Chapter 7: Objective Analysis (OBSGRID)

Table of Contents

Introduction

The goal of objective analysis in meteorological modeling is to improve meteorological analyses (the *first guess*) on the mesoscale grid by incorporating information from observations. Traditionally, these observations have been "direct" observations of temperature, humidity, and wind from surface and radiosonde reports. As remote sensing techniques come of age, more and more "indirect" observations are available for researchers and operational modelers. Effective use of these indirect observations for objective analysis is not a trivial task. Methods commonly employed for indirect observations include three-dimensional or four-dimensional variational techniques ("3DVAR" and "4DVAR", respectively), which can be used for direct observations as well.

This chapter discusses the objective analysis program, OBSGRID. Discussion of variational techniques (*WRFDA*) can be found in Chapter 6 of this User's Guide.

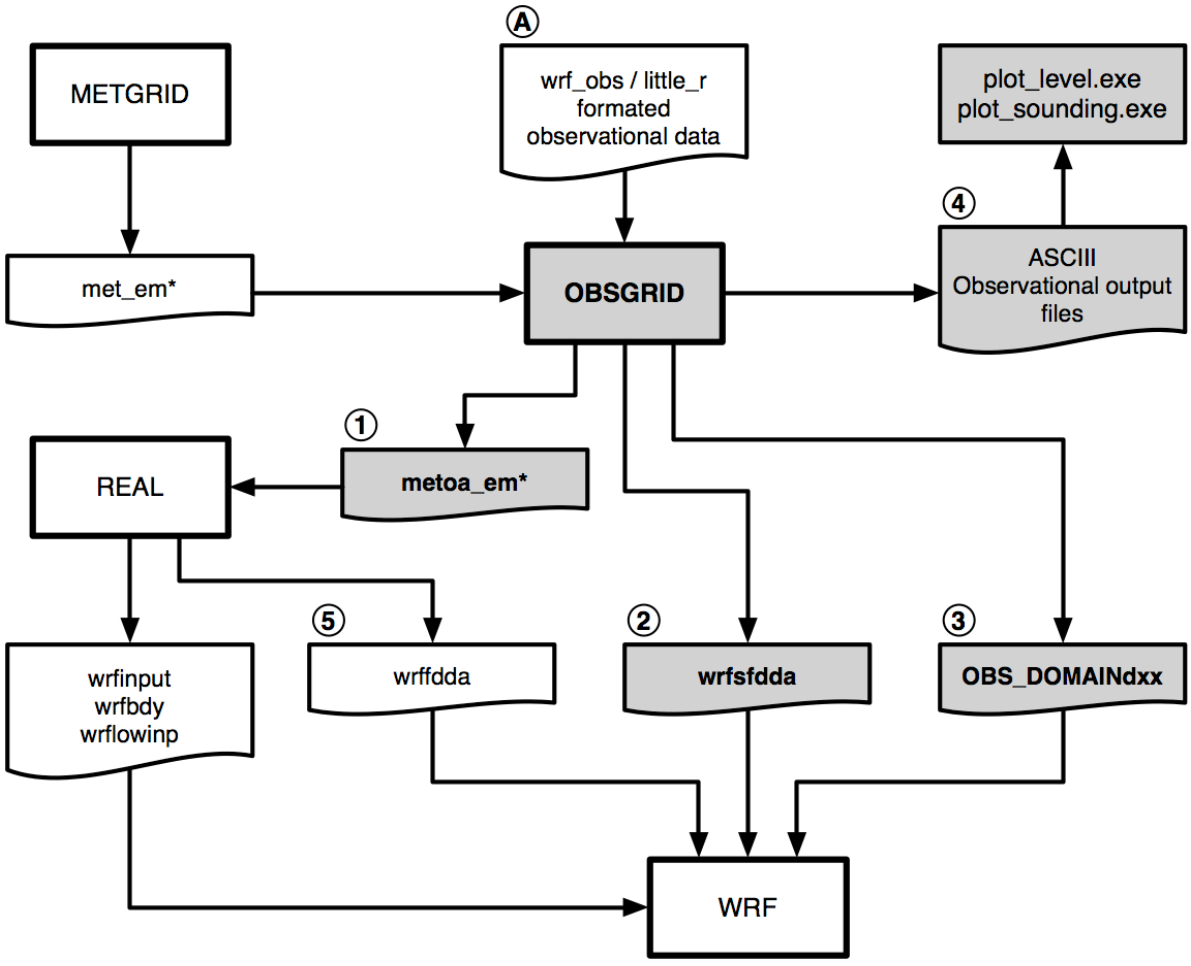
The analyses input to OBSGRID as the first guess are analyses output from the METGRID part of the WPS package (*see Chapter 3 of this User's Guide for details regarding the WPS package*).

