I400ADG - 1 = Diagnostics about analysis will be written to KFILDO.
0 = Most diagnostics will not be written. This occupies the same position as WNDTHR when this is not a SLP analysis.

Record Type 2 - Format (6I8) Mesh Lengths

MSHPAS(J,L) - Nominal mesh length in km for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. Since MESH cannot equal 0, if MSHPAS(J,) is zero, pass J is skipped.

Record Type 3 - Format (6F8.0) Error Criteria

ER1(J,L,M) - Error criteria in units of the data being analyzed for each pass (J=1,6), for each first guess option (L=1,4), and for each month of the year (M=1,12). Only NPASS values are read and used from each of 48 records. The value M for the month being processed will be used.

Record Type 4 - Format (6I8) Type of Correction

NTYPE(J,L) - Type of correction to make for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. The values of NTYPE (,) are explained in the section "Analysis Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985). If NTYPE(J,) = 0, pass J is skipped.

Record Type 5 - Format (6F8.0) Smoothing Parameter

B(J,L) - Smoothing parameter for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. The smoothing procedure is explained in the section "Smoothing Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985).

Record Type 6 - Format (6F8.0) Radius of Influence

R(J,L) - Radius of influence in grid units for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. Generally speaking, a data value affects a gridpoint only when it is within R(,) grid units of the gridpoint. Note that the actual distance depends on the gridlength being used on that pass. This is further explained in the section "Analysis Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985).

Record Type 7 - Format (6I8) Type of Interpolation

ITRPLQ(J,L) - The type of interpolation to be performed whenever interpolation is necessary for each pass (J=1,NPASS) and each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. If one pass of an analysis is on a grid of, say, 80 km, and the next pass is on 40-km, then interpolation is necessary to arrive at the "guess" for the next pass. ITRPLQ(,) is also used as necessary to arrive at thickened grids for the subsetted (see Data Output section) area. The values are:
1 = bilinear,

2 = biquadratic.

Record Type 8 - Format (6F8.0) R Star Modification of Radius of Influence

RSTAR(J,L) - The fraction of R(,) to use when the datum is outside the grid for each pass (J=1,6) and first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. That is, R(,) may be 8 grid units, but since the interpolation into the grid to get the "analysis" at that time becomes extrapolation outside the grid, it may be desirable to cut the 8 to, say, 2. This can be done by using RSTAR(,) = .25.

Record Type 9 - Format (6F8.0) Multiplicative Constant for Gridprinting

SMULT(J) - The multiplicative factor to use before gridprinting the data for each pass (J=1,6). Only NPASS values are read and used. That is, if visibility is in miles and fractions, it may be desirable to use SMULT( ) = 100 so that the quarter mile values can be seen.

Record Type 10 - Format (6I8) Additive Factor for Gridprinting

SADD(J) - The additive factor to use before gridprinting the data for each pass (J=1,6). Only NPASS values are read and used. Usually this is zero.

Record Type 11 - Format (6F8.0) Origin for Gridprinting

ORIGIN(J) - The origin when gridprinting the data for each pass (J=1,6). Only NPASS values are read and used. That is, the base contour will be at ORIGIN( ).

Record Type 12 - Format (6F8.0) Contour Interval for Gridprinting

CINT(J) - The contour interval when gridprinting the data for each pass (J=1,6). Only NPASS values are read and used. CINT( ) is in reference to the original units, not after the multiplication by SMULT( ).

Record Type 13 - Format (6I8) Gridprint Option Unsmoothed Grids

NPRT(J) - Gridprinting option for the unsmoothed analysis for each pass(J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 14 - Format (6I8) Gridprint Option for Smoothed Grids

JPRT(J) - Gridprinting option for the smoothed analyses for each pass (J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = gridprint,

0 = otherwise.

Record Type 15 - Format (6I8) TDLPACK Option for Unsmoothed Grids

NTDL(J) - TDLPACKing option for the unsmoothed analyses for each pass (J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = TDLPACK and write,
0 = otherwise.

Record Type 16 - Format (6I8) TDLPACK Option for Smoothed Grids

JTDL(J) - TDLPACKing option for the smoothed analyses for each pass (J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = TDLPACK and write,
0 = otherwise.

Record Type 17 - Format (6I8) Weights for Wind Correction for SLP

WNDWT(J,L) - For SLP only, fractional weight to apply to correction estimated by the wind and geostrophic relationship for each pass (J=1,6) and first guess option (L=1,4). For instance, 1.0 means apply full weight; 0.5 means apply 50% of the full weight. Only NPASS values are read (and are read even for variables not SLP) and used (for SLP only) from each of 4 records.

DATA INPUT:

Most gridpoint and vector data needed by U400A have been stored by U150 into the MOS-2000 Internal Storage System. GFETCH is used to retrieve the data. Final grids for eventual writing to Unit Number KFILIO are packed and written there by GSTORE, called by subroutine PAWING. However, if a variable has to be computed, the External Random Access file system might have to be accessed; provision is made for this access.

DATA OUTPUT

There are five forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). These generally pertain to all analysis modules, and are explained in U150. Differences are that the discontinuous variables do not have quality controlled observational data written, because there is no effective error checking possible. See the Data Output section of U150.

EXAMPLE CONTROL FILE: 'U400ASLP.CN'

An example exists as file 'U400ASLP.CN' in directory 'lamp2000/dru150/standard_CN' on ice. The easiest way to set up a run is to take an existing control file and modify it.

OUTPUT:

Output is generally explained in the U150 writeup.

RESTRICTIONS

See U150.

COMMENTS

Comments in U150 apply to U400A.

Although the primary purpose of U150 and U400A is to produce packed gridpoint data, writing of such data is not done if KFILIO = 0 (see Record Type 8). A secondary, but important, purpose is to write vector records of observations in which all but the tossed obs have been set to 9999; however, this file will not be written in KFILQC = 0. That is, no output unit number and file name have been provided. A similar record is written to the same file in which all but the "questionable" obs have been set to 9999. Questionable is defined as not meeting a threshold which is a fraction of the primary error criteria ER1 read in U400AXXX.CN. This feature can be used for checkout runs. Note that rendering the unit number = 0 is done by omitting the file in the input .CN file, not by leaving the file name there and setting the unit number = 0.

The modules of U150 (e.g., U400A) operate on grids of variable dimension. The grid length is specified at each stage of the process. For instance, the first guess and first pass of the sea level pressure analysis may be at 80 km.--the mesh length of the NCEP grid available for a first guess. Subsequent passes of the analysis may use other mesh lengths. U400A interpolates either linearly or quadratically to obtain a denser grid, or will use alternate grid points to thin a gird. It is important to note that the value of MESH is the mesh length of the active grid at that point and that XP( ) and YP( ) contain the IX and JY positions of the stations relative to that grid. That is, MESH and XP( ) and YP( ) follow the grid. Or you could think of the grid and XP( ) and YP( ) following MESH--same difference. Subroutines exist for these mesh length transformations which always have to be done in steps of a factor of two.

It is assumed that the grid that is to furnish a first guess (e.g., an NCEP AVN model or previous LAMP forecast) is large enough to encompass the primary LAMP grid. The LAMP grid is defined by a lower left latitude and longitude, a "basic" mesh length equal to a quarter Bedient, and an NX by NY extent; the pole position is also known or implied, being redundant to the other information. The first step in obtaining a first guess (assuming it is from a grid and not an average of observations or a constant) is to position the "LAMP" grid within the donor gird at the donor gird's mesh length (subroutine POSGRD is used). The next step is to make the guess the desired mesh length (subroutine SIZEGR is used).

VARIABLES ACCOMMODATED BY U400A

U150 calls U400A once per cycle. Furnished to U400A is the list of 4-word MOS-2000 variable identifiers for which processing is to be done by some module of U150. U400A has resident a table that contains the variable IDs that U400A is engineered to handle. There is a corresponding table which gives the names of the '.CN' control file for each variable.

U400A searches the table for each U150 input ID and processes those variables that reside in its table ITABLE(,NVAL). For each variable processed (analyzed), its specific '.CN' file is read (e.g.,

'U400ASLP.CN') for sea level pressure). This control file was explained earlier under "CONTROL FILE INPUT."

If it is desired that U400A handle an additional variable, it should be sufficient to:

(1) Increase NVAL in the PARAMETER statement,
(2) Include the Id in ITABLE(,NVAL),
(3) Include in DOTCN(NVAL) the control file name, and
(4) Provide the control file appropriate to the variable.

Currently, U400A is ready to handle four variables. The first guess fields from NCEP or LAMP must be at the correct cycle and projection t.

```
Variable
  Sea Level Pressure:      001201005  0  0  0
    Observations           701200000
  0  0  0
    First guess from NCEP  001200008  0  t  0
    First guess from LAMP  001201005  0  t  0

  Surface Temperature      002301005  2  0  0
    Observations           702000000  0  0  0
    First guess from NCEP  002001008  0  t  0
    First guess from LAMP  002001005  2  t  0

  Surface Dew Point        003301005  2  0  0
    Observations           703100000  0  0  0
    First guess from NCEP  calculated
    First guess from LAMP  003101005  2  t  0

  Saturation Deficit       003410005  0  0  0
    Observations           703100000  0  0  0   dew point
                           708500000  0  0  0   weather
    First guess from NCEP  calculated
    First guess from LAMP  003410000  0  t  0
```

TDLPACK OUTPUT

See the appendix for a listing of both gridpoint and vector packed output to the various files.

ERROR CHECKING PROCEDURE

Generally, an observation is not used if it differs from the current analysis (defined to be the quadratically interpolated value from the grid) by ER1 units of the variable being analyzed for that pass and first guess option. However, if it meets 1.5*ER1 and one of its two nearest neighbors also differs from the analysis by ER1*F (where F is hardwired to 0.6) and in the same direction, then, the observation is accepted. The closest neighbor assisting the observation being checked will also not be discarded if it meets the 1.5*ER1 criterion. This process (a specialized "buddy check") requires nearest neighbors to be found (a very time-consuming process) only infrequently.

Another check is made for sea level pressure only when the pressure is < 1013 mb and the observation is about to be tossed. Before a station is

tossed, its pressure is estimated from each of its two closest neighbors by using the neighbor's pressure and surface wind. If the wind is < WNDTHR, the wind is not used and the check is not made. A factor of 2 is applied to the calculated geostrophic gradient to account for the reduction in surface wind due to friction. If the estimate from either station is within ER1 mb, the station is kept. This check does not affect the acceptance of the neighbors. A main purpose of this check is to attempt to keep good observations near hurricanes and other intense lows.

As an added feature, for sea level pressure only, if the observation is < 970 mb and the observation is 50 mb lower than the analysis, then 100 mb is added to the observation, and if the result is within ER1 mb, the observation is modified and accepted. Also, if the observation is > 1040 mb, and the observation is 50 mb higher than the analysis, then 100 mb is subtracted from the observation, and if the result is within ER1 mb, the observation is modified and accepted. Otherwise, the observation is left alone and not accepted. This check was important in the manual observation days when it was easy to tabulate an observation as, say, 940.5, instead of 1040.5, but may not be very important today.

As explained in the U150 writuep, a record can be written for each pass in which all except the tossed obs have been set to 9999. This allows GEMPAK plotting routines to isolate by color the tossed obs vs the good obs. In addition, a record can be written for each analysis pass in which all except "questionable" obs have been set to 9999. This allows GEMPAK plotting routines to isolate by color those observations which, while not tossed, are questionable. "Questionable" is defined by pass as a function of ER1, the fraction being hardwired in subroutine BCD. The "ER1Q" thresholds depend on variable being analyzed and pass number and have been hardwired to values between 0.51 and 0.75 of the ER1 criteria read in U400AXXX.CN for the 6 possible passes. These records, with an ID that varies by pass number, are for tuning the analysis procedure.

In addition, a single record can be written for each analyzed variable in which the obs tossed on <u>the last pass</u> have been set to 9999. This record has a CCCFFF ID scheme of 71CFFF, where the "1" represents quality controlled obs,

ANALYSIS PROCEDURE

The values of NTYPE(,) (see Record Type 4) are explained below:

0 = Skip this pass. Not usually used.

1 = Correction applied to each gridpoint is the sum of the individual differences between observation and analysis grid values interpolated to stations divided by the number of stations within a radius R of the gridpoint being corrected.

$$C_1 = \frac{1}{n} \sum_{i=1}^{n} (O_{x,y} - A_{x,y})_i \qquad d_i \leq R$$

R = Radius of influence in grid units.
n = Number of observations within radius R of gridpoint being corrected.
$O_{x,y}$ = Observed station value at point x, y.
$A_{x,y}$ = Analyzed value of the variable interpolated from gridpoints to the point x, y.
d_i = Distance in grid units between the observation x, y and the gridpoint.

2 = Correction applied to each gridpoint is the same as type 1 correction except the distance of each station from a gridpoint is considered in the correction for that station. When a type 2 correction is being used, lots of distant stations, even with a few close stations, will result in small corrections being applied to a gridpoint.

$$C_2 = \frac{1}{n} \sum_{i=1}^{n} \frac{R^2 - d_i^2}{R^2 + d_i^2} (O_{x,y} - A_{x,y})_i \qquad d_i < R$$

3 = Correction applied to each gridpoint is the same as type 2 correction except the sum of the distances of each station from a gridpoint is also considered in the correction.

$$C_3 = \frac{\frac{1}{n} \sum_{i=1}^{n} \frac{R^2 - d_i^2}{R^2 + d_i^2} (O_{x,y} - A_{x,y})_i}{\frac{1}{n} \sum_{i=1}^{n} \frac{R^2 - d_i^2}{R^2 + d_i^2}} \qquad d_i < R$$

By having the sum of weights in the denominator, a larger correction is applied to a gridpoint than with a type 2 correction. This may not produce good results when n is very small and R is large. Therefore, when NTYPE(,) = 3 is being used, for those gridpoints where n = 1 and R > 3, a type 2 correction is made. In other words, a Type 3 correction can be made for any n when R < 3 and for any R when n > 1.

For SLP only, in order to try to get and preserve low pressure in tight lows and troughs, the surface wind at a station is used together with its pressure to make a correction according to the geostrophic relationship under certain circumstances. For such a correction to be applied, the values of WNDWT and WNDGRD must not be zero and the pressure at the gridpoint for which the correction is to be made must be $\leq$ 1013 mb. In addition, the gridpoint value must be a minimum in either the X or Y direction and the gradient over 240 km must exceed .012 mb/km. This is calculated by finding the difference between the gridpoint and the minimum of the two end points, each 240 km away. This difference must be exceed .012 mb/km. If the gridpoint is to have a wind correction, the neighboring point in either direction also gets a weight of 75 percent of the full correction. A gridpoint is allowed a maximum of three wind corrections per pass.

For saturation deficit (SD), the values are scaled before analysis so that the large positive values will not overwhelm the important values near zero. After the analysis, these are then rescaled. Values representing saturation are set to -0.06 to preserve a negative value when packing to one decimal place and later a zero line for gridprinting. In addition, for SD, reflectivity and fractional coverage of radar data on a 10-km grid are used to override the analysis in areas where the analysis is > 0. If the fractional coverage (FC) is 1, SD is set = -.06. Otherwise, SD is set SD = SD(1.-FC). That is, SD is reduced when FC > 0. This simple algorithm is applied first for the primary radar data, then when FC = 0, the backup data are applied in the same manner. "Primary" radar data are defined as the most recent radar data available for the particular run within the past hour. "Backup" data are those data also within the past hour that are prior to, and closest to in time, the primary data. Note that while reflectivity data are returned into the subroutine SDOVER where the override is done, the current algorithm does not use this data set. If it were desired to change the algorithm to use reflectivity data, it could be inserted into SDOVER with no other change. Although both reflectivity and fractional coverage data are expected, their absence does not stop the SD analysis from completing--there will just be no radar override.

SMOOTHING PROCEDURE

The smoothing parameter B(,) (see Record Type 5) is explained below:

$$S_{i,j} = \frac{A_{i,j} + b\bar{A}_{i,j}}{1 + b}$$

where:

$S_{i,j}$ = Smoothed gridpoint values at point i, j.
$A_{i,j}$ = Value of gridpoint at point i, j before smoothing.
$\underline{b}$ = Smoothing parameter.
$A_{i,j} = (1/4)\ (A_{i+1,j}+A_{i-1,j}+A_{i,j+1}+A_{i,j-1})$

Low values produce the least smoothing and higher values produce heavier smoothing. A value of zero indicates no smoothing on this pass.

When NSMTYP = 1 is used successively on several passes, areas with no data will be smoothed considerably from the first guess, which may be quite good over the ocean, for instance. In order to control this, NSMTYP = 2 can be used, which does not correct a point unless one of the values involved in the smoothing was changed by the analysis (subroutine SMOTHN). NSMTYP = 3 gives a 9-point smoothing on the last pass only; otherwise it is the same as NSMTYP = 2. NSMTYP = 4 is specialized for sea level pressure, and attempts to keep lows and troughs from being smoothed. If the pressure is $\leq$ 1013 mb, the point is the lowest in a sequence of points in eight directions, and the gradient is $\geq$ 0.012 mb/km over 240 km, then the point is not smoothed. In addition, the neighboring point is smoothed only 0.15 of the value it would normally be smoothed.

NONSYSTEM ROUTINES USED

Use the load line in file 'lamp2000/dru150', dataset 'u150.com' on ice.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: lamp2000/u400alib. The driver is in lamp2000/dru150.

Appendix
LAMP U150/U400A Packed Output

Sea Level Pressure

Unit No. KFILIO -- Archive Gridpoint

```
 001201005         0       0   0        U400A SLP analysis last pass (final)
```

Unit No. KFILOG -- Disposable Gridpoint

```
 001201005    990000       0   0  U400A SLP first guess
 001201005     p0000       0   0  U400A SLP analysis, pass p
 001201005     p0000       0 010  U400A SLP analysis, pass p, smoothed
```

Unit No. KFILOV -- Disposable Vector

```
 701200000     p0000       0   0  U400A SLP obs tossed, pass p
 701200000    1p0000       0   0  U400A SLP obs questionable, pass p
```

Unit No. KFILQC -- Archive Vector

```
 711200000         0       0   0  U400A SLP QC obs (sans tossed, last pass)
```

Temperature

Unit No. KFILIO -- Archive Gridpoint

```
 002301005         2       0   0        U400A Temp analysis last pass (final)
```

Unit No. KFILOG -- Disposable Gridpoint

```
 002301005    990002       0   0  U400A Temp first guess
 002301005     p0002       0   0  U400A Temp analysis, pass p
 002301005     p0002       0 010  U400A Temp analysis, pass p, smoothed
```

Unit No. KFILOV -- Disposable Vector

```
 702000000     p0002       0   0  U400A Temp obs tossed, pass p
 702000000    1p0002       0   0  U400A Temp obs questionable, pass p
```

Unit No. KFILQC -- Archive Vector

```
 712000000         2       0   0  U400A Temp QC obs (sans tossed, last pass)
```

Dew Point

Unit No. KFILIO -- Archive Gridpoint

```
 003301005         2       0   0        U400A Dew Pt analysis last pass (final)
```

Unit No. KFILOG -- Disposable Gridpoint

```
 003301005    990002       0   0  U400A Dew Pt first guess
 003301005     p0002       0   0  U400A Dew Pt analysis, pass p
```

```
 003301005     p0002       0 010  U400A Dew Pt analysis, pass p, smoothed
```

Unit No. KFILOV -- Disposable Vector

```
 703100000     p0002       0   0  U400A Dew Pt obs tossed, pass p
 703100000    1p0002       0   0  U400A Dew Pt obs questionable, pass p
```

Unit No. KFILQC -- Archive Vector

```
 713100000         2       0   0  U400A Dew Pt QC obs (sans tossed, last pass)
```

Saturation Deficit

Unit No. KFILIO -- Archive Gridpoint

```
 003410005         0       0   0  U400A SD analysis, (final, w/radar)
```

Unit No. KFILOG -- Disposable Gridpoint

```
 003410005    990000       0   0  U400A SD first guess (scaled)
 003410005     p0000       0   0  U400A SD analysis, pass p (scaled)
 003410005     p0000       0 010  U400A SD analysis, pass p, smoothed (scaled)
 003410005         0       0   0  U400A SD analysis, (final, w/radar)
 7076xx005         0       0   0  U400A Radar, primary reflectivity
 7077xx005         0 1000000   0  U400A Radar, backup reflectivity
 7077xx005         0       0   0  U400A Radar, primary reflectivity
 7076xx005         0 1000000   0  U400A Radar, backup reflectivity
```

Unit No. KFILOV -- Disposable Vector

```
 003350008         0       0   0  U400A Precip water (interpolated)
 409020000         0       0   0  U400A Terrain (interpolated)
 001000008 110000500       0   0  U400A 1000-500 mb thickness (interpolated)
 703411000         0       0   0  U400A Saturation thickness (interpolated)
 708000000         0       0   0  U400A Saturation deficit, (scaled)
 703410000     p0000       0   0  U400A SD obs tossed, pass p
 703410000    1p0000       0   0  U400A SD obs questionable, pass p
```

Unit No. KFILQC -- Archive Vector

```
 713410000         0       0   0  U400A SD QC obs (sans tossed, last pass)
```

Notes pertaining to U400A:

1) For radar data, the xx in 6xx and 7xx can be 00, 15, 30, or 45, meaning those minutes after the hour. For backup, the RR position in ID(2) is used, which in this case does not mean a full hour.
2) Saturation deficit when scaled is so indicated; otherwise, it is not.
3) Unscaled saturation deficit "observations" (the values being analyzed) never exist unscaled. The corresponding analyses are available as disposable grids, without radar data.
4) The radar data are available on a 10-km grid. For them to be used, the last pass analysis must be at 10 kilometers. No provision is made for the last pass analysis to be at, say, 80 km, and interpolation done before the radar data are used.

5) The final saturation deficit analysis is not available unscaled without radar data unless radar are not being used.
6) Use of JP(,N), where N is the variable being analyzed:
   JP(1,N) controls gridprinting (IP22 must be $\geq$ 1)
   JP(2,N) controls writing to KFILOG:
      IFSTGS must be $\geq$ 2 for first guess to be tdlpacked
      NTDL(LP) must be $\geq$ 1 for analysis after pass LP to be tdlpacked
      JP(2,N) must be $\geq$ 1 for analysis for variable N to be tdlpacked
   JP(3,N) must be $\geq$ 1 for printing (not tdlpacking) of vector data destined for KFILOV and KFILQC
7) Writing to any file (KFILIO, KFILOG, KFILOV, KFILQC) will be done only when the file unit number is $\geq$ 0.

U400B

ANALYZES U AND V WIND COMPONENTS AND WIND SPEED

Harry R. Glahn
December 1, 2000

PURPOSE: To produce objective analyses of u and v wind components and of wind speed (s), on a variable-sized computational grid. All three analyses are done at one time, and if an observation is thrown out for u, v, or s, it is thrown out for all. Analyses are performed with the Bergthorssen-Cressman-Doos successive approximation techniques with first guess fields provided from an NCEP model, a LAMP forecast from a previous hour, a constant field, or the average of all data values to be used in the analysis. Data to be analyzed are error checked as specified in a control file.

U400B is called from U150 and is controlled by a control file 'U400BWND.CN'. 'U400BWND.CN' is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below (e.g., IP22).

CONTROL FILE INPUT: 'U400BWND.CN' (Unit = KFILDI)

Record Type 1 - Format (8I4,F8.0,17X,2I4) Overall Control

This record contains the number of analysis passes to perform and other information which is not pass specific.

NPASS - Number of analysis passes to make, maximum of 6. However, one or more of these can be omitted (e.g., the first) by setting a corresponding NTYPE(,) = 0 (see below).

IFSTGS - Controls gridprinting to unit number IP22 or TDLPACKing and writing of the first guess to unit number KFILOG of the subsetted area:
0 = first guess is neither gridprinted nor TDLPACKed,

1 = the first guess is to be gridprinted,
2 = first guess is to be TDLPACKed and written,
3 = first guess is to be both gridprinted and TDLPACKed and written.

IGUESS(J) - The type of first guess to use in priority order (J=1,4).
1 = A constant (see GUESS),
2 = An NCEP model forecast,
3 = A LAMP forecast from a previous hour,
4 = The average of all data values to be used in the analysis.
For example, if IGUESS(1) = 2, then the first priority for first guess is an NCEP forecast. If IGUESS(2) = 1, the second priority is the constant GUESS (see below). If one source is not available, the next priority is tried.

IBACKN - The maximum number of cycles to go back from the "current" cycle for an NCEP forecast to be used as a first guess at 6-h inter-

vals. For instance, if IBACKN = 2 and the current run of the NCEP forecast model is not available, U400B will look for 2 previous runs at 6-hr intervals. By setting IBACKN = 2, then either the 6-h or 12-h previous run could be used in that priority order. When a previous run is used, the projection(s) are adjusted accordingly. The "current run" of the NCEP forecast model is defined to be the cycle of the LAMP analysis, mod 6. That is, if the LAMP analysis is for 05Z, the current NCEP model run is defined as 00Z. No assumption is made as to when the model actually runs. If only the current run is desired (no lookback), set IBACKN = 0. If IBACKN < 0, an NCEP forecast is not used even when IGUESS( ) = 2.

IBACKL - The maximum number of cycles to go back for a LAMP forecast to be used as a first guess at 1-h intervals. For instance, if IBACKL = 2, and the 1-h forecast from the previous run of the LAMP forecast model is not available, U400B will look for 1 more previous run. When a previous run of LAMP is used, the projection(s) are adjusted accordingly (e.g., a 2-h forecast is used from a LAMP run 2 hours earlier). Note that IBACKN refers to the number of runs to look for back from the "current" run; IBACKL refers to the total number of runs to look back. If IBACKL $\leq$ 0, a LAMP forecast is not used even when IGUESS( ) = 3.

GUESS - The constant value to use as a first guess when IGUESS( ) = 1.

NSMTYP - Type of smoothing.
1 = Normal 5-point.
2 = Same as 1, except no change is made unless one or more of the points to contribute to the new value has been changed.
A value of zero defaults to 1.

I400ADG - 1 = Diagnostics about analysis will be written to KFILDO.
0 = Most diagnostics will not be written.

Record Type 2 - Format (6I8) Mesh Lengths

MSHPAS(J,L) - Nominal mesh length in km for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. Since MESH cannot be zero, if MESH(J,) is given that value, pass J is skipped.

Record Type 3 - Format (6F8.0) U, V, and S Error Criteria

ER1(J,L,M) - Error criteria in kt for each pass (J=1,6), for each first guess option (L=1,4) and month (M=1,12). Only NPASS values are read and used from each of 48 records. This is used for each of u, v, and s. The value M for the month being processed will be used.

Record Type 4 - Format (6F8.0) Direction Error Criteria

ERD(J,L) - Error criteria in degrees for each pass (J=1,6), for each first guess option (L=1,4) for wind direction. Only NPASS values are read and used from each of 48 records.

Record Type 5 - Format (6F8.0) Minimum Wind Speed for Direction Error

FLSL(J,L) - Minimum wind speed in kt for the direction error criterion to be used for each pass (J=1,6), for each first guess option (L=1,4). Only NPASS values are read and used from each of 48 records.

Record Type 6 - Format (6I8) Type of Correction

NTYPE(J,L) - Type of correction to make for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. NTYPE(,) applies to each of u, v, and s. The values of NTYPE (,) are explained in the section "Analysis Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985). If NTYPE(J,) = 0, pass J is skipped.

Record Type 7 - Format (6F8.0) Smoothing Parameter

B(J,L) - Smoothing parameter for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. B(,) applies to each of u, v, and s. The smoothing procedure is explained in the section "Smoothing Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985).

Record Type 8 - Format (6F8.0) Radius of Influence

R(J,L) - Radius of influence in grid units for each pass (J=1,6) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. Generally speaking, a data value affects a gridpoint only when it is within R(,) grid units of the gridpoint. Note that the actual distance depends on the gridlength being used on that pass. This is further explained in the section "Analysis Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985).

Record Type 9 - Format (6I8) Type of Interpolation

ITRPLQ(J,L) - The type of interpolation to be performed whenever interpolation is necessary for each pass (J=1,NPASS) and each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. If one pass of an analysis is on a grid of, say, 80 km, and the next pass is on 40-km, then interpolation is necessary to arrive at the "guess" for the next pass. ITRPLQ(,) is also used as necessary to arrive at thickened grids for the subsetted area (see Data Output section). The values are:
1 = bilinear,
2 = biquadratic.

Record Type 10 - Format (6F8.0) R Star Modification of Radius of Influence

RSTAR(J,L) - The fraction of R(,) to use when the datum is outside the grid for each pass (J=1,6) and each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records.

That is, R(,) may be 8 grid units, but since the interpolation into the grid to get the "analysis" at that time becomes extrapolation outside the grid, it may be desirable to cut the 8 to, say, 2. This can be done by using RSTAR(,) = .25.

Record Type 11 - Format (6F8.0) Multiplicative Constant for Gridprinting

SMULT(J) - The multiplicative factor to use before gridprinting the data for each pass (J=1,6). Only NPASS values are read and used.

Record Type 12 - Format (6I8) Additive Factor for Gridprinting

SADD(J) - The additive factor to use before gridprinting the data for each pass (J=1,6). Only NPASS values are read and used. Usually this is zero.

Record Type 13 - Format (6F8.0) Origin for Gridprinting

ORIGIN(J) - The origin when gridprinting the data for each pass (J=1,6). Only NPASS values are read and used. That is, the base contour will be at ORIGIN( ).

Record Type 14 - Format (6F8.0) Contour Interval for Gridprinting

CINT(J,L) - The contour interval when gridprinting the data for each pass (J=1,6). Only NPASS values are read and used. CINT( ) is in reference to the original units, not after the multiplication by SMULT( ).

Record Type 15 - Format (6I8) Gridprint Option Unsmoothed Grids

NPRT(J) - Gridprinting option for the unsmoothed analysis for each pass(J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 16 - Format (6I8) Gridprint Option for Smoothed Grids

JPRT(J) - Gridprinting option for the smoothed analyses for each pass (J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 17 - Format (6I8) TDLPACK Option for Unsmoothed Grids

NTDL(J) - TDLPACKing option for the unsmoothed analyses for each pass (J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = TDLPACK and write,
0 = otherwise.

Record Type 18 - Format (6I8) TDLPACK Option for Smoothed Grids

JTDL(J) - TDLPACKing option for the smoothed analyses for each pass (J=1,6) for the subsetted area. Only NPASS values are read and used.
1 = TDLPACK and write,
0 = otherwise.

DATA INPUT:

Most gridpoint and vector data needed by U400B have been stored by U150 into the MOS-2000 Internal Storage System. GFETCH is used to retrieve the data. Final grids for eventual writing to Unit Number KFILIO are packed and written there by GSTORE, called by subroutine PAWING.

DATA OUTPUT

There are five forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). These generally pertain to all analysis modules, and are explained in U150. Differences are that the discontinuous variables do not have quality controlled observational data written, because there is no effective error checking possible. See the Data Output section of U150.

EXAMPLE CONTROL FILE: 'U400BWND.CN'

An example exists as file 'U400BWND.CN' in directory 'lamp2000/dru150/standard_CN' on ice. The easiest way to set up a run is to take an existing control file and modify it.

OUTPUT:

Output is generally controlled by U150.

RESTRICTIONS

See U150.

COMMENTS

Comments in U150 apply to U400B.

Although the primary purpose of U150 and U400B is to produce packed gridpoint data, writing of such data is not done if KFILIO = 0 (see Record Type 8). A secondary, but important, purpose is to write vector records of observations in which all but the tossed obs have been set to 9999; however, this file will not be written in KFILQC = 0. That is, no output unit number and file name have been provided. A similar record is written to the same file in which all but the "questionable" obs have been set to 9999. Questionable is defined as not meeting a threshold which is a fraction of the primary error criteria ER1 read in U400BWND.CN. This feature can be used for checkout runs. Note that rendering the unit number = 0 is done by omitting the file in the input .CN file, not by leaving the file name there and setting the unit number = 0.

The modules of U150 (e.g., U400B) operate on grids of variable dimension. The grid length is specified at each stage of the process. For instance, the first guess and first pass of the wind analysis may be at 80 km.--the mesh length of the NCEP grid available for a first guess. Subsequent passes of the analysis may use other mesh lengths. U400B interpolates either linearly or quadratically to obtain a denser grid, or will use alternate grid points to thin a gird. It is important to note that the value of MESH is the mesh length of the active grids (u, v, and s) at that point and that XP( ) and YP( ) contain the IX and JY positions of the stations relative to that grid. That is, MESH and XP( ) and YP( ) follow the grid. Or you could think of the grid and XP( ) and YP( ) following MESH--same difference. Subroutines exist for these mesh length transformations which always have to be done in steps of a factor of two.

It is assumed that the grid that is to furnish a first guess (e.g., an NCEP AVN model or previous LAMP forecast) is large enough to encompass the primary "LAMP" grid. The LAMP grid is defined by a lower left latitude and longitude, a "basic" mesh length equal to a quarter Bedient, and an NX by NY extent; the pole position is also known or implied, being redundant to the other information. The first step in obtaining a first guess (assuming it is from a grid and not an average of observations or a constant) is to position the "LAMP" grid within the donor gird at the donor gird's mesh length (subroutine POSGRD is used). The next step is to make the guess the desired mesh length (subroutine SIZEGR is used).

It is necessary to analyze speed in addition to u and v, or the speed would be biased to low values.

The analyses of u, v, and s for any particular pass have the same mesh length MESH.

Most input parameters (e.g., SMULT( )) are applied to each of u, v, and s.

A final adjustment is made to the wind speed analysis by setting each gridpoint to the maximum of either (a) its analyzed value or (b) the speed computed from the analyzed component values. This adjustment is made mainly to eliminate possible negative values in the speed analysis. Significant changes can occur in data sparse regions. The adjustment is made either to unsmoothed or smoothed values depending on whether the final pass is smoothed or not. The adjustment is made after calculation of goodness of fit statistics, but before any gridprinting or TDLPACKing for the subsetted area, and, of course, before the final writing to unit number KFILIO. A word of caution--if NTYPE( ) = 0 for the last pass, this step will be skipped; however, this would be an unusual situation.

VARIABLES ACCOMMODATED BY U400B

U150 calls U400B once per cycle. Furnished to U400B is the list of 4-word MOS-2000 variable identifiers for which processing is to be done by some module of U150. U400B has resident a table that contains the variable IDs that U400B is engineered to handle.

U400B searches the table for each U150 input ID and processes u, v, and s if any of those variables are in the ID list. There is a '.CN' file for wind which suffices for all three variables

U400B handles three variables, u, v, and s. The first guess fields from NCEP or LAMP must be at the correct cycle and projection t. For the NCEP first guess, the speed is calculated from u and v. The units of the input observations are kt for speed and degrees, earth oriented, for direction. The units for the output analyses are kt for speed and kt for the grid oriented wind components.

```
Variable
  U wind component:           004010005   0  0  0
  V wind component            004110005   0  0  0
  Wind speed                  004210005   0  0  0
    Observations (dir)        704200000
   0  0  0
                 (speed)      704210000   0  0  0
    First guess from NCEP (u) 004020008   0  t  0
                          (v) 004120008   0  t  0
    First guess from LAMP (u) 004020005   0  t  0
                          (v) 004120005   0  t  0
                          (s) 004211005   0  t  0
```

ERROR CHECKING PROCEDURE

A check is made between the current speed analysis (quadratically interpolated to the station location) and the observation, according to the error criterion ER1(,) kt, and also when the observed speed is ≥ FLSL(,), between the current analysis direction (direction computed from interpolated u and v components) and observed direction according to the error criterion in ERD(,). Whenever a station is about to be discarded because of either of the checks (speed and direction), a check is made of its two closest neighbors. If either of its two neighbors also disagrees with the analysis by greater than the direction criterion or 60 percent of the speed error criterion (and in the same direction for speed), the station is kept if it meets the direction error criterion and 1.5 times the speed error criterion. Also, the assisting neighbor is kept if it meets 1.5 times the speed error criterion. (Note there is no change in the direction error criterion for any of these checks.) This process (a specialized "buddy check") requires nearest neighbors to be found (a very time consuming process) only infrequently. Units are kt for speed, and degrees for direction.

As explained in the U150 writeup, a record can be written for each variable (u, v, and s) for each pass in which all except the tossed obs have been set to 9999. This allows GEMPAK plotting routines to isolate by color the tossed obs vs the good obs. In addition, a record can be written for each variable and analysis pass in which all except "questionable" obs have been set to 9999. This allows GEMPAK plotting routines to isolate by color those observations which, while not tossed, are questionable. "Questionable" is defined by pass as a function of ER1 and ERD, the fraction being hardwired in subroutine BCDW. The "ER1Q" thresholds for speed are .5, .5, .5, .6, .6. and .6 of the ER1 criteria read in U400BWND.CN, respectively, for the 6 possible passes; the "ERDQ" thresholds for wind direction are .8, .8, .8, .9, .9. and .9 of the ERD criteria read in U400BWND, respectively, for the 6 possible passes. These records, with an ID that varies by pass number, are for tuning the analysis procedure.

In addition, a single record can be written for each analyzed variable in which the obs tossed on the last pass have been set to 9999. These records have a CCCFFF ID scheme of 71CFFF, where the "1" represents quality controlled obs. In addition to writing records for u, v, and s, a quality controlled record can be written for wind direction provided the ID for wind direction has been read in the 'U150.CN' control file, Record Type 17.

ANALYSIS PROCEDURE

The wind components (u and v) and the speed are all analyzed at the same time. In error checking, if an ob is tossed out, it is tossed for all three analyses. The values of NTYPE(,) (see Record Type 4) are applied to each of u, v, and s, and are explained below:

0 = Skip this pass. Not usually used.

1 = Correction applied to each gridpoint is the sum of the individual differences between observation and analysis grid values interpolated to stations divided by the number of stations within a radius R of the gridpoint being corrected.

$$C_1 = \frac{1}{n} \sum_{i=1}^{n} (O_{x,y} - A_{x,y})_i \qquad d_i \leq R$$

R = Radius of influence in grid units.
n = Number of observations within radius R of gridpoint being corrected.
$O_{x,y}$ = Observed station value at point x, y.
$A_{x,y}$ = Analyzed value of the variable interpolated from gridpoints to the point x, y.
d_i = Distance in grid units between the observation x, y and the gridpoint.

2 = Correction applied to each gridpoint is the same as type 1 correction except the distance of each station from a gridpoint is considered in the correction for that station. When a type 2 correction is being used, lots of distant stations, even with a few close stations, will result in small corrections being applied to a gridpoint.

$$C_2 = \frac{1}{n} \sum_{i=1}^{n} \frac{R^2 - d_i^2}{R^2 + d_i^2} (O_{x,y} - A_{x,y})_i \qquad d_i < R$$

3 = Correction applied to each gridpoint is the same as type 2 correction except the sum of the distances of each station from a gridpoint is also considered in the correction.

$$C_3 = \frac{\frac{1}{n}\sum_{i=1}^{n} \frac{R^2 - d_i^2}{R^2 + d_i^2} (O_{x,y} - A_{x,y})_i}{\frac{1}{n}\sum_{i=1}^{n} \frac{R^2 - d_i^2}{R^2 + d_i^2}} \qquad d_i < R$$

By having the sum of weights in the denominator, a larger correction is applied to a gridpoint than with a type 2 correction. This may not produce good results when n is very small and R is large. Therefore, when NTYPE(,) = 3 is being used, for those gridpoints where n = 1 and R GE 3, a type 2 correction is made.

SMOOTHING PROCEDURE

The smoothing parameter B(,) (see Record Type 5), applied to u, v, and s, is explained below:

$$S_{i,j} = \frac{A_{i,j} + b\bar{A}_{i,j}}{1 + b}$$

where:
$S_{i,j}$ = Smoothed gridpoint values at point i, j.
$A_{i,j}$ = Value of gridpoint at point i, j before smoothing.
b = Smoothing parameter.
$\bar{A}_{i,j} = (1/4)\ (A_{i+1,j}+A_{i-1,j}+A_{i,j+1}+A_{i,j-1})$

Low values produce the least smoothing and higher values produce heavier smoothing. A value of zero indicates no smoothing on this pass.

NONSYSTEM ROUTINES USED

Use the load line in file 'lamp2000/dru150', dataset 'u150.com' on ice.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: lamp2000/u400blib. The driver is in lamp2000/lamp/dru150.

U400D

ANALYZES DISCONTINUOUS VARIABLES

Harry R. Glahn
December 1, 2000

PURPOSE: To produce objective analyses of discontinuous variables, such as ceiling height, visibility, and weather type, on a variable-sized computational grid. Analyses are performed with the closest neighbor approach; that is, the gridpoint is given the value of the closest station. A first guess field can be provided from a constant or from a previous LAMP forecast.

U400D is called from U150 and is controlled by a set of control files 'U400DXXX.CN', where XXX pertains to a specific variable (e.g., XXX = VIS for visibility). 'U400DXXX.CN' is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below (e.g., IP22).

CONTROL FILE INPUT: 'U400DXXX.CN' (Unit = KFILDI)

Record Type 1 - Format (4I4,12X,I4,F8.0,1X,A16) Overall Control

This record contains the number of analysis passes to perform and other information which is not pass specific.

NPASS - Number of analysis passes to make, maximum of 2. However, one of these can be omitted (e.g., the first) by setting a corresponding NCLOS(,) = 0 (see below). Pass 1 is for using the previous hour's data; Pass 2 is for on-time data.

IFSTGS - Controls gridprinting to unit number IP22 or TDLPACKing and writing of the first guess to unit number KFILOG of the subsetted area:
0 = first guess is neither gridprinted nor TDLPACKed,

1 = the first guess is to be gridprinted,
2 = first guess is to be TDLPACKed and written,
3 = first guess is to be both gridprinted and TDLPACKed and written.

IGUESS(J) - The type of first guess to use in priority order (J=1,2).
1 = A constant (see GUESS),
2 = Not implemented,
3 = A LAMP forecast from a previous hour,
4 = Not implemented.
For example, if IGUESS(1) = 3, then the first priority for first guess is a previous LAMP forecast. If IGUESS(2) = 1, the second priority is the constant GUESS (see below). IGUESS(2) could be zero, in which an analysis would not be performed if a previous LAMP forecast were not available.

IBACKL - The maximum number of cycles to go back for a LAMP forecast to be used as a first guess at 1-h intervals. For instance, if

IBACKL = 2, and the 1-h forecast from the previous run of the LAMP forecast model is not available, U400A will look for another previous run. When a previous run of LAMP is used, the projection(s) are adjusted accordingly (e.g., a 2-h forecast is used from a LAMP run 2 hours earlier). If IBACKL $\leq$ 0, a LAMP forecast is not used even when IGUESS( ) = 3.

GUESS - The constant value to use as a first guess when IGUESS( ) = 1.

TITLE - A 16-character title for this variable, the variable being defined by the 4-word ID read by U150. (CHARACTER*16)

Record Type 2 - Format (6I8) Mesh Lengths

MSHPAS(J,L) - Nominal mesh length in km for each pass (J=1,2) for each possible first guess option (L=1,4). Only NPASS values are read and used from each of 4 records.

Record Type 3 - Format (6I8) Type of Correction

NCLOS(J,L) - Controls whether the pass will be made or not, and for Pass 2 how the correction will be made pass (J=1,2) for each possible first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. The values of NCLOS(,) are as follows:

For Pass 1
If NCLOS(1,) = 0, pass 1 is skipped,
If NCLOS(1,) NE 0, pass 1 is made.

For Pass 2
If NCLOS(2,) = 0, pass 2 is skipped. In this case, the first guess would be used or the 1-h old obs. IF NCLOS(2,) NE 0, then the value of NCLOS(2,) is the number of gridpoints closer the past observation needs to be to override the current observation. X means X grid lengths at mesh length MSHPAS(2,L).

Record Type 4 - Format (6F8.0) Smoothing Parameter

B(J,L) - Smoothing parameter for each pass (J=1,2) for each first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. The smoothing procedure is explained in the section "Smoothing Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985). While B( ) > 0 is possible, it is expected this will be zero because the smoothing routine does not deal with missing values.

Record Type 5 - Format (6F8.0) Radius of Influence

R(J,L) - Radius of influence in grid units for each pass (J=1,2) for each possible first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. Generally speaking, a data value affects a gridpoint only when it is within R(,) grid units of the gridpoint. Note that the actual distance depends on the gridlength being used on that pass. This is further

explained in the section "Analysis Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985).

Record Type 6 - Format (6I8) Type of Interpolation

ITRPLQ(J,L) - The type of interpolation to be performed whenever interpolation is necessary for each pass (J=1,2) and each possible first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. If one pass of an analysis is on a grid of, say, 80 km, and the next pass is on 40-km, then interpolation is necessary to arrive at the "guess" for the next pass. ITRPLQ(,) is also used as necessary to arrive at thickened grids for the subsetted (see Data Output section) area. The values are:
1 = bilinear,
2 = biquadratic.

Record Type 7 - Format (6F8.0) R Star Modification of Radius of Influence

RSTAR(J,L) - The fraction of R(,) to use when the datum is outside the grid for each pass (J=1,2) and first guess option (L=1,4). Only NPASS values are read and used from each of 4 records. That is, R(,) may be 8 grid units, but since the interpolation into the grid to get the "analysis" at that time becomes extrapolation outside the grid, it may be desirable to cut the 8 to, say, 2. This can be done by using RSTAR(,) = .25.

Record Type 8 - Format (6F8.0) Multiplicative Constant for Gridprinting

SMULT(J) - The multiplicative factor to use before gridprinting the data for each pass (J=1,2). Only NPASS values are read and used. That is, if visibility is in miles and fractions, it may be desirable to use SMULT( ) = 100 so that the quarter mile values can be seen.

Record Type 9 - Format (6I8) Additive Factor for Gridprinting

SADD(J) - The additive factor to use before gridprinting the data for each pass (J=1,2). Only NPASS values are read and used. Usually this is zero.

Record Type 10 - Format (6F8.0) Origin for Gridprinting

ORIGIN(J) - The origin when gridprinting the data for each pass (J=1,2). Only NPASS values are read and used. That is, the base contour will be at ORIGIN( ).

Record Type 11 - Format (6F8.0) Contour Interval for Gridprinting

CINT(J,L) - The contour interval when gridprinting the data for each pass (J=1,2). Only NPASS values are read and used. CINT( ) is in reference to the original units, not after the multiplication by SMULT( ).

Record Type 12 - Format (6I8) Gridprint Option Unsmoothed Grids

NPRT(J) - Gridprinting option for the unsmoothed analysis for each pass(J=1,2) for the subsetted area. Only NPASS values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 13 - Format (6I8) Gridprint Option for Smoothed Grids

JPRT(J) - Gridprinting option for the smoothed analyses for each pass (J=1,2) for the subsetted area. Only NPASS values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 14 - Format (6I8) TDLPACK Option for Unsmoothed Grids

NTDL(J) - TDLPACKing option for the unsmoothed analyses for each pass (J=1,2) for the subsetted area. Only NPASS values are read and used.
1 = TDLPACK and write,
0 = otherwise.

Record Type 15 - Format (6I8) TDLPACK Option for Smoothed Grids

JTDL(J) - TDLPACKing option for the smoothed analyses for each pass (J=1,2) for the subsetted area. Only NPASS values are read and used.
1 = TDLPACK and write,
0 = otherwise.

DATA INPUT:

Most gridpoint and vector data needed by U400D have been stored by U150 into the MOS-2000 Internal Storage System. GFETCH is used to retrieve the data. Final grids for eventual writing to Unit Number KFILIO are packed and written there by GSTORE, called by subroutine PAWING. However, if a variable has to be computed, the External Random Access file system might have to be accessed; provision is made for this access.

As a special feature, because ceiling height is not observed and is not on the hourly archive, it is computed with subroutine OBSCIGHT, which is in the directory u201lib. This happens automatically when the ceiling height ID in included in the 'U150.CN' control file, Record Type 17.

DATA OUTPUT

There are three forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). These are explained in the U150 writeup in the Data Output section, Subsections A, B, and D. There is no vector output corresponding to Subsections C and E. Quality control on discontinuous observations with the analysis procedure is not feasible.

EXAMPLE CONTROL FILE: 'U400DVIS.CN'

An example exists as file 'U400DVIS.CN' in directory 'lamp2000/dru150' on ice. The easiest way to set up a run is to take an existing control file and modify it.

OUTPUT:

Output is generally controlled by U150, except for the gridded output for A and B discussed above.

RESTRICTIONS

See U150.

COMMENTS

Comments in U150 apply to U400D.

Although the primary purpose of U150 and U400D is to produce packed gridpoint data, writing of such data is not done if KFILIO = 0 (see Record Type 8 for U150). That is, no output unit number and file name have been provided. This feature can be used for checkout runs.

The modules of U150 (e.g., U400D) operate on grids of variable dimension. The grid length is specified at each stage of the process. For instance, the first guess and first pass of the visibility analysis may be at 80 km. Subsequent passes of the analysis may use other mesh lengths. U400D interpolates either linearly or quadratically to obtain a denser grid, or will use alternate grid points to thin a gird. It is important to note that the value of MESH is the mesh length of the active grid at that point and that XP( ) and YP( ) contain the IX and JY positions of the stations relative to that grid. That is, MESH and XP( ) and YP( ) follow the grid. Or you could think of the grid and XP( ) and YP( ) following MESH--same difference. Subroutines exist for these mesh length transformations which always have to be done in steps of a factor of two.

It is assumed that the grid that is to furnish a first guess (i.e., a previous LAMP forecast) is large enough to encompass the "LAMP" grid. The primary LAMP grid is defined by a lower left latitude and longitude, a "basic" mesh length equal to a quarter Bedient, and an NX by NY extent; the pole position is also known or implied, being redundant to the other information. The first step in obtaining a first guess (assuming it is from a grid and not an average of observations or a constant) is to position the "LAMP" grid within the donor gird at the donor gird's mesh length (subroutine POSGRD is used). The next step is to make the guess the desired mesh length (subroutine SIZEGR is used).

VARIABLES ACCOMMODATED BY U400D

U150 calls U400D once per cycle. Furnished to U400D is the list of 4-word MOS-2000 variable identifiers for which processing is to be done by some module of U150. U400D has resident a table that contains the variable IDs that U400D is engineered to handle. There is a corresponding table which gives the names of the '.CN' control file for each variable.

U400D searches the table for each U150 input ID and processes those variables that reside in its table ITABLE(,NVAL). For each variable processed (analyzed), its specific '.CN' file is read (e.g., 'U400DVIS.CN' for visibility). This control file was explained earlier under "CONTROL FILE INPUT."

If it is desired that U400D handle an additional variable, it should be sufficient to:

(1) Increase NVAL in the PARAMETER statement,
(2) Include the ID in ITABLE(,NVAL),
(3) Include in DOTCN(NVAL) the control file name, and
(4) Provide the control file appropriate to the variable.

Currently, U400D is ready to handle 38 variables. The first guess fields from LAMP must be at the correct cycle and projection t.

Variable				
Ceiling Height:	008000005	0	0	0
Computed by OBSCIGHT				
First guess from LAMP	008000005	0	t	0
Control file	U400DCIG.CN			
Visibility	008100005	0	0	0
Observations	708100000	0	0	0
First guess from LAMP	008100005	0	t	0
Control file	U400DVIS.CN			
Total cloud cover	008311005	0	0	0
Computed by SFCTCLD				
First guess from LAMP	008311005	0	t	0
Control file	U400DSKY.CN			
Amount lowest cloud lyr	008320005	0	0	0
Observations	708320000	0	0	0
First guess from LAMP	008320005	0	t	0
Control file	U400DCA1.CN			
Height lowest cloud lyr	008321005	0	0	0
Observations	708321000	0	0	0
First guess from LAMP	008321005	0	t	0
Control file	U400DCH1.CN			
Amount 2nd cloud lyr	008322005	0	0	0
Observations	708322000	0	0	0
First guess from LAMP	008322005	0	t	0
Control file	U400DCA2.CN			
Height 2nd cloud lyr	008323005	0	0	0
Observations	708323000	0	0	0
First guess from LAMP	008323005	0	t	0
Control file	U400DCH2.CN			

Amount 3rd cloud lyr	008324005	0	0	0
Observations	708324000	0	0	0
First guess from LAMP	008324005	0	t	0
Control file	U400DCA3.CN			
Height 3rd cloud lyr	008325005	0	0	0
Observations	708325000	0	0	0
First guess from LAMP	008325005	0	t	0
Control file	U400DCH3.CN			
Amount 4th cloud lyr	008326005	0	0	0
Observations	708326000	0	0	0
First guess from LAMP	008326005	0	t	0
Control file	U400DCA4.CN			
Height 4th cloud lyr	008327005	0	0	0
Observations	708327000	0	0	0
First guess from LAMP	008327005	0	t	0
Control file	U400DCH4.CN			
Amount 5th cloud lyr	008328005	0	0	0
Observations	708328000	0	0	0
First guess from LAMP	008328005	0	t	0
Control file	U400DCA5.CN			
Height 5th cloud lyr	008329005	0	0	0
Observations	708329000	0	0	0
First guess from LAMP	008329005	0	t	0
Control file	U400DCH5.CN			
Amount 6th cloud lyr	008330005	0	0	0
Observations	708330000	0	0	0
First guess from LAMP	008330005	0	t	0
Control file	U400DCA6.CN			
Height 6th cloud lyr	008331005	0	0	0
Observations	708331000	0	0	0
First guess from LAMP	008331005	0	t	0
Control file	U400DCH6.CN			
First weather group	008500005	0	0	0
Observations	708500000	0	0	0
First guess from LAMP	008500005	0	t	0
Control file	U400DWX1.CN			
Second weather group	008510005	0	0	0
Observations	708510000	0	0	0
First guess from LAMP	008510005	0	t	0
Control file	U400DWX2.CN			
Third weather group	008520005	0	0	0
Observations	708520000	0	0	0
First guess from LAMP	008520005	0	t	0
Control file	U400DWX3.CN			

Cloud amount 0-1 hdr ft	008350005		0	0	0
Computed by CLDCAT					
First guess from LAMP	008350005		0	t	0
Control file	U400DA01.CN				
Cloud height 0-1 hdr ft	008355005	0001	0	0	
Conputed by CLDCAT					
First guess from LAMP	008355005	0001	t	0	
Control file	U400DH01.CN				
Cloud amount 2-4 hdr ft	008350005	020004	0	0	
Computed by CLDCAT					
First guess from LAMP	008350005	020004	t	0	
Control file	U400DA04.CN				
Cloud height 2-4 hdr ft	008355005	020004	0	0	
Computed by CLDCAT					
First guess from LAMP	008355005	020004	t	0	
Control file	U400DH04.CN				
Cloud amount 5-9 hdr ft	008350005	050009	0	0	
Computed by CLDCAT					
First guess from LAMP	008350005	050009	t	0	
Control file	U400DA09.CN				
Cloud height 5-9 hdr ft	008355005	050009	0	0	
Computed by CLDCAT					
First guess from LAMP	008355005	050009	t	0	
Control file	U400DH09.CN				
Cloud amount 10-15 hdr ft	008350005	100015	0	0	
Computed by CLDCAT					
First guess from LAMP	008350005	100015	t	0	
Control file	U400DA15.CN				
Cloud height 10-15 hdr ft	008355005	100015	0	0	
Computed by CLDCAT					
First guess from LAMP	008355005	100015	t	0	
Control file	U400DH15.CN				
Cloud amount 16-30 hdr ft	008350005	160030	0	0	
Computed by CLDCAT					
First guess from LAMP	008350005	160030	t	0	
Control file	U400DA30.CN				
Cloud height 16-30 hdr ft	008355005	160030	0	0	
Computed by CLDCAT					
First guess from LAMP	008355005	160030	t	0	
Control file	U400DH30.CN				
Cloud amount 31-45 hdr ft	008350005	310045	0	0	
Computed by CLDCAT					
First guess from LAMP	008350005	310045	t	0	
Control file	U400DA45.CN				

```
Cloud height 31-45 hdr ft  008355005  310045  0  0
  Computed by CLDCAT
  First guess from LAMP    008355005  310045  t  0
  Control file             U400DH45.CN

Cloud amount 46-65 hdr ft  008350005  460065  0  0
  Computed by CLDCAT
  First guess from LAMP    008350005  460065  t  0
  Control file             U400DA65.CN

Cloud height 46-65 hdr ft  008355005  460065  0  0
  Computed by CLDCAT
  First guess from LAMP    008355005  460065  t  0
  Control file             U400DH65.CN

Cloud amount 66-120 hdr ft 008350005  660120  0  0
  Computed by CLDCAT
  First guess from LAMP    008350005  660120  t  0
  Control file             U400DA1X.CN

Cloud height 66-120 hdr ft 008355005  660120  0  0
  Computed by CLDCAT
  First guess from LAMP    008355005  660120  t  0
  Control file             U400DH1X.CN

Cloud amount > 120 hdr ft  008350005  1219999  0  0
  Computed by CLDCAT
  First guess from LAMP    008350005  1219999  t  0
  Control file             U400DA99.CN

Cloud height > 120 hdr ft  008355005  1219999  0  0
  Computed by CLDCAT
  First guess from LAMP    008355005  1219999  t  0
  Control file             U400DH99.CN

Obscuration height hdr ft  008350005  0000000  0  0
  Computed by CLDCAT
  First guess from LAMP    008350005  0000000  t  0
  Control file             U400DA00.CN

Obscuration amount hdr ft  008355005  0000000  0  0
  Computed by CLDCAT
  First guess from LAMP    008355005  0000000  t  0
  Control file             U400DH00.CN
```

Subroutine CLDCAT can be consulted for details for the cloud variables in specific layers, subroutine OBSCIGHT for ceiling height, and SFCTCLD for total cloud cover.

Subroutine ECLWEA sets a "missing" 9999 value for a cloud layer to 8888 when there was a valid report for the lowest cloud layer and for a missing weather report when there was a valid report for weather group one. This is a second kind of missing; it distinguishes "real" missings from just no report at that level.

ERROR CHECKING PROCEDURE

Error checking is limited to range checks:

Ceiling--0 to 400, where values are in hundreds of ft. Earlier processing has set values > 400 to 400. In addition, values of 888 representing unlimited ceilings are set to 400.

Visibility--0 to 40, where values are in miles. Earlier processing has set values > 40 to 40.

Total cloud cover--0 to 10, where values are coded

ANALYSIS PROCEDURE

The values of NCLOS(,) (see Record Type 4) are explained below:

0 = Skip this pass.

Pass 1:

The closest station to each gridpoint within the radius R(1,) is found and the gridpoint given that value. If no station is within R(1,) of the gridpoint, the gridpoint retains its first guess value.

Pass 2:

The closest station to each gridpoint within the radius R(2,) is found. If that station is not NCLOS(2,) grid units farther away than the value put at the gridpoint on Pass 1, it is used. If not, the original value (1-h old value) is used.

For instance, if NCLOS(2,) = 1, and the value placed at a gridpoint on pass 1 was 3 gridpoints away, the closest value within a radius of 4 gridlengths would be placed at that gridpoint on pass 2. If none exists, the original value is retained.

A viable option might be to use a previous hour's LAMP as a first guess and skip pass 1. If the previous LAMP forecast was not available, the process could fall back on using 2 hours of data. Relevant values would be: IGUESS(1)=3; IGUESS(2)=1; NCLOS(1,3)=0; NCLOS(1,1)=1; NCLOS(2,3)>0, NCLOS(2,1)>0.

SMOOTHING PROCEDURE

The smoothing parameter B(,) (see Record Type 5) is explained below:

$$S_{i,j} = \frac{A_{i,j} + b\bar{A}_{i,j}}{1 + b}$$

where:
$S_{i,j}$ = Smoothed gridpoint values at point i, j.
$A_{i,j}$ = Value of gridpoint at point i, j before smoothing.

<u>b</u> = Smoothing parameter.

$A_{i,j} = (1/4)\ (A_{i+1,j}+A_{i-1,j}+A_{i,j+1}+A_{i,j-1})$

Low values produce the least smoothing and higher values produce heavier smoothing. A value of zero indicates no smoothing on this pass.

<u>NONSYSTEM ROUTINES USED</u>

Use the load line in file 'lamp2000/dru150', dataset 'u150.com' on ice.

<u>LANGUAGE</u>: FORTRAN77 with some HP extensions.

<u>LOCATION</u>: lamp2000/u400dlib. The driver is in lamp2000/dru150.

U450

FORECASTS SEA LEVEL PRESSURE

Harry R. Glahn
June 1, 2001

PURPOSE: To produce forecasts of sea level pressure (SLP) from 1 to a maximum of 25 hours on a variable-sized computational grid. The forecast model used is basically that of Reed, although various modifications have been made. Initialization is with the sea level pressure analysis produced by U400A, and NCEP model 500-mb (and possibly 1000-mb) fields are used as the advecting agent and development term in this Lagrangian model.

U450 is called from U150 and is controlled by a control file 'U450.CN', which is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below [e.g., IP(22)].

CONTROL FILE INPUT: 'U450.CN' (Unit = KFILDI)

Record Type 1 - Format (I4,5F4.0,2I4,1X,A16,4I4,F4.0,I4) Overall Control

This record contains the maximum number of hours to make forecasts for and other information which is not projection specific.

NPROJH - Maximum projection in hours to make forecasts for, starting with 1 hour, maximum of 25. Certain projections can be omitted (see MAKEF( ) below). This is only for efficiency of computation and could be set to 24 permanently.

C405F - The factor to multiply 0.405 by to compute the terrain term TERRM( ). Now, normally 0.5.

C55HCF - The factor to multiply 0.55 by to compute the 500-mb height change. Normally 1 for the basic model.

C55AF - The factor to use to compute the 500-mb advecting winds. Normally 0.55 for the basic model.

C00AF - The factor to use to compute the 1000-mb advecting winds. Normally 0.0 for the basic model.

C163GF - The factor to multiply 163. by to compute the terrain term G( ). Normally 1 for the basic model.

IBACKN - The maximum number of cycles to go back for an NCEP forecast at 6-h intervals. For instance, if IBACKN = 2 and the current run of the NCEP forecast model is not available, U450 will look for 2 previous runs at intervals of 6 hours earlier. By setting IBACKN = 2, then either the 6-h or 12-h previous run could be used in that priority order. When a previous run is used, the projection(s) are adjusted accordingly. The "current run" of the NCEP forecast model is defined to be the cycle of the LAMP

analysis, mod 6. That is, if the LAMP analysis is for 05Z, the current NCEP model run is defined as 00Z. No assumption is made as to when the model actually runs.

IBACKL - The maximum number of cycles to go back for a LAMP forecast at 1-h intervals to use as an analysis. For instance, if IBACKL = 1, and the current analysis is not available, U450 will look for a forecast from 1 previous run. IBACKL will usually be 0; U450 is not fully developed for IBACKL > 0.

TITLE - A 16-character title for this variable, the variable being defined by the 4-word ID read by U150. (CHARACTER*16)

MESH - The nominal mesh length to use in the run. Note that this is not variable by projection. Normally 80 to match the anticipated mesh length of the sea level pressure analysis.

ITRPLQ - The method of interpolation when adjusting an input grid to the mesh size used in the run. Note that this is not variable by projection. The values are:
1 = bilinear,
2 = biquadratic.

N50SM - The number of times the 500-mb heights are smoothed with a square 25-point smoother on a 1-Bedient grid for purposes of computing the advecting winds. The standard version uses 1. (See C55AF.)

N00SM - The number of times the 1000-mb heights are smoothed with a square 25-point smoother on a 1-Bedient grid for purposes of computing the advecting winds. The standard version uses 0. (See C00AF.)

CFILT - The fraction of the advecting wind reduction to apply. A "1" means apply the full reduction; 0 means do not reduce the winds. The standard version uses 1.

I450DG - Set to 1 when the diagnostic information listed in Sections "TDLPACK Grids for Checkout or Quality Control" and "GRIDPRINT for Checkout or Quality Control" are to be written to Unit Nos. KFILOG (TDLPACK) and/or IP(22) (Gridprint); zero otherwise.

Record Type 2 - Format (25F4.0) Smoothing Parameter

B(J) - Smoothing parameter for each projection (J=1,25) used to smooth the SLP forecasts. Only NPROJH values are read. The smoothing procedure is basically that explained in the section "Smoothing Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985); however, it has been made selective in that strong gradients are not smoothed. The routine used is SMOTHCM. Use B( ) = 4 for all passes. See Comments section.

Record Type 3 - Format (25F4.0) Radius of Influence

R(J) - Radius of influence in grid units for each projection (J=1,25). Only NPROJH values are read. (R( ) is not currently implemented. It could be used to determine how far outside the grid to extrapolate when the trajectory end points do not fall inside the grid.)

Record Type 4 - Format (25F4.0) Multiplicative Constant for Gridprinting

SMULT(J) - The multiplicative factor to use before gridprinting the data for each projection (J=1,25). Only NPROJH values are read and used.

Record Type 5 - Format (25F4.0) Additive Factor for Gridprinting

SADD(J) - The additive factor to use before gridprinting the data for each projection (J=1,25). Only NPROJH values are read and used. Usually this is zero.

Record Type 6 - Format (25F4.0) Origin for Gridprinting

ORIGIN(J) - The origin when gridprinting the data for each pass (J=1,25). Only NPROJH values are read and used. That is, the base contour will be at ORIGIN( ).

Record Type 7 - Format (25F4.0) Contour Interval for Gridprinting

CINT(J) - The contour interval when gridprinting the data for each pass (J=1,25). Only NPROJH values are read and used. CINT( ) is in reference to the original units, not after the multiplication by SMULT( ).

Record Type 8 - Format (6I8) Projections to Make Forecasts For

MAKEF(J) - Projections to make forecasts for (J=1,25). Only NPROJH values are read and used. A value of 1 for projection J means make a forecast will be made for projection J; a zero means to not make a forecast. Note that NPROJH is the maximum projection that a forecast can be made for, and MAKEF( ) controls the actual projections.

Record Type 9 - Format (6I8) Gridprint Option for SLP Pressure Forecasts

NPRT(J) - Gridprinting option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 10 - Format (6I8) TDLPACK Option for SLP Forecasts

NTDL(J) - TDLPACKing option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = TDLPACK and write,

0 = otherwise.

DATA INPUT:

Most gridpoint data needed by U450 have been stored by U150 into the MOS-2000 Internal Storage System. GFETCH is used to retrieve the data. Final grids for eventual writing to Unit Number KFILIO are packed and written there by GSTORE, called by subroutine PAWING. However, the terrain heights are retrieved from the External Random Access; provision is made for this access.

DATA OUTPUT

There are three forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). These are Types A, B, and D explained in U150. See the Data Output section of U150.

EXAMPLE CONTROL FILE: 'U450.CN'

An example exists as file 'U450.CN' in directory 'lamp2000/dru150' on ice. The easiest way to set up a run is to take an existing control file and modify it.

OUTPUT:

Output is generally explained in the U150 writeup.

RESTRICTIONS

See U150.

COMMENTS

Comments in U150 apply to U450.

Although the primary purpose of U150 and U450 is to produce packed gridpoint forecasts of sea level pressure, writing of such data is not done if KFILIO = 0 (see Record Type 8 of U150). That is, no output unit number and file name have been provided. Note that rendering the unit number = 0 is done by omitting the file in the input .CN file, not by leaving the file name there and setting the unit number = 0.

The U450 module of U150 accepts grids of variable dimension. The grids are put on the grid with a mesh length MESH; this is not changed throughout the run.

It is assumed that the grid that is to furnish input (e.g., an NCEP AVN model) is large enough to encompass the primary LAMP grid (the exception is the radar data). The LAMP grid is defined by a lower left latitude and longitude, a "basic" mesh length equal to a quarter Bedient, and an NX by NY extent; the pole position is also known or implied, being redundant to the other information. The first step is to position the "LAMP" grid within the input grid at the input grid's mesh length (subroutine POSGRD is used). The next step is to make the input grid the desired mesh length (subroutine SIZEGR is used).

Testing has shown that the forecasts were generally slightly more noisy than the analyses. In order to improve scores, the 1-h forecast is not smoothed, the 2-h forecast is smoothed once, the 3-h forecast is smoothed twice, and all other projections are smoothed 3 times with B( ). Normally, B( ) = 4.

It has been difficult to program U450 with exactly the same formulations as the currently running model. The external documentation is not complete and detailed, and is not always consistent with the code and its internal comments. The program that produced the terrain field used in the model could not be found, and there was probably some processing done when the file was written. However, it is believed the formulation is very close.

The grids input to U450 have no missing (9999) values. These are model fields, LAMP analyses, and terrain. Also, the trajectory end positions in subroutines FORCST and TRJCT are limited to one gridlength outside the grid. U450 through subroutine INTRQ interpolates or extrapolates into the grids so that missing values are not produced.

VARIABLES ACCOMMODATED BY U450

U150 calls U450 once per cycle. Furnished to U450 is the list of 4-word MOS-2000 variable identifiers for which processing is to be done by some module of U150. U450 has resident a table that contains the single variable ID that U450 is engineered to handle, 001201005. There is a corresponding (one element) table which gives the name of the '.CN' control file, 'U450.CN'.

U450 searches the table for each U150 input ID and processes those variables that reside in its table ITABLE( ). Only one is accommodated, and the associated fields needed are:

Variable					
Sea Level Pressure:	001201005	0	0	0	
500-mb heights from NCEP	001000008	500	t	0	
1000-mb heights from NCEP	001000008	1000	t	0	(not used when C00AF=0)
Elevations (terrain)	409020000	0	0	0	

The only variables normally written to the archive file, Unit No. KFILIO, are the SLP forecasts for each hour t indicated in MAKEF( ). They are written with the smoothing parameter S = 0 in the ID even though the forecasts may have been smoothed. This is consistent with the final analyses (e.g., see U400A) being written with S = 0 even though they may have been smoothed.

TDLPACK GRIDS FOR CHECKOUT OR QUALITY CONTROL

TDLPACK grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed below will be written to the dataset whose name has been provided to OUTDIS and Unit No. KFILOG from the control file 'U150.CN', Record Type 9, provided (1) KFILOG is not zero and (2) I450DG is not zero:

409020005	0	0	0	Terrain height in m
409020005	0	0	60	Terrain height in m smoothed over 25 1-Bedient points
409021005	0	0	0	Terrain height field in m <u>used in the model</u> after multiplicative factors and height adjustment has been applied
001000005	0	0	0	Initial 1000-mb height in m derived from SLP analysis
409120005	0	0	0	Constant "G" term in m <u>used in the model</u>
409220005	0	0	0	Full conservative quantity in m <u>used in the model</u>
001000008	500	t	0	NCEP 500-mb heights in m
001000005	500	t	60	500-mb heights in m smoothed over 25 1-Bedient points N50SM (see Record Type 1) times
001000008	1000	t	0	NCEP 1000-mb heights in m
001000005	1000	t	60	1000-mb heights in m smoothed over 25 1-Bedient points N00SM (see Record Type 1) times
004000005	0	t	60	U-advective winds in grid units per half hour
004100005	0	t	60	V-advective winds in grid units per half hour
001201005	0	t	10	Sea level forecasts for hour t. Note that smoothing is indicated because these really are smoothed and to distinguish the record ID from that on Unit No. KFILIO (the archive file), which may have the same characteristics except a different mesh length.

Note that all gridpoint data written by LAMP have the DD = LAMPNO (= 5) even though the data came from NCEP. This is to alleviate difficulties in following programs. Projection is represented by "t" above, and will occur for projections 0 through NPROJH (see Record Type 1).

GRIDPRINT FOR CHECKOUT OR QUALITY CONTROL

The same grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed in the section above will be gridprinted and written to Unit No. IP(22) from the control file 'U150.CN', Record Type 1, provided (1) IP(22) is not zero and (2) I450DG is not zero. All are contoured at intervals of 100 meters, except the advective U and V winds which are first multiplied by 100 then contoured at units of 10.

SMOOTHING PROCEDURE

The smoothing parameter B(,) (see Record Type 2) is explained below:

$$S_{i,j} = \frac{A_{i,j} + b\bar{A}_{i,j}}{1 + b}$$

where:

$S_{i,j}$ = Smoothed gridpoint values at point i, j.

$A_{i,j}$ = Value of gridpoint at point i, j before smoothing.

<u>b</u> = Smoothing parameter.

$A_{i,j} = (1/4)\ (A_{i+1,j}+A_{i-1,j}+A_{i,j+1}+A_{i,j-1})$

Low values produce the least smoothing and higher values produce heavier smoothing. A value of zero indicates no smoothing.

As stated earlier, this has been modified to not smooth in areas of pressure < 1013 mb where there is a large gradient. This is to fit the low pressure areas and troughs better, and especially in analyses of hurricanes.

NONSYSTEM ROUTINES USED

Use the load line in file 'lamp2000/dru150', dataset 'u150.com' on ice.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: lamp2000/u450lib. The driver is in lamp2000/dru150.

BIBLIOGRAPHY:

Glahn, H. R., 1980: Plans for the development of a Local AFOS MOS Program (LAMP). Preprints Eighth Conference on Weather Forecasting and Analysis, Denver, Amer. Meteor. Soc., 302-305.

_____, and D. A. Lowry, 1972: An operational Subsynoptic Advection Model (SAM). J. Appl. Meteor., 11, 578-585.

_____, _____, and G. W. Hollenbaugh, 1969: An operational subsynoptic advection model. ESSA Technical Memorandum WBTM TDL 23, Environmental Science Services Administration, U.S. Department of Commerce, 33 pp.

_____, and D. A. Unger, 1982: Some results from the Local AFOS MOS Program (LAMP). Preprints 9th Conference on Weather Forecasting and Analysis, Seattle, Amer. Meteor. Soc., 364-371.

_____, and _____, 1986: A Local AFOS MOS Program (LAMP) and its application to wind prediction. Mon. Wea. Rev., 114, 1313-1329.

Reed, R. J., 1963: Experiments in 1000-mb prognosis. National Meteorological Center Tech. Memorandum No. 26, Washington, D. C., 42 pp.

Unger, D. A., 1982: The sea level pressure prediction model of the Local AFOS MOS Program. NOAA Technical Memorandum NWS TDL 70, National Oceanographic and Atmospheric Administration, U.S. Department of Commerce, 33 pp.

_____, 1983: An evaluation of the sea level pressure and moisture models of LAMP. Preprints, Sixth Conference on Numerical weather Prediction, Omaha, Amer. Meteor. Soc., 252-258.

U454

FORECASTS DISCONTINUOUS VARIABLES

Harry R. Glahn
September 24, 2001

PURPOSE: To produce advective forecasts of both continuous and discontinuous variables from 1 to a maximum of 25 hours on a variable-sized computational grid. The forecast model used is purely advective. Initialization is with nearest neighbor analyses produced by U400D or from a radar grid. NCEP AVN model 1000-, 850-, 700-, and/or 500-mb winds can used as the advecting agents.

U454 is called from U150 and is controlled by several control files 'U454XXX.CN', where XXX is specific to a variable (e.g., CIG = ceiling height), which is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below [e.g., IP(22)].

CONTROL FILE INPUT: 'U454XXX.CN' (Unit = KFILDI)

Record Type 1 - Format (2I4,8X,F4.0,4X,2I4,1X,A16,2I4,3F4.0,4X,5F4.0,I4)
Overall Control

This record contains the maximum number of hours to make forecasts for and other information which is not projection specific. A control file is used for each variable that is in the U150.CN control file. All of these are read before any forecasts are made. All variables in Record Type 1 through 10 are variable specific except NPROJH, MESHW, IBACKN, and IBACKL; for those, the last one read will be used. To avoid confusion and possible mistakes, it is best to make these variables the same in all .CN files.

NPROJH - Maximum projection in hours to make forecasts for, starting with 1 hour, maximum of 25. Certain projections can be omitted (see MAKEF( ) below). This is only for efficiency of computation and could be set to 24 permanently. NPROJH is not variable by weather element.

MESHW - The nominal mesh length to use in computing advecting winds and trajectories. MESHW is not necessarily equal to MESH (see below). MESHW is not variable by weather element.

C00AF - The factor to use to compute the NCEP 1000-mb advecting winds.

IBACKN - The maximum number of cycles to go back for an NCEP forecast at 6-h intervals. For instance, if IBACKN = 2 and the current run of the NCEP forecast model is not available, U454 will look for 2 previous runs at intervals of 6 hours earlier. By setting IBACKN = 2, then either the 6-h or 12-h previous run could be used in that priority order. When a previous run is used, the projection(s) are adjusted accordingly. The "current run" of the NCEP forecast model is defined to be the cycle of the LAMP analysis, mod 6. That is, if the LAMP analysis is for 05Z, the

current NCEP model run is defined as 00Z. No assumption is made as to when the model actually runs.

IBACKL - The maximum number of cycles to go back for a LAMP forecast at 1-h intervals to use as an analysis. For instance, if IBACKL = 1, and the current analysis is not available, U454 will look for a forecast from 1 previous run. IBACKL will usually be 0; U454 is not fully developed for IBACKL > 0.

TITLE - A 16-character title for this variable, the variable being defined by the 4-word ID read by U150. (CHARACTER*16)

MESH - The nominal mesh length to use for the forecast variables in the run. Note that this is not variable by projection. (See MESHW above.) Forecasts will be written to the archive at this mesh length. Initial fields (tau = 0) will be written at the mesh length available, either from an analysis in U400A or U400D or from the radar archive.

ITRPLQ - The method of interpolation when adjusting an input grid to the mesh size used in the run. The values are:
1 = bilinear,
2 = biquadratic.
This is not used for interpolation into grids for trajectory end points. Interpolation into wind fields is biliner.

C85AF - The factor to use to compute the NCEP 850-mb advecting winds.

C70AF - The factor to use to compute the NCEP 700-mb advecting winds.

C50AF - The factor to use to compute the NCEP 500-mb advecting winds.

N00SM - The number of times the NCEP 1000-mb U and V components of the wind are smoothed with a square 5-point smoother on a 1 Bedient grid for purposes of computing the advecting winds. If N00SM $\geq$ 10, the number of smoothings is mod 10 on a 1/2-Bedient grid. (See C00AF.)

N85SM - The number of times the NCEP 850-mb U and V components of the wind are smoothed with a square 5-point smoother on a 1 Bedient grid for purposes of computing the advecting winds. If N85SM $\geq$ 10, the number of smoothings is mod 10 on a 1/2-Bedient grid. (See C85AF.)

N70SM - The number of times the NCEP 700-mb U and V components of the wind are smoothed with a square 5-point smoother on a 1 Bedient grid for purposes of computing the advecting winds. If N70SM $\geq$ 10, the number of smoothings is mod 10 on a 1/2-Bedient grid. (See C70AF.)

N50SM - The number of times the NCEP 500-mb U and V components of the wind are smoothed with a square 5-point smoother on a 1 Bedient grid for purposes of computing the advecting winds. If N50SM $\geq$ 10, the number of smoothings is mod 10 on a 1/2-Bedient grid. (See C50AF.)

CFILT - The fraction of the advecting wind reduction to apply. A "1" means apply the full reduction; 0 means do not reduce the winds. The standard version uses 1.

I454DG - Set to 1 when the diagnostic information listed in Sections "TDLPACK Grids for Checkout or Quality Control" and "GRIDPRINT for Checkout or Quality Control" are to be written to Unit Nos. KFILOG (TDLPACK) and/or IP(22) (Gridprint); zero otherwise.

Record Type 2 - Format (25F4.0) Smoothing Parameter

B(J) - Smoothing parameter for each projection (J=1,25) used to smooth the forecasts. Only NPROJH values are read. While B( ) > 0 is possible, it is expected this will be zero because the smoothing routine does not deal with missing values.

Record Type 3 - Format (25F4.0) Radius of Influence

R(J) - Radius of influence in grid units for each projection (J=1,25). Only NPROJH values are read. R( ) determines how far from the trajectory end point one looks to find a non missing value.

Record Type 4 - Format (25F4.0) Multiplicative Constant for Gridprinting

SMULT(J) - The multiplicative factor to use before gridprinting the data for each projection (J=1,25). Only NPROJH values are read and used.

Record Type 5 - Format (25F4.0) Additive Factor for Gridprinting

SADD(J) - The additive factor to use before gridprinting the data for each projection (J=1,25). Only NPROJH values are read and used. Usually this is zero.

Record Type 6 - Format (25F4.0) Origin for Gridprinting

ORIGIN(J) - The origin when gridprinting the data for each pass (J=1,25). Only NPROJH values are read and used. That is, the base contour will be at ORIGIN( ).

Record Type 7 - Format (25F4.0) Contour Interval for Gridprinting

CINT(J) - The contour interval when gridprinting the data for each pass (J=1,25). Only NPROJH values are read and used. CINT( ) is in reference to the original units, not after the multiplication by SMULT( ).

Record Type 8 - Format (6I8) Projections to Make Forecasts For

MAKEF(J) - Projections to make forecasts for (J=1,25). Only NPROJH values are read and used. A value of 1 for projection J means make a forecast will be made for projection J; a zero means to not make a forecast. Note that NPROJH is the maximum projection that a forecast can be made for, and MAKEF( ) controls the actual projections.

Record Type 9 - Format (6I8) Gridprint Option for Forecasts

NPRT(J) - Gridprinting option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 10 - Format (6I8) TDLPACK Option for Forecasts

NTDL(J) - TDLPACKing option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = TDLPACK and write,
0 = otherwise.

DATA INPUT:

Most gridpoint data needed by U454 have been stored by U150 into the MOS-2000 Internal Storage System. GFETCH is used to retrieve the data. Final grids for eventual writing to Unit Number KFILIO are packed and written there by GSTORE, called by subroutine PAWING. However, the terrain heights are retrieved from the External Random Access; provision is made for this access. The radar grid is read from a TDLPACK sequential file with DD = 4.

DATA OUTPUT

There are three forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). These are Types A, B, and D explained in U150. See the Data Output section of U150.

EXAMPLE CONTROL FILE: 'U454XXX.CN'

An example exists as file 'U454RFL.CN' in directory '/lamp2000/dru150' on ice. The easiest way to set up a run is to take an existing control file and modify it.

OUTPUT:

Output is generally explained in the U150 writeup.

RESTRICTIONS

See U150.

COMMENTS

Comments in U150 apply to U454.

Although the primary purpose of U150 and U454 is to produce packed gridpoint forecasts, writing of such data is not done if KFILIO = 0 (see Record Type 8 of U150). That is, no output unit number and file name have been provided. Note that rendering the unit number = 0 is done by omitting the file in the input .CN file, not by leaving the file name there and setting the unit number = 0.

The U454 module of U150 accepts grids of variable dimension. The grids are put on the grid with a mesh length MESH or MESHW as appropriate; MESH and MESHW are constant for any particular advected variable, but are read in for each .CN control file. MESHW is constant for all variables for U454; the last one read from a .CN file is used. MESH can vary by variable. This means each variable can be advected at the resolution desired, but the advecting wind is calculated with the same mesh length for all variables. It is likely MESH will be used as small as 10 km for some variables (e.g., 10-km radar data) and larger for other variables. Since the advecting winds are quite smooth with respect to a grid size of a few tens of kilometers, MESHW can probably be used as 80 km. Use of MESHW > 10 km reduces computation measurably.

It is assumed that the grid that is to furnish input (e.g., an NCEP AVN model) is large enough to encompass the primary LAMP grid (the exception is the radar data). The LAMP grid is defined by a lower left latitude and longitude, a "basic" mesh length equal to a quarter Bedient, and an NX by NY extent; the pole position is also known or implied, being redundant to the other information. The first step is to position the "LAMP" grid within the input grid at the input grid's mesh length (subroutine POSGRD is used). The next step is to make the input grid the desired mesh length (subroutine SIZEGR is used).

The earlier versions of LAMP produced a forecast by going back 1 hour at a time, making a forecast, then going back from that forecast another hour, etc. U454 goes all the way back to the initial analysis for each projection. This alleviates the noise introduced by multiple trajectory end point determinations, but requires more computation.

Any combination of NCEP winds from the four standard height levels can be used for any variable to be advected. However, it is expected no more than two will be needed for any one variable. For instance, for the 4600 to 6500 ft. level, the 850-mb winds might be used, while for the levels at 4500 and below, some combination of 1000- and 850-mb winds might be used.

The smoothing parameters NxxSM read in Record Type 1 operate the same way no matter what the values of MESH and MESHW.

Because MESHW was added recently, some control files may still contain a zero in that location. In that case, U454 provides a **** diagnostic and sets MESHW = MESH.

The grids input to U454 can have missing values, such as cloud analyses. The trajectory end point in XPOS( ) and YPOS( ) are constrained through subroutine FCST and TRJCT to one gridlength outside the grid. Because of the discontinuous nature of the initial grids to be forecast, the subroutine NEARGR finds the closest value to the trajectory end point (within a search radius in the .CN file). If the trajectory endpoint is $\geq$ .501 gridlength outside the grid, a missing (9999) is returned.

Smoothing is possible, but is not recommended, not only because of the discontinuous nature of the fields, but because the smoothing routine does not consider missing values. The continuous fields temperature and dew point could be smoothed, but even with them, the boundary points might

have propagated several gridlengths inside the grid. If smoothing becomes necessary, a different smoothing routine will be needed.

VARIABLES ACCOMMODATED BY U454

U150 calls U454 once per cycle. Furnished to U454 is the list of 4-word MOS-2000 variable identifiers for which processing is to be done by some module of U150. U454 has resident a table that contains the variable IDs that U454 is engineered to handle. There is a corresponding table which gives the name of the '.CN' control file, 'U454XXX.CN'.

U454 searches the table for each U150 input ID and processes those variables that reside in its table ITABLE( ). The associated fields that may be needed besides the initial analysis and radar and terrain input are:

Variable

1000-mb GRID-U wind from NCEP	004000008	1000	t	0
850-mb GRID-U wind from NCEP	004000008	850	t	0
700-mb GRID-U wind from NCEP	004000008	700	t	0
500-mb GRID-U wind from NCEP	004000008	500	t	0
1000-mb GRID-V wind from NCEP	004100008	1000	t	0
850-mb GRID-V wind from NCEP	004100008	850	t	0
700-mb GRID-V wind from NCEP	004100008	700	t	0
500-mb GRID-V wind from NCEP	004100008	500	t	0

The only variables normally written to the archive file, Unit No. KFILIO, are the forecasts for each hour t indicated in MAKEF( ). They are written with the smoothing parameter S = 0 in the ID even though the forecasts may have been smoothed. This is consistent with the final analyses (e.g., see U400D) being written with S = 0 even though they may have been smoothed.

Currently, U454 is ready to handle 49 variables. The fields must be available from U400A, U400D, or the radar archive.

Variable

Ceiling Height:	008000005	0	0	0
Control file	U454CIG.CN			
Visibility	008100005	0	0	0
Control file	U454VIS.CN			
Total cloud cover	008311005	0	0	0
Control file	U454SKY.CN			
Amount lowest cloud lyr	008320005	0	0	0
Control file	U454CA1.CN			
Height lowest cloud lyr	008321005	0	0	0
Control file	U454CH1.CN			
Amount 2nd cloud lyr	008322005	0	0	0
Control file	U454CA2.CN			

```
Height 2nd cloud lyr       008323005       0  0  0
  Control file             U454CH2.CN

Amount 3rd cloud lyr       008324005       0  0  0
  Control file             U454CA3.CN

Height 3rd cloud lyr       008325005       0  0  0
  Control file             U454CH3.CN

Amount 4th cloud lyr       008326005       0  0  0
  Control file             U454CA4.CN

Height 4th cloud lyr       008327005       0  0  0
  Control file             U454CH4.CN

Amount 5th cloud lyr       008328005       0  0  0
  Control file             U454CA5.CN

Height 5th cloud lyr       008329005       0  0  0
  Control file             U454CH5.CN

Amount 6th cloud lyr       008330005       0  0  0
  Control file             U454CA6.CN

Height 6th cloud lyr       008331005       0  0  0
  Control file             U454CH6.CN

First weather group        008500005       0  0  0
  Control file             U454WX1.CN

Second weather group       008510005       0  0  0
  Control file             U454WX2.CN

Third weather group        008520005       0  0  0
  Control file             U454WX3.CN

Grid elevations
          409020005       0  0  0
  Control file             U454ELE.CN

Temperature                0023010005      2  0  0
  Control file             U454TMP.CN

Dew point                  0033010005      2  0  0
  Control file             U454DEW.CN

Radar reflectivity H+00    7076000004      0  0  0
  Control file             U454RFL.CN

Radar reflectivity H+15    7076150004      0  0  0
  Control file             U454RFL.CN

Radar reflectivity H+30    7076300004      0  0  0
  Control file             U454RFL.CN

Radar reflectivity H+45    7076450004      0  0  0
  Control file             U454RFL.CN
```

Radar coverage H+00	7077000004		0	0	0
Control file	U454RRF.CN				
Radar coverage H+15	7077150004		0	0	0
Control file	U454RRF.CN				
Radar coverage H+30	7077300004		0	0	0
Control file	U454RRF.CN				
Radar coverage H+45	7077450004		0	0	0
Control file	U454RRF.CN				
Cloud amount 0-1 hdr ft	008350005		0	0	0
Control file	U454A01.CN				
Cloud height 0-1 hdr ft	008355005	0001	0	0	
Control file	U454H01.CN				
Cloud amount 2-4 hdr ft	008350005	020004	0	0	
Control file	U454A04.CN				
Cloud height 2-4 hdr ft	008355005	020004	0	0	
Control file	U454H04.CN				
Cloud amount 5-9 hdr ft	008350005	050009	0	0	
Control file	U454A09.CN				
Cloud height 5-9 hdr ft	008355005	050009	0	0	
Control file	U454H09.CN				
Cloud amount 10-15 hdr ft	008350005	100015	0	0	
Control file	U454A15.CN				
Cloud height 10-15 hdr ft	008355005	100015	0	0	
Control file	U454H15.CN				
Cloud amount 16-30 hdr ft	008350005	160030	0	0	
Control file	U454A30.CN				
Cloud height 16-30 hdr ft	008355005	160030	0	0	
Control file	U454H30.CN				
Cloud amount 31-45 hdr ft	008350005	310045	0	0	
Control file	U454A45.CN				
Cloud height 31-45 hdr ft	008355005	310045	0	0	
Control file	U454H45.CN				
Cloud amount 46-65 hdr ft	008350005	460065	0	0	
Control file	U454A65.CN				
Cloud height 46-65 hdr ft	008355005	460065	0	0	
Control file	U454H65.CN				
Cloud amount 66-120 hdr ft	008350005	660120	0	0	
Control file	U454A1X.CN				

```
Cloud height 66-120 hdr ft 008355005  660120  0  0
  Control file             U454H1X.CN

Cloud amount > 120 hdr ft  008350005 1219999  0  0
  Control file             U454A99.CN

Cloud height > 120 hdr ft  008355005 1219999  0  0
  Control file             U454H99.CN

Obscuration height hdr ft  008350005  000000  0  0
  Control file             U454A00.CN

Obscuration amount hdr ft  008355005  000000  0  0
  Control file             U454H00.CN
```

Note that the same .CN file is used for all four radar reflectivity fields and also the same one for all four radar coverage fields.

TDLPACK GRIDS FOR CHECKOUT OR QUALITY CONTROL

TDLPACK grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed below needed in U454 will be written to the dataset whose name has been provided to OUTDIS and Unit No. KFILOG from the control file 'U150.CN', Record Type 9, provided (1) KFILOG is not zero and (2) I454DG is not zero:

004000008	1000	t	0	NCEP 1000-mb grid-oriented U wind in m/s when used in advection.
004000008	850	t	0	NCEP 850-mb grid-oriented U wind in m/s when used in advection.
004000008	700	t	0	NCEP 700-mb grid-oriented U wind in m/s when used in advection.
004000008	500	t	00	NCEP 500-mb grid-oriented U wind in m/s when used in advection.
004100008	1000	t	0	NCEP 1000-mb grid-oriented V wind in m/s when used in advection.
004100008	850	t	0	NCEP 850-mb grid-oriented V wind in m/s when used in advection.
004100008	700	t	0	NCEP 700-mb grid-oriented V wind in m/s when used in advection.
004100008	500	t	00	NCEP 500-mb grid-oriented V wind in m/s when used in advection.
004mmm005	777	t	80	Filtered (if desired) U-advective winds in grid units per half hour; mmm is 199 plus the number of the height combination having been computed.
004nnn005	777	t	80	Filtered (if desired) V-advective winds in grid units per half hour; nnn is 399 plus the number of the height combination having been computed.
004mmm005	777	t	80	Unfiltered Earth-oriented U-advective winds in m/s; mmm is 599 plus the number of the height combination having been computed.
004nnn005	777	t	80	Unfiltered Earth-oriented V-advective winds in m/s; nnn is 799 plus the number of the height combination having been computed.
004540005	000	t	0	Upstream X-positions of trajectories.
004550005	000	t	0	Upstream Y-positions of trajectories.

xxxxxx005	yyy	t	0	Forecasts for hour t, where xxxxxx is the CCCFFF of the variable being forecasted and yyy is the LLLL UUUU of the variable or else zero. (The first two words in the ID are the first two words of the variable being forecast.)
707xxx005	000	0	0	The initial radar fields (tau = 0)
001000005	004mmm005	t	0	Summed heights for CLAM; for mmm, see above.

Note that gridpoint data written by LAMP coming directly from NCEP have the DD = NCEPNO (= 8), while data modified or computed by LAMP have the DD = LAMPNO (=5). Projection is represented by "t" above, and will occur for projections 0 through NPROJH (see Record Type 1). To emphasize, if a variable is not used, it will not be written. For instance, if the 1000 mb height is not used in the advection, it will not be written.

Also note that while most data are written with DD = LAMPNO = 5, <u>forecasts</u> from U454 are written with DD = LAMPNO + INCDD. This allows the forecasts from one run to be distinguished from forecasts of another run for verification purposes.

The mmm and nnn in the above IDs can make the identification of the fields written to the disposable file difficult. Since essentially any combination of up to five fields can be used, having a hard definition of the specific combination is not feasible. If only one combination is used in a checkout run, then mmm = 200 and nnn = 400 for the final filtered grid-oriented advective winds in units of gridlengths per half hour, and mmm=600 and nnn=800 for intermediate earth-oriented unfiltered advective winds in units of m/s. If two combinations are used (one for each of one or more variables) then additionally mmm = 201 and nnn = 401 for the final advective winds, and mmm=601 and nnn=801 for the intermediate winds. The combinations can be identified by the order in which they were computed.

GRIDPRINT FOR CHECKOUT OR QUALITY CONTROL

The same grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed in the section above will be gridprinted and written to Unit No. IP(22) from the control file 'U150.CN', Record Type 1, provided (1) IP(22) is not zero and (2) I454DG is not zero. All are contoured at intervals provided in the MOS-2000 constants file, except some are adjusted for the grid mesh.

SMOOTHING PROCEDURE

The smoothing parameter B(,) (see Record Type 2) is explained below:

$$S_{i,j} = \frac{A_{i,j} + b\bar{A}_{i,j}}{1 + b}$$

where:

$S_{i,j}$ = Smoothed gridpoint values at point i, j.
$A_{i,j}$ = Value of gridpoint at point i, j before smoothing.
<u>b</u> = Smoothing parameter.
$\bar{A}_{i,j} = (1/4)\ (A_{i+1,j}+A_{i-1,j}+A_{i,j+1}+A_{i,j-1})$

Low values produce the least smoothing and higher values produce heavier smoothing. A value of zero indicates no smoothing. The smoothing routine used is SMTH5.

NONSYSTEM ROUTINES USED

Use the load line in file '/lamp2000/dru150', dataset 'u150.com' on ice.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: /lamp2000/u454lib. The driver is in /lamp2000/dru150.

U452

FORECASTS SATURATION DEFICIT

Harry R. Glahn
June 4, 2001

PURPOSE: To produce forecasts of saturation deficit (SD) from 1 to a maximum of 25 hours on a variable-sized computational grid. The forecast model used is basically SLYH, although various modifications have been made. Initialization is with the saturation deficit analysis produced by U400A, and NCEP model 500-mb and 1000-mb fields are used as the advecting agent and the 1000- to 500-mb thickness is used as the development term in this Lagrangian model. In this version, each forecast is developed from a full trajectory from projection tau back to the initial analysis (start time) (see U453).

U452 is called from U150 and is controlled by a control file 'U452.CN', which is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below [e.g., IP(22)].

CONTROL FILE INPUT: 'U452.CN' (Unit = KFILDI)

Record Type 1 - Format (I4,4F4.0,4X,2I4,1X,A16,5I4,F4.0,I4) Overall Control

This record contains the maximum number of hours to make forecasts for and other information which is not projection specific.

NPROJH - Maximum projection in hours to make forecasts for, starting with 1 hour, maximum of 25. Certain projections can be omitted (see MAKEF( ) below). This is only for efficiency of computation and could be set to 24 permanently.

CPMA - The factor to multiply PMA( ) by to get the final PMA( ) terrain term [see references (e.g., MWR 1986) for definition of PMA]. Normally 1 for the basic model; a larger value would give a larger effect.

C50TF - The factor to multiply the thickness change term by in computing thickness change. Normally 1 for the basic model; a larger value would give a larger effect.

C33AF - The factor to use to compute the 500-mb portion of the advecting winds. Normally 0.33 for the basic model.

C00AF - The factor to use to compute the 1000-mb portion of the advecting winds. Normally 0.5 for the basic model.

IBACKN - The maximum number of cycles to go back for an NCEP forecast at 6-h intervals. For instance, if IBACKN = 2 and the current run of the NCEP forecast model is not available, U452 will look for 2 previous runs at intervals of 6 hours earlier. By setting IBACKN = 2, then either the 6-h or 12-h previous run could be used in that priority order. When a previous run is used, the

projection(s) are adjusted accordingly. The "current run" of the NCEP forecast model is defined to be the cycle of the LAMP analysis, mod 6. That is, if the LAMP analysis is for 05Z, the current NCEP model run is defined as 00Z. No assumption is made as to when the model actually runs.

IBACKL - The maximum number of cycles to go back for a LAMP forecast at 1-h intervals. For instance, if IBACKL = 2, and the 1-h forecast from the previous run of the LAMP forecast model is not available, U452 will look for 2 previous runs. When a previous run of LAMP is used, the projection(s) are adjusted accordingly (e.g., a 2-h forecast is used from a LAMP run 2 hours earlier). Note that both IBACKN and IBACKL refer to the total number of runs to look for, IBACKN at 6-h intervals and IBACKL at 1-h intervals. (This may not be useful in U452.)

TITLE - A 16-character title for this variable, the variable being defined by the 4-word ID read by U150. (CHARACTER*16)

MESH - The nominal mesh length to use in the run. Note that this is not variable by projection. Normally 10 to match the anticipated mesh length of the saturation deficit analysis.

ITRPLQ - The method of interpolation when adjusting an input grid to the mesh size used in the run. Note that this is not variable by projection. The values are:
1 = bilinear,
2 = biquadratic.

N50SM - The number of times the 500-mb heights are smoothed with a square 25-point smoother on a 1- or 1/2-Bedient grid for purposes of computing the advecting winds (see C33AF). When N50SM < 10, smoothing is on a 1-Bedient grid; when N50SM $\geq$ 10, smoothing is on a 1/2-Bedient grid N50SM-10 times.

N00SM - The number of times the 1000-mb heights are smoothed with a square 25-point smoother on a 1/4-Bedient grid for purposes of computing the advecting winds. (See C00AF.)

NTTSM - The number of times the 1000-mb heights are smoothed over 25 points on a 1/4-Bedient grid.

CFILT - The fraction of the advecting wind reduction to apply. A "1" means apply the full reduction; 0 means do not reduce the winds. (Reduction not currently implemented.)

I452DG - Set to 1 when the diagnostic information listed in Sections "TDLPACK Grids for Checkout or Quality Control" and "GRIDPRINT for Checkout or Quality Control" are to be written to Unit Nos. KFILOG (TDLPACK) and/or IP(22) (Gridprint); zero otherwise.

Record Type 2 - Format (25F4.0) Smoothing Parameter

B(J) - Smoothing parameter for each projection (J=1,25) used to smooth the SD forecasts. Only NPROJH values are read. The smoothing procedure is that explained in the section "Smoothing Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985).

Record Type 3 - Format (25F4.0) Radius of Influence

R(J) - Radius of influence in grid units for each projection (J=1,25). Only NPROJH values are read. R( ) currently is not used.

Record Type 4 - Format (25F4.0) Multiplicative Constant for Gridprinting

SMULT(J) - The multiplicative factor to use before gridprinting the SD data for each projection (J=1,25). Only NPROJH values are read and used.

Record Type 5 - Format (25F4.0) Additive Factor for Gridprinting

SADD(J) - The additive factor to use before gridprinting the SD data for each projection (J=1,25). Only NPROJH values are read and used. Usually this is zero.

Record Type 6 - Format (25F4.0) Origin for Gridprinting

ORIGIN(J) - The origin when gridprinting the SD data for each pass (J=1,25). Only NPROJH values are read and used. That is, the base contour will be at ORIGIN( ).

Record Type 7 - Format (25F4.0) Contour Interval for Gridprinting

CINT(J) - The contour interval when gridprinting the SD data for each pass (J=1,25). Only NPROJH values are read and used. CINT( ) is in reference to the original units, not after the multiplication by SMULT( ).

Record Type 8 - Format (6I8) Projections to Make Forecasts For

MAKEF(J) - Projections to make forecasts for (J=1,25). Only NPROJH values are read and used. A value of 1 for projection J means make a forecast will be made for projection J; a zero means to not make a forecast. Note that NPROJH is the maximum projection that a forecast can be made for, and MAKEF( ) controls the actual projections.

Record Type 9 - Format (6I8) Gridprint Option for SD Pressure Forecasts

NPRT(J) - Gridprinting option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 10 - Format (6I8) TDLPACK Option for SD Forecasts

NTDL(J) - TDLPACKing option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = TDLPACK and write,
0 = otherwise.

DATA INPUT:

Most gridpoint data needed by U452 have been stored by U150 into the MOS-2000 Internal Storage System. GFETCH is used to retrieve the data. Final grids for eventual writing to Unit Number KFILIO are packed and written there by GSTORE, called by subroutine PAWING. However, the terrain heights are retrieved from the External Random Access; provision is made for this access.

DATA OUTPUT

There are three forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). These are Types A, B, and D explained in U150. See the Data Output section of U150.

EXAMPLE CONTROL FILE: 'U452.CN'

An example exists as file 'U452.CN' in directory 'users24/glahn/lamp/dru150' on ice. The easiest way to set up a run is to take an existing control file and modify it; DO NOT START FROM SCRATCH.

OUTPUT:

Output is generally explained in the U150 writeup.

RESTRICTIONS

See U150.

COMMENTS

Comments in U150 apply to U452.

Although the primary purpose of U150 and U452 is to produce packed gridpoint forecasts of saturation deficit, writing of such data is not done if KFILIO = 0 (see Record Type 8 of U150). That is, no output unit number and file name have been provided. Note that rendering the unit number = 0 is done by omitting the file in the input .CN file, not by leaving the file name there and setting the unit number = 0.

The U452 module of U150 accepts grids of variable dimension. The grids are put on the grid with a mesh length MESH; this is not changed throughout the run.

It is assumed that the grid that is to furnish input (e.g., an NCEP AVN model) is large enough to encompass the primary LAMP grid. The LAMP grid is defined by a lower left latitude and longitude, a "basic" mesh length equal to a quarter Bedient, and an NX by NY extent; the pole position is

also known or implied, being redundant to the other information. The first step is to position the "LAMP" grid within the input grid at the input grid's mesh length (subroutine POSGRD is used). The next step is to make the input grid the desired mesh length (subroutine SIZEGR is used).

It has been difficult to program U452 with exactly the same formulations as the currently running model. The external documentation is not complete and detailed, and is not always consistent with the code and its internal comments. The program that produced the terrain field used in the model could not be found, and there was probably some processing done when the file was written. However, it is believed the formulation is very close.

VARIABLES ACCOMMODATED BY U452

U150 calls U452 once per cycle. Furnished to U452 is the list of 4-word MOS-2000 variable identifiers for which processing is to be done by some module of U150. U452 has resident a table that contains the single variable ID that U452 is engineered to handle, 00341005. There is a corresponding (one element) table which gives the name of the '.CN' control file, 'U452.CN'.

U452 searches the table for each U150 input ID and processes those variables that reside in its table ITABLE( ). Only one is accommodated, and the associated fields needed are:

Variable				
Saturation Deficit:	003410005	0	0	0
500-mb heights from NCEP	001000008	500	t	0
1000-mb heights from NCEP	001000008	1000	t	0

The only variables normally written to the archive file, Unit No. KFILIO, are the SD forecasts for each hour t indicated in MAKEF( ) and provided the U150.CN file contains the SD variable with that projection t. They are written with the smoothing parameter S = 0 in the ID even though the forecasts may have been smoothed. This is consistent with the final analyses (e.g., see U400A) being written with S = 0 even though they may have been smoothed.

TDLPACK GRIDS FOR CHECKOUT OR QUALITY CONTROL

TDLPACK grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed below will be written to the dataset whose name has been provided to OUTDIS and Unit No. KFILOG from the control file 'U150.CN', Record Type 9, provided (1) KFILOG is not zero and (2) I450DG is not zero:

409020005	0	0	0	Terrain height in m
409020005	0	0	60	Terrain height in m smoothed over 25 1-Bedient points
409023005	0	0	0	Terrain height field in m <u>used in the model</u> after multiplicative factors and height adjustment has been applied
409240005	0	0	0	Full conservative quantity in m <u>used in the model</u>
001000008	500	t	0	NCEP 500-mb heights in m
001000005	500	t	60	500-mb heights in m smoothed over 25 1-Bedient points N50SM (see Record Type 1) times

001000008	1000	t	0	NCEP 1000-mb heights in m
001000005	1000	t	60	1000-mb heights in m smoothed over 25 1-Bedient points N00SM (see Record Type 1) times
004000005	500	t	60	U-advective winds in grid units per half hour
004100005	500	t	60	V-advective winds in grid units per half hour
003410005	0	t	0	Saturation deficit forecasts for hour t

The convention used is that when the data come directly from NCEP, except that it has been put on the current LAMP grid, the model number DD in the variable ID is NCEP's number (8 currently for the AVN), but after processing for LAMP, it is given DD = 5 for LAMP. Projection is represented by "t" above, and will occur for projections 0 through NPROJH (see Record Type 1).

GRIDPRINT FOR CHECKOUT OR QUALITY CONTROL

The same grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed in the section above will be gridprinted and written to Unit No. IP(22) from the control file 'U150.CN', Record Type 1, provided (1) IP(22) is not zero and (2) I450DG is not zero. All are contoured at intervals of 100 meters, except the advective U and V winds which are first multiplied by 100 then contoured at units of 10.

The parameters SMULT( ), SADD( ), ORIGIN( ), and CINT( ) apply only to SD output.

SMOOTHING PROCEDURE

The smoothing parameter B(,) (see Record Type 2) is explained below:

$$S_{i,j} = \frac{A_{i,j} + b\bar{A}_{i,j}}{1 + b}$$

where:
$S_{i,j}$ = Smoothed gridpoint values at point i, j.
$A_{i,j}$ = Value of gridpoint at point i, j before smoothing.
b = Smoothing parameter.
$A_{i,j} = (1/4)\ (A_{i+1,j}+A_{i-1,j}+A_{i,j+1}+A_{i,j-1})$

Low values produce the least smoothing and higher values produce heavier smoothing. A value of zero indicates no smoothing.

NONSYSTEM ROUTINES USED

Use the load line in file 'users24/glahn/lamp/dru150', dataset 'u150.com' on ice.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: users24/glahn/lamp/u452lib. The driver is in users24/glahn/lamp/dru150.

BIBLIOGRAPHY:

Glahn, H. R., 1980: Plans for the development of a Local AFOS MOS Program (LAMP). Preprints Eighth Conference on Weather Forecasting and Analysis, Denver, Amer. Meteor. Soc., 302-305.

_____, and D. A. Lowry, 1972: An operational Subsynoptic Advection Model (SAM). J. Appl. Meteor., 11, 578-585.

_____, _____, and G. W. Hollenbaugh, 1969: An operational subsynoptic advection model. ESSA Technical Memorandum WBTM TDL 23, Environmental Science Services Administration, U.S. Department of Commerce, 33 pp.

_____, and D. A. Unger, 1982: Some results from the Local AFOS MOS Program (LAMP). Preprints 9th Conference on Weather Forecasting and Analysis, Seattle, Amer. Meteor. Soc., 364-371.

_____, and _____, 1986: A Local AFOS MOS Program (LAMP) and its application to wind prediction. Mon. Wea. Rev., 114, 1313-1329.

Unger, D. A., 1983: An evaluation of the sea level pressure and moisture models of LAMP. Preprints, Sixth Conference on Numerical Weather Prediction, Omaha, Amer. Meteor. Soc., 252-258.

U453

FORECASTS SATURATION DEFICIT

Harry R. Glahn
January 24, 2002

PURPOSE: To produce forecasts of saturation deficit (SD) from 1 to a maximum of 25 hours on a variable-sized computational grid. The forecast model used is basically SLYH, although various modifications have been made. Initialization is with the saturation deficit analysis produced by U400A, and NCEP model 500-mb and 1000-mb fields are used as the advecting agent and the 1000- to 500-mb thickness is used as the development term in this Lagrangian model. In this version, forecasts are made in hourly steps; a forecast for projection tau is made by going back to the previous analysis (for tau = 1) or forecast at tau-1 (see U452).

U453 is called from U150 and is controlled by a control file 'U453.CN', which is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below [e.g., IP(22)].

CONTROL FILE INPUT: 'U453.CN' (Unit = KFILDI)

Record Type 1 - Format (I4,4F4.0,3I4,1X,A16,5I4,F4.0,I4) Overall Control

This record contains the maximum number of hours to make forecasts for and other information which is not projection specific.

NPROJH - Maximum projection in hours to make forecasts for, starting with 1 hour, maximum of 25. Certain projections can be omitted (see MAKEF( ) below). This is only for efficiency of computation and could be set to 24 permanently.

CPMA - The factor to multiply PMA( ) by to get the final PMA( ) terrain term [see references (e.g., MWR 1986) for definition of PMA]. Normally 1 for the basic model; a larger value would give a larger effect.

C50TF - The factor to multiply the thickness change term by in computing thickness change. Normally 1 for the basic model; a larger value would give a larger effect.

C33AF - The factor to use to compute the 500-mb portion of the advecting winds. Normally 0.33 for the basic model.

C00AF - The factor to use to compute the 1000-mb portion of the advecting winds. Normally 0.5 for the basic model.

MESHW - The nominal mesh length to use in computing advecting winds and trajectories. MESHW is not necessarily equal to MESH (see below).

IBACKN - The maximum number of cycles to go back for an NCEP forecast at 6-h intervals. For instance, if IBACKN = 2 and the current run of the NCEP forecast model is not available, U453 will look for 2 previous runs at intervals of 6 hours earlier. By setting IBACKN = 2, then either the 6-h or 12-h previous run could be used in that priority order. When a previous run is used, the projection(s) are adjusted accordingly. The "current run" of the NCEP forecast model is defined to be the cycle of the LAMP analysis, mod 6. That is, if the LAMP analysis is for 05Z, the current NCEP model run is defined as 00Z. No assumption is made as to when the model actually runs.

IBACKL - The maximum number of cycles to go back for a LAMP forecast at 1-h intervals to use as an analysis. For instance, if IBACKL = 1, and the current analysis is not available, U453 will look for a forecast from 1 previous run. IBACKL will usually be 0; U453 is not fully developed for IBACKL > 0.

TITLE - A 16-character title for this variable, the variable being defined by the 4-word ID read by U150. (CHARACTER*16)

MESH - The nominal mesh length to use in the run for the forecast variables (see MESHW above). Note that this is not variable by projection. Normally 10 to match the anticipated mesh length of the saturation deficit analysis.

ITRPLQ - The method of interpolation when adjusting an input grid to the mesh size used in the run. Note that this is not variable by projection. The values are:
1 = bilinear,
2 = biquadratic.
This is not used for interpolation into grids for trajectory end points. Interpolation into wind fields is biliner.

N50SM - The number of times the 500-mb heights are smoothed with a square 25-point smoother on a 1- or 1/2-Bedient grid for purposes of computing the advecting winds (see C33AF). When N50SM < 10, smoothing is on a 1-Bedient grid; when N50SM $\geq$ 10, smoothing is on a 1/2-Bedient grid N50SM-10 times.

N00SM - The number of times the 1000-mb heights are smoothed with a square 25-point smoother on a 1/4-Bedient grid for purposes of computing the advecting winds. (See C00AF.)

NTTSM - The number of times the 1000-mb heights are smoothed over 25 points on a 1/4-Bedient grid.

CFILT - The fraction of the advecting wind reduction to apply. A "1" means apply the full reduction; 0 means do not reduce the winds. (Reduction not currently implemented.)

I452DG - Set to 1 when the diagnostic information listed in Sections "TDLPACK Grids for Checkout or Quality Control" and "GRIDPRINT for Checkout or Quality Control" are to be written to Unit Nos. KFILOG (TDLPACK) and/or IP(22) (Gridprint); zero otherwise.

Record Type 2 - Format (25F4.0) Smoothing Parameter

B(J) - Smoothing parameter for each projection (J=1,25) used to smooth the SD forecasts. Only NPROJH values are read. The smoothing procedure is that explained in the section "Smoothing Procedure" and in NOAA Technical Memorandum NWS TDL 75, March 1985).

Record Type 3 - Format (25F4.0) Radius of Influence

R(J) - Radius of influence in grid units for each projection (J=1,25). Only NPROJH values are read. R( ) currently is not used.

Record Type 4 - Format (25F4.0) Multiplicative Constant for Gridprinting

SMULT(J) - The multiplicative factor to use before gridprinting the SD data for each projection (J=1,25). Only NPROJH values are read and used.

Record Type 5 - Format (25F4.0) Additive Factor for Gridprinting

SADD(J) - The additive factor to use before gridprinting the SD data for each projection (J=1,25). Only NPROJH values are read and used. Usually this is zero.

Record Type 6 - Format (25F4.0) Origin for Gridprinting

ORIGIN(J) - The origin when gridprinting the SD data for each pass (J=1,25). Only NPROJH values are read and used. That is, the base contour will be at ORIGIN( ).

Record Type 7 - Format (25F4.0) Contour Interval for Gridprinting

CINT(J) - The contour interval when gridprinting the SD data for each pass (J=1,25). Only NPROJH values are read and used. CINT( ) is in reference to the original units, not after the multiplication by SMULT( ).

Record Type 8 - Format (6I8) Projections to Make Forecasts For

MAKEF(J) - Projections to make forecasts for (J=1,25). Only NPROJH values are read and used. A value of 1 for projection J means make a forecast will be made for projection J; a zero means to not make a forecast. Note that NPROJH is the maximum projection that a forecast can be made for, and MAKEF( ) controls the actual projections.

Record Type 9 - Format (6I8) Gridprint Option for SD Pressure Forecasts

NPRT(J) - Gridprinting option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = gridprint,
0 = otherwise.

Record Type 10 - Format (6I8) TDLPACK Option for SD Forecasts

NTDL(J) - TDLPACKing option for the forecasts for each projection (J=1,25) for the subsetted area. Only NPROJH values are read and used.
1 = TDLPACK and write,
0 = otherwise.

DATA INPUT:

Most gridpoint data needed by U453 have been stored by U150 into the MOS-2000 Internal Storage System. GFETCH is used to retrieve the data. Final grids for eventual writing to Unit Number KFILIO are packed and written there by GSTORE, called by subroutine PAWING. However, the terrain heights are retrieved from the External Random Access; provision is made for this access.

DATA OUTPUT

There are three forms of data output, besides the diagnostics and other information on Units KFILDO and IP( ). These are Types A, B, and D explained in U150. See the Data Output section of U150.

EXAMPLE CONTROL FILE: 'U453.CN'

An example exists as file 'U453.CN' in directory 'lamp2000/dru150' on ice. The easiest way to set up a run is to take an existing control file and modify it.

OUTPUT:

Output is generally explained in the U150 writeup.

RESTRICTIONS

See U150.

COMMENTS

Comments in U150 apply to U453.

Although the primary purpose of U150 and U453 is to produce packed gridpoint forecasts of saturation deficit, writing of such data is not done if KFILIO = 0 (see Record Type 8 of U150). That is, no output unit number and file name have been provided. Note that rendering the unit number = 0 is done by omitting the file in the input .CN file, not by leaving the file name there and setting the unit number = 0.

The U453 module of U150 accepts grids of variable dimension. The grids are put on the grid with a mesh length MESH; this is not changed throughout the run, except MESHW is used for advecting winds and trajectories.

It is assumed that the grid that is to furnish input (e.g., an NCEP AVN model) is large enough to encompass the primary LAMP grid. The LAMP grid is defined by a lower left latitude and longitude, a "basic" mesh length equal to a quarter Bedient, and an NX by NY extent; the pole position is

also known or implied, being redundant to the other information. The first step is to position the "LAMP" grid within the input grid at the input grid's mesh length (subroutine POSGRD is used). The next step is to make the input grid the desired mesh length (subroutine SIZEGR is used).

It has been difficult to program U453 with exactly the same formulations as the currently running model. The external documentation is not complete and detailed, and is not always consistent with the code and its internal comments. The program that produced the terrain field used in the model could not be found, and there was probably some processing done when the file was written. However, it is believed the formulation is very close.

VARIABLES ACCOMMODATED BY U453

U150 calls U453 once per cycle. Furnished to U453 is the list of 4-word MOS-2000 variable identifiers for which processing is to be done by some module of U150. U453 has resident a table that contains the single variable ID that U453 is engineered to handle, 00341005. There is a corresponding (one element) table which gives the name of the '.CN' control file, 'U453.CN'.

U453 searches the table for each U150 input ID and processes those variables that reside in its table ITABLE( ). Only one is accommodated, and the associated fields needed are:

Variable				
Saturation Deficit:	003410005	0	0	0
500-mb heights from NCEP	001000008	500	t	0
1000-mb heights from NCEP	001000008	1000	t	0

The only variables normally written to the archive file, Unit No. KFILIO, are the SD forecasts for each hour t indicated in MAKEF( ) and provided the U150.CN file contains the SD variable with that projection t. They are written with the smoothing parameter S = 0 in the ID even though the forecasts may have been smoothed. This is consistent with the final analyses (e.g., see U400A) being written with S = 0 even though they may have been smoothed.

TDLPACK GRIDS FOR CHECKOUT OR QUALITY CONTROL

TDLPACK grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed below will be written to the dataset whose name has been provided to OUTDIS and Unit No. KFILOG from the control file 'U150.CN', Record Type 9, provided (1) KFILOG is not zero and (2) I450DG is not zero:

409020005	0	0	0	Terrain height in m
409020005	0	0	60	Terrain height in m smoothed over 25 1-Bedient points
409023005	0	0	0	Terrain height field in m <u>used in the model</u> after multiplicative factors and height adjustment has been applied
409240005	0	0	0	Full conservative quantity in m <u>used in the model</u>
001000008	500	t	0	NCEP 500-mb heights in m
001000005	500	t	60	500-mb heights in m smoothed over 25 1-Bedient points N50SM (see Record Type 1) times

001000008	1000	t	0	NCEP 1000-mb heights in m
001000005	1000	t	60	1000-mb heights in m smoothed over 25 1-Bedient points N00SM (see Record Type 1) times
004000005	500	t	60	U-advective winds in grid units per half hour
004100005	500	t	60	V-advective winds in grid units per half hour
003410005	0	t	0	Saturation deficit forecasts for hour t

The convention used is that when the data come directly from NCEP, except that it has been put on the current LAMP grid, the model number DD in the variable ID is NCEP's number (8 currently for the AVN), but after processing for LAMP, it is given DD = 5 for LAMP. Projection is represented by "t" above, and will occur for projections 0 through NPROJH (see Record Type 1).

GRIDPRINT FOR CHECKOUT OR QUALITY CONTROL

The same grids over the subset area (see Record Type 3 of U150 writeup) of the variables listed in the section above will be gridprinted and written to Unit No. IP(22) from the control file 'U150.CN', Record Type 1, provided (1) IP(22) is not zero and (2) I450DG is not zero. All are contoured at intervals of 100 meters, except the advective U and V winds which are first multiplied by 100 then contoured at units of 10.

The parameters SMULT( ), SADD( ), ORIGIN( ), and CINT( ) apply only to SD output.

SMOOTHING PROCEDURE

The smoothing parameter B(,) (see Record Type 2) is explained below:

$$S_{i,j} = \frac{A_{i,j} + b\bar{A}_{i,j}}{1 + b}$$

where:
$S_{i,j}$ = Smoothed gridpoint values at point i, j.
$A_{i,j}$ = Value of gridpoint at point i, j before smoothing.
b = Smoothing parameter.
$A_{i,j} = (1/4)\ (A_{i+1,j}+A_{i-1,j}+A_{i,j+1}+A_{i,j-1})$

Low values produce the least smoothing and higher values produce heavier smoothing. A value of zero indicates no smoothing.

NONSYSTEM ROUTINES USED

Use the load line in file 'lamp2000/dru150', dataset 'u150.com' on ice.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: lamp2000/u452lib. The driver is in lamp2000/dru150.

BIBLIOGRAPHY:

Glahn, H. R., 1980: Plans for the development of a Local AFOS MOS Program (LAMP). Preprints Eighth Conference on Weather Forecasting and Analysis, Denver, Amer. Meteor. Soc., 302-305.

_____, and D. A. Lowry, 1972: An operational Subsynoptic Advection Model (SAM). J. Appl. Meteor., 11, 578-585.

_____, _____, and G. W. Hollenbaugh, 1969: An operational subsynoptic advection model. ESSA Technical Memorandum WBTM TDL 23, Environmental Science Services Administration, U.S. Department of Commerce, 33 pp.

_____, and D. A. Unger, 1982: Some results from the Local AFOS MOS Program (LAMP). Preprints 9th Conference on Weather Forecasting and Analysis, Seattle, Amer. Meteor. Soc., 364-371.

_____, and _____, 1986: A Local AFOS MOS Program (LAMP) and its application to wind prediction. Mon. Wea. Rev., 114, 1313-1329.

Unger, D. A., 1983: An evaluation of the sea level pressure and moisture models of LAMP. Preprints , Sixth Conference on Numerical Weather Prediction, Omaha, Amer. Meteor. Soc., 252-258.2

U120

CONVERTS .HDF RADAR DATA TO TDLPACK

Mark A. Armstrong
Jerome P. Charba
September 17, 2001

PURPOSE: U120 is a FORTRAN executable program that processes raw radar reflectivity data obtained from WSI in HDF (Hierarchical Data Format) format and derives two new radar variables and writes them to a file in TDL format. The raw data consist of reflectivity levels (or categories) within latitude-longitude grid cells (approximately 2 km on a side) over the CONUS. The two archived radar variables are derived on an approximate 10-km polar stereographic grid (11.906 km mesh at the standard latitude of 60° N.) that spans the CONUS.

Prior to the specification and archiving of two derived radar variables [see “DATA OUTPUT” and “VARIABLE LIST” (below) for a description of these radar variables], pre-processing is performed on the raw data. The pre-processing consists of applying two types of diagnostic data that specify where the raw reflectivity values are changed to “missing” values. The first data type is a constant 2-km (gridded) occultation field that involves geographic coverage by the radar network and locations of topographic-obstruction contamination of the radar measurements. The second information type, which is included with each HDF file, is textual information that describes the status of real-time operations at each radar site.

The TDLPACK output grids can be written in one of two output modes controlled by the user. In one mode, which is used for developing an archive, a single file containing grids covering the CONUS is written. A second output mode option involves writing two files, with each file containing data for one-half of the CONUS grid. The latter option must be used for mapping the data with GEMPAK because of an upper limit restriction on the number of gridpoints in GEMPAK.

Control File Input: ‘RADAR.CN’

Record Type 1 - Format (I2,1X,I2) **Output Control**

IP(J) - Each value (J=1,2) assigns a unit number to a diagnostic output (print) file. The value for J=1 specifies the unit number for the primary print file (KFILDO). All error and diagnostic messages are output to this file. The value for J=2 is the print output unit number for the file containing occultation and radar site operational status information. Specifically, this file contains information from the constant occultation file (read internally in the code) and the real-time operational status information for each radar site read from the .hdf file.

Record Type 2 - Format (I1,1X,I1) **File Organization Control**

This record type assigns values to two variables that specify how the TDLPACK output files are organized, the number of output file names read, and the number of input files to be read.

IALL - 0 = output written as two half-grids (two files written). This option is used for mapping the data with GEMPAK.

1 = output is written as one grid. This option is used for developing an MDL radar data archive, as described in TDL ON 00-1.

IMON - The number of .hdf input files that are provided in Record Type 4.

Record Type 3 - Format (A60) or (A60,/,A60) **Gridpoint Output Files**

When IALL = 0 (see above), MEASTOUT (defined below) is assigned by the first line of this record type and MWESTOUT by the second line. When IALL = 1, MALLOUT is assigned by the first and only line of this record type.

MEASTOUT - Output file in TDLPACK format, containing data for the eastern half of the 10-km CONUS grid. (A60)

MWESTOUT - Output file in TDLPACK format, containing data for the western half of the 10-km CONUS grid. (A60)

MALLOUT - Output file in TDLPACK format, containing data for the entire grid. (A60)

Record Type 4 - Format (A60) **Gridpoint Input Files**

One or more records containing the .hdf input filenames to be processed, the number of which is specified by IMON.

HDFFIL - .hdf input filename, in the form U{hhmm}Z{yy}.{ddd}_refl_2km.hdf, where hh = hour, mm = minute, yy = last two digits of the year, and ddd = Julian day of the year. The HDFFIL files must be in this form and in the directory from which the program is run. (A25)

DATA INPUT:

All input data, comprised of raw radar reflectivity-level values, are obtained from WSI in HDF format. These .hdf input files must be located in the directory from which the program is being executed.

DATA OUTPUT:

The output archive files are comprised of gridded data records in TDLPACK format for the 10-km archive grid. For each .hdf input file, which applies to some 15-min time, two data records pertaining to the 10-km gridboxes are archived. The first record contains the maximum WSI (2-km)

radar reflectivity-level value and the second contains the fractional coverage of non-zero (2-km) reflectivity-level values.

COMMENTS:

U120 is comprised of several basic steps. (1) The control file is read, which defines the input and output files to process. (2) The radar-site status information is extracted from the .hdf input file being processed, and a temporary file containing this information is written. (3) Another temporary file containing the raw gridded radar data is created in binary form from the input .hdf file by an internally-defined program named hdftor8.old. (4) The information in the constant 2-km, gridded occultation file is read and combined with information in the radar site status file (from step 2) to form a 2-km grid of flags that indicates where the raw gridded reflectivity values are to be set to “missing.” (5) The flag array is applied to the raw reflectivity grid (obtained from the binary file created in step 3) to create the pre-processed reflectivity data on the 2-km grid. (6) The two radar reflectivity variables on the 10-km archive grid are derived and output in TDLPACK format.

The process described above from Step (2) on is repeated for each .hdf input file.

VARIABLE LIST:

The derived radar reflectivity fields written to the output file (with the four-word identifiers to the right) are as follows (“reflectivity” denotes reflectivity level and h denotes the two-digit hour):

Reflectivity: h + 0 min	707600004	0	0	0
Reflectivity: h + 15 min	707615004	0	0	0
Reflectivity: h + 30 min	707630004	0	0	0
Reflectivity: h + 45 min	707645004	0	0	0
Nonzero reflectivity coverage: h + 0 min	707700004	0	0	0
Nonzero reflectivity coverage: h + 15 min	707715004	0	0	0
Nonzero reflectivity coverage: h + 30 min	707730004	0	0	0
Nonzero reflectivity coverage: h + 45 min	707745004	0	0	0

LANGUAGE: FORTRAN-77 with an aliased C function fread (6.8 mb with each call to fread).

LOCATION: /users24/armstron/drurad/

U455

FORECASTS DISCONTINUOUS VARIABLES

Harry R. Glahn
September 24, 2001

PURPOSE: To produce advective forecasts of both continuous and discontinuous variables from 1 to a maximum of 25 hours on a variable-sized computational grid. The forecast model used is purely advective. Initialization is with nearest neighbor analyses produced by U400D or from a radar grid. LAMP model 1000-mb and NCEP model 1000-, 850-, 700-, and/or 500-mb fields can used as the advecting agents. Geostrophic winds are used, which are calculated from, usually highly, smoothed heights. The LAMP 1000-mb forecast is converted from the SLP forecast of U450.

U455 is called from U150 and is controlled by several control files 'U454XXX.CN', where XXX is specific to a variable (e.g., CIG = ceiling height), which is explained below. An understanding of the 'U150.CN' control file will be needed to understand certain variables mentioned below [e.g., IP(22)].

U455 runs the same way as U454. In U150, the control variable IU454 is set to 1 when U454 is to be run (no change), but set to 2 when U455 is to be run. The control files such as U454XXX.CN are used without a name change. So, the only change actually needed to switch from U454 to U455 is IU454 (still called by that name in U150--U150 is not changed).

The difference between U454 and U455 is that in U454 the values controlling the height of the winds to use for advection (CLOAF, C00AF, C85AF, C70AF, and C50AF) are absolutes; those values are used for that variable over the whole grid. In U455, the winds are kept at an approximate height above a highly smoothed terrain field. The height parameters (listed above) pertain to a height above sea level (1013 mb), and where the terrain is above that level, a height above that asked for is used. For this purpose, 850 mb is considered at 5000 ft, 700 mb at 10,000 ft, and 500 mb at 18,000 ft.

As an example, when C00AF is not zero and all other height parameters are, then below 5000 ft, a weighted combination of 1000 mb and 850 mb winds are used. At 5000 ft, only 850 mb winds are used. In a like manner, when the terrain is between 5000 and 10000 ft, a weighted combination of 850 and 700 mb winds are used. The smoothed heights over the West reach a height of slightly above 7000 ft.

When only C50AF is non zero, U454 and U455 give the same result; only 500 mb winds are used.

RESTRICTIONS:

Only two levels of winds can be used, and they must be contiguous. That is, 850 and 700 mb can be used, but not 850 and 500 mb. Since the LAMP and NCEP 1000 mb winds are blended for the first few hours, 1000 mb is

treated as one level. For instance, CL0AF, C00AF, and C85AF can be used, or just C00AF and C85AF. As with U454, if CL0AF is not zero, then C00AF must be non zero.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: users24/glahn/lamp/u455lib. The driver is in users24/glahn/lamp/dru150.

ugem.ksh

PROCESSES DATA REQUESTS FOR GEMPAK DISPLAY

Judy E. Ghirardelli
March 23, 2001

PURPOSE: ugem.ksh is a Korn shell script which acts as a driver interface to GEMPAK to produce GEMPAK images of any gridded or vector data contained in GEMPAK binary files. U203 can be used in conjunction with (prior to) ugem.ksh to produce the GEMPAK binary files from any TDLPACK file, or once U203 has been run, ugem.ksh can be run separately. This script will produce either X Window images on a PC running Exceed or workstation, gif images, postscript images, or Vector Graphics (N-AWIPS Product Generation). Other file types may be added as needed. There is an overlay capability where one can overlay gridded graphics with vector graphics. The possible graphical capabilities for gridded data are: contour plot, contour and color fill plot, color fill plot, plot grid values (of scalars or wind barbs) at the grid points with contours overlaid, inventory the GEMPAK binary grid file. The possible graphical capabilities for vector graphics are: plot station values (of scalars or wind barbs) with station ids, plot station values (of scalars or wind barbs) with markers on the stations location, plot station values (of scalars or wind barbs) with station locations blank, plot station values (of scalars or wind barbs) at the station locations, and inventory the GEMPAK binary vector file. New capabilities will be added as needed. Note that running ugem.ksh with no command line arguments tells the script to use as a control file a file called UGEM.CN which should be found in the local directory from which the user is running. However, if the user wishes to run with a control file named otherwise, the name of the control file can be supplied as the first and only command line argument to ugem.ksh (e.g., ugem.ksh UGEM_judy.CN). If the script is supplied a command line argument, it accepts that as the control file to use. If no command line argument is supplied, it uses UGEM.CN.

CONTROL FILE INPUT: 'UGEM.CN'

Record Type 1 - Format (A72) Binary Input Data Control

This record contains the name of the binary GEMPAK file from which GEMPAK will read the requested data. The actual file name should be all in lower case. Note of importance: the data in the GEMPAK binary file are uniquely identified ONLY via the date and the parm identifier of the data. Therefore you cannot have two grids, for example, in the same binary GEMPAK grid file that have both the same date and same parm identifier. The combination of those two must be unique for each field in the binary GEMPAK file.

binfil - Path and name of binary GEMPAK file from which to read the data. This can be either a file of gridded data or vector data. Gridded data and vector data cannot co-exist on the same binary file. Also, gridded data of different grid resolutions or area

definitions cannot co-exist on the same binary file. If the binary file of interest resides in the same directory as UGEM.CN, no path is necessary, and the code will look for the binary file in the current directory from which you are running ugem.ksh. The entire length of binfil cannot exceed 72 characters. U203 INTERACTION: If UGEM.CN is created directly from u203, this record is determined in u203 and written appropriately to UGEM.CN. DO NOT USE A BLANK OR A "+" FOR ONE OF THE CHARACTERS.

Record Type 2 - Format (A15,1X,A4,1X,I2,1X,A39,1X,F3.1,1X,A4,1X,I2,1X,A32,1X, F7.2,1X,A1) GEMPAK Control Inputs

This record contains the necessary inputs to control the GEMPAK routines.

date - Date of data in GEMPAK data file which you would like to plot. Format is YYMMDD/HHHHFTTT where YY is the two digit year, MM is the two digit month, DD is the two digit day, HHHH is the two digit hour * 100 (e.g.,l200, 0000). F indicates that this is a forecast time. The TTT which follows is the forecast projection in hours. If the data to be plotted is an analysis or an observation, and therefore the projection is 000, one can leave off the F000, and GEMPAK will know it is an analysis of an observation. This date is NOT the valid time. If one has data from the 12Z cycle of a model, and one wishes to plot the 6 hour projection, the date would be "991231/1200F006." This equates to a valid time of 991231/1800 (which GEMPAK would denote as 991231/1800V006 if one wished the date to show up in valid time format). This script assumes the date is either in forecast time format, or represents a 000 hour projection. U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is determined in u203 and written appropriately to UGEM.CN. (Columns 1-15).

parm - Four character name identifying the data field in the binary file which you wish to plot. This id name is set by the user in U203. This can be a meaningful name (TMPF), or otherwise (JUDY, JUNK). In order to plot derived fields, pre-defined parameter names must be used. For example:

BRBK = will plot wind barbs in knots, provided either SKNT (wind sped in knots) and DRCT (wind direction), or UKNT (u component of wind in knots) and VKNT (v component of wind in knots) appear in the binary file. If they appear in the binary file, and the parm of BRBK is specified in the UGEM.CN file, parms of SKNT, DRCT, VKNT, and UKNT do not need to appear in the UGEM.CN file. If the script sees the BRBK parm, it will know to look for either the U and V parms OR the speed and direction parms. Alternately, if the script sees either of the two components (U and V, or direction and speed), it will know to calculate BRBK for wind barbs.

The same is true for the wind parameters in units of m/s - BRBM, UWND, VWND, SPED.

See /lamp2000/drugem/gparm.txt (for gridded data) or sfparm.txt (for station data) for lists of pre-defined parms that GEMPAK will automatically recognize and derive from other fields if those fields are available in the binary GEMPAK file. U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 17-20)

gemcon - GEMPAK control number. This number instructs the script to call different GEMPAK routines to do the plotting of the field requested (by “date” and “parm”). Values 1-10 are for routines dealing with gridded data, and values 11-20 are for routines dealing with station data. gemcon values follow:

(01) = Contour plot of gridded data.
(02) = Contour and color fill plot of gridded data
(03) = Color fill (no contours) plot of gridded data
(04) = Plot wind barbs at grid points
(05) = Contour plot overlaid with grid points values
(06) = Plot grid point values
(07) = Not used
(08) = Not used
(09) = Inventory of a binary GEMPAK file of gridded data. Provides the following information: grid definition information, dates of the data in the file, parm identifiers of the data in the file, and sample grid values. Output goes to ftninv in the current directory.
(11) = Plot station scalar fields (station id plotted at station).
(12) = Plot station wind barbs (station id plotted at station)
(13) = Plot station scalar fields (station marker plotted at station).
(14) = Plot station wind barbs (station marker plotted at station)
(15) = Not used
(16) = Plot station wind barbs (no station id or marker plotted at station)
(17) = Plot station scalar fields at station
(18) = Not used
(19) = Inventory of a binary GEMPAK file of station data. Returns to file ftninv the following: Returns dates and SFPARM values (called “parm” in our script). No station information or data values are returned. Output goes to ftninv.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 22-23)

area - Map area over which to plot the data. This is adjustable “on the fly” if you select “device” below to be “XW” which gives you interactive options. If you select XW for “device”, this is the area over which GEMPAK will first plot (and then you can modify the area). If you select any other device, this is the final area that will be plotted on the hard copy of the plot.

This can be either a GEMPAK abbreviation (e.g., US, TX); an areal box determined by the supplied lower left latitude, lower left longitude, upper right latitude, upper right longitude; or an areal abbreviation interpreted by the script.

GEMPAK abbreviation: All the standard GEMPAK area state abbreviations are accepted (US, MD, TX, etc.). For a list of all abbreviations GEMPAK recognizes, see /users24/gempak1/gempak/tables/stns/geog.tbl. One can expand or reduce the standard area displayed by the GEMPAK abbreviation by appending a + or - to the area. “A suffixed + or * will decrease the extent of the geographic area. A suffixed - will increase the extent of the geographic area.” (GEMPAK documentation) For example, to show more than the standard GEMPAK defined TX area, set the area to TX-. To show less than the standard GEMPAK defined TX area, set the area to TX+. To show even more area, set the area to TX–-. To show even less, set the area to TX++. The state abbreviations are not always defined by GEMPAK to be what you expect, so this feature has limited usefulness. Note that you cannot successfully display the area AK (Alaska) on the north polar stereographic map orientation. The next option can be used to plot Alaska.

Latitude/Longitude box: If a user defined area is desired, the syntax is: lower left lat; lower left long; upper right lat; upper right lon (e.g., 6.987;-147.064;37.147;-16.848). Note that west longitude should be negative, however expressing west longitude as an east longitude is also acceptable (e.g., -100 longitude W is also 260 longitude east).

Special pre-defined areas the script recognizes: If you specify “area” as the following, the script will interpret them as follows for either gridded or station data:

```
ALL =   AVN GRID AREA  6.987;-147.064;37.147;-16.848
EAST=   EASTERN CONUS 29.900;-102.700;41.300;-67.000
WEST=   WESTERN CONUS 28.600;-121.400;49.400;-87.000
```

U203 INTERACTION: When plotting gridded data, the user can enter “ALL” for the area in U203.CN. U203 will calculate the latitude and longitude box coordinates which cover the entire grid, and copy those to UGEM.CN so that the user can easily view the entire grid without knowing *a priori* what the grid area is. This feature cannot work for station data in U203. If any other area other than ALL is entered in U203.CN, it will merely get copied to UGEM.CN. (Columns 25-63)

<u>dens</u> - This controls the density of the data to display. The possible values differ depending on the type of data you are plotting. They are:

For gridded data (equivalent to SKIP in GEMPAK):

0 = plot all grid points (no grid points skipped)
N = (where N is an integer greater than 0) will skip every N grid points. For example, “dens” set to 1 will skip every other grid point for plotting.

For station data (equivalent to FILTER in GEMPAK):

0 = plot all station values (no stations filtered out)
X = (where X is a real value between 0 and 1, exclusive) will allow for some data overlap. X values closer to 0 will have less filtering; values closer to 1 will have more filtering
N = (where N is an integer greater than or equal to 1) will plot station values more widely spaced apart

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 65-67)

<u>device</u> - This controls the type of GEMPAK plot produced. The available options are to display the plot on the screen as it plots it in an X window, or you can output the plot to an output file. Note that it is not recommended to produce black background images for printing purposes because the black background will use a great deal of printer toner. If you intend to print a plot it is advised to create an image with a white background. Note that if you request that the images be output to a file, they will be named certain names, as seen below. <u>If files with those names already exist in the current directory, they will be overwritten.</u> The options are:

XW = image will display in an X window, which the user can interact with via an interactive script driven menu. This provides a user-friendly interface for changing the display area, and getting a low resolution printout of what appears on the screen, but with a white background. In the future, the interactive menu will have more options so that one can also easily get a high resolution printout of the display also. The background color for any images displaying on the X window is black.
GF = each image will be saved as a gif file in the current directory with names gempak_1.gif, gempak_2.gif, etc. The background color will be black.
GFW= same as GF, except the image background will be white. The output naming convention is also the same as for GF.
PS1= each image will be saved as a postscript file, landscape orientation, monochrome color scheme with the following names: gempak_1.ps, gempak_2.ps, etc. The background color will be white.
PSC= each image will be saved as a postscript file, landscape orientation, 32 color scheme with the following names: gempak_1.ps, gempak_2.ps, etc. The background color will be white.

PSP= each image will be saved as a postscript file, landscape orientation, 20 grays scheme with the following names: gempak_1.ps, gempak_2.ps, etc. The background color will be white.

PS = each image will be saved as a postscript file with the following names: gempak_1.ps, gempak_2.ps, etc. The background color will be white. However, I am uncertain as to the color scheme this will use. The documentation is not clear, and this has not been investigated.

VG = each image will be saved as a vector graphics file (N-AWIPS Product Generation) with the following names: gempak_1.vgf, gempak_2.vgf, etc. This option has not yet been tested, and the background color has not been determined.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 69-72)

end_image - This value controls whether to overlay subsequent fields over the current field.

0 = this field will be plotted with the following requested field overlaid on it.

1 = this field will be plotted, and no subsequent fields will be overlaid on it.

GEMPAK will continue to overlay fields on top of other fields as long as end_image is 0 (up to a limit). As soon as the end_image value of 1 is encountered, the overlaying stops with that field. For example, if you wish to plot 4 fields, and the end_image values of those fields are: 1, 0, 1, 1, this script would plot the first field on an image, then it would plot the 2nd field overlaid on the 3rd image, and then it would plot the 4th image. You would have a total of 3 images created (one with two fields on it). If you were displaying to a gif file, for example, this would create 3 gif files.

For gridded overlays, you can overlay up to five fields. The color scheme follows (note that GFW and PSC will have white backgrounds, while XW and GF will have black backgrounds):

	GFW	PSC	XW, GF
Overlay 1	Black	Black	White
Overlay 2	Red	Red	Green
Overlay 3	Blue	Blue	Light Blue
Overlay 4	Green	Green	Yellow
Overlay 5	Magenta	Magenta	Pink

For station overlays, you can overlay up to four scalar fields if you are displaying the data around the station location.

	GFW	PSC	XW, GF
Value 1	Black	Black	White
Value 2	Red	Red	Green
Value 3	Blue	Blue	Light Blue
Value 4	Dark Green	Dark Green	Yellow

If you are displaying the data **at** the station location, up to three overlays are allowed.

	GFW	PSC	XW, GF
Value 1	Apricot	Apricot	Apricot
Value 2	Dark Green	Dark Green	Cyan
Value 3	Purple	Purple	Magenta

This latter option is useful for plotting regular observations vs. throw-outs vs. marginals from U150. If you overlay fields **at** the station location, the data from the different overlays should be mutually exclusive, or identical, or else you will get more than one value at the same location, and it will be illegible.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 74-75)

title - This is the title to be shown for the plot. The script will append the date and the parm of the data. The title will display at the bottom of the plot. For gridded data, one title per gridded field is displayed when overlaying, and each title color will match the color of the grid contours. For the gridded data, this assists the user in knowing which contours are for which fields. If station data were requested, the color of title does not relate to the station values, and only one title is shown describing the station data. The title must be in quotes.

NOTE: GEMPAK has special characters it can interpret as follows in the title. If you do not intend for them to be interpreted as below, avoid using them in the title:

^ = Display the date/time of the data in Forecast format (990131/1200F0012).
~ = Display the date/time of the data in Valid format (990201/0000V0012).
@ = Displays the vertical level in the title. We are not using vertical level.
_ = Displays the grid parm in the title.
$ = Displays the non-zero scaling factor. Currently this is not used by default.
# = Displays the grid point location.

The ugem.ksh script will append the Grid function (parm) and the Forecast format date/time to the user supplied title. For each gridded field requested, if more than one gridded field is requested, it will display (overlay) up to five different gridded fields, and will display the five distinct titles. However, for station data, it will list all the information in one title, and only the title from the last station plot overlay will be shown.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. U203 adds the quotes to the title as required by UGEM.CN. (Columns 77-108)

cint - This is the contour interval to use to contour gridded data. If station data is plotted, this field is not used.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 110-116)

scale - This is the value to scale the gridded data by. If non-zero, the gridded data are multiplied by 10**scale. This feature is not yet in use. When this is implemented, the data will be scaled by the scale factor, and the contour interval cint will also be multiplied by 10**scale. The absolute value of scale will be limited to be less than 5.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 118-124)

nts_flag - This is the flag which indicates whether an nts file is specified in the following record. 0 indicates that no nts file follows this record (in which case the next line represents either the next binary file to access for the next plot, or the terminator of “ 999999" (three spaces and six 9's)). 1 indicates that an nts file does follow this record.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 126-126)

Record Type 3 - Format (A72) GEMPAK nts file

This record is used to identify the GEMPAK file of settings, called an nts file. If the last field of the Record Type 2 is 1, the script will expect an nts file to be listed here. If it is 0, it will not.

nts - This is the path and filename of the nts file to restore. The filename should be in lower case. If no nts file is to be used, this record is not included, and the last field of Record Type 2 is 0. Useful examples of settings one might wish to set in an nts file can be found below.

U203 INTERACTION: If UGEM.CN is created directly from u203, this variable is merely copied from U203.CN to UGEM.CN. (Columns 1-72)

Record Type 4 - Format (A9) Terminator

Record types 1 through 2 (or 3 if an nts file is desired) are repeated until the terminator record is found. The terminator line consists of a

single line containing three spaces followed by six 9's (“ 999999"). If the terminator is not found either after the nts file line (Record Type 3), or after the GEMPAK control line (Record Type 2) which contains a “0" for the nts flag, the script assumes that more data is being requested for plotting, and it tries to read the next Record types 1 through 2 (or 3). No information found below the terminator line is processed.

U203 INTERACTION: Currently U203 does NOT place a terminator record at the end of the UGEM.CN file. If U203 is run to create the UGEM.CN file, the user must manually enter the terminator record at the end of the file.

nts FILE INPUT: (Name read from UGEM.CN)

Record Type 1 - GEMPAK parameter settings

When the user wishes to override a GEMPAK setting which is hard-coded in the ugem.ksh script or a subscript, they can override those settings by specifying the new setting in the nts file. The GEMPAK parameter name must appear in capitals in columns 1-8. The setting you wish to set the parameter to must begin in column 9 (e.g., COLORS 5). This file has one line per parameter you wish to override.

Useful settings (you can use none, one, or any combination of these):

(1) COLORS Overrides the scripts colors. Color options are:

1	VANILLA (for X Window and gif)
1	BLACK (for postscript)
2	RED
3	GREEN
4	BLUE
5	YELLOW
6	CYAN
7	MAGENTA
8	BROWN
9	CORAL
10	APRICOT
11	PINK
12	DKPINK
13	MDVIOLET
14	MAROON
15	FIREBRICK
16	ORRED
17	ORANGE
18	DKORANGE
19	GOLD
20	DKYELLOW
21	LWNGREEN
22	MDGREEN
23	DKGREEN
24	GRPBLUE
25	LTBLUE
26	SKY
27	MDCYAN
28	VIOLET

29	PURPLE
30	PLUM
31	VANILLA (for postscript)
31	WHITE (for X Window and gif)
32	WHITE (for postscript)
32	BLACK (for X Window and gif)

Example: If you are plotting grid contours, and wish to change the default line color for a particular field, you could 1) change the nts_flag in Record Type 2 to a “1", 2) add a Record Type 3 below Record Type 2 which listed an nts file path and filename, and 3) edit this nts file to have in it:

```
COLORS   5
```

This will change the line color to yellow.
The COLORS option is valid for any kind of plotting (any GEMCON value).

(2) TITLE Overrides script title. Format is:

```
title color / title line location / title string | short title
```

Title color: see above description of COLORS for color information.

Title line location: specifies the line on which the title will be written. The value of the title line has the following meanings:

0	bottom line
-n	n lines from bottom
+n	n lines from top

Title string: the title to be written. This does not have to be surrounded by quotes, as is the title in the UGEM.CN. See the description of title in Record Type 2 above for built-in GEMPAK codes which are recognized in title.

Example:

```
TITLE    5/-2/This is a test
```

This will print the title in YELLOW, 2 lines from the bottom. The title will say: This is a test.

(3) CINT Overrides the contour interval for grid contouring. CINT can be any real number.

Example:

```
CINT    10
```

This will change the contour interval to contour the field every 10.

Minimum and Maximum contour values can also be specified if desired. Interested parties should read /users24/gempak1/gempak/help/cint.hl2.

"A list of two or more contour levels may be entered using semicolons to separate the individual values." (GEMPAK documentation) Example:

```
CINT    2;4;8
```

This will cause GEMPAK to drawn the contours of 2, 4, and 8.

(4) SCALE Overrides the scaling used for plotting grid point values and contouring grids. Format:

scalar value / vector value

where the scalar value is used to scale the scalar fields, and the vector value is used to scale the vector fields (i.e., wind barbs). We have used only the scalar value for scaling so far in our nts file use for ugem.ksh. If SCALE is set, the data are multiplied by 10**SCALE (e.g., a SCALE of 1 will cause the data to be multiplied for plotting purposes by 10). Note that if the data are scaled, the contour interval (CINT) used will be relative to the new scaled data. To maintain the same contouring you had before scaling, you should multiply the CINT by 10**SCALE also. Therefore, if you had original values contoured every 4, and you now wish to plot the values to the nearest tenth place, and also contour those values, you should do the following in your nts file:

```
SCALE   1
CINT    40
```

<u>Eventually</u>, the field of "scale" in Record Type 2 will be used and this will not have to be overwritten via the nts file. The code will automatically read the field "scale" from Record Type 2, use it as the SCALE, and multiply CINT by 10**scale.

(5) CONTUR Overrides the smoothing used for contouring. Format (from the CONTUR help documentation):

subbox factor / number of smoothing passes

"Each grid box will be subdivided into the number of subboxes specified by the subbox factor. Increasing this value will produce smoother contours, but will increase contouring times and make metafiles larger. The new default value in GEMPAK 5.2 is 0.

A simple three-point smoothing filter is used if the number of smoothing passes is greater than 0. The default is 0."

ugem.ksh uses a CONTUR value of 2. Our previous release of GEMPAK defaulted to a CONTUR value of 7. A CONTUR value of 0 has proven to provide jagged contours. A CONTUR value of 0 produces the same result as a CONTUR value of 1. This parameter is not typically overwritten, but this is how one would do it. Example:

CONTUR 0

(6) FINT Overrides the color fill interval for gridded data (gemcon of 02 or 03). FINT can be any real number. Example:

FINT 4

"A list of two or more fill levels may be entered using semicolons to separate the individual values." (GEMPAK documentation) Example:

FINT 2;4;6;8

This will cause the color fill contours to be 2, 4, 6, and 8.

If gemcon is set to 02 or 03 in UGEM.CN, the FINT which the script uses is the same as cint from Record Type 2. If one wishes FINT to be different from CINT, an nts file must be used.

(7) FLINE Overrides the colors used in the color fill of gridded data (gemcon of 02 or 03). From the GEMPAK documentation:

"FLINE is the list of colors to be used for contour fill. The number of fill colors needed is one greater than the number of fill levels in FINT. The number may be entered as a list of color numbers separated by semicolons or a range of colors. More information on color selection can be found in the help for COLORS."

If gemcon is set to 02 or 03 in UGEM.CN, the FLINE which the script uses is:

FLINE = 31;30;29=242:62:149;11;18=224:227:219;27-22;3;19;17;16;2;13

This produces roughly a rainbow color scheme where the lowest values are white, to pink, to blues, to greens, to yellow, to orange, to red. If your data and FINT indicate only two contours to fill with color, they will be the lowest two above (white and plum). If the data and FINT indicate more contours to fill than there are FLINE parameters for, it will repeat them.

(8) LATLON Overrides the default setting to NOT print out latitude and longitude lines. To plot lat and lon lines, in the nts file set LATLON to any of the colors 1-32 listed above. Example:

LATLON 5 Will plot latitude and longitude lines in yellow, every 10 degrees. To change the line type, and frequency of lines, see the documentation at /users24/gempak1/gempak/help/latlon.hl2.

(9) LINE Overrides the default line color and type for contour plotting. Example:

LINE 5/2 will use lines that are yellow, and short dashed, for contours. For more information see /users24/gempak1/gempak/help/line.hl2.

(10) MAP Overrides the default map color for plotting. Example:

MAP 5 will plot the map lines as yellow.

(11) MARKER Overrides the default marker color, type, thickness, etc. for the station location. See /users24/gempak1/gempak/help/marker.hl2 for more information on the options available.

(12) TEXT Overrides the default text size and font. The script uses a TEXT setting of 0.7/22/hw for contouring gridded data, and 0.6/22/hw for station plots. The first value controls the size, where the smaller number indicates a smaller size text font. Example:

TEXT 0.9/22/hw

For more information, see /users24/gempak1/gempak/help/text.hl2

DATA INPUT:

All data input can be either gridded data or station data which have been put into GEMPAK's binary file format. U203 takes any data from TDLPACK format and converts it to GEMPAK's binary file format. Those files can then be input into ugem.ksh for plotting purposes.

DATA OUTPUT

There are various forms of data output, depending on what the user has requested in the UGEM.CN control file.

A. Diagnostics - to screen

ugem.ksh will output to the screen (standard output) basic information to let the user know at what step in the process the script is at. Diagnostic errors will also appear on the screen preceded by four asterisks (****) as in the MOS system.

B. Diagnostics - to file ftngem

ugem.ksh will output GEMPAK run stream output to a local file called ftngem. Usually it is unnecessary to look at this file. However, there are times when GEMPAK will produce an error, but the script will not be able to detect that it had an error, and indicate the code finished successfully when it did not. To avoid confusion this can cause, at the end of the ugem.ksh script, the ftngem file is searched for GEMPAK errors, which begin with a [and end with a]. If the script finds such a GEMPAK error, it will output it to the screen. At this point it might be necessary to investigate the ftngem file. Also if the script is not doing what one expects it to do, the ftngem file may provide clues as to what is going wrong.

C. GEMPAK binary file inventory results

If the user indicated an inventory was desired (gemcon of 09 for gridded data, 19 for station data), the inventory is output to a file called ftninv. If an inventory of both a gridded binary file and a station binary file is desired (gemcons of 09 and 19) both inventories will be output to the ftninv file. An inventory of gridded data will indicate the following information:

- o Name of GEMPAK binary file being inventoried
- o Number of grids in gridfile
- o First time and last time of data in the file
- o For each grid found in the grid file, the following information will be output to the ftninv file:

  - grid number location in grid file
  - GDATTIM which is the time of the data
  - LEVEL which is the level of the data (we do not note the level, so this should be 0 -1)
  - IVCOORD which is the vertical coordinate system used. We use 0 for surface
  - GFUNC which is equivalent to the “parm” above
  - NX which is the number of grid points in the x direction

  - NY which is the number of grid points in the y direction
  - First 10 values in grid
  - Last 10 values in grid

An inventory of station data will indicate the following information:

- o Name of GEMPAK station file being inventoried
- o Number of “parm” fields in the file
- o A list of the parm fields in the file, specifying what the parm is for each field
- o Number of date/times in the station file
- o List of date/times in the station file

Note that if the user requests more than one inventory per ugem.ksh run, all inventories will be output to the same ftninv file. If an ftninv file already exists, it will be overwritten.

D. Images - X Window output

If the user specifies the device of XW in UGEM.CN, the fields requested will be displayed to an X Window. In this case, the user will be given options to interact with the X Window display. Once the fields are plotted for that image (all fields processed until an “end_image” of 1 is encountered), the following interactive menu will display on the standard output from which you invoked the ugem.ksh script:

XWindow interactive options:

1) resize to a new area.
2) resize to the original area.
3) send hardcopy of window to printer, and then continue.
4) continue.

5) exit.
Enter number 1 through 5 now >>

These options are explained below:

1 This allows the user to change the area being displayed on the X Window. If selected, another menu is displayed:

Zooming is desired. Zooming options:
1) specify area via latitude and longitude, or standard abbreviation.
2) user cursor to grab new area.
Enter number 1 through 2 now >>

If Option 1 is selected, the user will see:

Enter new area (MD, MD+, MD++, FL-, FL--, lat1;lon1;lat2;lon2)
>>

Enter the new area, and hit enter. The field requested will now display over the new area specified. See the description of “area” in Record Type 2 for more information about acceptable areas to enter.

If Option 2 is selected, the user will see:

Move cursor over map. When cursor becomes crosshairs, left click and drag

Do as instructed, and the field requested will now display over the new area you have just “boxed” off with the cursor.

After the new area displays, the interactive main menu will display again to the standard output.

2 This allows the user to change the area being displayed on the X Window back to the original area specified in UGEM.CN. This is useful if you have zoomed in to a very small area, and would like to zoom out again. The field requested will now display over the original area specified. After the new area displays, the interactive main menu will display again to the standard output.

3 This allows the user to get a hardcopy of what is on the screen. The current option produces a low resolution black and white copy which is sent to the printer near Matt Sardi’s desk (hp4000). In the future, there will be an option to produce a higher resolution hard copy from the screen, and options to send this to other printers.

When a user selects 3, they will see :

Hardcopy output desired.
Wait until the cursor turns to crosshairs on GEMPAK window.
Then click on window. Printout will go to printer by Sardi.

request id is hp4000-909 (standard input) (for example)

However, because of the nature of this screen dump, you must first make sure that the X Window is unobscured by other windows before selecting 3 or else you will also get those other overlapping windows in your hard copy of the window.

4 This allows the user to continue on to the next field requested for plotting. If all the requested fields have been plotted, the script will now end.

5 This allows the user to gracefully exit the script.

E. Images - Image file output

If the user specifies the device of GF, GFW, PSC, or VG, the requested fields will be plotted to a file of that type (gif, ps, or vgf) and output in the current directory. See the description of “device” for Record Type 2 for more information.

F. Miscellaneous files

Whenever GEMPAK is run, it will create a file called gemglb.nts and one called last.nts. These files record the last settings used in GEMPAK. At the beginning of every ugem.ksh run, they are deleted, and then made again. Usually it is unnecessary to look at this file. However, when problems arise, or if the user wishes to later run GEMPAK manually, these files can be useful.

EXAMPLE CONTROL FILE: 'UGEM.CN'

An example does not currently exist, but could be made if that is needed.

RESTRICTIONS

Currently, U203 does not preserve the order of the fields requested for plotting in U203.CN. For instance, if you would like grids overlaid with station information, and then more grids overlaid with other station data, this order may not be reflected in the UGEM.CN which u203 creates. If no overlays are desired, this does not affect anything other than the order in which single (non-overlaid) fields are displayed. However, if overlaying is desired, the non-preservation of the order will have an effect since ugem.ksh is dependent on the order of the “end_image” values being in the order the user intended them to be in. Therefore, until this is corrected, the user may have to re-order the UGEM.CN file to reflect the order they desire.

In U203.CN, if the user wishes, for example, temperature grids overlaid with temperature observations on one plot, and then temperature grids overlaid with dewpoint observations on another plot (for whatever odd reason), then U203 believes that the second temperature grid request is a duplicate (I believe). It will not process the data twice, and it will not write the data out to the same binary GEMPAK file twice, which is fine. However, it also will not create an entry for the duplicate in the UGEM.CN file. Without that entry in the UGEM.CN file, the ugem.ksh will

not know that the temperature field is being requested for the second plot. I.e., if the user is requesting duplicate fields to be plotted via U203, they must manually alter the UGEM.CN file to reflect that.

GEMPAK does unpredictable things with file names of upper case. Please use all lower case for specifying both file names and paths to avoid possible problems.

Wind barb plotting:

Set “parm” in Record Type 2 to BRBK for wind barbs in knots (or BRBM for m/s). This will plot wind barbs in knots, provided either SKNT (wind sped in knots) and DRCT (wind direction), or UKNT (u component of wind in knots) and VKNT (v component of wind in knots) appear in the binary file. If they appear in the binary file, and the parm of BRBK is specified in the UGEM.CN file, parms of SKNT, DRCT, VKNT, and UKNT do not need to appear in the UGEM.CN file. If the script sees the BRBK parm, it will know to look for either the U and V parms OR the speed and direction parms. (The same is true for the wind parameters in units of m/s - BRBM, UWND, VWND, SPED).

Another way to plot wind barbs is to just specify either the UKNT and VKNT, or the DRCT and SKNT. If the ugem.ksh script sees these pairs of parms, it automatically sets parm internally to “BRBK”, which tells GEMPAK to calculate the wind barbs. In this case, the title and nts_flag (and associated nts file) from the VKNT or SKNT record are used. The title and nts_flag from either the UKNT or DRCT are ignored.

One limitation is that wind is defined either by the U and V components, or by the direction and speed, and the fields have required “parm” identifiers that must be used in order to calculate wind barbs from these components (see above paragraph). This introduces a limitation because you cannot have two different kinds of wind in the same GEMPAK binary file for the same date. For example, if you had quality controlled winds and put them in the binary GEMPAK file with parms of UKNT and VKNT, and then had unacceptable winds (throw-outs) and tried to put them also in the same binary GEMPAK file with parms of UKNT and VKNT, GEMPAK would not be able to distinguish these winds from the first winds because the parms would not be unique. Yet the parms must be these if GEMPAK can recognize these fields are needed to compute wind barbs. If you identified the first as UKNT and VKNT, and the second as DRCT and SKNT, it would still be a problem because these parms would both be describing winds between which GEMPAK could not distinguish. The only work around for this is to run U203 for as many different kinds of winds as you wish to plot. <u>If the winds are identified by the same date and parms, but are located in separate GEMPAK binary files, this is not a problem</u> because the user can tell ugem.ksh via UGEM.CN which winds to get from which binary files.

COMMENTS

None.

WRITING NEW COMPUTATIONAL SUBROUTINES

New ‘subscript’ routines can be written as needed. The existing routines, such as gridcontour.ksh, can be used as templates.

NONSYSTEM ROUTINES USED

g_inv, v_inv, gridcontour.ksh, stationplot,ksh

LANGUAGE: Korn shell, which invokes other korn shells, GEMPAK executables, and some FORTRAN 77 code (for the GEMPAK binary file inventories).

LOCATION: ugem.ksh. The script is in /lamp2000/drugem/.

CUT

CUTS A GRID

Harry R. Glahn
December 1, 2000

PURPOSE: To accept a grid of data in an array and to put a portion of that grid into another array.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL CUT(KFILDO,XIN,NXIN,NYIN,NXPIN,NYPIN,XOUT,NXOUT,NYOUT,NXPOUT,NYPOUT)
```

KFILDO - Unit number of output (print) file. (INPUT)

XIN(IX,JY)- The input array (IX=1,NXIN) (JY=1,NYIN). (INPUT)

NXIN - The first dimension of XIN(,). (INPUT)

NYIN - The second dimension of XIN(,). (INPUT)

NXPIN - The north pole IX-position in gridpoint units of the grid in XIN(,). (INPUT)

NYPIN - The north pole JY-position in gridpoint units of the grid in XIN(,). (INPUT)

XOUT(IX,JY)- The output array (IX=1,NXOUT) (JY=1,NYOUT). (OUTPUT)

NXOUT - The first dimension of XOUT(,). (INPUT)

NYOUT - The second dimension of XOUT(,). (INPUT)

NXPOUT - The north pole IX-position in gridpoint units of the grid in XOUT(,). (INPUT)

NYPOUT - The north pole JY-position in gridpoint units of the grid in XOUT(,). (INPUT)

EXAMPLE

```
DATA KFILDO/12/
...
DIMENSION XIN(13,14),XOUT(11,10)

CALL CUT(KFILDO,XIN,13,14,40,50,XOUT,11,10,42,54)
```

The data in XIN(13,14) is cut to size XOUT(11,10) in such a way that the (1,1) position of XIN(,) and XOUT(,) are the same.

OUTPUT:

A diagnostic message will be written to Unit No. KFILDO if the output grid is not within the input grid.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

DENSER

MAKES A GRID DENSER BY LINEAR INTERPOLATION

Harry R. Glahn
December 1, 2000

PURPOSE: To accept a grid of data in an array and to insert by bi-linear interpolation a point between each original point and to return that resulting grid. The boundaries of the grid are not changed. The dimensions of the input grid must be odd, as must be the dimensions of the output grid.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL DENSER(KFILDO,P,NX,NY,ND2X3)
```

KFILDO - Unit number of output (print) file. (INPUT)

P(IX,JY) - The input and output grids (IX=1,NX) (JY=1,NY). (INPUT/OUTPUT)

NX - On input, the first dimension of the input grid P(,). Also, the first dimension of P(,) on output. (INPUT/OUTPUT)

NY - On input, the second dimension of the input grid P(,). Also, the second dimension of P(,) on output. (INPUT/OUTPUT)

ND2X3 - The overall dimension of P( ), which must be $\geq$ NX*NY. (INPUT)

EXAMPLE

```
    DIMENSION P(ND2X3)

    DATA KFILDO/12/
    DATA NX/13/,
   1     NY/15/
    ...

    CALL DENSR(KFILDO,P,NX,NY,725)
```

The data in P(13,15) are interpolated to a grid P(25,29).

OUTPUT:

A diagnostic message will be written to Unit No. KFILDO if ND2X3 is not large enough. This latter test is made because P( ) may be singly dimensioned ND2X3 in the calling program, and a NX*NY grid is to be returned.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

DENSRQ

MAKES A GRID DENSER BY QUADRATIC INTERPOLATION

Harry R. Glahn
December 1, 2000

PURPOSE: To accept a grid of data in an array and to insert by bi-quadratic interpolation a point between each original point and to return that resulting grid. The boundaries of the grid are not changed. The dimensions of the input grid must be odd, as must be the dimensions of the output grid.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL DENSRQ(KFILDO,P,NX,NY,ND2X3)
```

KFILDO - Unit number of output (print) file. (INPUT)

P(IX,JY) - The input and output grids (IX=1,NX) (JY=1,NY). (INPUT/OUTPUT)

NX - On input, the first dimension of the input grid P(,). Also, the first dimension of P(,) on output. (INPUT/OUTPUT)

NY - On input, the second dimension of the input grid P(,). Also, the first dimension of P(,) on output. (INPUT/OUTPUT)

ND2X3 - The overall dimension of P( ), which must be $\geq$ NX*NY. (INPUT)

EXAMPLE

```
      DIMENSION P(ND2X3)

      DATA KFILDO/12/
      DATA NX/13/,
     1     NY/15/
      ...

      CALL CUT(KFILDO,P,NX,NY,725)
```

The data in P(13,15) are interpolated to a grid P(25,29).

OUTPUT:

A diagnostic message will be written to Unit No. KFILDO if ND2X3 is not large enough. This latter test is made because P( ) may be singly dimensioned ND2X3 in the calling program, and a NX*NY grid is to be returned.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

CHKGRD

CHECKS GRID CHARACTERISTICS

Harry R. Glahn
December 1, 2000

PURPOSE: To check the map projection, orientation, and latitude of mesh length specification for a grid whose characteristics are in IS2( ). The mesh length is expected to be in increments of Bedients divided by some power of 2 (range = 381 to 1.48783125 km).

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL CHKGRD(KFILDO,ID,NPROJ,ORIENT,XLAT,IS2,ND7,IER)
```

KFILDO - Unit number of output (print) file. (INPUT)

ID(J) - 4-word ID of the variable being checked (J=1,4). Used for diagnostic print only. (INPUT)

NPROJ - The number of the map projection being used. (INPUT)

ORIENT - Orientation in degrees west longitude. This is the longitude of the grid column which is oriented exactly north-south. (INPUT)

XLAT - The latitude at which the grid length applies. (INPUT)

IS2(J) - The TDLPACK Section 2 (J=1,12). (INPUT)

ND7 - Dimension of IS2( ). (INPUT)

IER - Error code). (OUTPUT)
0 = good return.
199 = grid did not check.

EXAMPLE

```
DATA KFILDO/12/
...
DIMENSION IS2(54)

CALL CHKGRD(KFILDO,ID,5,120,60,IS2,54,IER)
```

The grid characteristics in IS2( ) are checked with the map projection (5), the orientation (120) and the latitude of mesh length specification (60). The mesh length is checked for one of the allowable values, which is 381 successively divided by 2.

OUTPUT:

A diagnostic message will be written to Unit No. KFILDO if values in IS2( ) do not check with what is expected.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

CLOS

FINDS TWO CLOSEST NEIGHBORS TO A STATION LOCATION

Harry R. Glahn
December 1, 2000

PURPOSE: To find the two closest neighbors to a given station location.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL CLOS(XP,YP,LTAG,NSTA,K,LS1,LS2)
```

XP(K) - IX-position of station on a grid (K=1,NSTA). (INPUT)

YP(K) - JX-position of station on a grid (K=1,NSTA). (INPUT)

LTAG(K) - Don't use this station if LTAG(K) > 0 (K=1,NSTA). (INPUT)

NSTA - Number of values in XP( ), YP( ), and LTAG( ). (INPUT)

K - Location of station in XP( ), YP( ), and LTAG( ) for which the closest stations are desired. (INPUT)

LS1 - Position in XP( ), YP( ), and LTAG( ) of the closest station to station K. (OUTPUT)

LS2 - Position in XP( ), YP( ), and LTAG( ) of the second closest station to station K. (OUTPUT)

EXAMPLE

```
DIMENSION XP(100),YP(100,LTAG(100)

CALL CLOS(XP,YP,LTAG,100,K,LS1,LS2)
```

The locations in the lists XP( ), YP( ), and LTAG( ) of the closest station and second closest station to station K are returned in LS1 and LS2, respectively.

OUTPUT:

None.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

FLTAG

SETS LTAG TO INDICATE USE OF DATA

Harry R. Glahn
December 1, 2000

PURPOSE: To set LTAG( ) to 0 if datum for station K can be used in the analysis, or a non-zero value if the station is outside the grid.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL FLTAG(KFILDO,DATA,XP,YP,LTAG,NSTA,NX,NY,RMAX)
```

KFILDO - Unit number of output (print) file. (INPUT)

DATA(K) - Data for stations (K=1,NSTA). (INPUT)

XP(K) - IX-position of station on a grid (K=1,NSTA). (INPUT)

YP(K) - JX-position of station on a grid (K=1,NSTA). (INPUT)

LTAG(K) - Indicates the use of the data in DATA( ).
0 = use data.
1 = Station is outside radius of influence for the area being analyzed, or missing datum.
2 = Station location unknown
(OUTPUT)

NX - Number of gridpoints in the IX direction. (INPUT)

NY - Number of gridpoints in the JX direction. (INPUT)

RMAX - The maximum distance outside the grid in terms of gridlengths for which the datum outside the grid will be used. Note that the actual distance depends on the gridlength. (INPUT)

EXAMPLE

```
DIMENSION XP(100),YP(100,LTAG(100),DATA(100)

CALL FLTAG(KFILDO,DATA,XP,YP,LTAG,100,40,50,8)
```

The locations in LTAG( ) will be set to the values indicated above based on the input data. A station will not be used if it is outside the grid by > 8 gridlengths.

OUTPUT:

None.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

NEWXY

CALCULATES GRID POSITIONS

Harry R. Glahn
December 1, 2000

PURPOSE: To calculate new IX and JY positions in XP( ) and YP( ), respectively, given initial positions in XPIN( ) and YPIN( ) and the grid lengths associated with XP( ), YP( ) and with XPIN( ) and YPIN( ).

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL NEWXY(KFILDO,MESHIN,XPIN,YPIN,MESHOUT,XP,YP,NSTA)
```

KFILDO - Unit number of output (print) file. (INPUT)

MESHIN - Mesh length for the input locations in XPIN( ) and YPIN( ). (INPUT)

XPIN(K) - IX-position of station on a grid with a gridlength MESHIN (K=1,NSTA). (INPUT)

YPIN(K) - JY-position of station on a grid with a gridlength MESHIN (K=1,NSTA). (INPUT)

MESHOUT - Mesh length for the output locations in XP( ) and YP( ). (INPUT)

XP(K) - IX-position of station on a grid with a gridlength of MESHOUT (K=1,NSTA). (INPUT)

YP(K) - JY-position of station on a grid with a gridlength of MESHOUT (K=1,NSTA). (INPUT)

NSTA - Number of values in XPIN( ), YPIN( ), XP( ), and YP( ). (INPUT)

EXAMPLE

```
DIMENSION XPIN(100,YPIN(100),XP(100),YP(100

DATA KFILDO/12/

CALL NEWXY(KFILDO,MESHIN,XPIN,YPIN,MESHOUT,XP,YP,50)
```

The 50 values in the variables XPIN( ) and YPIN( ) with a mesh length MESHIN are transformed into XP( ) and YP( ), respectively, with a mesh length of MESHOUT.

OUTPUT:

None.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

SIZEGR

THINS OR THICKENS A GRID OVER THE SAME AREA

Harry R. Glahn
December 1, 2000

PURPOSE: To interpolate (thicken) or thin a grid so that it fits within the same area, but with a different mesh length.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL SIZEGR(KFILDO,P,NX,NY,NXP,NYP,MESHIN,MESHOUT,ITRPLQ,ND2X3)
```

KFILDO - Unit number of output (print) file. (INPUT)

P(IX,JY) - The input array (IX=1,NX) (JY=1,NY). Also, the output array (IX=1,NX) (JY=1,NY), where NX and NY are adjusted. (INPUT/OUTPUT)

NX - On input, the first dimension of the input grid P(,) on input. Also, on output, the first dimension of the output grid P(,). (INPUT/OUTPUT)

NY - On input, the second dimension of the input grid P(,) on input. Also, on output, the first dimension of the output grid P(,). (INPUT/OUTPUT)

NXP - The north pole position of the input grid in the IX direction on input. Also, the north pole position of the output grid in the IX direction on output. (INPUT/OUTPUT)

NYP - The north pole position of the input grid in the JY direction. Also, the north pole position of the output grid in the JY direction on output. (INPUT/OUTPUT)

ITRPLQ - Type of interpolation to go from one mesh length to one of half that:
1 = bilinear.
2 = biquadratic.
Not used when the grid is to be thinned (MESHIN $\leq$ MESHOUT).
(INPUT)

ND2X3 - The overall dimension of P( ), which must be $\geq$ NX*NY on both input and output. (INPUT)

EXAMPLE

```
      DIMENSION P(ND2X3)

      DATA KFILDO/12/
      DATA NXIN/13/,
     1     NYIN/15/,
     2     NXP/40/
     3     NYP/50/
      DATA MESHIN/80/,
```

```
1     MESHOUT/40/
2     ND2X3/195/
 ...

 CALL CUT(KFILDO,P,NX,NY,NXP,NYP,MESHIN,MESHOUT,1,ND2X3)
```

The data in P(13,15) are bilinearly interpolated to a grid P(25,29). The pole positions are adjusted accordingly. NX and NY are returned as 25 and 29, respectively.

OUTPUT:

Diagnostic messages, especially with /D option. Other diagnostics from called subroutines.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

DENSER, THINER, MXHXMS.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

THINER

THINS A GRID BY ELIMINATING ALTERNATE GRIDPOINTS

Harry R. Glahn
December 1, 2000

PURPOSE: To accept a grid of data in an array and to insert by bi-linear interpolation a point between each original point and to return that resulting grid. The boundaries of the grid are not changed. The dimensions of the input grid must be odd, as must be the dimensions of the output grid.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL THINER(KFILDO,P,NX,NY)
```

KFILDO - Unit number of output (print) file. (INPUT)

P(IX,JY) - The input and output grids (IX=1,NX) (JY=1,NY). (INPUT/OUTPUT)

NX - On input, the first dimension of the input grid P(,). Also, the first dimension of P(,) on output. (INPUT/OUTPUT)

NY - On input, the second dimension of the input grid P(,). Also, the first dimension of P(,) on output. (INPUT/OUTPUT)

EXAMPLE

```
      PARAMETER (ND2X3=13*17)

      DIMENSION P(ND2X3)

      DATA KFILDO/12/
      DATA NX/13/,
     1     NY/17/
      ...

      CALL CUT(KFILDO,P,NX,NY)
```

The data in P(13,17) are thinned to a grid P(7,9).

OUTPUT:

A diagnostic message will be written to Unit No. KFILDO if NX or NY is not odd.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

SMOTH

SMOOTHS A GRID

Harry R. Glahn
December 1, 2000

PURPOSE: To accept a grid of data in an array, to smooth it, and furnish it back in the same array.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL SMOTH(P,WORK,NX,NY,BQ)
```

P(IX,JY) - The input and output grids (IX=1,NX) (JY=1,NY). (INPUT/OUTPUT)

WORK(IX,JY) - Work array (IX=1,NX) (JY=1,NY). (INTERNAL)

NX - On input, the first dimension of the input grid P(,). Also, the second dimension of P(,) on output. WORK(,) must have dimensions $\geq$ NX*NY. (INPUT)

NY - On input, the second dimension of the input grid P(,). Also, the first dimension of P(,) on output. (INPUT)

BQ - The smoothing parameter. (INPUT)

EXAMPLE

```
DIMENSION P(500),WORK(500)

DATA KFILDO/12/
DATA NX/13/,
1     NY/15/
DATA BQ/1./
...

CALL SMOTH(P,WORK,NX,NY,BQ)
```

The data in P(13,15) are smoothed.

OUTPUT:

None.

RESTRICTIONS:

None other than stated above.

COMMENTS:

The smoothed value at point P equals the original value plus BQ/4 times the sum of the surrounding 4 points, all divided by BQ+1. A modified procedure is used for the outer rows and columns that assumes (not strictly for the 4 corner points) a linear extrapolation from the row

(column) just inside the outer row (column) to the row (column) just outside the outer row (column).

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

TRNSFR

TRANSFERS DATA FROM ONE ARRAY TO ANOTHER

Harry R. Glahn
December 1, 2000

PURPOSE: To transfer data from one array to another.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL TRNSFR(WORK,P,ND)
```

WORK(J) - The input array (J=1,ND) (INPUT)

P(J) - Output array (J=1,ND). (OUTPUT)

ND - The number of values in WORK( ) to transfer to P( ). (INPUT)

EXAMPLE

```
DIMENSION P(500),WORK(500)
...

CALL TRNSFR(WORK,P,400)
```

400 values from WORK( ) will be put into P( ).

OUTPUT:

None.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

XMSMSH

FINDS NOMINAL MESH LENGTH FROM ACTUAL MESH LENGTH

Harry R. Glahn
December 1, 2000

PURPOSE: To find the nominal (integer) mesh length as a function of the actual mesh length, and to find the floating point mesh length which is a true halving for each nominal mesh length. Nominal mesh lengths between 320 and 1 are accommodated.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL XMSMSH(KFILDO,XMESH,MESH,XMESHN)
```

KFILDO - Unit number of output (print) file. (INPUT)

XMESH - Actual mesh length. (INPUT)

MESH - Nominal mesh length. (OUTPUT)

XMESHN - Real representation of nominal mesh length. MESH and XMESHN are the same from 320 to 5, but from then on, the integer values are not halves of the next one. (OUTPUT)

EXAMPLE

```
CALL XMSMSH(KFILDO,5.951325,MESH,XMESHN)
```

The value MESH = 5 is returned. XMESHN = 5. = 320/(2**6), which corresponds to 5.951325 = 381/2**6 is also returned. XMESH is needed because this is the mesh length stored by TDLPACK, while the shorthand value of MESH is used in most places in LAMP.

OUTPUT:

None.

RESTRICTIONS:

Actual mesh lengths between XMESH = 320. and 1.48783125 inclusive are accommodated. If XMESH is not one of the accommodated values, a diagnostic is produced.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

NEWSIZ

CALCULATES A NEW GRID SIZE AND POLE POSITION

Harry R. Glahn
December 1, 2000

PURPOSE: To calculate a new grid size and pole position based on an old grid size, old pole position, old mesh length, and new mesh length.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL NEWSIZ(KFILDO,MESHIN,MESHOUT,NXIN,NYIN,NXPIN,NYPIN,
            NXOUT,NYOUT,NXPOUT,NYPOUT,ND2X3)
```

KFILDO - Unit number of output (print) file. (INPUT)

MESHIN - Nominal mesh length of old grid. (INPUT)

MESHOUT - Nominal mesh length of new grid. (INPUT)

NXIN - Size of old grid in IX direction. (INPUT)

NYIN - Size of old grid in JY direction. (INPUT)

NXPIN - Pole position in the IX direction of the old grid. (INPUT)

NYPIN - Pole position in the JY direction of the old grid. (INPUT)

NXOUT - Size of new grid in IX direction. (OUTPUT)

NYOUT - Size of new grid in JY direction. (OUTPUT)

NXPOUT - Pole position in the IX direction of the new grid. (OUTPUT)

NYPOUT - Pole position in the JY direction of the new grid. (OUTPUT)

ND2X3 - Size of array that NXOUT*NYOUT must fit into.

EXAMPLE

```
CALL NEWSIZ(KFILDO,80,40,100,50,110,55,NXOUT,NYOUT,NXPOUT,NYPOUT,5000O)
```

Given a grid of mesh length 80 km of size 100 by 50 with a pole position of 110,55, the values of NXOUT and NYOUT contain the dimensions of a grid with mesh length of 40 km and a pole position of NXPOUT, NYPOUT. If the grid were to occupy more than 50000 positions, a diagnostic would be produced and NEWSIZ would halt.

OUTPUT:

None.

RESTRICTIONS:

Mesh lengths are expected to be such that one can be obtained from the other by dividing or multiplying by a power of two. Also, NXIN, NYIN, NXPIN, and NYPIN are expected to be odd; however, this is not checked.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

POSGRD

POSITIONS A GRID WITHIN A LARGER GRID

Harry R. Glahn
December 1, 2000

PURPOSE: To position an output grid within an input grid, both with the same mesh length, and to adjust NX, NY, NXP, and NYP. The output grid will cover the "LAMP" area, which is defined as a quarter Bedient grid.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL POSGRD(KFILDO,P,MESHB,ALATL,ALONL,NXL,NYL,NXPL,NYPL,
            MESH,XMESH,XLAT,ORIENT,NXIN,NYIN,ALATIN,ALONIN,
            NXOUT,NYOUT,NXPOUT,NYPOUT,ND2X3)
```

KFILDO - Unit number of output (print) file. (INPUT)

P(IX,JY) - The input grid (IX=1,NXIN) (JY=1,NYIN) and output grid (IX=1,NXOUT) (JY=1,NYOUT). (INPUT/OUTPUT)

MESHB - Nominal mesh length of LAMP grid at a quarter Bedient. ALATL, ALONL, NXL, NYL, NXPL, and NYPL all refer to MESHB. (INPUT)

ALATL - Latitude of lower left corner of LAMP Grid. (INPUT)

ALONL - Longitude of lower left corner of LAMP Grid. (INPUT)

NXL - NX extent of the LAMP grid at mesh length MESHB. (INPUT)

NYL - NY extent of the LAMP grid at mesh length MESHB. (INPUT)

NXPL - Pole position in the IX direction of the LAMP grid at mesh length MESHB. (INPUT)

NYPL - Pole position in the JY direction of the LAMP grid at mesh length MESHB. (INPUT)

MESH - Nominal mesh length of the input and output grids. (INPUT)

XMESH - Nominal mesh length of the input and output grids in floating point form. MESH and XMESH are equal at 5 and above, but the next two lower values of XMESH are 2.5 and 1.25, whereas the corresponding values of MESH are 2 and 1. (OUTPUT)

XLAT - North latitude in degrees at which the mesh lengths are specified. (INPUT)

ORIENT - Grid orientation--west longitude in degrees at which the grid columns are vertical. (INPUT)

NXIN - Size of the input grid in the IX direction. (INPUT)

NYIN - Size of the input grid in the JY direction. (INPUT)

ALATIN - Latitude of lower left corner of the input grid. (INPUT)

ALONIN - Longitude of lower left corner of the input grid. (INPUT)

NXOUT - Size of output grid in IX direction with nominal mesh length = MESH. (OUTPUT)

NYOUT - Size of output grid in JY direction with nominal mesh length = MESH. (OUTPUT)

NXPOUT - Pole position in the IX direction of the output grid with nominal mesh length = MESH. (INPUT)

NYPOUT - Pole position in the JY direction of the output grid with nominal mesh length = MESH. (INPUT)

ND2X3 - Size of combined dimension of P(,). (INPUT)

EXAMPLE

```
CALL POSGRD(KFILDO,P,80,ALATL,ALONL,100,50,110,55,40,XMESH,60.,105.,
            100,50,6.9872,147.0644,NXOUT,NYOUT,NXPOUT,NYPOUT,5000O)
```

The "LAMP" grid is 80 km (one quarter Bedient), has a lower left corner at ALATL, ALONL, and is of extent 100 by 50 with a pole position of 110,55. (Note that there is redundancy in specification.) The mesh length of the input and output grids is 40 km. The input grid has a lower left corner at 6.9872 deg. north latitude and 147.0644 deg. west longitude. When the LAMP grid is cut out of the input grid, it will have an extent of NXOUT, NYOUT, with a pole position of NXPOUT, NYPOUT. The overall dimension of P( ) is ND2X3.

OUTPUT:

None.

RESTRICTIONS:

The output grid must be within the input grid.

NONSYSTEM ROUTINES USED:

MSHXMS, PSLLIJ, CUT.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

XYCOMP

COMPUTES X AND Y POSITIONS OF STATIONS

Harry R. Glahn
December 1, 2000

PURPOSE: To compute XI and YJ positions of stations (or locations) in terms of a grid specified by XMESHL, XLATLL, and XLONLL.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL XYCOMP(KFILDO,IP,CCALL,NAME,XMESHL,XLAT,ORIENT,XLATLL,XLONLL,
            STALAT,STALON,XI,YJ,NSTA)
```

KFILDO - Unit number of output (print) file. (INPUT)

IP15 - Indicates whether (=1) or not (=0) the list of stations and their IX,JY positions on the grid will be printed to the file whose unit number is IP15. (INPUT)

CCALL(K) - 8-character call letters (K=1,NSTA) to use for printout when IP15 is used. (INPUT) (CHARACTER*8)

NAME(K) - 20 character station names (K=1,NSTA) to use for printout when IP15 is used. (INPUT) (CHARACTER*20)

XMESHL - Actual mesh length in km of LAMP grid at a quarter Bedient. (INPUT)

XLAT - Latitude at which the mesh length is specified. (INPUT)

ORIENT - Grid orientation--longitude at which the grid columns are vertical. (INPUT)

XLATLL - Latitude of lower left corner of the grid. (INPUT)

XLONLL - Longitude of lower left corner of the grid. (INPUT)

STALAT(K) - North latitude of stations. (INPUT).

STALON(K) - West longitude of stations. (INPUT).

XI - The XI position on the grid for station K (K=1,NSTA). (OUTPUT)

YJ - The YJ position on the grid for station K (K=1,NSTA). (OUTPUT)

NSTA - Number of values in CCALL( ), NAME( ), STALAT( ), and STALON( ). (INPUT)

EXAMPLE

```
CALL XYCOMP(KFILDO,IP,CALL,NAME,80.,60.,120.,XLATLL,XLONLL,
            STALAT,STALON,XI,YJ,NSTA)
```

The unit number for printing is IP15, when it is not zero. The Call letters and names of the stations are in CCALL( ) and NAME( ), respectively. The mesh length is 80 km, which is specified at 60 deg. N. The orientation is 120 deg. W. longitude. The grid has a lower left corner of XLATLL, XLONLL. The XI and YJ positions of the stations in CCALL( ) will be returned in XI( ) and YJ( ), respectively, for NSTA stations.

OUTPUT:

None.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

PSLLIJ.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

ITRP

INTERPOLATES INTO GRID

Harry R. Glahn
December 1, 2000

PURPOSE: To interpolate biquadratically into a grid P(,) at a point, and return the value. If the point is not within the grid, a linear extrapolation to the point is provided.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL ITRP(P,NX,NY,BX,BY,BB)
```

P(IX,JY) - The grid is in P(,) (IX=1,NX) (JY=1,NY). (INPUT)

NX - The NX extent of the grid in P(,). (INPUT)

NY - The NY extent of the grid in P(,). (INPUT)

BX - The position in the IX direction of the point to interpolate to. (INPUT)

BY - The position in the JY direction of the point to interpolate to. (INPUT)

BB - The interpolated value. (OUTPUT)

EXAMPLE

```
DIMENSION P(100,100)

CALL ITRP(P,100,100,30,40,BB)
```

The value from the grid at the point at IX=30 and JY=40 in the grid in P(,) of dimensions 100 by 100 is returned in BB.

OUTPUT:

None.

RESTRICTIONS:

None other than stated above. Normally, the point BX,BY would be within the NX by NY grid. If it is not, extrapolation rather than interpolation is done.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

MSHXMS

FINDS ACTUAL MESH LENGTH FROM NOMINAL MESH LENGTH

Harry R. Glahn
December 1, 2000

PURPOSE: To find the actual mesh length as a function of the nominal mesh length, and to find the actual floating point nominal length which is a true halving for each nominal mesh length. Nominal mesh lengths between 320 and 1 are accommodated.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL XMSMSH(KFILDO,MESH,XMESHN,XMESH)
```

KFILDO - Unit number of output (print) file. (INPUT)

MESH - Nominal mesh length. (INPUT)

XMESHN - Real representation of nominal mesh length. MESH and XMESHN are the same from 320 to 5, but from then on, the integer values are not halves of the next one. (OUTPUT)

XMESH - Actual mesh length. (OUTPUT)

EXAMPLE

```
CALL XMSMSH(KFILDO,80,XMESHN,XMESH)
```

The value XMESH = 95.25 is returned, along with XMESHN = 80. XMESH is needed because this is the mesh length stored by TDLPACK, while the shorthand value MESH is used in most places in LAMP.

OUTPUT:

None.

RESTRICTIONS:

Nominal mesh lengths between MESH = 320. and 1 inclusive are accommodated; these correspond to 381 and 1.48783125, respectively. If XMESH is not one of the accommodated values, a diagnostic is produced.

NONSYSTEM ROUTINES USED:

None.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

PAWOTG

TDLPACKS AND WRITES GRID TO SEQUENTIAL FILE

Harry R. Glahn
December 1, 2000

PURPOSE: To prepare for packing, pack, and write a grid record to a TDLPACK sequential file.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL PAWOTG(KFILDO,KFILOG,IP16,NDATE,ID,ITAUH,ITAUM,MODNO,NSEQ,ISCALD,
            NPROJ,ALAT,ALON,ORIENT,MESH,XLAT,NX,NY,RECORD,IA,IC,IPACK,ND5,
            MINPK,IS0,IS1,IS2,IS4,ND7,IPLAIN,NCHAR,XMISSP,XMISSS,LX,
            IOCTET,NTOTBY,NTOTRC,L3264B,L3264W,IER)
```

KFILDO - Unit number of output (print) file. (INPUT)

KFILOG - The TDLPACK sequential unit number to write to. (INPUT)

IP16 - Indicates whether (>0) or not (=0) a statement will be output to IP16 when a sequential file is written through PAWOTG. (INPUT)

NDATE - The date of the data to be written in form yyyymmddhh. (INPUT)

ID(J) - The 4-word MOS ID of the data to be written (J=1,4). (INPUT)

ITAUH - The projection of the data to be written in hours. (INPUT)

ITAUM - The projection of the data to be written in minutes. (INPUT)

MODNO - The number of the model that produced the grid of data. (INPUT)

NSEQ - Sequence number of the grid. (INPUT)

ISCALD - Decimal scaling factor to use in packing the data. (INPUT)

NPROJ - Map projection of the grid of data. (INPUT)

ALAT - Latitude of the lower left corner of the grid in degrees. (INPUT)

ALON - Longitude of the lower left corner of the grid in degrees. (INPUT)

ORIENT - Longitude of grid orientation in degrees. (INPUT)

MESH - Nominal mesh length of grid in km. (INPUT)

XLAT - Latitude in degrees at which MESH is correct. (INPUT)

NX - IX extent of the grid. (INPUT)

NY - JY extent of the grid. (INPUT)

RECORD(J) - Grid of data for packing and writing (J=1,ND5). (INPUT)

IA(J) - Work array (J=1,ND5). (INTERNAL)

IC(J) - Work array (J=1,ND5). (INTERNAL)

IPACK(J) - Work array (J=1,ND5). (INTERNAL)

ND5 - Dimension of RECORD( ), IA( ), IC( ), and IPACK( ). (INPUT)

MINPK - Minimum size of groups to pack in TDLPACK. Normally equals 14. (INPUT)

IS0(J) - Work array (J=1,ND7). (INTERNAL)

IS1(J) - Work array (J=1,ND7). (INTERNAL)

IS2(J) - Work array (J=1,ND7). (INTERNAL)

IS4(J) - Work array (J=1,ND7). (INTERNAL)

ND7 - Dimension of IS0( ), IS1( ), IS2( ), and IS4( ), usually 54. (INPUT)

IPLAIN(I,J) - Name of the variable to pack (I=1,L3264W) (J=1,4). This is CHARACTER data in an INTEGER array. IPLAIN(,) is likely EQUIVALENCed to PLAIN*32 in the calling routine. (INPUT)

NCHAR - The number of characters to plain language in IPLAIN(,) to pack with the data, maximum of 32. (INPUT)

XMISSP - Primary missing value indicator. (INPUT).

XMISSS - Secondary missing value indicator. (INPUT).

LX - The number of groups in the packed data (the number of second order minima). (OUTPUT)

IOCTET - The total packed message size in octets (bytes). (OUTPUT)

NTOTBY - The total number of bytes in the file after writing this record. (INPUT/OUTPUT)

NTOTRC - The total number of records in the file after writing this record. (INPUT/OUTPUT)

L3264B - Integer word length in bits of machine being used (either 32 or 64). (INPUT)

L3264W - Number of machine length words in 64 bits (either 1 or 2). (INPUT)

IER - Status return.
0 = good return.
51 = ND5 ≤ NX*NY
Other values come from called subroutines and should be treated as fatal in calling program. (OUTPUT)

EXAMPLE

```
      PARAMETER (L3264B=32)
      PARAMETER (L3264W=64/L3264B)
      PARAMETER (ND5=10000)
      PARAMETER (ND7=54)

      CHARACTER*32 PLAIN

      DIMENSION ID(4)
      DIMENSION RECORD(ND5),IA(ND5),IC(ND5),IPACK(ND5)
      DIMENSION IPLAIN(L3264W,4)
      DIMENSION IS0(ND7),IS1(ND7),IS2(ND7),IS4(ND7)

      DATA KFILDO/12/,
     1     KFILOG/30/,
     2     IP16/16/
      DATA ID/001200005,0,0,0/
      DATA PLAIN/'SEA LEVEL PRESSURE              '/

      EQUIVALENCE (IPLAIN,PLAIN)

      CALL PAWOTG(KFILDO,KFILOG,IP16,2000010100,ID,6,0,5,0,1,
                  5,ALAT,ALON,105.,MESH,60.,NX,NY,RECORD,IA,IC,IPACK,ND5,
                  14,IS0,IS1,IS2,IS4,ND7,IPLAIN,18,9999.,0.,LX,
                  IOCTET,NTOTBY,NTOTRC,L3264B,L3264W,IER)
```

The data for packing are furnished in RECORD( ), and IA( ), IC( ), and IPACK( ) are work arrays. The primary missing data indicator is 9999, and there is no secondary missing value indicator. The grid has a lower left corner at ALAT, ALON, has a mesh length of MESH at 60 degrees north latitude, is oriented with 105 degrees west longitude, has an extent of NX by NY, is on a polar stereographic map projection (NPROJ=5), and has a sequence number of zero. The data to be written are identified in ID((4) and PLAIN as sea level pressure, and 18 characters of plain data are to be packed. A 32-bit machine is being used. The unit number for the print file is 12 and the packed data are to be written to unit number 30. The date of the data is January 1, 2000, at 0000 UTC. The minimum group size to use in packing is 14. The user may want to know the byte size of the record (IOCTET), the total number of bytes on the file (NTOTBY), the total number of records on the file (NTOTRC), and/or the number of groups used in packing the data (LX). The user will be reminded on unit number 16 that a record has been written.

OUTPUT:

Diagnostic for ND5 $\leq$ NX*NY. Possibly other diagnostics from called subroutines.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

DATPRS, MSHXMS, PACK2D, WRITEP, UNPKBG.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

PAWING

TDLPACKS AND WRITES GRID TO MOS-2000 INTERNAL FILE

Harry R. Glahn
December 1, 2000

PURPOSE: To prepare for packing, pack, and write a grid record to the TDLPACK MOS-2000 Internal Random Access file system.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
CALL PAWING(KFILDO,KFIL10,NDATE,ID,ITAUH,ITAUM,MODNO,NSEQ,ISCALD,
            NPROJ,ALAT,ALON,ORIENT,MESHB,MESH,XLAT,NX,NY,
            RECORD,IA,IC,IPACK,ND5,MINPK,
            LSTORE,ND9,LITEMS,CORE,ND10,NBLOCK,
            IS0,IS1,IS2,IS4,ND7,IPLAIN,NCHAR,XMISSP,XMISSS,LX,
            IOCTET,L3264B,L3264W,IER)
```

KFILDO - Unit number of output (print) file. (INPUT)

KFIL10 - The MOS-2000 internal random access unit number to write to. (INPUT)

NDATE - The date of the data to be written in form yyyymmddhh. (INPUT)

ID(J) - The 4-word MOS ID of the data to be written (J=1,4). (INPUT)

ITAUH - The projection of the data to be written in hours. (INPUT)

ITAUM - The projection of the data to be written in minutes. (INPUT)

MODNO - The number of the model that produced the grid of data. (INPUT)

NSEQ - Sequence number of the grid. (INPUT)

ISCALD - Decimal scaling factor to use in packing the data. (INPUT)

NPROJ - Map projection of the grid of data. (INPUT)

ALAT - Latitude of the lower left corner of the grid in degrees. (INPUT)

ALON - Longitude of the lower left corner of the grid in degrees. (INPUT)

ORIENT - Longitude of grid orientation in degrees. (INPUT)

MESHB - Nominal mesh length of a quarter Bedient grid in km. (INPUT)

MESH - Nominal mesh length of grid in km. (INPUT)

XLAT - Latitude in degrees at which MESHB and MESH are correct. (INPUT)

NX - IX extent of the grid. (INPUT)

NY - JY extent of the grid. (INPUT)

RECORD(J) - Grid of data for packing and writing (J=1,ND5). (INPUT)

IA(J) - Work array (J=1,ND5). (INTERNAL)

IC(J) - Work array (J=1,ND5). (INTERNAL)

IPACK(J) - Work array (J=1,ND5). (INTERNAL)

ND5 - Dimension of RECORD( ), IA( ), IC( ), and IPACK( ). (INPUT)

MINPK - Minimum size of groups to pack in TDLPACK. Normally equals 14. (INPUT)

LSTORE(I,J) - Where the keys for the data for the internal storage system are stored (I=1,12) (J=1,ND9). (INPUT/OUTPUT)

ND9 - The first dimension of LSTORE(,). The number of keys that can be stored in LSTORE(,). (INPUT)

LITEMS - The number of items stored in LSTORE(,). (INPUT/OUTPUT)

CORE(J) - The core storage (internal memory) available for storing data for which the keys are in LSTORE(,) (J=1,ND10). (INPUT/OUTPUT)

ND10 - Dimension of CORE( ). (INPUT)

NBLOCK - The block size in words of the MOS-2000 random disk file. (INPUT)

IS0(J) - Work array (J=1,ND7). (INTERNAL)

IS1(J) - Work array (J=1,ND7). (INTERNAL)

IS2(J) - Work array (J=1,ND7). (INTERNAL)

IS4(J) - Work array (J=1,ND7). (INTERNAL)

ND7 - Dimension of IS0( ), IS1( ), IS2( ), and IS4( ), usually 54. (INPUT)

IPLAIN(I,J) - Name of the variable to pack (I=1,L3264W) (J=1,4). This is CHARACTER data in an INTEGER array. IPLAIN(,) is likely EQUIVALENCed to PLAIN*32 in the calling routine. (INPUT)

NCHAR - The number of characters to plain language in IPLAIN(,) to pack with the data, maximum of 32. (INPUT)

XMISSP - Primary missing value indicator. (INPUT).

XMISSS - Secondary missing value indicator. (INPUT).

LX - The number of groups in the packed data (the number of second order minima). (OUTPUT)

IOCTET - The total packed message size in octets (bytes). (OUTPUT)

L3264B - Integer word length in bits of machine being used (either 32 or 64). (INPUT)

L3264W - Number of machine length words in 64 bits (either 1 or 2). (INPUT)

IER - Status return.
0 = good return.
51 = ND5 $\leq$ NX*NY
Other values come from called subroutines and should be treated as fatal in calling program. (OUTPUT)

EXAMPLE

```
      PARAMETER (L3264B=32)
      PARAMETER (L3264W=64/L3264B)
      PARAMETER (ND9=100)
      PARAMETER (ND5=10000)
      PARAMETER (ND7=54)

      CHARACTER*32 PLAIN

      DIMENSION ID(4)
      DIMENSION RECORD(ND5),IA(ND5),IC(ND5),IPACK(ND5)
      DIMENSION IPLAIN(L3264W,4)
      DIMENSION IS0(ND7),IS1(ND7),IS2(ND7),IS4(ND7)
      DIMENSION LSTORE(12,ND9)

      DATA KFILDO/12/,
     1     KFILOG/30/
      DATA ID/001200005,0,0,0/
      DATA PLAIN/'SEA LEVEL PRESSURE              '/
      ...
      EQUIVALENCE (IPLAIN,PLAIN)

      CALL PAWING(KFILDO,KFIL10,2000010100,ID,6,0,5,0,1,
                  5,ALAT,ALON,105.,MESHB,MESH,60.,NX,NY,
                  RECORD,IA,IC,IPACK,ND5,14,
                  LSTORE,ND9,LITEMS,CORE,ND10,NBLOCK,
                  IS0,IS1,IS2,IS4,ND7,IPLAIN,18,9999.,0.,LX,
                  IOCTET,L3264B,L3264W,IER)
```

The data for packing are furnished in RECORD( ), and IA( ), IC( ), and IPACK( ) are work arrays, all of dimension ND5, which must be $\geq$ NX*NY.

The primary missing data indicator is 9999, and there is no secondary missing value indicator. The grid has a lower left corner at ALAT, ALON, has a mesh length of MESH at 60 degrees north latitude, is oriented with 105 degrees west longitude, has an extent of NX by NY, is on a polar stereographic map projection (NPROJ=5), and has a sequence number of zero. The quarter Bedient mesh length is MESHB (=80). The data to be written are identified in ID((4) and PLAIN as sea level pressure, and 18 characters of plain data are to be packed. A 32-bit machine is being used. The unit number for the print file is 12 and the packed data are to be written to unit number 30. The date of the data is January 1, 2000, at 0000 UTC. It is from model number 5 (LAMP) and its projection is 6 hours and 0 minutes. The scaling is 1, so that for sea level pressure, tenths of mb are retained. The minimum group size to use in packing is 14. The user may want to know the byte size of the record (IOCTET), and/or the number of groups used in packing the data (LX). LSTORE(,), CORE( ), and associated variables as well as IS0( ), IS1( ), IS2( ), and IS4( ) are standard variables in MOS and LAMP programs.

OUTPUT:

Diagnostic for ND5 $\leq$ NX*NY. Possibly other diagnostics from called subroutines.

RESTRICTIONS:

None other than stated above.

NONSYSTEM ROUTINES USED:

DATPRS, MSHXMS, PACK2D, WRITEP, UNPKBG GSTORE.

LANGUAGE: FORTRAN 77

LOCATION: LAMP-2000 Library

CLDCAT

SPECIFIES CLOUD AMOUNT AND HEIGHT IN HEIGHT CATEGORIES

Jerome P.Charba
Jon Flatley
March 15, 2002

PURPOSE: Subroutine CLDCAT is used to assign METAR-reported “cloud layer” observations of amount (coded) and height (100s ft) to categories of cloud height (subsequently referred to as cloud bins). The routine accommodates any number of cloud bins, the top and bottom boundaries of which are specified in the variable ID (in hundreds of ft). These are referred to as the regular bins, as CLDCAT can also process two special bins which are used to specify the “amount” and height for partial obscuration and total obscuration. The CLDCAT CALL statement below is appropriate for direct insertion into program U150/Subroutine U400D. For a corresponding call in program U203/Subroutine OPTION, two array names (DATA( ) and LTAG( )) must be changed as indicated in the appropriate comments below. Information on U150/U400D and U203/OPTION is available in LAMP-2000 software documentation.

The MOS-2000 variable IDs used with CLDCAT have the form:

708FFF000 0LLLLUUUU 000000000 000

FFF has values of 350 and 355 for cloud amount and height, respectively. LLLL and UUUU are height values (in 100s of ft) for the bottom and top, respectively, of the bin. UUUU for the top bin, which has no upper bound, has the value 9999. For the special obscuration bins LLLL and UUUU both have the values 0000. The following are example IDs:

708350000 000050009 000000000 000 = cloud amount in 500-900 ft bin

708355000 001219999 000000000 000 = cloud height in top bin whose bottom is 12100 ft

708350000 000000000 000000000 000 = obscuration amount.

CALL AND EXPLANATION OF FORMAL PARAMETERS:

```
      CALL CLDCAT(KFILDO,KFIL10,
     1            LD,LDPARS,KDATE,DATA,LTAG,ND1,NSTA,
     2            IPACK,IWORK,FD2,FD3,FD4,ND5,
     3            LSTORE,ND9,LITEMS,CORE,ND10,
     4            NBLOCK,NFETCH,
     5            IS0,IS1,IS2,IS4,ND7,
     6            ISTAV,L3264B,IER)
```

KFILDO - Unit number for output (print) file (INPUT)

KFIL10 - Unit number of random access internal data storage (INPUT)

LD(K) - The basic integer predictor ID (J=1,4). This is the same as ID(J) except that the portions pertaining to processing are omitted:
B = LDPARS(3),
T = LDPARS(8),
I = LDPARS(13),
S = LDPARS(14),
G = LDPARS(15), and THRESH( ).
LD( ) is used to identify what variable is returned by this subroutine. (INPUT)

LDPARS(J) - The parsed individual components of the parsed predictor ID corresponding to LD( ) (J=1,15). (INPUT)
J=1--CCC (class of variable)
J=2--FFF (subclass of variable)
J=3--B (binary indicator)
J=4--DD (data source, model number)
J=5--V (vertical application)
J=6--LBLBLBLB (bottom of cloud bin, 0 if only 1 layer)
J=7--LTLTLTLT (top of cloud bin)
J=8--T (transformation)
J=9--R (run time offset, always + and back in time)
J=10--OT (time application)
J=11--OH (time period in hours)
J=12--TAU (projection in hours)
J=13--I (interpolation type)
J=14--S (smoothing indicator)
J=15--G (grid indicator)

KDATE - The date/time for which assigned cloud variable is needed. (INPUT)

DATA(K) - Assigned cloud variable for bin, as returned by subroutine. (For a corresponding call in U203/OPTION, this array must be named SDATA( )) (K=1,ND1). (OUTPUT)

LTAG(K) - Work array used to check status of cloud assignments to a bin. (For a corresponding call in U203/OPTION, this array must be named ISDATA( )) (K=1,ND1). (INTERNAL)
= 0, a cloud variable was not assigned to bin
= 1, a preliminary cloud variable was assigned to bin
= 2, a final cloud variable was assigned to bin.

ND1 - Maximum number of stations that can be dealt with. (INPUT)

NSTA - Number of stations or locations being dealt with. (INTERNAL)

IPACK(J) - Work array (J=1,ND5). (INTERNAL)

IWORK(J) - Work array (J=1,ND5). (INTERNAL)

FD2(J) - Work array containing ingested cloud layer height (J=1,ND5). (INTERNAL)

FD3(J) - Work array containing ingested cloud layer amount (J=1,ND5). (INTERNAL)

FD4(J) - Work array containing assigned cloud amount for bin (J=1,ND5). (INTERNAL)

ND5 - Dimension of IPACK( ), IWORK( ), FD2( ), FD3( ), and FD4( ). (INPUT)

LSTORE(L,J)- Array containing information about data fetched from MOS-2000 Internal File System (L=1,12) (J=1,LITEMS). (INPUT-OUTPUT).
L=1,4--The four IDs for the fetched data.
L=5 --Location of stored data. When in CORE( ), this is the location in CORE( ) where the data start.
L=6 --The number of four-byte words stored.
L=7 --= 2 for data packed in TDL GRIB format, = 1 otherwise.
L=8 --The date/time of the data in format YYYYMMDDHH.
L=9 --Number of times data have been fetched.
L=10 --Number of the SLAB in DIR(, ,L) and in NGRIDC(,L) defining the characteristics of the grid.
L=11 --The number of the predictor in the sorted list in ID(,N) (N=1,NPRED) for which this variable is needed, when it is needed only once from LSTORE(,). When it is needed more than once, the value is set = 7777.
L=12 --Used initially in establishing MOSTORE(,). Later used as a way of determining whether to keep this variable.

ND9 - The second dimension of LSTORE(,). (INPUT)

LITEMS - The number of items (columns) in LSTORE(,) that may have been used in this run. (INPUT)

CORE(J) - The array in which to store or retrieve data identified in LSTORE(,) (J=1,ND10). When CORE( ) is full, data are stored on disk. (OUTPUT)

ND10 - Dimension of CORE( ). (INPUT)

NBLOCK - The block size in words of the MOS-2000 random access file. (INPUT)

NFETCH - Incremented each time GFETCH is entered, which is a running count from the beginning of the program. This count is maintained should the user need it (diagnostics, etc.). (INTERNAL)

IS0 - MOS-2000 GRIB section 0 IDs (J=1,3). (INTERNAL)

IS1 - MOS-2000 GRIB section 1 IDs (J=1,22+). (INTERNAL)

IS2 - MOS-2000 GRIB section 2 IDs (J=1,12). (INTERNAL)

IS4 - MOS-2000 GRIB section 4 IDs (J=1,4). (INTERNAL)

ND7 - Dimension of IS0, IS1, and IS4. Not all locations are used. (INPUT)

ISTAV - = 1 since the data returned are stations values. (OUTPUT)

L3264B - Integer word length in bits of the machine being used (either 32 or 64). (INPUT)

IER - I/O status for GFETCH
0 = Good return
52 = NWORDS for GFETCH not equal to NSTA
103 = Entered with incorrect ID.
178 = Not good assignment of values.
See GFETCH for other values. (INTERNAL-OUTPUT)

EXAMPLE:

```
      PARAMETER (L3264B=32)
      PARAMETER (L3264W=64/L3264B)
      PARAMETER (NBLOCK=6400/L3264B)
      PARAMETER (MAXSTA=3500)
      PARAMETER (ND1=MAX(NBLOCK,MAXSTA))
      PARAMETER (ND2=1089,ND3=641)
      PARAMETER (ND7=54)
      PARAMETER (ND9=8000)
      PARAMETER (ND2X3=MAX(ND1,ND2*ND3))
      PARAMETER (ND5=ND2X3)
      PARAMETER (ND10=50000)
C
      REAL FD2(ND5),FD3(ND5),FD4(ND5),DATA(ND1),CORE(ND10)
C
      INTEGER KD(4),LD(4),MD(4),LDPARS(15)
      INTEGER IPACK(ND5),IWORK(ND5)
      INTEGER LSTORE(12,ND9)
      INTEGER IS0(ND7),IS1(ND7),IS2(ND7),IS4(ND7),LTAG(ND1)
C
      DATA KFILDO/12/,KFIL10/99/,KDATE/99021112/,
     1     LD/008350005,000020004,0,0/
      ...

      CALL PRSID1(KFILDO,LD,LDPARS)
      CALL CLDCAT(KFILDO,KFIL10,
     1            LD,LDPARS,KDATE,DATA,LTAG,ND1,NSTA,
     2            IPACK,IWORK,FD2,FD3,FD4,ND5,
     3            LSTORE,ND9,LITEMS,CORE,ND10,
     4            NBLOCK,NFETCH,
     5            IS0,IS1,IS2,IS4,ND7,
     6            ISTAV,L3264B,IER)
```

For this example, the default output unit number is 12, and the default unit number for random access internal data storage is 99. This example is taken from CLDCAT usage in U150/U400D, which explains the very large

values of ND2 and ND3 (which are x- and y-grid dimensions for a 10-km grid covering the grid used in MDL's AVN grid archive). Note that ND2 and ND3 ultimately control the value of ND5, which is the dimension of the work arrays FD2( ), FD3( ), FD4( ), IPACK( ), and IWORK( ). Much smaller values for these parameters could be used for most other applications where CLDCAT is called. ND1, the maximum number of stations to be dealt with, is the dimension of DATA( ), which contains the data returned by CLDCAT. ND10, which is assigned the very large value of 50000, is the dimension of CORE( ); this large value assures minimal usage of disk storage. KD( ), LD( ), MD( ) are dimensioned 4 and LDPARS( ) is dimensioned 15, which are standard values for the MOS-2000 variable ID system. LSTORE must be set up with dimensions, but is used only internally by GSTORE. ND9 is the second dimension of LSTORE, and upon return from GFETCH, LITEMS will be the number of items in LSTORE, with a maximum of ND9. The full identification of the data, when they had been packed in TDLPACK format, will be returned in IS0( ), IS1( ), IS2( ), and IS4( ) as appropriate. The data returned by CLDCAT, which is stored in DATA( ), is for the variable in LD( )(008350005,000020004,0,0), the cloud amount for the 200-400 ft bin, and for the date KDATE (99021112). CLDCAT is called from a 32-bit machine. IER will be zero upon return unless a problem has been encountered.

RESTRICTIONS:

CLDCAT is applicable to the MDL's METAR data archive, packed in MOS-2000 format, beginning December 1, 1996, 00Z. This archive is described in Chapter 13 of TDL Office Note 00-1. Based on this archive, cloud coverage for all station types must be available for the first "cloud layer." Otherwise, 9999 (missing) is returned for both cloud amount and height in all bins. Note, however, that 9999 is also returned for cloud height in all bins when a station reports "clear" or when a particular bin contains no cloud layer.

COMMENTS:

The cloud amount and height within a bin is determined from observations of up to six cloud layers contained in the MDL hourly archive. The six layers contained in the MDL archive is the maximum number of cloud layers reported by manual stations. ASOS reports a maximum of only three cloud layers, but when such automated observations are augmented by human observations additional cloud layers can be reported. For each reported cloud layer the cloud amount and base height is provided.

The reported cloud amount in the hourly reports is summarized in Table 1.

Table 1. The relationship between sky coverage, cloud coverage, and coded value.

Sky Coverage	Cloud Coverage	Coded Value
Clear (ASOS)	= 0	0
Clear (Manual)	= 0	1
Few	> 0 - <= 2/8	2
Scattered	>= 3/8 - <= 4/8	3
Broken	>= 5/8 - < 8/8	6
Overcast	= 8/8	8
Partial Obscuration	Assume unknown	9
Total Obscuration	Assume unknown	10

The cloud base height of each cloud layer is reported in hundreds of ft.

Subroutine CLDCAT is called separately for each bin and cloud variable (height or amount) desired. Rules that govern the assignment of cloud amount and height for an individual CLDCAT call are:

1. When missing (9999) is reported in the first layer, the requested bin is assigned 9999 for both amount and height.

2. When clear (coded 0 or 1) is reported in the first layer, the requested bin is assigned clear (0 for amount and 9999 for height).

3. When the requested bin does not contain a cloud layer, clear (0 for amount and 9999 for height) is assigned.

4. When the requested bin is a regular type and a single cloud layer falls within it, the coded value is assigned for amount and the reported height for height. Exceptions to this rule apply when the layer is partial or total obscuration; clear (0 for amount and 9999 for height) is assigned for partial obscuration and unknown (9997 for both amount and height) is assigned for total obscuration.

5. When an overcast or total obscuration layer is present below a regular bin, 9997 (unknown) is assigned for both amount and height.

6. When two or more cloud layers fall within a regular bin, several situations can arise:

   - When the first cloud layer contains partial obscuration, this layer is ignored and the assigned amount and height is based on the remaining layers;

   - When the highest cloud amount code is 6 or less, the layer with highest value is used to assign the amount and height;

   - When a lower layer contains an amount code of 6 (broken) and an upper layer contains an amount code of 8 (overcast), the assigned amount is 8 and the assigned height is taken from the broken layer.

7. Regarding the obscuration bin, 1 is assigned as the amount for partial obscuration and 2 for total obscuration. Otherwise the assigned amount is 0. The corresponding height is the reported height of the obscuration, which in the case of partial obscuration is always 0. Otherwise the assigned height is 9999.

It should be noted that for cloud height, 9999 is assigned for both truly missing and clear; thus, the two conditions are not distinguished. However, for any particular map time, the two conditions can be differentiated by comparing the corresponding cloud amount and height fields, as the amount field contains 0 in areas of clear.

It is also worthy to note that subroutine CLDCAT produces cloud amount and height fields that are consistent with those produced by two other cloud processing subroutines in the MOS-2000 library. In particular, regarding the cloud amount field, CLDCAT produces results consistent with the total sky cover used in MOS-2000, as produced by subroutine SFCTCLD (see TDL Office Note 00-2). In fact, when CLDCAT is applied by specifying a single cloud bin that spans the full depth of the atmosphere, the two subroutines yield essentially the same cloud amount field. The only difference appears in the manner total obscuration is treated; CLDCAT assigns 9997 (unknown) and SFCTCLD assigns 8 for this condition. (Of course, since CLDCAT returns the total obscuration for the amount variable in the special obscuration bin, a “true” unknown cloud amount can be differentiated from obscuration by comparing the cloud amount in the corresponding regular and obscuration bins.) In regards to cloud height, CLDCAT yields results that are consistent with the ceiling height used in MOS-2000, as produced by subroutine OBSCIGHT (/home21/tdllib/u201lib/obscight.f on HPs). Again, based on a single cloud bin encompassing the full depth of the atmosphere, CLDCAT yields a cloud height which is identical to the ceiling height returned by OBSCIGHT when broken (cloud amount code 6) and/or overcast (cloud amount code 8) is reported. Similar to the situation for cloud amount, the two routines treat height of total obscuration differently; CLDCAT returns 9997 whereas OBSCIGHT returns the reported height of the total obscuration. (Of course, since CLDCAT returns the height of the total obscuration for the height variable of the special obscuration bin, a “true” unknown cloud height can be differentiated from the obscuration by comparing the cloud height in the corresponding regular and obscuration bins.)

NONSYSTEM ROUTINES USED:

GFETCH

LANGUAGE: FORTRAN 77 with HP extensions

LOCATION: File cldcat.f is in the LAMP library /users24/glahn/lamp/lamplib.

U860

VERIFIES GRIDDED FORECASTS OF CONTINUOUS VARIABLES

David Rudack
Harry R. Glahn
J. Paul Dallavalle
Jerome P. Charba
March 15, 2002

PURPOSE: U860 computes verification scores for forecasts of continuous variables represented on a rectangular grid. The full set of scores are described in the APPENDIX. User specified scores are computed at each point of the entire input grid, and the full set of scores accommodated by U860 are computed over subsetted grid areas specified by the user. The forecast input data are gridpoint fields from numerical models, and the verification data are analyses of observed data. Both data types must be on an identical grid and in TDLPACK format. Because the forecast grids are defined internally in the packed files, the data can originate from a single model or multiple models. In addition, constant forecast-type data, also in TDLPACK format, can be input from random access files.

U860 uses a driver program, called DRU860, which provides for setting the fixed dimensions of array variables by PARAMETER statements. The passing of these variables to the main subroutine (called U860) and subsequent subroutines (where they are used with variable dimensions) eliminates they need for multiple versions. U860 can run on either a 32-bit or a 64-bit wordlength machine, as specified in a PARAMETER statement in the driver.

As indicated above, U860 performs two types of score computations. In one type, at least one scoring measure is computed at each gridpoint of the full input grid, and the ensuing gridded field(s) is/are output in TDLPACK format. This type of gridded scoring output can subsequently be displayed using GEMPAK. In a second type of score computation, the full set of available scores are computed from data aggregated over rectangular areas of the input grid. These grid rectangles, which are specified in a user-supplied control file or alternatively in inline U860.CN control records, must be applicable to whatever grid is used for input, and they can entail the full grid and/or any subset(s) thereof. (Note that all input grids, containing both forecast and observed data, must be identical. These scores are output in ASCII form to the default output file and to a user-specified output file. (Provision for writing the latter scores to a separate ASCII file has been made, but the actual writing of this file has not been implemented.)

Some familiarity with other MOS-2000 program write-ups will be necessary for a full understanding of this document.

CONTROL FILE INPUT: 'U860.CN' (Unit = KFILDI)

Record Type 1 - Format (A4,25I3) **Output Control**

This record contains unit numbers for a run, and it can also control the "default" output file number. KFILDI is the input unit number, as specified in DRU860.

IPINIT - four characters, usually a user's initials plus a run number, which are appended to print output filenames that begin with 'U860.' The control over the unit numbers for these files and the contents of the files is contained in the description of IP(J) (see below). The run number allows for multiple runs of U860 that write output files with unique names. For example, with IPINIT = 'DER2' and IP(2) = 40, the file name for Unit No. 40 = 'U860DER240'. DO NOT USE A BLANK FOR ANY OF THE CHARACTERS. (CHARACTER*4)

IP(J) - Each value (J=1,25) indicates whether (>0) or not (=0) certain information will be written to an output file. When IP( ) > 0, the value indicates the unit number for the file. The IP( ) values should not be the same as any other unit numbers used in U860 except possibly KFILDO (the default output file); however, a value of one IP( ) can be the same as that for another IP( ). The files contain ASCII text, generally for diagnostic purposes. The output files contain information as described below for individual values J:

(1) = All error diagnostics plus other information not specifically identified with other IP( ) numbers. When IP(1) is read as nonzero, KFILDO, the default output file unit number, will be set to IP(1). When IP(1) is read as zero, KFILDO will be used unchanged, as specified in a DRU860 DATA statement (= 12). Changing the default unit number allows for multiple runs of U860 or other programs within the same directory without overwriting output files.

(2) = The input dates in IDATE( ) exactly as they are read in. When input errors occur, print will be to the default output file unit KFILDO as well as to unit IP(2).

(3) = The output dates in IDATE( ), which may be extended by date spanning. When errors occur, output will be to the default output file unit KFILDO as well as to unit IP(3).

(4) = The station list (call letters only). If input errors occur, the station list will be written to the default output file unit KFILDO as well as to unit IP(4).

(5) = The station directory information. If input errors occur in the station list, the list will be written to the default output file unit KFILDO as well as to unit IP(5).

(6) = The variable IDs exactly as they are read in. This output is useful for checkout; for routine operation, IP(7), IP(8), and/or IP(9) may be more suitable.

(7) = The variable ID list in summary form. If input errors occur, the predictor list will be written to the default output file unit KFILDO as well as to unit IP(7).

(8) = The variable ID list in summary form. This list includes the parsed ID's in IDPARS(,). (IDPARS(,) contains the 15 components of each ID.)

(9) = The variable ID list in summary form. This differs from the print in IP(8) in that IP(9) does not include the

parsed ID's in IDPARS(,), but rather includes the information taken from the predictor constant file on unit KFILCP (see below).

(10) = The variable IDs for the first day (Day 1) as read from the archive tapes. This is simply a list of all the packed gridpoint fields on the input files.

(11) = The variable IDs of the archived input fields that are needed in U860, in the order they appear on the archive files for Day 1.

(12) = The I,J positions of the stations on the NGRID grids, together with the call letters and names. NGRID, at this point, is the number of different grid specifications comprising the input data. That is, NGM archived data and MRF archived data will have different grid specifications. After the data are read for the first day in the date list, the I,J positions are computed and saved under the IP(12) unit number. If, during the U860 run, a different grid definition is encountered, it is accommodated automatically, but the corresponding I,J positions are not saved in IP(12). In addition, when IP(12) ≠ 0, each directory record read from a vector input will be output to unit IP(12). For input of hourly data, this creates voluminous output.

(13) = Gridpoint fields. When the variable list indicates gridpoint values are to be written for viewing [JP(1,)>0, Record Type 13], they will be written to unit IP(13).

(14) = Gridpoint fields. When the variable list indicates gridpoint values are to be contoured (“gridprinted”) and written for viewing [JP(2,)>0, Record Type 13], they will be written to unit IP(14).

(16) = Diagnostics for constant and linearization routines (e.g., stations in threshold lists that are not being dealt with in this run).

(17) = The values of IFIND( ), ISTAV( ), and ITIME( ) after Day 1. (See Comments section for explanation.)

(18) = The variables saved in the MOS-2000 Internal Storage System after Day 1.

(23) = Information concerning opening and closing of sequential files.

Record Type 2 - Format (A72) **Run Identification**

This record is used to describe the run.

RUNID - 72 characters of information to describe the run. (CHARACTER*72)

Record Type 3 - Format (7(I10/),F10.0) **Control Parameters**

This record contains a variety of control parameters for the run. Note that each parameter is on a separate line. A brief description may be entered following each.

NSKIP - The number of errors that can occur on Day 1 before U860 aborts. Day 3 is usually completed before the abort actually occurs so

that the user can see additional results. When no data are found for Day 1, U860 continues and processes data for all subsequent days.

JSTOP - The total number of errors that can occur before the program aborts (see Comments section).

INCCYL - The increment in hours between date/times that are put into IDATE( ) (see Record Type 5) as a result of date spanning in subroutine DATPRO. This variable is also used in determining what data are saved in the Internal MOS-2000 Storage System. The date/times in Record Type 5 must be such that all could be arrived at by successively adding INCCYL to the first date/time in IDATE( ).

PXMISS - The value to substitute in the output for the secondary missing value of 9997. Note that the value allows the 9997 to be maintained if desired, set to zero in the case of the implied value of near zero for forecasts, or even set to some other value. If PXMISS is set to 9999 or 9997, then 9997 values in the gridded input data are treated as a missing in the scoring. Note that this control does not apply to missing values other than 9997.

Record Type 4 - Format (I3,4XA60) **Date List File**

This record (plus the terminator record) identifies the data set from which the date list is read. Records are read until the terminator KFILDT( ) = 99 is reached. The maximum number of records excluding the terminator record = 1. This Record Type 4 is read by subroutine RDSNAM.

KFILDT - Unit number for the file containing the input date list.

NAMDT - Filename containing the date list. When KFILDT = KFILDI (where KFILDI is the input unit number as specified in DRU860), NAMDT is not used and can be read as ‘DEFAULT’; in this case the date list is read in Record Type 5. (CHARACTER*60)

Record Type 5 - Format (7I10) **Date List**

This group of records specifies the date/times for which verification scores are computed. If KFILDT (read in Record Type 4) ≠ KFILDI, this Record Type 5 is omitted. It is important to note the date/times provided apply to the date/time of the “observation” variable, which is the same as the valid time for a forecast variable being scored. Also, all dates/times for a given verification must apply to a single time of the day. Therefore, if several forecast projections are verified in a single run they would be based on varying model initial times.

IDATE(J) - Initial date list, which may contain negative values indicating date spanning. When a negative occurs, all dates between this value and the previous date are filled in at the increment of hours specified in INCCYL. This input date list is modified in subroutine DATPRO to contain the complete date list with the dates in the spans filled in (J=1,NDATES).

Dates are input as YYMMDDHH and then modified to YYYYMMDDHH. This list is read by subroutine RDI, wherein zero values found in the input are removed. The terminator is 99999999. The maximum number of dates excluding the terminator is ND8, which is set in DRU860. Data for the first date in the list need not be available to get a successful run. The date/times put into Record Type 5 must be such that all of them could be obtained by successively adding INCCYL (read in Record Type 1) to the first date/time in IDATE( ).

Record Type 6 - Format (2I3,1XA60) **Gridpoint Input Data Files**

This group of records identifies the datasets from which the gridpoint data are read. Records are read until the terminator KFILIN( ) = 99 is reached. Maximum number of records excluding the terminator record = ND6, where the latter is set in DRU860. This Record Type 6 is read by subroutine RDSNAM. During reading this record type J = 1,NUMIN. If all input data are alternatively read from a constant file, only the terminator is necessary.

KFILIN(J) - Unit number for the input gridpoint file. KFILIN( ) must be < 80 for gridpoint data. Also note that 44 through 49 are reserved for random access files.

MODNUM(J) - Source of gridpoint data for this file (e.g., 5 = LAMP gridpoint data, 7 = eta model, 8 = aviation model). This number is used to match with the "DD" in the variable ID.

NAMIN(J) - Filename containing the model input data. (CHARACTER*60)

Note that when the intent is to read the datasets in sequence they should be read in the corresponding order and all have the same unit number. That is, if 2 years of data are to be used and one year is on one dataset and the other year on another, then the first should immediately precede the second in the list and both should have the same unit number. Actually, the same unit number is not mandatory but is highly recommended.

Record Type 7 - Format (I3,4XA60) **Random Access Files**

This group of records specifies random access datasets from which constant (input) data are read. Records are read until the terminator KFILRA( ) = 99 is reached. The maximum number of records excluding the terminator record = ND12 (ND12 $\leq$ 6). This Record Type 7 is read by subroutine RDSNAM. During reading this record type J = 1,NUMRA. If no data are read from a constant file, only the terminator is necessary.

KFILRA(J) - Unit number for the random access constant file. (J=1,NUMRA) Unit numbers must be in the range of 44 to 49; see "Restrictions" for more information.

RACESS(J) - Filename corresponding to KFILRA(J). (J=1,NUMRA) (CHARACTER*60)

Record Type 8a - Format(I3,4X,A60) **Subset Area(s) For Score Computation**

Record Type 8a (plus the terminator record) specifies the file containing grid area(s) over which are all scores accommodated by U860 are computed. Such records are read until the terminator KFILRR = 99 is reached. The maximum number of records excluding the terminator record = 1. This Record Type 8a is read by subroutine RDSNAM. The file will be opened as "OLD".

KFILRR - Unit number for the input file.

IREGION - Filename containing this information. (CHARACTER*60)

The file above contains one or more records, each with the format,

Format(4I8),

which specifies a rectangular grid area that can encompass the full input grid or any subset thereof. Specific information contained in each record is:

NXMIN - The western boundary of the grid area.
NXMAX - The eastern boundary of the grid area.
NYMIN - The southern boundary of the grid area.
NYMAX - The northern boundary of the grid area.

For example, if the input grid consists of the full AVN (137X81) archive grid, and one would like to compute verification scores only for Alaska, the following values would define the grid area:

43 61 54 21

The terminator record 99999999 follows one or more of these records. Note that the maximum number of points (where two points, the lower left and upper right, define a single rectangular grid area) used in defining of the verification areas is defined by ND13, which is set in DRU860.

When KFILRR is set to KFILDI in Record Type 8a, the grid areas are not contained in the file IREGION. For this option, IREGION can be set to 'DEFAULT', and instead records specifying the grid areas immediately follow with the next record in U860.CN. With this option the format and content of the records specifying the grid areas is identical to that in the above file. The terminator record is identical as well.

Record Type 8b - Format (I3,4XA60) **ASCII Output File**

This record (plus the terminator record) specifies the ASCII output file containing the verification scores for grid areas. Records are read until the terminator (KFILAS = 99) is reached. The maximum number of records excluding the terminator record = 1. This Record Type 8b is read by subroutine RDSNAM. This file will be opened as "NEW". If these ASCII data are not to be saved, only the terminator is necessary; in that case a file is not opened and data are not written.

KFILAS - Unit number for the output file.

ASCIFL - Filename containing this data. (CHARACTER*60)

Record Type 8c - Format (I3,4XA60) **Gridded TDLPACK Output File**

This record (plus the terminator record) identifies the TDLPACK output file containing verification scores at each gridpoint over the entire input grid. Records are read until the terminator (KFILGO = 99) is reached. The maximum number of records excluding the terminator record = 1. This Record Type 8c is read by subroutine RDSNAM. This file will be opened as “NEW”. If packed data are not to be saved, only the terminator is necessary; in that case a file is not opened and data are not written.

KFILGO - Unit number for the output file.

OUTGRD - Filename for this data. (CHARACTER*60)

Record Type 9 - Format (I3,4XA60) **Station and Location Files**

This pair of records (plus the terminator record) identifies the file(s) from which the station (or location) information is obtained. While this Record Type 9 is not used in computing the verification scores, it is necessary to make U860 functional. Records are read until the terminator KFILD( ) = 99 is reached. The maximum number of records excluding the terminator record = 2. This Record Type 9 is read by subroutine RDSNAM. During reading this record type J = 1,2.

KFILD(J) - Unit number for the file containing station call letters (J=1) and the station dictionary which holds the latitudes, longitudes, WBAN numbers, elevations, and names for each possible station (J=2). KFILD(1) can be the default input file number, KFILDI, in which case DIRNAM(1) is not used. Also, KFILD(1) can equal KFILD(2), in which case the call letters list will consist of all stations on the dictionary file; note that both a list of stations and a dictionary cannot exist on unit KFILD(1) = KFILD(2). (Normally, the directory would not reside on KFILDI, but is accommodated if KFILDI = KFILD(2). The input would follow Record Type 10.)

DIRNAM(J) - Filename matching KFILD(J). When KFILD( ) = KFILDI, DIRNAM( ) is not used and can be read as ‘DEFAULT’. (CHARACTER*60)

Record Type 10 - Format (7(A8,1X)) **Station List**

This group of records identifies the stations. If KFILD(1) ≠ KFILDI (the default input file) in Record Type 9, this group is omitted, and the information is taken from Record Type 9. Note that call letters for only one station is necessary to make U860 functional.

CCALL(K) - Call letters (or other 8-character location designator) of stations (or locations) for which output values are desired (K=1,NSTA). The call letters should (normally) be left justified, and if a full 8 characters are not present, CCALL( ) will be blank-filled (b) on the right, e.g., 'OKCbbbbb' could be 'OKC' or 'OKCb'. The terminator is '99999999'. The maximum number of stations excluding the terminator is ND1. (CHARACTER*8)

Record Type 11 - Format (I3,4XA60) **Variable File**

This record (plus the terminator record) identifies the file from which the variable IDs are input. Records are read until the terminator (KFILP = 99) is reached. The maximum number of records excluding the terminator record = 1. This Record Type 11 is read by subroutine RDSNAM. To verify a particular variable, the ID record for the "observation" variable must be first in the triplet set, the forecast variable ID record second, and the score ID record last. This triplet set of records will produce the score corresponding to the ID record provided at each point of the input grid, which is then output in TDLPACK form to an output file (see Record Type 8c). It also results in the computation of all scores specified in the APPENDIX for the grid subset areas provided, which are output in ASCII format to KFILDO (the default output file) and to the file whose name is in ASCIFL when KFILAS≠0 (see Record Type 8b). A discussion for Record Type 11 under CONTROL FILE INPUT provides details of how a triplet set is formulated. The APPENDIX provides a discussion of the score ID record content and format for all scores.

KFILP - Unit number for the variable IDs.

PRENAM - Filename corresponding to KFILP. When KFILP = KFILDI, PRENAM is not used and can be read as 'DEFAULT'. (CHARACTER*60)

Record Type 12 - Format (I3,4XA60) **Variable Constants File**

This record (plus the terminator record) identifies the file from which the variable constants are to be taken. (Variable constants include plain language description and gridprinting information.) Records are read until the terminator (KFILCP = 99) is reached. Maximum number of records excluding the terminator record = 1. This Record Type 12 is read by subroutine RDSNAM.

KFILCP - Unit number for the constants.

CONNAM - Filename corresponding to KFILCP. (CHARACTER*60)

Record Type 13 - Format (I9,1XI9,1XI9,1XI3,1XA1,1XI4,1XI3,4X3I2) **Variable List**

This group of records, which is comprised of information identical to that contained in the file read in Record Type 11, contains the variable IDs used in the scoring. When KFILP ≠ KFILDI (Record Type 11), this group is omitted, and the variable list is taken from the file read in Record Type 11. Records are read until the terminator [ID(1,) = 999999] is reached. The maximum number of records excluding the terminator record = ND4, which is set in DRU860. This Record Type 13 is read by subroutine RDPRED1. During reading of this record type N=1,NPRED.

ID(J,N) - The first 3 (J=1,3) words of the variable ID plus the last 3 digits of the 4th word, followed in order by the components of a threshold value (zero for continuous variables) consisting of (1) sign (either minus, or plus or blank for plus, read as A1), (2) 4 digits to follow a decimal point, and (3) 3 digits representing the power of 10 by which to multiply the decimal value just read. For easy reading (only) (1) and (2) above can

be separated by a decimal point and (2) and (3) separated by an "E". From these values, the 4th ID word (J=4) is composed.

JP(J,N) - For each variable N, JP(,N) indicates print or no print when ≠ 0 or = 0, respectively, for gridpoint values (J=1) or "zebra" gridprint map (J=2). These values combined with IP(13) and IP(14), allow easy control of output. For instance, all variables could have JP(2,) ≠ 0 and IP(14) ≠ 0 for a diagnostic run. All such print could be turned off by setting IP(14) = 0. Alternatively, IP(14) could be ≠ 0 and a selected gridprint obtained by changing JP(2,) from 0 to ≠ 0.

See Record Type 11 for further details concerning the precise formulation of the ID variables.

CONTROL FILE INPUT: (Name read from U860.CN, Record Type 4) (Unit = KFILDT)

Record Type 1 - Format (7I10) **Date List**

When the date list is not provided in file U860.CN Record Type 4 (KFILDT ≠ KFILDI, where KFILDT is read in Record Type 4 and KFILDI is the input unit number as specified in DRU860), this group of records determines the date/times for which data are to be input and processed. If KFILDT = KFILDI, this date list file is omitted.

IDATE(J) - Initial date list, which may contain negative values indicating date spanning. When a negative occurs, all dates between this value and the previous date are filled in at the increment of hours specified in INCCYL. This input date list is modified in subroutine DATPRO to contain the complete date list with the dates in the spans filled in (J=1,NDATES). Dates are input as YYMMDDHH and then modified to YYYYMMDDHH. This list is read by subroutine RDI, wherein the latter removes zero values found in the input. The terminator is 99999999. The maximum number of dates excluding the terminator is ND8, which is set in DRU860.

CONTROL FILE INPUT: (Name read from U860.CN, Record Type 8a) (Unit = KFILRR)

Record Type 1 - Format (4I4) **Grid areas for Score Computation**

When the grid areas for score computation is not provided in U860.CN (KFILRR ≠ KFILDI in Record Type 8a and KFILDI is the unit number as specified in DRU860), this group of records specifies the grid areas of concern. The format and content of the records (including the terminator record) is identical to that as discussed previously for Record Type 8a.

CONTROL FILE INPUT: (Name read from U860.CN, first record of Record Type 9) (Unit = KFILD(1))

Record Type 1 - Format (7(A8,1X)) **Station List**

This file is not needed when KFILD(1) = KFILDI, where KFILD(1) is read in Record Type 9 and KFILDI is the input unit number defined in DRU860. In this case, the call letters are read from KFILDI. It is also not needed when KFILD(1) = KFILD(2); for this highly unusual case, both the station

list and the associated information are read according to the format for the subsequent control file description for Unit No. KFILD(2).

CCALL(K) - Call letters (or other 8-character location designator) of stations (or locations) desired (K=1,NSTA). This list is read with subroutine RDC, which eliminates any blanks found in the input. Terminator is '99999999'. Maximum number of stations excluding the terminator is ND1. (See Record Type 9, Control File 'U860.CN', unit KFILDI for additional information.) (CHARACTER*8)

CONTROL FILE INPUT: (Name read from U860.CN, second record of Record Type 9) (Unit = KFILD(2))

Record Type 1 - Format **Station Locations**
(A8,1XA8,1XA17,4XA2,8XI5,1XA1,F7.4,1XA1,F8.4,9XA8,3(1XA8),12XI5)

This group of records provides information about the stations (or locations). It is needed unless KFILD(1) = KFILD(2) = KFILDI, where KFILD(2) is read in Record Type 9; for this highly unusual case, both the station list and the associated information are read from the input control file KFILDI. When KFILD(1) = KFILD(2) ≠ KFILDI, this control file is the same as the one above and both the station list and the associated information are taken from this file. If both files are needed (the usual case; KFILD(1) ≠ KFILD(2) ≠ KFILDI), the call letters here are matched with those in the station list and the appropriate information extracted.

CCALLD(K,J) - Call letters (or other character location designator) of stations (or locations) (J=1). As stated above, these call letters are matched with those in the station list unless KFILD(1) = KFILD(2), in which case these call letters comprise the station list and at the completion of reading this file, K=1,NSTA. When NEW = 1, CCALLD(K,1) is read from the first field (A8) and CCALLD(K,2) is read from the second field (A4). When NEW ≠ 1, CCALLD(K,1) is read from the second field and CCALLD(K,2) is read from the first field.

NAME(K) - 20-character name of station. This is used for visual identification of the station in certain output. Format is A17,4XA2; this provides for a 17-character name, 4 blanks, and a 2-character state abbreviation. Note that the last three characters in the "name" field in the directory are not used. (CHARACTER*20)

NELEV(K) - Elevation of the station. Format is I5.

SIGNLA - Sign of the latitude of the station, read as either "S" or "N". When read as "S", the latitude is set negative indicating South Latitude. Format is A1.

XLAT - Latitude in degrees. Format is F7.4.

SIGNLO - Sign of the longitude of the station, read as either "E" or "W". When read as "E", the longitude is modified to make all longitudes West. That is, longitude will be in the

range from 0 through 360 and be in degrees West over the United States. Format is A1.

LONDD - Longitude in degrees west. Format is F8.4.

CCALLD(K,J) - Call letters of substitute stations (or locations) (J=3,6) (K=1,NSTA).

IWBAN(K) - The WBAN number of the station. Format is I5.

The number of stations in this directory is not limited, except when it also constitutes the list to be used U860, in which case the number of entries is limited by ND1-1. No terminator is used.

CONTROL FILE INPUT: (Name read from U860.CN, Record Type 11) (Unit = KFILP)

Record Type 1 - Format (I9,1XI9,1XI9,1XI3,1XA1,1XI4,1XI3,4X3I2) **Variable List**

This group of records contains the triplet sets of variable ID records when this information is not provided in U860.CN (KFILP ≠ KFILDI, where KFILP is read in Record Type 11 and KFILDI is a unit number specified in DRU860). In each triplet set, U860 treats the first variable ID record in the set as the "observation," the second variable (record) as the forecast, and the third record is a score ID. It is critical that the order of the first two ID records is correct since U860 will still run to completion if the order is reversed. The score ID record (third record) specifies the score to be computed at each gridpoint of the input grid and packed to file KFILGO (see Record Type 8c). The formulation and content of this score ID variable is discussed in the APPENDIX.

When KFILP = KFILDI in Record Type 11, this file is omitted, and the variable list is taken from input file KFILDI (file 'U860.CN', Record Type 13). Records are read until the terminator ID(1,) = 999999 is reached. The maximum number of records excluding the terminator record = ND4. This Record Type 1 is read by subroutine RDPRED1. Upon completion of reading this record type, N=1,NPRED.

For illustration, the following is a example of two triplet sets of IDs in the appropriate format:

```
001201005 000000000 000000000 000  .0000E+00     0 0 0          LAMP SLP
ANALYSIS
001201005 000000000 006000006 000  .0000E+00     0 0 0          LAMP 6H
FCST SLP
910608005 000000000 000000006 000  .0000E+00     0 0 0          MABSE LAMP
SLP FCST (MB)
001201005 000000000 000000000 000  .0000E+00     0 0 0          LAMP SLP
ANALYSIS
001200008 000000000 012000012 000  .0000E+00     0 0 0          AVN 6H SLP
FCST
910601008 000000000 000000012 000  .0000E+00     0 0 0          MEAN OF
AVN FCST SLP(MB)
```

The first triplet set specifies the computation of the mean absolute error (for individual gridpoints of the full input grid) for a 6-h LAMP SLP

forecast. Note that if the time contained in the date/time list (file 'U860.CN', Record Types 4 or 5) is, say, 18Z the initial time for the LAMP forecast would be 12Z. The second triplet set of IDs specifies the computation of the mean of the AVN 12-h forecast of SLP at each point of the input grid. Note that in this case the AVN model initial time would be 06Z.

It is important to note that all scores specified in the APPENDIX for the input subset grid areas are computed regardless of how many triplet sets of IDs are provided. An essential point, however, is that at least one triplet set must be provided for U860 to proceed with either type (i.e., for individual gridpoints or for subset grid areas) of score computations.

CONTROL FILE INPUT: (Name read from U860.CN, Record Type 12) (Unit = KFILCP)

Record Type 1 - Format
(3(I9,1X),I3,2XA32,I3,F11.4,F10.4,F9.2,F8.2,2XA12) **Variable Constants**

This group of records, corresponding to Record Type 12 in U860.CN, contains information about the input variables [ID(J,) in Record Type 13] and IDTEMP(J) (see below). This file should be universally useable by all MOS-2000 users and is expected to be a separate file; that is, while KFILD(2) in Record Type 9 could = KFILDI, it would be very unusual for that to be the case. Note that the format matches that for a file input to other programs such as U600, U660, and U850.

IDTEMP(1) - First word of the variable ID, either with or without the "B" and "DD". This is matched with all variables read for this run, both with and without the "B" and "DD". When there is a match, the constant information with IDTEMP(1) is stored as indicated below.

IDTEMP(J) - These 3 words (J=2,4) are currently not used, but are meant to correspond to the ID words 2-4.

PLAINT - When IDTEMP(1) matches an ID(1,N), PLAINT is stored in PLAIN(N). These 32 characters are used for visual identification of variables in certain output. Although 32 characters are allowed, the first 5 are reserved for a height indicator (e.g., 1000-), and those after character 23 may be overwritten for vertical or time processing. This generally leaves 18 characters besides height, smoothing, and other processing indicators. (CHARACTER*32)

ISCALT - When IDTEMP(1) matches an ID(1,N), ISCALT is stored in ISCALD(N). This variable is the scaling constant (power of 10) to use when packing MOS-2000 data, in this case, the verification scores.

SMULTT - When IDTEMP(1) matches an ID(1,N), SMULTT is stored in SMULT(N). This variable is the multiplicative factor (power of 10) to use when gridprinting the gridpoint data.

SADDT - When IDTEMP(1) matches an ID(1,N), SADDT is stored in SADD(N). This variable is the additive factor to use when gridprinting the gridpoint data.

CONTT - When IDTEMP(1) matches an ID(1,N), CONTT is stored in CINT(N). This variable is the contour interval to use when gridprinting the data. It applies to the units in which the data exist on the packed input tapes, not after manipulation by SMULTT and SADDT.

ORIGNT - When IDTEMP(1) matches an ID(1,N), ORIGNT is stored in ORIGIN(N). This variable is the contour origin to use when gridprinting the data It applies to the units in which the data exist on the packed input tapes, not after manipulation by SMULTT and SADDT.

UNITST - When IDTEMP(1) matches an ID(1,N), UNITST is stored in UNITST(N). These characters define the units of the data after application of SMULTT and SADDT and are used in the visual inspection of the data in gridprinted maps. (CHARACTER*12)

Even when a match is found, the rest of the ID's in ID(1,) are checked because there might be more than one match.

Other processing occurs with the reading of these records in RDPRED1, much of it associated with the plain language description. See Chapter 4 of Office Note 00-1, Variable Identification, for details.

DATA INPUT:

All archived model data input to U860 must be in the MOS-2000 TDLPACK format read from sequential files. The unit numbers and file names are provided in KFILIN(J) and NAMIN(J) (J=1,NUMIN), respectively, of U860.CN, Record Type 6. Constant data are input from the random access MOS-2000 External File System (U860.CN, Record Type 7); these data are also in TDLPACK format, except for the call letters (directory) record. Reading is done with standard FORTRAN unformatted reads, and unpacking is done with subroutine UNPACK and its associated subroutine UNPKBG. Details of these data inputs are provided below.

A. SEQUENTIAL GRIDPOINT DATA

One or more sources of gridpoint data are accommodated, with the filenames and unit numbers provided in NAMIN(J) and KFILIN(J), respectively, as obtained from the control file 'U860.CN', Record Type 6. Each file is closed when an EOF is reached.

The definition of the grid to which the data in each record pertain is contained in that same record. A check is always made to assure that the correct grid information is used. In fact, not only can the data sources have different grid definitions, the definition within a particular source can change once or many times. For instance, it might be normal for the grid definition to change during the archive period. Beware, though, that each time the grid definition changes

when reading records that are to be used, extra computation must be done. Normally, ND11 (see "Setting up the Driver DRU860") will be ≥ the number of input grid data sources (unit numbers). **It is important to note that U860 will only verify "observations" and forecasts having identical grid characteristics.** If the grid definitions between the observations and forecasts are not identical U860 immediately terminates.

For gridpoint data, the unit number KFILIN( ) must be < 80, and the model number MODNUM( ) in Record Type 6 must match the model number(s) (DD) in Record Type 11 or 13 of the IDs from which data are input for that unit (see Record Type 6). Also, the RR in the latter records function as "lookback" feature.

B. RANDOM ACCESS GRIDDED AND VECTOR DATA

Up to five sources of station-oriented random access data and one source of gridpoint random access data are accommodated. These data must reside in the MOS-2000 External Random Access File System. These data are accessed with prescribed unit numbers, depending on the type of data; see "Restrictions" for more information.

DATA OUTPUT:

U860 outputs three forms of data, besides the diagnostics and other information on Units KFILDO and IP( ).

A. ASCII FOR VIEWING OR PRINTING

Gridprinted maps of a specified "gridpoint score" for each variable for which JP(1,) ≠ 0 without contours and/or for which JP(2,) ≠ 0 with contours are written to Units IP(13) and IP(14), respectively, when those units numbers ≠ 0 (see Record Type 13). This makes for easy control; all such print can be eliminated by changing the IP( ) values, and there is still control on each variable J through JP(,J).

B. ASCII FILE CONTAINING STATISTICAL SCORES FOR GRID AREAS(S)

All scores computed by U860 for the subset grid areas (see APPENDIX) can be written to KFILAS. The data are written to the dataset whose name is in ASCIFL and unit number in KFILAS, as specified from the control file 'U860.CN', Record Type 8b, unless KFILAS = 0, in which case data are not output.

When KFILAS ≠ 0, the ASCII information written to this file takes the following form:

(a) Header descriptors: (Projection, Variable, Area, Score, and Value)

Format - (A1,3X,A10,4X,A8,3X,A4,7X,A5,11X,5A,/A1)

(b) Score measures:

1. Mean of observations
2. Variance of observations

3. Standard deviation of observations
4. Mean of forecasts
5. Variance of forecasts
6. Standard deviation of forecasts
7. Correlation coefficient of forecasts
8. Bias of forecasts
9. Mean absolute error of forecasts
10. Root-mean-squared error of forecasts
11. S1 score of forecasts

Format - 11((7X,I2,11X,I2,7X,I2,7X,A11,8X,F8.3)/)

C. STATISTICAL SCORES WRITTEN TO KFILDO FOR SUBSET AREAS(S)

Even when KFILAS = 0 and KFILGO = 0, all scores accommodated by U860 (see APPENDIX) for the input subset grid areas are written to the default output file KFILDO. These scores appear near the end of the KFILDO file.

D. SCORES IN SEQUENTIAL GRIDPOINT MOS-2000 FORMAT

Scores for individual gridpoints of the entire input grid, which are specified in either Record Type 11 or 13, are packed in MOS-2000 TDLPACK format. They are packed with subroutine PACK2D and its associated subroutines by calling PACKG. Since the scores are generally computed over many dates and the packing of the scoring data requires a date for each record, the last date in the input list is used. Thus, the last date in the date list is written to KFILGO. The data are written to the dataset whose name was provided to OUTGRD and unit number KFILGO defined from the control file 'U860.CN', Record Type 8c, unless KFILGO = 0, in which case data are not output to this file.

EXAMPLE CONTROL FILE: ‘U860.CN’

Two example control files named ‘U860.CN’ and ‘U860_AREAS-IN-FILE.CN’ appear in directory /lamp2000/dru860/standard_CN/'. The first example file specifies the grid areas for verification in a separate file, while the latter file specifies these grid areas in in-line records. The easiest way to set up a U860 run is to take one of these control files and modify it as needed; DO NOT START FROM SCRATCH.

OUTPUT:

U860 output that can be printed can be put into one or more files, as described above under control file 'U860.CN', Record Type 1 and the definition of KFILDO in the driver DRU860. All errors will be contained in this default output file (KFILDO as possibly modified by IP(1)) as well as possibly to other files as defined by IP( ). Every effort has been made to notify the user of problems and potential problems and to proceed under user control without jeopardizing data.

RESTRICTIONS

Most restrictions are associated only with variables in PARAMETER statements in the driver which control array sizes. In some instances, machine wordlength is a factor, and 32-bit and 64-bit machines have been provided for through PARAMETER statements and the setting of L3264B to either 32 or 64. Only northern hemisphere polar stereographic and Lambert gridpoint data are accommodated. Formats and other guidelines in other MOS-2000 documents are followed.

HP workstations accommodate optionally-compiled Fortran statements containing a "D" in Column 1. The IBM-SP does not allow this. It is best to replace the "D" with, for instance, "C****" in Columns 1-5. This way, the possible optional statements can be spotted and be readily made operative.

The TDLPACK format for gridpoint data provides for a primary missing data indicator (=9999), although gridpoint data are generally not expected to contain missing values. If a primary missing value is found during unpacking of gridpoint data (for either the observations or forecasts or both), it is flagged by U860. Upon completion U860 notifies the user (with a message in the KFILDO output file) that primary missing values have been found. If primary missing values are encountered, U860 ensures that the gridpoints containing the missing data do not enter into the scoring for that particular day’s data. If secondary missing values (=9997) are encountered during processing, the user has the option of treating them as missing values, leaving them unchanged, or substituting a different value that is used as a valid data value in the scoring. Regardless of the choice, the new value is substituted by setting PXMISS in the U860 control file. When PXMISS is set to 9999, the 9997 value is treated as missing, but if PXMISS is set to 9997, this value is treated as a valid datum. **It is important to note that gridpoints with missing values encountered in the scoring of one forecast variable are also removed in the scoring for a subsequent forecast variable and vice versa.** This approach ensures a matched sample in the scoring of all forecast variables in a particular U860 run. Thus, if matched samples among forecast variables are not needed, each forecast variable should be scored in a separate U860 run. **It is also important to note that if comparative scores are wanted for separate forecast experiments involving the same variable (e.g., some basic model parameters are altered from one forecast experiment to another), the variable ID must be unique for each forecast experiment involved.** Provision for a unique variable ID in this case is accommodated in the U150.CN control file, wherein a U150 variable called INCDD is set. Setting this variable to a nonzero value is used to provide a unique DD value in the associated TDLPACK gridpoint output files (see APPENDIX for variable ID structure).

For each U860 run, all input data must have the same grid characteristics because interpolation of the observed or forecast grids to different grids is not accommodated by U860. If the input grid characteristics do not match, the program terminates and notifies the user of the mismatch.

The various arrays in U860 have specific uses. Arrays FD1( ), FD2( ), ... FD7( ) have been provided as general work arrays. Arrays FDVERT( ) and FDTIME( ) have been reserved for use by subroutines VERTP and TIMEP, respectively. FDSINS( ) and FDMS( ) have been reserved for use by

routines involving the sine of the latitude and the map scale factor for the input grid. L1DATA( ) and SDATA1( ) are reserved for use by linearization and constant routines. If modifications or additions are made to U860, care must be taken that information in the work arrays is not overwritten by new called routines. For instance, if a new routine were to call OPTN2 and the latter called VORTH, then VORTH would use arrays FD1( ), FD2( ), and FD3( ). Thus, the data the new calling routine had stored in these arrays could be overwritten.

Presently, U860 accommodates the verification of continuous scalar variables represented on grids. Extra capabilities, such as grid interpolation or variable transformations are not present in U860. If verification of discontinuous fields is needed, U870 should be used.

Regarding the gridded data input to U860, it not known *a priori* whether the computational elements of the input variable ID (e.g., "B" or "RR") have already been satisfied in the input data or whether these elements must still be satisfied. U860 initially attempts to find the exact variable ID in the input data, and if it does it is assumed these elements are already satisfied. If U860 does not find the exact field but instead finds a more basic version, it will perform the additional operations needed. If neither field is found, of course the program will abort. Also, if the input data contain missing gridpoint values (an unlikely possibility)

When the runtime offset, RR, and MODNUM( ) are not zero, meaning that data from a previous model run are needed, then data are carried forward in the MOS-2000 Internal Storage System. In the storing of data, as the file is read from a previous cycle to the current one, some data from an intermediate cycle may be saved that are not needed. Since U860 verifies data as a function of valid time, RR must be greater than zero in order to retrieve forecast data. Consequently, in the 3rd word of the forecast ID, RR = $\tau\tau\tau$.

The unit numbers for the random access files (see Record Type 7) must be in the range 44 through 49. (Note that U860 will not write to a random access file.) These unit numbers are used for the following purposes, where the CCC values in ID(1) denote the following:

Unit No.	CCC Range	Use
44	400-499	Gridpoint data
45	400-499	"True" constants (rel. freq., means, etc.)
46	500-599	1-d and 2-d constants, probably for U201
47	800-899	Thresholds for best category forecasts, etc.
48	200-299	Forecasts read only
49	200-299	Forecasts read/write

COMMENTS

Most error messages, which begin with ****, are output for printing to the file with unit number designated by IP( ) (see Record Type 1), and a count of the messages is kept for matching with NSKIP and JSTOP (see Record Type 3). Missing data will usually not be counted as an error, but a diagnostic message is provided. It is not always obvious what is truly an error as opposed to something that might be expected to happen

occasionally; therefore, the count cannot be considered absolute. When a field is to be computed in OPTION and a field needed for the computation cannot be found, a diagnostic message (possibly "****PREDICTOR NOT IDENTIFIED IN OPTN2") and in addition "****ALL VALUES IN FIELD _____ ARE MISSING IN PKMS99," will be provided. While these diagnostic messages could be eliminated, it is thought best to keep a watchful eye for such errors. A good practice is to provide a set of dates to U860 for which it is known valid data exist. Then any diagnostic is really unexpected. At the end of Day 1 and at the completion of the run, the number of "errors" and the number of missing variables (data records) are printed. Note that an error may result in missing data values and be counted for both.

On Day 1, GFETCH is entered directly at least once for every variable ("Day 1 is a term that has come to be used for the "first case." It is actually the first cycle of the first day.) A return of IER = 47 (which means data were not found in GFETCH) is not counted as an error in PRED81 for Day 1. It is counted as an error in PRED82, because GFETCH should not be entered directly when OPTION is needed.

Computational routines are written and available for U860. For instance, if verification is to be performed on dewpoint and this field is not provided, the subroutine DEWPT can be invoked through OPTION.

Some information that may seem repetitive is provided for printout on each run. However, it is believed the user should monitor this information and investigate apparent abnormalities. For instance, subroutine GCPAC prints compression information for the first 3 days. This will let the user determine whether the CORE( ) space provided for storage is far larger, or smaller, than needed, etc. If CORE( ) is small, extensive disk access will be necessary. If it is large, then other users or swapping space for the current run may be impacted. Generally, it is hoped CORE( ) can be about the size to hold the intermediate storage after Day 1.

It is unlikely that when no variable in a run has a particular model number, that the file containing that model will be needed; therefore, the file is closed. Closing the file keeps the data from being read when not needed. A highly unlikely exception occurs when a subroutine like COMBIN could use data from a model that was not indicated by the model number in the variable ID. If this is the case, include a variable with the (otherwise omitted) model number to force the file to remain open. This variable can be a “dummy” (see OPTION subroutine) to keep it from being written.

It is generally not necessary that each variable be present for Day 1 for that variable to be used on subsequent days (see the subsequent section "Characteristics of the U860 Lookback Feature").

A count is kept of each time a missing value indicator is found when unpacking grids. While missing values are generally unexpected, at the end of a run the missing count is printed if non-zero (see “Restrictions”).

Keep in mind that the writing of verification scores for gridpoints is not done if KFILGO = 0 (see Record Type 8c). That is, no output unit number and filename have been provided in this record type. If KFILGO ≠ 0,

scores are packed and can then be graphically displayed by running U203 and GEMPAK.

The variables IFIND( ), ISTAV( ), and ITIME( ), which can be printed for each input variable in table form under IP(17) control, are set after processing Day 1 data and have the following meanings:

IFIND = 1 When the variable can be found directly from the input and does not have to be computed through OPTION.
= 0 Variable has been identified in OPTION and will be computed there.
= 2 Variable has been identified in OPTION but missing values are returned Therefore, it is not yet known whether this variable will be input directly or computed through OPTION.

ISTAV = 1 When the data have vector format either on initial input or on return from OPTION. (Not applicable to U860)
= 0 Variable computed from gridded data.
= 2 Variable has not been found yet and it is still not known whether its input will be vector or gridded.

ITIME = 1 When the time offset RR is operative.
= 0 Otherwise. This may (also) occur when RR = 0.

CHARACTERISTICS OF THE U860 LOOKBACK FEATURE

U860 attempts to save data from previous cycles (date/times previous to NDATE, the date/time being processed). Such data are denoted by RR > 0 in ID(3) of the forecast variable (second variable ID of a triplet set). Consider the following conditions:

- The variable needed <u>is in</u> the ID list input by the user in Record Type 13.

  - All data needed are found for Day 1

    RR = 0 No data are saved from one NDATE to the next.

    RR > 0 Data for the variable with RR > 0 are saved for the number of hours indicated by RR.

  - Some data needed are not found for Day 1

    RR = 0 Not a problem.

    RR > 0 Usually not a problem.

- The variable needed <u>is not in</u> the ID list input by the user, but is needed by a computational subroutine.

  - All data needed are found for Day 1

    RR $\geq$ 0 Shouldn't be a problem, <u>provided the subroutine looks for all variables it will need for subsequent cycles</u>. This may require the subroutine to ask for data not needed for

the particular cycle being processed (e.g., max temperature is not calculated for each cycle).

- Some or all data needed for Day 1 are not found.

  RR $\geq$ 0 Shouldn't be a problem; U860 continues to process data and calculate scores for all subsequent days. All acceptable processed dates in U860 are listed near the bottom of KFILDO.

While considerable effort has been made to make the lookback feature robust, and to not require the user to specify (possibly in a separate list) all the variables that might be needed in all subroutines, not all situations could be handled. It is believed that all will be well when all data needed for Day 1 are present. It should be rare when one would need to make a run that would require data that were not available for the first date/time.

SETTING UP THE DRIVER DRU860

The preparation of the driver for a particular U860 run is relatively painless; it consists of using a template driver and modifying certain PARAMETER statements as necessary. These statements set values of:

L3264B - Set to 32 for a 32-bit machine (e.g., the HP's) or 64 for a 64-bit machine (e.g., some options on the IBM-SP).

ND1 - Maximum number of stations (or points) that can be dealt with. Note that this does not include the number of stations in the directory (read on Unit No. KFILD(2)) unless, of course, the station directory is to be used as the station list. ND1 must also be GE NBLOCK (see below).

ND2 - ND2*ND3 is the maximum size of the grid that can be dealt with. ND2 and ND3 are set separately to highlight the possible dimension of the grids. However, in the called routines, the size is only limited by the product, not each dimension individually.

ND3 - See ND2.

ND4 - The maximum number of variables that can be entered into 'U860.CN'.

ND5 - Dimension of IPACK( ), IWORK( ), and DATA( ). ND2X3 and ND5 should be set in the driver DRU860:

```
ND2X3 = MAX(ND1,ND2*ND3,ND12)
ND5=ND2x3
```

ND6 - Maximum number of sequential file input sources (e.g., models) that can be dealt with. If data from a model are on two files, then this would be counted as two, not one, etc.

ND7 - The size of ISO( ), IS1( ), IS2( ), and IS4( ). This would normally be 54.

ND8 - The maximum number of date/times that can be used. This is the "extended" date list, not just the abbreviated values read in.

ND9 - The maximum number of fields (variables) stored in the MOS-2000 Internal Storage System. Since all fields are stored for Day 1, ND9 must be large enough to hold all records of the date/time of Day 1 from all input files.

ND10 - The number of words of storage provided in the variable CORE( ) for the MOS-2000 Internal Storage System. When this is filled, a scratch disk file is used. Too small a number will result in more disk accesses than necessary (although caching may alleviate that); too large a number will result in wasted memory and possible excess paging.

ND11 - The maximum number of grid combinations that can be dealt with. For instance, if the NGM and AVN data are being used, and the grids are different, then ND11 would be $\geq$ 2.

ND12 - Maximum number of random access files (MOS-2000 External File System) that can be used. Limit is 6.

ND13 - Maximum number of points used in defining subset areas (see description for Record Type 8a).

Do not change the computation in U860 for the variable L3264W, and probably not for NBLOCK. NBLOCK is the block size in words for disk records for the MOS-2000 Internal Storage System. (Experimentation may determine that a larger value is desirable.) Note that ND1 must be GE NBLOCK.

The user can see from the template what impact each of these parameters has on storage from its usage in the DIMENSION statements. Some have relatively little effect (e.g., ND7), while others have considerable effect (e.g., ND2*ND3). Every effort has been made to insure that the array variables will not be overflowed if values too small are used; however, some danger may still exist.

The "startup" control file "U860.CN" is opened in DRU860, so that this aspect of U860 can be rather easily changed without recompiling or having a separate version of the main subroutine U860. This will be different on the IBM-SP where unit/file assignments are made differently, and possibly when U860 is run in batch mode.

TYPES OF COMPUTATIONS IN U860 USING GRIDDED DATA

U860 will perform the basic functions specified in the variable ID (TDL Office Note 00-1), as summarized below. It is expected only polar stereographic and Lambert grid archives will be used, and only those grid projections are accommodated.

PROCESSING INDICATORS

DD - Model number
1 = Trajectory
5 = LAMP
6 = NGM
7 = Eta
8 = AVN
9 = MRF

V - Vertical Processing [Same field (CCCFFF) and same projection (ttt)], but with different levels (UUUU and LLLL)
VERTP
0 = None (not used)
1 = Difference (UUUU-LLLL) (not used)
2 = Sum (not used)
3 = Mean (not used)

RR, O, HH - Time Operation on Fields (1) and (2)
TIMEP
Field (1) = Run time offset (NDATE - RR), projection tau + HH
Field (2) = Run time offset (NDATE - RR), projection tau
1 = Mean (not used)
2 = Difference (1)-(2) (not used)
3 = Maximum (not used)
4 = Minimum (not used)
Field (1) = Run time offset (NDATE - RR), projection tau + HH
Field (2) = Current run time, projection tau
5 = Mean (not used)
6 = Difference (2)-(1) (not used)
7 = Maximum (not used)
8 = Minimum (not used)

S - Smoothing
1 = 5-point
SMTH5
2 = 9-point
SMTH9
3 = 25-point
SMTH25

The processing is always done in the following order:

V,
RR, O, HH,
and S,

When **OPTION** is entered, the data returned are also gridpoint (in which case S is operative).

VARIABLES HANDLED IN U860

If direct model output such as temperature is to be verified, U860 internally converts the temperatures from Kelvin to Fahrenheit. The same can be said if dewpoint is to be verified by using subroutine DEWPT.

CCCFFF	Routine	Write-up No.	Description
003100	DEWPT	2.50	Dew point on an isobaric surface
003101	DEWPT	2.50	Dew point on a constant height surface
003106	DEWPT	2.50	Dew point on a sigma surface

WRITING NEW COMPUTATIONAL SUBROUTINES

New computational subroutines can be written as needed. Existing routines, such as **VORTH**, can be used as templates. Generally, the steps to be followed are:

1) Include the subroutine in the u860lib directory, and see that it is included in the makefile.

2) Enter the subroutine name with its calling sequence in the switching subroutine **OPTION** and probably in **OPTN2** with the proper IF tests. Generally, this switching would be based on IDPARS(1) and/or IDPARS(2).

3) Use the work arrays FD1( ), FD2( ), etc. as needed.

4) The new routine can call **OPTN2** (or even **OPTION**) if computed variables are needed for which routines are already available. If this is done, the complete **OPTN2** calling sequence must be used for the new subroutine, so that it can be passed to **OPTN2**.

5) If **OPTN2** is called or **OPTION** recalled, care must be taken to not overlap the use of the arrays FD1( ), FD2( ), etc. That is, if **VORTH** were to be called by **OPTN2**, it would use arrays FD1( ), FD2( ), and FD3( ). If these arrays were used by the new subroutine, they would be overwritten when **OPTN2** and **VORTH** are called. However, data are never carried from variable to variable in FD1( ), etc. That is, each time a new variable is dealt with in **OPTION**, each of the FDi is available for use. It is only in the computation of a particular variable that the work space must be managed by the user. Arrays NELEV( ), STALAT( ), and STALON( ) hold the station elevations, latitudes, and longitudes, respectively, in the order of the stations in CCALL(,). These arrays can be used in any subroutine, and, of course, they shouldn't be modified. Other "constant" data must be obtained from the MOS-2000 External Direct Access Storage System by calling subroutine CONST. Station elevations, if desired, can also be obtained through **CONST**, if desired.

(6) When a computational subroutine returns gridded data, that routine must set ISTAV = 0.

(7) As a special case, when data for a particular hour cannot be available (e.g., max/min temperature for some particular hour of the day), the computational subroutine can set IER = -1 to indicate a record is not to be written for that variable for that date/time.

(8) **OPTION** will produce a diagnostic message as appropriate when a called subroutine does not return IER = 0. The computational subroutine probably does not need an additional diagnostic. Also, the subroutine

should not return IER = 47 (indicating missing data), but another selected from the error table for MOS-2000. **OPTION**, **PRED81**, and **PRED82** will then be able to handle the situation.

NONSYSTEM ROUTINES USED

Use the load line in file /lamp2000/dru860/u860.com to create the executable program.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: u860lib. The driver is the file /lamp2000/dru860/dru860.f.

APPENDIX

SCORE DESCRIPTION

As indicated in the discussion for Record Type 11 and Record Type 13, the computation of scores is controlled by ID variables arranged in “triplet” groups. Each triplet, which produces a single score for a single variable, is also called a group. When a group appears in U860.CN, the associated score is computed at each gridpoint of the input grid. Further, when one or more verification grid areas are provided in U860.CN (see Record Type 8a), all scores specified below are computed for each area. Note also, U860 checks for primary missing gridpoint values (9999 values) in the forecast and verifying fields, and when they are encountered for a specified scoring area U860 removes them for all scoring groups. (U860 also accommodates secondary missing values, as discussed previously.) Thus, a matched verification sample is insured for all groups for the area in a U860 run.

VERIFICATION SCORES:

U860 computes verification scores for continuous scalar forecast variables, such as MSL pressure or surface temperature. Therefore, threshold values, which are used transform continuous forecast or observed variables into binary variables, are not used with this program. The scores and related statistics for an rectangular area with N gridpoints in the i (east-west) direction, M gridpoints in the j (north-south) direction, and S the total number of gridpoints (N x M) are as follows:

S = sample size.

$$\bar{o} = \frac{1}{S}\sum_{i=1}^{N}\sum_{j=1}^{M} o_{i,j}$$ = mean of observations.

$$\sigma_o = (\sigma_o{}^2)^{1/2}$$ = variance of observations.

$$\sigma_o^2 = \frac{1}{S}\sum_{i=1}^{N}\sum_{j=1}^{M} \left(o_{i,j} - \bar{o} \right)^2$$ = standard deviation of observations.

$$\bar{f} = \frac{1}{S}\sum_{i=1}^{N}\sum_{j=1}^{M} f_{i,j}$$ = mean of forecasts.

$$\sigma_f^2 = \frac{1}{S}\sum_{i=1}^{N}\sum_{j=1}^{M} \left(f_{i,j} - \bar{f} \right)^2$$ = variance of *forecasts*.

$$\sigma_f = (\sigma_f{}^2)^{1/2}$$ = standard deviation of forecasts.

$$\bar{f} - \bar{o}$$ = bias (or mean error) of forecasts.

$$MAE = \frac{1}{S}\sum_{i=1}^{N}\sum_{j=1}^{M}\left| f_{i,j} - o_{i,j} \right|$$ = mean absolute error of forecasts.

$$RMSE = \left\{ \frac{1}{S}\left[\sum_{i=1}^{N}\sum_{j=1}^{M} \left(f_{i,j} - o_{i,j} \right)^2 \right] \right\}^{1/2}$$ = root mean square error of forecasts.

$$S1 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} \left| \left(f_{i+1,j} - f_{i-1,j} + f_{i,j+1} - f_{i,j-1} \right) - \left(o_{i+1,j} - o_{i-1,j} + o_{i,j+1} - o_{i,j-1} \right) \right|}{\sum_{i=1}^{M}\sum_{j=1}^{N} \max\left\{ \left| \left(f_{i+1,j} - f_{i-1,j} + f_{i,j+1} - f_{i,j-1} \right) \right|, \left| \left(o_{i+1,j} - o_{i-1,j} + o_{i,j+1} - o_{i,j-1} \right) \right| \right\}}$$

= S1 score.

In words, the S1 score provides a measure of the vector error in the horizontal gradient of a continuous scalar forecast variable. More specifically, the magnitude of the vector (forecast) error is expressed as a percentage of the magnitude of the forecast or observed horizontal gradient of the variable, whichever is greater. When applied to any scalar variable, the S1 score has a range of 0 to 200, where 0 represents no horizontal gradient vector error and 200 represents the worst case scenario where the forecast and observed gradient vectors are equal in magnitude but opposite in sign. When applied to forecast MSL pressure fields, a “very good” forecast yields an S1 of about 30, and a “useless” forecast yields an S1 of about 70.

$$r_{xy} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j} y_{i,j} - \left(1/S\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\right)}{\left[\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}^2 - \left(1/S\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M} x_{i,j}\right)^2\right]^{1/2}\left[\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}^2 - \left(1/S\right)\left(\sum_{i=1}^{N}\sum_{j=1}^{M} y_{i,j}\right)^2\right]^{1/2}}$$

= correlation coefficient (Pearson)

<u>SCORE VARIABLE ID FORMAT:</u>

The MOS-2000 variable ID formulation for a U860 score is:

9ccfff0DD 000000000 000000ttt 000

where 9 indicates a U860 score, cc the meteorological variable, fff the verification measure, DD the data source, and ttt the forecast projection. (Note that the second ID word is identical to that for the corresponding MOS-2000 variable while the third and fourth words each retain only the last three digits of the MOS-2000 formulation.) The cc value specifies the variable verified, where:

10 = MSL pressure (mb)

20 = temperature (degrees F)
30 = dew point (degrees F)

The fff value specifies the score measure or related statistic, as follows:

601 - 999 = any score or related statistic for gridded fields.
601 = mean of the observed variable
602 = variance of the observed variable
603 = standard deviation of the observed variable
604 = mean of the forecast variable
605 = variance of the forecast variable
606 = standard deviation of the forecast variable
607 = mean algebraic error (bias) of the forecast variable
608 = mean absolute error of the forecast variable
610 = root mean squared error of the forecast variable
611 = correlation coefficient of the forecast variable
700 = S1 score (used for MSL pressure forecasts)

As an example, the four-word variable ID for the mean absolute error of a LAMP 3-h MSL pressure forecasts is:

910608005 000000000 000000003 000

Note that the LAMP model cycle from which the 3-h forecast is issued is specified in the HH portion of the date-time used in the TDLPACK record.

The ID variable for the mean absolute error of an AVN 6-h MSL pressure forecast would be:

910608008 000000000 000000006 000

Note that to insure that both LAMP 3-h and AVN 6-h forecast are valid for the same clock time, the LAMP forecast would be based on a LAMP cycle which is HH-3 while the AVN 6-h forecast would be based on an AVN cycle of HH-6 (see “**Variable List**” under “CONTROL FILE INPUT”).

U870

VERIFIES GRIDDED FORECASTS OF DISCONTINUOUS VARIABLES

David Rudack
Harry R. Glahn
J. Paul Dallavalle
Jerome P. Charba
July 12, 2002

PURPOSE: U870 verifies forecasts of discontinuous variables represented on a rectangular grid and presents scoring results in the form of contingency tables and standard statistical measures (see the APPENDIX). The scoring measures are computed over the entire input grid (by default for each gridpoint), and they, along with the corresponding contingency tables, can also be computed over subsetted grid areas through user control. The forecast input data are gridpoint fields from numerical models, and the verification data are analyses of observed data. Both data types must be on the same grid and in TDLPACK format. Because the gridded forecast data are defined internally in the packed files, they can originate from a single model or multiple models. In addition, constant forecast-type data, also in gridded TDLPACK format, can be input from random access files.

U870 uses a driver program, called DRU870, which provides for setting the fixed dimensions of array variables by PARAMETER statements. The passing of these variables to the main subroutine (called U870) and subsequent subroutines, where they are used with variable dimensions, eliminates the need for multiple versions of the program. U870 is written to run on either a 32-bit or a 64-bit word-length machine. This is also accomplished through PARAMETER statements in the driver.

The contingency tables and corresponding scores for each subsetted grid area are presented in ASCII form in the U870 output file. If specific verification scores are desired, the proper score ID must be placed in ‘U870.CN’.

Some familiarity with other MOS-2000 program write-ups will be necessary for a full understanding of this document.

CONTROL FILE INPUT: 'U870.CN' (Unit = KFILDI)

Record Type 1 - Format (A4,25I3) **Output Control**

This record contains unit numbers for a run, and it can also control the "default" output file number. KFILDI is the input unit number, as specified in DRU870.

IPINIT - four characters, usually a user's initials plus a run number, which are appended to print output filenames that begin with ‘U870.’ The control over the unit numbers for these files and the contents of the files is contained in the description of

IP(J) (see below). The run number allows for multiple runs of U870 that write output files with unique names. For example, with IPINIT = 'DER2' and IP(2) = 40, the file name for Unit No. 40 = 'U870DER240'. DO NOT USE A BLANK FOR ANY OF THE CHARACTERS. (CHARACTER*4)

IP(J) - Each value (J=1,25) indicates whether (>0) or not (=0) certain information will be written to an output file. When IP( ) > 0, the value indicates the unit number for the file. The IP( ) values should not be the same as any other unit numbers used in U870 except possibly KFILDO (the default output file), although a value of one IP( ) can be the same as that for another IP( ). The files contain ASCII text, generally for diagnostic purposes. The output files contain information as described below for individual values J:

(1) = All error diagnostics plus other information not specifically identified with other IP( ) numbers. When IP(1) is read as nonzero, KFILDO, the default output file unit number, will be set to IP(1). When IP(1) is read as zero, KFILDO will be used unchanged, as specified in a DRU870 DATA statement (= 12). Changing the default unit number allows for multiple runs of U870 or other programs within the same directory without overwriting output files.

(2) = The input dates in IDATE( ) exactly as they are read in. When input errors occur, print will be to the default output file unit KFILDO as well as to unit IP(2).

(3) = The output dates in IDATE( ), which may be extended by date spanning. When errors occur, output will be to the default output file unit KFILDO as well as to unit IP(3).

(4) = The station list (call letters only). If input errors occur, the station list will be written to the default output file unit KFILDO as well as to unit IP(4).

(5) = The station directory information. If input errors occur in the station list, the list will be written to the default output file unit KFILDO as well as to unit IP(5).

(6) = The variable IDs exactly as they are read in. This output is useful for checkout; for routine operation, IP(7), IP(8), and/or IP(9) may be more suitable.

(7) = The variable ID list in summary form. If input errors occur, the predictor list will be written to the default output file unit KFILDO as well as to unit IP(7).

(8) = The variable ID list in summary form. This list includes the parsed IDs in IDPARS(,). (IDPARS(,) contains the 15 components of each ID.)

(9) = The variable ID list in summary form. This differs from the print in IP(8) in that IP(9) does not include the parsed ID's in IDPARS(,), but rather includes the information taken from the predictor constant file on unit KFILCP (see below).

(10) = The variable IDs for the first day (Day 1) as read from the archive tapes. This is simply a list of all the packed gridpoint fields on the input files.

(11) = The variable IDs of the archived input fields that are needed in U870, in the order they appear on the archive files for Day 1.

(12) = The I,J positions of the stations on the NGRID grids, together with the call letters and names. NGRID, at this point, is the number of different grid specifications comprising the input data. That is, NGM archived data and MRF archived data will have different grid specifications. After the data are read for the first day in the date list, the I,J positions are computed and saved under the IP(12) unit number. If, during the U870 run, a different grid definition is encountered, it is accommodated automatically, but the corresponding I,J positions are not saved in IP(12). In addition, when IP(12) ≠ 0, each directory record read from a vector input will be output to unit IP(12). For input of hourly data, this creates voluminous output.

(13) = Gridpoint fields. When the variable list indicates gridpoint values are to be written for viewing [JP(1,)>0, Record Type 3], they will be written to unit IP(13).

(14) = Gridpoint fields. When the variable list indicates gridpoint values are to be contoured and written for viewing [JP(2,)>0, Record Type 3], they will be written to unit IP(14).

(16) = Diagnostics for constant and linearization routines (e.g., stations in threshold lists that are not being dealt with in this run).

(17) = The values of IFIND( ), ISTAV( ), and ITIME( ) after Day 1. (See Comments section for explanation.)

(18) = The variables saved in the MOS-2000 Internal Storage System after Day 1.

(23) = Information concerning opening and closing of sequential files.

Record Type 2 - Format (A72) **Run Identification**

This record is used to describe the run.

RUNID - 72 characters of information to describe the run. (CHARACTER*72)

Record Type 3 - Format (7(I10/),F10.0) **Control Parameters**

This record contains a variety of control parameters for the run. Note that each parameter is on a separate line. A brief description may be entered following each

NSKIP - The number of errors that can occur on Day 1 before U870 aborts. Day 3 is usually completed before the abort actually occurs so that the user can see additional results. When no data are found for Day 1, U870 continues and processes data for all subsequent days.

JSTOP - The total number of errors that can occur before the program aborts (see Comments section).

INCCYL - The increment in hours between date/times that are put into IDATE( ) (see Record Type 5) as a result of date spanning in subroutine DATPRO. This variable is also used in determining what data are saved in the Internal MOS-2000 Storage System. The date/times in Record Type 5 must be such that all could be arrived at by successively adding INCCYL to the first date/time in IDATE( ).

NXGMIN - X-coordinate of the lower-left corner of the input grid specified with respect to (1,1) of the full (137X81) LAMP archive grid at 80-km mesh.

NXGMAX - X-coordinate of the upper-right corner of the input grid specified with respect to (1,1) of the full (137X81) LAMP archive grid at 80-km mesh.

NYGMIN - Y-coordinate of the lower-left corner of the input grid specified with respect to (1,1) of the full (137X81) LAMP archive grid at 80-km mesh.

NYGMAX - Y-coordinate of the upper-right corner of the input grid specified with respect to (1,1) of the full (137X81) LAMP archive grid at 80-km mesh.

XMISSS - The value to substitute in the output for the secondary missing value of 9997. Note that the value allows the 9997 to be maintained if desired, set to zero in the case of the implied value of near zero for forecasts, or even set to some other value. If XMISSS is set to 9999 or 9997, 9997 values in the gridded input data are treated as a missing in the scoring. Note that this control does not apply to missing values other than 9997.

Record Type 4 - Format (I3,4XA60) **Date List File**

This record (plus the terminator record) identifies the data set from which the date list is read. Records are read until the terminator KFILDT( ) = 99 is reached. The maximum number of records excluding the terminator record = 1. This Record Type 4 is read by subroutine RDSNAM.

KFILDT - Unit number for the file containing the input date list.

NAMDT - Filename containing the date list. When KFILDT = KFILDI (where KFILDI is the input unit number as specified in DRU870), NAMDT is not used and can be read as ‘DEFAULT’; in this case the date list is read in Record Type 5. (CHARACTER*60)

Record Type 5 - Format (7I10) **Date List**

This group of records specifies the date/times for which verification scores are computed. If KFILDT (read in Record Type 4) ≠ KFILDI, this Record Type 5 is omitted.

IDATE(J) - Initial date list, which may contain negative values indicating date spanning. When a negative occurs, all dates between this value and the previous date are filled in at the increment of hours specified in INCCYL. This input date list is modified in subroutine DATPRO to contain the complete date list with the dates in the spans filled in (J=1,NDATES). Dates are input as YYMMDDHH and then modified to YYYYMMDDHH. This list is read by subroutine RDI , wherein zero values found in the input are removed. The terminator is 99999999. The maximum number of dates excluding the terminator is ND8, which is set in DRU870. For a successful U870 run, data for the first date in the list need not be available. The date/times put into Record Type 5 must be such that all of them could be obtained by successively adding INCCYL (read in Record Type 1) to the first date/time in IDATE( ).

Record Type 6 - Format (2I3,1XA60) **Gridpoint Input Data Files**

This group of records identifies the datasets from which the gridpoint data are read. Records are read until the terminator KFILIN( ) = 99 is reached. Maximum number of records excluding the terminator record = ND6, where the latter is set in DRU870. This Record Type 6 is read by subroutine RDSNAM. During reading this record type J = 1,NUMIN. If all input data are instead read from a constant file, only the terminator is necessary.

KFILIN(J) - Unit number for the input gridpoint file. KFILIN( ) must be < 80 for gridpoint data. Also note that values 97, 99, and 44 through 49 are reserved for other uses.

MODNUM(J) - Source of gridpoint data for this file (e.g., 5 = LAMP gridpoint data, 7 = eta model, 8 = aviation model). This number is used to match with the "DD" in the variable ID.

NAMIN(J) - Filename containing the model input data. (CHARACTER*60)

Note that when the intent is to read the datasets in sequence, they should be read in the corresponding order and all have the same unit number. That is, if 2 years of data are to be used and one year is on one dataset and the other year on another, then the first should immediately precede the second in the list and both should have the same unit number. Actually, the same unit number is not mandatory but is highly recommended.

Record Type 7 - Format (I3,4XA60) **Random Access Files**

This group of records specifies random access datasets from which constant (input) data are read. Records are read until the terminator KFILRA( ) = 99 is reached. The maximum number of records excluding the

terminator record = ND12 (ND12 $\leq$ 6). This Record Type 7 is read by subroutine RDSNAM. During reading this record type J = 1,NUMRA. If no data are read from a constant file, only the terminator is necessary.

KFILRA(J) - Unit number for the random access constant file. (J=1,NUMRA) Unit numbers must be in the range of 44 to 49; see "Restrictions" for more information.

RACESS(J) - Filename corresponding to KFILRA(J). (J=1,NUMRA) (CHARACTER*60)

Record Type 8a - Format (I3,4XA60) **Subset Area(s) For Score Computation**

Record Type 8a (plus the terminator record) specifies the file containing grid area(s) over which are all scores accommodated by U870 are computed. Such records are read until the terminator KFILRR = 99 is reached. The maximum number of records excluding the terminator record = 1. This Record Type 8a is read by subroutine RDSNAM. The file will be opened as “OLD”.

KFILRR - Unit number for the input file.

IREGION - Filename containing this information. (CHARACTER*60)

The file above contains one or more records, each with the format,

Format(4I8),

which specifies a rectangular grid area (in terms of the lower left and upper right corners) over which the verification is to be done. Note that the grid area can encompass the full input grid or any subset thereof. Specific information contained in each record is:

NXMIN - The western-most gridpoint coordinate (in the IX direction) of the verification area specified in terms of the (137X81) AVN archive grid.
NXMAX - The eastern-most gridpoint coordinate (in the IX direction) of the verification area specified in terms of the (137X81) AVN archive grid.
NYMIN - The southern-most gridpoint coordinate (in the JY direction) of the verification area specified in terms of the (137X81) AVN archive grid.
NYMAX - The northern-most gridpoint coordinate (in the JY direction) of the verification area specified in terms of the (137X81) AVN archive grid.

It is important to note that the four grid coordinates that define the verification area are specified in terms of the 137X81 AVN archive grid (80-km mesh) regardless of the actual input grid used. This configuration imposes two restrictions on the input grid. One is its areal domain must equal or lie within the AVN archive grid domain. Another is the grid mesh must be less than or equal to the 80-km mesh used for the AVN archive grid. In fact, U870 converts the coordinates for the verification areas,

which (again) are defined in terms of 80-km AVN archive grid, into the equivalent coordinates for the actual grid mesh used in the input file.

As an example, if the input grid is identical to the AVN archive grid and one would like to compute verification scores only for Alaska, the following values would define the grid area:

43 61 54 21

Note that if a reduced grid area were used for input, the coordinates would still be specified in terms of the AVN archive grid. U870 determines the relationship between the AVN grid and the actual input grid in part from the information provided in Record Type 3. The terminator record 99999999 follows one or more of these records. Note that the maximum number of points (where two points, the lower-left and upper right, define a single rectangular grid area) used in defining of the verification areas is defined by ND13, which is set in DRU860.

When KFILRR is set to KFILDI in Record Type 8a, the grid areas are not contained in the file IREGION. For this option, IREGION can be set to 'DEFAULT', and instead records specifying the grid areas follow the terminator for Record Type 8a. With this option the format and content of the records specifying the grid areas is identical to that in the above file. The terminator record is identical as well.

Record Type 8b - Format (I3,4XA60) **ASCII Output File**

This record (plus the terminator record) specifies the ASCII output file containing the verification scores for grid areas. Records are read until the terminator (KFILAS = 99) is reached. The maximum number of records excluding the terminator record = 1. This Record Type 8b is read by subroutine RDSNAM. This file will be opened as "NEW". If these ASCII data are not to be saved, only the terminator is necessary; in that case a file is not opened and data are not written.

KFILAS - Unit number for the output file.

ASCIFL - Filename containing this data. (CHARACTER*60)

Record Type 9 - Format (I3,4XA60) **Station and Location Files**

This pair of records (plus the terminator record) identifies the file(s) from which the station (or location) information is obtained. While this Record Type 9 is not used in calculating the verification scores, it is necessary to make U870 functional. Records are read until the terminator KFILD( ) = 99 is reached. The maximum number of records excluding the terminator record = 2. This Record Type 9 is read by subroutine RDSNAM. Upon completion of reading this record type J = 1,2.

KFILD(J) - Unit number for the file containing station call letters (J=1) and the station dictionary which holds the latitudes, longitudes, WBAN numbers, elevations, and names for each possible station (J=2). KFILD(1) can be the default input file number, KFILDI, in which case DIRNAM(1) is not used. Also, KFILD(1) can

equal KFILD(2), in which case the call letters list will consist of all stations on the dictionary file; note that both a list of stations and a dictionary cannot exist on unit KFILD(1) = KFILD(2). (Normally, the directory would not reside on KFILDI, but is accommodated if KFILDI = KFILD(2). The input would follow Record Type 10.)

DIRNAM(J) - Filename matching KFILD(J). When KFILD( ) = KFILDI, DIRNAM( ) is not used and can be read as 'DEFAULT'. (CHARACTER*60)

Record Type 10 - Format (7(A8,1X)) **Station List**

This group of records identifies the stations. If KFILD(1) ≠ KFILDI (the default input file), this group is omitted, and the information is taken from Record Type 9. Note that call letters for only one station is necessary to make U870 functional.

CCALL(K) - Call letters (or other 8-character location designator) of stations (or locations) for which output values are desired (K=1,NSTA). The call letters should (normally) be left justified, and if a full 8 characters are not present, CCALL( ) will be blank-filled (b) on the right, e.g., 'OKCbbbbb' could be 'OKC' or 'OKCb'. The terminator is '99999999'. The maximum number of stations excluding the terminator is ND1. (CHARACTER*8)

Record Type 11 - Format (I3,4XA60) **Variable File**

This record (plus the terminator record) identifies the file from which the variable IDs are input. Records are read until the terminator KFILP = 99 is reached. The maximum number of records excluding the terminator record = 1. This Record Type 11 is read by subroutine RDSNAM. To verify a particular variable, the "observation" variable ID record must be first in the "score group" followed by the forecast ID record and finally the score ID record(s). Note that a separate score ID record is needed for each score wanted. Also, since the variables are discontinuous, the observation and forecast IDs require a threshold value in the fourth word of the ID record. A discussion for Record Type 11 under CONTROL FILE INPUT provides details of how a score group is formulated. The APPENDIX provides a discussion of the score ID record content and format for all scores.

KFILP - Unit number for the variable IDs.

PRENAM - Filename corresponding to KFILP. When KFILP = KFILDI, PRENAM is not used and can be read as 'DEFAULT'. (CHARACTER*60)

Record Type 12 - Format (I3,4XA60) **Variable Constants File**

This record (plus the terminator record) identifies the file from which the variable constants are to be taken. (Variable constants include plain language description and gridprinting information.) Records are read until the terminator KFILCP = 99 is reached. Maximum number of records

excluding the terminator record = 1. This Record Type 12 is read by subroutine RDSNAM.

KFILCP - Unit number for the constants.

CONNAM - Filename corresponding to KFILCP. (CHARACTER*60)

Record Type 13 - Format (I9,1XI9,1XI9,1XI3,1XA1,1XI4,1XI3,4X3I2) **Variable List**

This group of records, which is comprised of information identical to that contained in the file read in Record Type 11, contains the variable IDs used in the scoring. When KFILP ≠ KFILDI (Record Type 11), this group is omitted, and the variable list is taken from the file read in Record Type 11. Records are read until the terminator [ID(1,) = 999999] is reached. Maximum number of records excluding the terminator record = ND4, which is set in DRU860. This Record Type 13 is read by subroutine RDPRED2. During reading of this record type N=1,NPRED.

ID(J,N) - The first 3 (J=1,3) words of the variable ID plus the last 3 digits of the 4th word, followed in order by the components of a threshold value (zero for continuous variables) consisting of (1) sign (either minus, or plus or blank for plus, read as A1), (2) 4 digits to follow a decimal point, and (3) 3 digits representing the power of 10 by which to multiply the decimal value just read. For easy reading (only) (1) and (2) above can be separated by a decimal point and (2) and (3) separated by an "E". From these values, the 4th ID word (J=4) is composed.

JP(J,N) - For each variable N, JP(,N) indicates print or no print when ≠ 0 or = 0, respectively, for gridpoint values (J=1) or "zebra" gridprint map (J=2). These values combined with IP(13) and IP(14), allow easy control of output. For instance, all variables could have JP(2,) ≠ 0 and IP(14) ≠ 0 for a diagnostic run. All such print could be turned off by setting IP(14) = 0. Alternatively, IP(14) could be ≠ 0 and a selected gridprint obtained by changing JP(2,) from 0 to ≠ 0.

CONTROL FILE INPUT: (Name read from U870.CN, Record Type 4) (Unit = KFILDT)

Record Type 1 - Format (7I10) **Date List**

When the date list is not provided in file U870.CN (KFILDT ≠ KJILDI, where KFILDT is read in Record Type 4 and KFILDI is the input unit number as specified in DRU870), this group of records determines the date/times for which data are to be input and processed. If KFILDT = KFILDI, this date list file is omitted.

IDATE(J) - Initial date list, which may contain negative values indicating date spanning. When a negative occurs, all dates between this value and the previous date are filled in at the increment of hours specified in INCCYL. This input date list is modified in subroutine DATPRO to contain the complete date list with the dates in the spans filled in (J=1,NDATES). Dates are input as YYMMDDHH and then modified to YYYYMMDDHH. This list is read by

subroutine RDI, wherein the latter removes zero values found in the input. The terminator is 99999999. The maximum number of dates excluding the terminator is ND8, which is set in DRU870.

CONTROL FILE INPUT: (Name read from U860.CN, Record Type 8a) (Unit = KFILRR)

Record Type 1 - Format (4I4) **Grid areas for Score Computation**

When the grid areas for score computation is not provided in U870.CN (KFILRR ≠ KFILDI in Record Type 8a and KFILDI is the unit number as specified in DRU860), this group of records specifies the grid areas of concern. The format and content of the records (including the terminator record) is identical to that as discussed previously for Record Type 8a.

CONTROL FILE INPUT: (Name read from U870.CN, Record Type 9) (Unit = KFILD(1))

Record Type 1 - Format (7(A8,1X)) **Station List**

This file is not needed when KFILD(1) = KFILDI, where KFILD(1) is read in Record Type 9 and KFILDI is the input unit number defined in DRU870. In this case, the call letters are read from KFILDI. It is also not needed when KFILD(1) = KFILD(2); for this highly unusual case, both the station list and the associated information are read according to the format for the subsequent control file description for Unit No. KFILD(2).

CCALL(K) - Call letters (or other 8-character location designator) of stations (or locations) desired (K=1,NSTA). This list is read with subroutine RDC, which eliminates any blanks found in the input. Terminator is '99999999'. Maximum number of stations excluding the terminator is ND1. (See Record Type 9, Control File 'U870.CN', unit KFILDI for additional information.) (CHARACTER*8)

CONTROL FILE INPUT: (Name read from U870.CN, Record Type 9) (Unit = KFILD(2))

Record Type 1 - Format **Station Locations**
(A8,1XA8,1XA17,4XA2,8XI5,1XA1,F7.4,1XA1,F8.4,9XA8,3(1XA8),12XI5)

This group of records provides information about the stations (or locations). It is needed unless KFILD(1) = KFILD(2) = KFILDI, where KFILD(2) is read in Record Type 9; for this highly unusual case, both the station list and the associated information are read from the input control file KFILDI. When KFILD(1) = KFILD(2) ≠ KFILDI, this control file is the same as the one above and both the station list and the associated information are taken from this file. If both files are needed (the usual case; KFILD(1) ≠ KFILD(2) ≠ KFILDI), the call letters here are matched with those in the station list and the appropriate information extracted.

CCALLD(K,J) - Call letters (or other character location designator) of stations (or locations) (J=1). As stated above, these call letters are matched with those in the station list unless KFILD(1) = KFILD(2), in which case these call letters comprise the station list and at the completion of reading this file, K=1,NSTA. When NEW = 1, CCALLD(K,1) is read from the first

field (A8) and CCALLD(K,2) is read from the second field (A4). When NEW ≠ 1, CCALLD(K,1) is read from the second field and CCALLD(K,2) is read from the first field.

NAME(K) - 20-character name of station. This is used for visual identification of the station in certain output. Format is A17,4XA2; this provides for a 17-character name, 4 blanks, and a 2-character state abbreviation. Note that the last three characters in the "name" field in the directory are not used. (CHARACTER*20)

NELEV(K) - Elevation of the station. Format is I5.

SIGNLA - Sign of the latitude of the station, read as either "S" or "N". When read as "S", the latitude is set negative indicating South Latitude. Format is A1.

XLAT - Latitude in degrees. Format is F7.4.

SIGNLO - Sign of the longitude of the station, read as either "E" or "W". When read as "E", the longitude is modified to make all longitudes West. That is, longitude will range from 0 through 360 and be in degrees West over the United States. Format is A1.

LONDD - Longitude in degrees west. Format is F8.4.

CCALLD(K,J) - Call letters of substitute stations (or locations) (J=3,6) (K=1,NSTA).

IWBAN(K) - The WBAN number of the station. Format is I5.

The number of stations in this directory is not limited, except when it also constitutes the list to be used U870, in which case the number of entries is limited by ND1-1. No terminator is used.

CONTROL FILE INPUT: (Name read from U870.CN, Record Type 11) (Unit = KFILP)

Record Type 1 - Format (I9,1XI9,1XI9,1XI3,1XA1,1XI4,1XI3,4X3I2) **Variable List**

This group of records contains the variable ID records for a score group. The first record of the score group contains IDs for the verifying “observation” and the second record contains the IDs for the forecast. The correct order of these ID records and associated thresholds is critical since U870 will run to completion even when the order is reversed. The third set of record(s) for the score group identifies the score(s)s to be computed and printed to the output file KFILDO. Note that a score ID record is needed for each score wanted. Also, if the score ID record(s) is(are) omitted, only the contingency table corresponding to the score group and area is written to the output file KFILDO.

When KFILP = KFILDI in Record Type 11, this file is omitted, and the variable list is taken from input file KFILDI (file ‘U870.CN’, Record Type 13. Each score group is separated by ‘888888', and separate score groups are read until the terminator ID(1,) = 999999 is reached. The maximum number of records excluding the terminator record = ND4. This

Record Type 1 is read by subroutine RDPRED2. During reading of this record type, N=1,NPRED. See Record Type 13, file 'U870.CN', unit KFILDI, for other details, as the format is the same as for that record type.

CONTROL FILE INPUT: (Name read from U870.CN, Record Type 12) (Unit = KFILCP)

Record Type 1 - Format
(3(I9,1X),I3,2XA32,I3,F11.4,F10.4,F9.2,F8.2,2XA12) **Variable Constants**

This group of records contains information about the variables, as defined by ID(J,) (read previously) and IDTEMP(J). This file should be universally useable by all MOS-2000 users and is expected to be a separate file; that is, while KFILD(2) could = KFILDI, it would be very unusual for that to be the case. Note that the format matches that for a file input to other programs such as U600, U660, and U850.

IDTEMP(1) - First word of the variable ID, either with or without the "B" and "DD". This is matched with all variables read for this run, both with and without the "B" and "DD". When there is a match, the constant information with IDTEMP(1) is stored as indicated below.

IDTEMP(J) - These 3 words (J=2,4) are currently not used, but are meant to correspond to the ID words 2-4.

PLAINT - When IDTEMP(1) matches an ID(1,N), PLAINT is stored in PLAIN(N). These 32 characters are used for visual identification of variables in certain output. Although 32 characters are allowed, the first 5 are reserved for a height indicator (e.g., 1000-), and those after character 23 may be overwritten for vertical or time processing. This generally leaves 18 characters besides height, smoothing, and other processing indicators. (CHARACTER*32)

ISCALT - When IDTEMP(1) matches an ID(1,N), ISCALT is stored in ISCALD(N). This variable is the scaling constant (power of 10) to use when packing MOS-2000 data, in this case, the verification scores.

SMULTT - When IDTEMP(1) matches an ID(1,N), SMULTT is stored in SMULT(N). This variable is the multiplicative factor (power of 10) to use when gridprinting the gridpoint data.

SADDT - When IDTEMP(1) matches an ID(1,N), SADDT is stored in SADD(N). This variable is the additive factor to use when gridprinting the gridpoint data.

CONTT - When IDTEMP(1) matches an ID(1,N), CONTT is stored in CINT(N). This variable is the contour interval to use when gridprinting the data. It applies to the units in which the data exist on the packed input tapes, not after manipulation by SMULTT and SADDT.

ORIGNT - When IDTEMP(1) matches an ID(1,N), ORIGNT is stored in ORIGIN(N). This variable is the contour origin to use when gridprinting the data It applies to the units in which the data exist on the packed input tapes, not after manipulation by SMULTT and SADDT.

UNITST - When IDTEMP(1) matches an ID(1,N), UNITST is stored in UNITST(N). These characters define the units of the data after application of SMULTT and SADDT and are used in the visual inspection of the data in gridprinted maps. (CHARACTER*12)

NVRBLS(M) - Total number of observation and forecast IDs in a score group (M=1,IGROUP).

NSCORES(M) - Total number of score IDs in a score group (M=1,IGROUP).

Other processing occurs with the reading of these records in RDPRED2, much of it associated with the plain language description. See Chapter 4 of Office Note 00-1, Variable Identification, for details.

DATA INPUT:

All archived model data input to U870 must be in the MOS-2000 TDLPACK format read from sequential files. The unit numbers and file names are provided in KFILIN(J) and NAMIN(J) (J=1,NUMIN), respectively, of ‘U870.CN’ Record Type 6. Constant data are input from the random access MOS-2000 External File System ('U870.CN', Record Type 7); these data are also in TDLPACK format, except for the call letters (directory) record. Reading is done with standard FORTRAN unformatted reads, and unpacking is done with subroutine UNPACK and its associated subroutine UNPKBG. Details of these data inputs are provided below.

A. SEQUENTIAL GRIDPOINT DATA

One or more sources of gridpoint data are accommodated, with the filenames and unit numbers provided in NAMIN(J) and KFILIN(J), respectively, as obtained from the control file 'U870.CN', Record Type 6. Each file is closed when an EOF is reached.

The definition of the grid to which the data in each record pertain is contained in that same record. In fact, not only can the data sources have different grid definitions, the definition within a particular source can change once or many times. For instance, it might be normal for the grid definition to change during the archive period. Beware, though, that each time the grid definition changes when reading records that are to be used, extra computation must be done. Normally, ND11 (see "Setting up the Driver DRU870") will be $\geq$ the number of input grid data sources (unit numbers).

It is important to note that U870 will only verify observations and forecasts having identical grid characteristics. If the grid definitions between the observations and forecasts are not identical for all dates in the date list, U870 notifies the user of the mismatch and terminates.

For gridpoint data, the unit number KFILIN( ) must be < 80, and the model number(s) MODNUM( ) in Record Type 6 must match the model number(s) (DDs) in Record Type 11 or 13. Also, the RR in the latter (variable ID) record types function as “lookback” feature.

B. RANDOM ACCESS GRIDDED AND VECTOR DATA

Up to five sources of station-oriented random access data and one source of gridpoint random access data are accommodated. These data must reside in the MOS-2000 External Random Access File System. These data are accessed with prescribed unit numbers, depending on the type of data; see "Restrictions" for more information.

DATA OUTPUT:

Along with the usual diagnostics and other information written to units KFILDO and IP( ), U870 outputs the contingency tables and statistical scores for each group and area. These scores can be found near the bottom of KFILDO.

A. ASCII FOR VIEWING OR PRINTING

Gridprinted maps of each variable for which JP(1,) ≠ 0 without contours and/or for which JP(2,) ≠ 0 with contours are written to Units IP(13) and IP(14), respectively, when those units numbers ≠ 0. This makes for easy control; all such print can be eliminated by changing the IP( ) values, and there is still control on each variable J through JP(,J).

B. ASCII FILE CONTAINING STATISTICAL SCORES FOR GRID AREAS(S)

All scores accommodated by U870 (see APPENDIX), which apply to the entire input grid or subset grid area(s) thereof, are written to KFILAS. The data are written to the dataset whose name is in ASCIFL and unit number in KFILAS, as specified from the control file 'U870.CN', Record Type 8b, unless KFILAS = 0, in which case data are not output.

When KFILAS ≠ 0 , the ASCII information written to this file takes the following form and format:

(a) Header descriptors: (Projection, Group, Area, Score, and Value)

Format - (A1,3X,A10,4X,A5,3X,A4,7X,A5,11X,5A,/A1)

(b) Score measures:

1.) Bias by category
2.) Percent of forecasts correct
3.) Heidke skill score
4.) Probability of detection
5.) False alarm ratio
6.) Threat score

Format - 6((7X,I2,11X,I2,7X,I2,7X,A4,6X,F7.3)/)

EXAMPLE CONTROL FILE: 'U870.CN'

Two example control files named 'U870.CN' and ‘U870_AREAS-IN-FILE.CN’ in directory '/lamp2000/dru870/drU870/standard_CN'. The first example file specifies the grid areas for verification in a separate file, while the latter file specifies these grid areas in in-line records. The easiest way to set up a run is to take such an existing control file and modify it; DO NOT START FROM SCRATCH.

OUTPUT:

Output that can be printed can be put into one or more files, as described above under control file 'U870.CN', Record Type 1 and the definition of KFILDO in the driver DRU870. All errors will be contained in the default output file (KFILDO as possibly modified by IP(1)) as well as possibly to other files as defined by IP( ). Every effort has been made to notify the user of problems and potential problems and to proceed under user control without jeopardizing data.

RESTRICTIONS

Most restrictions are associated only with variables in PARAMETER statements in the driver which control array sizes. In some instances, machine wordlength is a factor, and 32-bit and 64-bit machines have been provided for through PARAMETER statements and the setting of L3264B to either 32 or 64. Only northern hemisphere polar stereographic and Lambert gridpoint data are accommodated. Formats and other guidelines in other MOS-2000 documents are followed.

HP workstations accommodate optionally-compiled Fortran statements containing a "D" in Column 1. The IBM-SP does not allow this. It is best to replace the "D" with, for instance, "C****" in Columns 1-5. This way, the possible optional statements can be spotted and be readily made operative.

The TDLPACK format for gridpoint data provides for a primary missing data indicator (=9999), although gridpoint data are generally not expected to contain missing values. If a primary missing value is found during unpacking of gridpoint data (for either the observations or forecasts or both), it is flagged by U870. Upon completion U870 notifies the user (with a message in the KFILDO output file) that primary missing values have been found. If primary missing values are encountered, U870 ensures that the gridpoints containing the missing data do not enter into the scoring tabulations for that particular day’s data. If secondary missing values (=9997) are encountered during processing, the user has the option of treating them as missing values, leaving them unchanged, or substituting a different value that is used as a valid data value in the scoring. Regardless of the choice selected, the new value is substituted by setting XMISSS in the U870 control file. When XMISSS is set to 9999, the 9997 value is treated as missing, but if XMISSS is set to 9997, this value is treated as a valid datum. **It is important to note that gridpoints with missing values encountered in the scoring of one forecast variable are also removed in the scoring for a subsequent forecast variable and vice**

versa. This approach ensures a matched sample in the scoring of all forecast variables in a particular U870 run. Thus, if matched samples among forecast variables are not needed, each forecast variable should be scored in a separate U870 run. **It is also important to note that if comparative scores are wanted for separate forecast experiments for the same variable (e.g., some forecast model parameters are altered from one forecast experiment to another), the variable ID must be unique for each forecast experiment involved.** Provision for a unique variable ID in this case is accommodated in the U150.CN control file, wherein a U150 variable called INCDD is set. Setting this variable to a nonzero value is used to provide a unique DD value in the associated TDLPACK gridpoint output files (see APPENDIX for variable ID structure).

The various arrays in U870 have specific uses. Arrays FD1( ), FD2( ), ..., FD7( ) have been provided as general work arrays. Arrays FDVERT( ) and FDTIME( ) have been reserved for use by subroutines VERTP and TIMEP, respectively. FDSINS( ) and FDMS( ) have been reserved for use by routines involving the sine of the latitude and the map scale factor for the input grid. L1DATA( ) and SDATA1( ) are reserved for use by linearization and constant routines. If modifications or additions are made to U870, care must be taken that information in the work arrays is not overwritten by new called routines. For instance, if a new routine were to call OPTN2 and the latter called VORTH, then VORTH would use arrays FD1( ), FD2( ), and FD3( ). Thus, the data the new calling routine had stored in these arrays could be overwritten.

Regarding the gridded data input to U870, it not known *a priori* whether the computational elements of the input variable ID (e.g., "B" or "RR") have already been satisfied in the input data or whether these elements must still be satisfied. U870 initially attempts to find the exact variable ID in the input data, and if it does it is assumed these elements are already satisfied. If U870 does not find the exact field but instead finds a more basic version, it will perform the additional operations needed. If neither field is found, of course the program will abort.

When the runtime offset, RR, and MODNUM( ) are not zero, meaning that data from a previous model run are needed, then data are carried forward in the MOS-2000 Internal Storage System. In the storing of data, as the file is read from a previous cycle to the current one, some data from an intermediate cycle may be saved that are not needed. Since U870 verifies data as a function of valid time, RR must be greater than zero in order to retrieve forecast data. Consequently, in the 3rd word of the forecast ID, RR = $\tau\tau\tau$.

The unit numbers for the random access files must be in the range 44 through 49 (although U870 will not write to a random access file); those unit numbers are used for the following purposes, where the CCC values in ID(1) denote the following:

Unit No.	CCC Range	Use
44	400-499	Gridpoint data
45	400-499	"True" constants (rel. freq., means, etc.)
46	500-599	1-d and 2-d constants, probably for U201
47	800-899	Thresholds for best category forecasts, etc.

48	200-299	Forecasts read only
49	200-299	Forecasts read/write

COMMENTS

Most errors are output for printing starting with **** to the file with number designated by IP( ) (see Record Type 1) and a count is kept for matching with NSKIP and JSTOP (see Record Type 3). Missing data will usually not be counted as an error, but a diagnostic message is provided. It is not always obvious what is truly an error as opposed to something that might be expected to happen occasionally; therefore, the count cannot be considered absolute. When a field is to be computed in OPTION and a field needed for the computation cannot be found, a diagnostic message (possibly "****PREDICTOR NOT IDENTIFIED IN OPTN2") and in addition "****ALL VALUES IN FIELD...ARE MISSING IN PKMS99," will be provided. While these diagnostics could be eliminated, it is thought best to keep a watchful eye for such errors. A good practice to follow is to provide a set of dates to U870 for which it is known valid data exist. Then any diagnostic is really unexpected. At the end of Day 1 and at the completion of the run, the number of "errors" and the number of missing variables (data records) are printed. Note that an error may result in missing data values and be counted for both.

On Day 1, GFETCH is entered directly at least once for every variable ("Day 1 is a term that has come to be used for the "first case." It is actually the first cycle of the first day.) A return of IER = 47 (which means data were not found in GFETCH) is not counted as an error in PRED81 for Day 1. It is counted as an error in PRED82, because GFETCH should not be entered directly when OPTION is needed.

Computational routines are written and available for U870. Also, a few have been written primarily for illustrative purposes. For instance, if verification is to be performed on ceiling height and this field is not provided, the subroutine OBSCIGHT can be invoked through OPTION.

Some information that may seem repetitive is provided for printout on each run. However, it is believed the user should monitor this information and investigate apparent abnormalities. For instance, subroutine GCPAC prints compression information for the first 3 days. This will let the user determine whether the CORE( ) space provided for storage is far larger, or smaller, than needed, etc. If CORE( ) is small, extensive disk access will be necessary. If it is large, then other users or swapping space for the current run may be impacted. Generally, it is hoped CORE( ) can be about the size to hold the intermediate storage after Day 1.

It is unlikely that when no variable in a run has a particular model number, that the file containing that model will be needed; therefore, the file is closed. Closing the file keeps the data from being read when not needed. A highly unlikely exception occurs when a subroutine like COMBIN could use data from a model that was not indicated by the model number in the variable ID. If this is the case, include a variable with the (otherwise omitted) model number to force the file to remain open. This variable can be a “dummy” (see OPTION subroutine) to keep it from being written.

It is generally not necessary that each variable be present for Day 1 for that variable to be used on subsequent days (see the following section "Characteristics of the U870 Lookback Feature").

A count is kept of each time a missing value indicator is found when unpacking grids. While missing values are generally unexpected, at the end of a run the missing count is printed if non-zero.

The variables IFIND( ), ISTAV( ), and ITIME( ), which can be printed for each input variable in table form under IP(17) control, are set after processing Day 1 data and have the following meanings:

IFIND = 1 When the variable can be found directly from the input and does not have to be computed through OPTION.
= 0 Variable has been identified in OPTION and will be computed there.
= 2 Variable has been identified in OPTION but data returned are missing (?). Therefore, it is not yet known whether this variable will be input directly or computed through OPTION.

ISTAV = 1 When the data have vector format either on initial input or on return from OPTION. (Not applicable to U870)
= 0 Variable computed from gridded data.
= 2 Variable has not been found yet and it is still not known whether its input will be vector or gridded.

ITIME = 1 When the time offset RR is operative.
= 0 Otherwise. This may (also) occur when RR = 0.

CHARACTERISTICS OF THE U870 LOOKBACK FEATURE

U870 attempts to save data from previous cycles (date/times previous to NDATE, the date/time being processed). Such data are denoted by RR > 0 in ID(3) of the forecast variable (second variable ID of a score group). Consider the following conditions:

- The variable needed is in the ID list input by the user in Record Type 13.

  - All data needed are found for Day 1

    RR = 0 No data are saved from one NDATE to the next.

    RR > 0 Data for the variable with RR > 0 are saved for the number of hours indicated by RR.

  - Some data needed are not found for Day 1

    RR = 0 Not a problem.

    RR > 0 Usually not a problem.

- The variable needed <u>is not in</u> the ID list input by the user, but is needed by a computational subroutine.

  - All data needed are found for Day 1

    RR $\geq$ 0 Shouldn't be a problem, <u>provided the subroutine looks for all variables it will need for subsequent cycles</u>. This may require the subroutine to ask for data not needed for the particular cycle being processed (e.g., max temperature is not calculated for each cycle).

  - Some or all data needed for Day 1 are not found.

    RR $\geq$ 0 Shouldn’t be a problem; U870 continues to process data and calculate scores for all subsequent days. All acceptable processed dates in U870 are listed near the bottom of KFILDO.

While considerable effort has been made to make the lookback feature robust, and to not require the user to specify (possibly in a separate list) all the variables that might be needed in all subroutines, not all situations could be handled. It is believed that all will be well when all data needed for Day 1 are present. It should be rare when one would need to make a run that would require data that were not available for the first date/time.

SETTING UP THE DRIVER DRU870

The preparation of the driver for a particular U870 run is relatively painless; it consists of using a template driver and modifying certain PARAMETER statements as necessary. These statements set values of:

L3264B - Set to 32 for a 32-bit machine (e.g., the HP's) or 64 for a 64-bit machine (e.g., some options on the IBM-SP).

ND1 - Maximum number of stations (or points) that can be dealt with. Note that this does not include the number of stations in the directory (read on Unit No. KFILD(2)) unless, of course, the station directory is to be used as the station list. ND1 must also be GE NBLOCK (see below).

ND2 - ND2*ND3 is the maximum size of the grid that can be dealt with. ND2 and ND3 are set separately to highlight the possible dimension of the grids. However, in the called routines, the size is only limited by the product, not each dimension individually.

ND3 - See ND2.

ND4 - The maximum number of variables that can be entered into ‘U870.CN’.

ND5 - Dimension of IPACK( ), IWORK( ), and DATA( ). ND2X3 and ND5 should be set in the driver DRU870:

ND2X3 = MAX(ND1,ND2*ND3,ND12)
ND5=ND2x3

ND6 - Maximum number of sequential file input sources (e.g., models) that can be dealt with. If data from a model are on two files, then this would be counted as two, not one, etc.

ND7 - The size of ISO( ), IS1( ), IS2( ), and IS4( ). This would normally be 54.

ND8 - The maximum number of date/times that can be used. This is the "extended" date list, not just the values read in.

ND9 - The maximum number of fields (variables) stored in the MOS-2000 Internal Storage System. Since all fields are stored for Day 1, ND9 must be large enough to hold all records of the date/time of Day 1 from all input files.

ND10 - The number of words of storage provided in the variable CORE( ) for the MOS-2000 Internal Storage System. When this is filled, a scratch disk file is used. Too small a number will result in more disk accesses than necessary (although caching may alleviate that); too large a number will result in wasted memory and possible excess paging.

ND11 - The maximum number of grid combinations that can be dealt with. For instance, if the NGM and AVN data are being used, and the grids are different, then ND11 would be $\geq$ 2.

ND12 - Maximum number of random access files (MOS-2000 External File System) that can be used. Limit is 6.

ND13 - Maximum number of points used in defining subset areas (see description for Record Type 8a).

ND14 - Maximum number of score groups used in a run of U870.

Do not change the computation for the variable L3264W, and probably not for NBLOCK. NBLOCK is the block size in words for disk records for the MOS-2000 Internal Storage System. (Experimentation may determine that a larger value is desirable.) Note that ND1 must be GE NBLOCK.

The user can see from the template what effect each of these values has on storage from where it occurs in the DIMENSION statements. Some have relatively little effect (e.g., ND7), while others have considerable effect (e.g., ND2*ND3). Every effort has been made so that the variables will not be overflowed if values too small are used; however, some danger may still exist.

The "startup" control file "U870.CN" is opened in DRU870, so that this aspect of U870 can be rather easily changed without recompiling or having a separate version of the main (sub)routine U870. This will be different on the IBM-SP where unit/file assignments are made differently, and possibly when U870 is run in batch mode.

TYPES OF COMPUTATIONS IN U870 USING GRIDDED DATA

U870 will perform the basic functions specified in the variable ID (TDL Office Note 00-1), as summarized below. It is expected only polar stereographic and Lambert grid archives will be used, and only those grid projections are accommodated.

PROCESSING INDICATOR

DD - Model number
- 1 = Trajectory
- 5 = LAMP
- 6 = NGM
- 7 = Eta
- 8 = AVN
- 9 = MRF

V - Vertical Processing [Same field (CCCFFF) and projection (ttt)], but with different levels (UUUU and LLLL)

VERTP
- 0 = None (not used)
- 1 = Difference (UUUU-LLLL) (not used)
- 2 = Sum (not used)
- 3 = Mean (not used)

RR, O, HH - Time Operation on Fields (1) and (2)

TIMEP

Field (1) = Run time offset (NDATE - RR), projection tau + HH
Field (2) = Run time offset (NDATE - RR), projection tau
- 1 = Mean (not used)
- 2 = Difference (1)-(2) (not used)
- 3 = Maximum (not used)
- 4 = Minimum (not used)

Field (1) = Run time offset (NDATE - RR), projection tau + HH
Field (2) = Current run time, projection tau
- 5 = Mean (not used
- 6 = Difference (2)-(1) (not used)
- 7 = Maximum (not used)
- 8 = Minimum (not used)

S - Smoothing
- 1 = 5-point
  SMTH5
- 2 = 9-point
  SMTH9
- 3 = 25-point
  SMTH25

The processing is always done in the following order:

V,
RR, O, HH,
and S,

When **OPTION** is entered, the data returned are also gridpoint (in which case S is operative).

VARIABLES HANDLED IN U870

Direct model output or derived variables such as ceiling height or cloud amount may be verified using U870. If derived variables are to be processed, U870 calls the subroutine OPTION to derive these data fields, and it then proceeds to calculate the desired verification scores.

WRITING NEW COMPUTATIONAL SUBROUTINES

New computational routines can be written as needed. Existing routines, such as **VORTH**, can be used as templates. Generally, the steps to be followed are:

1) Include the routine in the u870lib directory, and see that it is included in the makefile.

2) Enter the routine name with its calling sequence in the switching routine **OPTION** and probably in **OPTN2** with the proper IF tests. Generally, this switching would be on IDPARS(1) and/or IDPARS(2).

3) Use the work arrays FD1( ), FD2( ), etc. as needed.

4) The new routine can call **OPTN2** (or even **OPTION**) if computed variables are needed for which routines are already written. If this is done, the complete **OPTN2** calling sequence must be used for the new routine, so that it can be passed to **OPTN2**.

5) If **OPTN2** is called or **OPTION** recalled, care must be taken to not overlap the use of the arrays FD1( ), FD2( ), etc. That is, if **VORTH** were to be called by **OPTN2**, it would use arrays FD1( ), FD2( ), and FD3( ). If these arrays were used by the new routine, they would be wiped out when **OPTN2** and **VORTH** were called. However, data are never carried from variable to variable in FD1( ), etc. That is, each time a new variable is dealt with in **OPTION**, each of the FDi( ) series is available for use. It is only in the computation of a particular variable that the work space must be managed by the user. Arrays NELEV( ), STALAT( ), and STALON( ) hold the station elevations, latitudes, and longitudes, respectively, in the order of the stations in CCALL(,). These arrays can be used in any subroutine, and, of course, they shouldn't be modified. Other "constant" data must be obtained from the MOS-2000 External Direct Access Storage System by calling subroutine **CONST**. Station elevations can also be obtained through **CONST**, if desired.

(6) When a computational routine returns gridded data, that routine must set ISTAV = 0.

(7) As a special case, when data for a particular hour cannot be available (e.g., max/min temperature at more than one hour each per day), the computational routine can set IER = -1 to indicate a record is not to be written for that variable for that date/time.

(8) **OPTION** will produce a diagnostic as appropriate when a called routine does not return IER = 0. The computational routine probably does not need an additional diagnostic. Also, the routine should not return IER = 47 (indicating missing data), but something else coordinated with the error table for MOS-2000. **OPTION**, **PRED87**, and **PRED88** will then be able to handle the situation.

NONSYSTEM ROUTINES USED

Use the load line in file /lamp2000/drU870/U870.com along with the libraries /home21/tdllib/u870lib and /home21/tdllib/moslib, all on blizzard.

LANGUAGE: FORTRAN77 with some HP extensions.

LOCATION: u870lib. The driver is /lamp2000/dru870/dru870.f.

APPENDIX

SCORE DESCRIPTION

As indicated in the discussion for Record Type 11 and 14, the computation of scores is controlled by ID variables comprising a score group. Each score group controls the scoring of a single forecast variable. U870 checks for primary (=9999) and secondary (9997) missing gridpoint values in the forecast field. If either of these values are encountered, U870 does not process that gridpoint value. However, if the user chooses to replace secondary values with 0.0, U870 treats these gridpoint values as any other valid forecast value.

VERIFICATION SCORES:

U870 computes verification scores for discontinuous scalar forecast fields such as cloud amount and ceiling height. The ID records that identify the forecast and observed variables also contains a threshold value that is used to convert these variables into categorical variables. When the conversion involves two ensuing categories only one ID record (that includes the associated threshold value) for each forecast and observed variable is needed. Also, a switch (‘B” value) in the ID record with the value of 1 (2) specifies that the event occurrence (1 value) applies to forecast or observed values greater than or equal (less than) to the threshold value.

When the forecast and observed are based on two categories and a particular sub-set grid area is specified, U870 forms a 2X2 contingency table based on points which have non-missing forecast and observed data (both). From the table, the following scores can be computed: Bias, Threat score, Probability of Detection, False Alarm Ratio, Percent Correct and Heidke Skill Score. Each of these scores is defined by an ID record that must be entered in Record Type 11 or 13.

If the number of non-missing points is “n”, the 2X2 contingency table has the following form:

		Observed event Yes	No	Total
Forecast	Yes	a	b	a + b
Event	No	c	d	c + d
	Total	a + c	b + d	n=a+b+c+d

The score measures computed from the table are defined as:

$$\text{Bias} = \frac{a + b}{a + c} \quad \text{(set to 998.0, when } a + c = 0\text{)}$$

$$\text{Threat score (TS)} = \frac{a}{a + b + c}$$

$$\text{Probability of detection (POD)} = \frac{a}{a + c}$$

$$\text{False alarm ratio (FAR)} = \frac{b}{a + b}$$

$$\text{Percent correct (PC)} = \frac{a + d}{a + b + c + d}$$

$$\text{SkillScore (HSS)} = \frac{2(ad - bc)}{(a + c)(c + d) + (a + b)(b + d)}$$

Note that when no observed events appear in the sample, U870 sets the bias value to 998.0. If the Heidke skill score cannot be determined (because the denominator in (f) equals zero), the HSS is set to 9999.0

When scoring for m categories, where m > 2, the forecasts and observations are tabulated in an m x m contingency table. In this case, the binary portion of the first ID word (‘B’) for the observations and forecasts is set to 3. Also, a separate ID record for these variables is needed for each threshold applied. With B=3, U870 computes only two scores, bias by category and the Heidke Skill Score. Again, if these scores are to be output, the appropriate ID records must be provided.

The contingency table for m x m categories contains information from which the scores in this group (bias and HSS) are computed. The element X_{ij} in the table is the number of times the forecast was in the jth category and the observation was in the ith category. The row and column totals are also shown with the subscript p.

		Forecast Category				
		1	2	...	m	Total
	1	X_{11}	X_{12}	...	X_{1m}	X_{1p}
Observed	2	X_{21}	X_{22}	...	X_{2m}	X_{2p}
	.	.	.		.	.
Category	.	.	.		.	.
	.	.	.		.	.
	m	X_{m1}	X_{m2}	...	X_{mm}	X_{mp}
	Total	X_{p1}	X_{p2}	...	X_{pm}	X_{pp}

$$\text{Bias}_i = \frac{X_{pi}}{X_{ip}} = \text{bias for category i.}$$

$$\text{HSS} = \frac{NC - E}{T - E} = \text{Heidke Skill Score, where}$$

$$\text{number correct (NC)} = \sum_{i=1}^{m} X_{ii}$$

$$T = X_{pp}$$

$$E = \sum_{i=1}^{m}(X_{ip}X_{pi})/T$$

Note that if (for some reason) a particular category has no event occurrences, U870 assigns the bias a value of 998.0 to that category. Similarly, if both the observations and forecasts are missing for a particular category, the bias for that category is assigned a value of 999.0.

SCORE ID FORMAT:

The variable ID structure for scores in U870 is:

9ccfff0DD 000000000 000000ttt 0000000000

where 9 indicates U870 output, cc the meteorological variable, fff the verification measure, DD the data source, and ttt the forecast projection (see Office Note 00-1). [Note that the second ID word is identical to the second ID word of the variable being verified, and the binary indicator (‘B’), RR, and threshold value (‘WXXXXYY’) is provided in the record IDs for the observations and forecasts.]

The cc value specifies the variable, where for example:

80 = Ceiling height (100s of feet)
82 = Sky cover (octas)

The fff value specifies the score or associated statistic, as follows:

800 = Bias by category
801 = Percent correct of forecasts
802 = Heidke skill score
803 = Probability of detection
804 = False alarm ratio
805 = Threat score

As an example, the four-word ID for the Heidke skill score corresponding to a 3-h forecast of total sky cover would be:

982802005 000000000 000000003 0000000000