OBSGRID capabilities include:

- Choice of Cressman-style or Multiquadric objective analysis.
- Various tests to screen the data for suspect observations.
- Procedures to input bogus data.
- Expanded Grid: OBSGRID has the capability to cut the input model domain down on output. This feature allows you to incorporate data from outside your intended grid to improve analyses near the boundaries. To use this feature, a user must create a larger domain than the final intended domain when running WPS.

Program Flow

OBSGRID is run directly after `metgrid.exe`, and uses the *met_em** output files from `metgrid.exe` as input. OBSGRID also requires additional observations (**A**) as input. The format of these observational files is described in the Observations Format section of this chapter.



Output from the objective analysis programs can be used to:

- Provide fields for Initial and Boundary conditions (1). Note that the files *metoa_em** are formatted identically to the *met_em** files from `metgrid.exe`. The only difference is that the fields in these files now incorporate observational information.
- Provide surface fields for surface-analysis-nudging *FDDA (2)*. Note, when using the *wrfsfdda* file as input to WRF, it is also recommended to use the 3-D fdda file *(wrffdda (5) – which is an optional output created when running* `real.exe`*)* as input to WRF.
- Provide data for observational nudging (3). Note: since version 3.1.1 of OBSGRID this file can be read directly by the observational nudging code and no longer needs to pass through an additional perl script.
- Provide ASCII and netCDF output (4). These files provide information regarding the observations used and the quality control flags assigned. The information in these files can also be plotted with the provided plotting utilities.

Source of Observations

OBSGRID reads observations provided by the user in formatted ASCII text files. This allows users to adapt their own data to use as input to the OBSGRID program. This format (*wrf_obs / little_r format*) is the same format used in the MM5 objective analysis program LITTLE_R (*hence the name*).

Programs are available to convert NMC ON29 and NCEP BUFR formatted files (*see below*) into the `wrf_obs / little_r` format. Users are responsible for converting other observations they may want to provide to OBSGRID into this format. A user-contributed (*i.e., unsupported*) program is available in the `utils/` directory for converting observation files from the GTS to `wrf_obs / little_r` format.

NCEP operational global surface and upper-air observation subsets, as archived by the Data Support Section (DSS) at *NCAR*.

- Upper-air data in NMC ON29 format *(from early 1970s to early 2000)* *http://rda.ucar.edu/datasets/ds353.4/*
- Surface data in NMC ON29 format *(from early 1970s to early 2000)* *http://rda.ucar.edu/datasets/ds464.0/*
- Upper-air data in NCEP BUFR format *(from 1999 to present)* *http://rda.ucar.edu/datasets/ds351.0/*
- Surface data in NCEP BUFR format *(from 1999 to present)* *http://rda.ucar.edu/datasets/ds461.0/*

The newer data *(ds351.0 and ds461.0)* is also available in the little_r format. From outside NCAR, this data can be download from the web, while it is available on the NCAR /glade

system for NCAR supercomputer users. This data is sorted into 6-hourly windows, which are typically too large for use in OBSGRID. To reorder this into 3-hourly windows:

- Get the little_r 6-hourly data
  - Non-NCAR super-computer users. Get the data directly from the above web sites. Combine (by using the Unix 'cat' command) all the surface and upper-air data into one large file called `rda_obs`.
  - NCAR super-computer users. Use the script `util/get_rda_data.csh`, to get the data and create the file `rda_obs`. You will need to edit this script to supply the date range that you are interested in.
- Compile the Fortran program `util/get_rda_data.f`. Place `rda_obs` file the in the top OBSGRID directory. Run the `util/get_rda_data.exe` executable. This executable will use the date range from namelist.oa, and create 3-hourly OBS:<date> files which are ready to use in OBSGRID.

NMC Office Note 29 can be found in many places on the World Wide Web, including: http://www.emc.ncep.noaa.gov/mmb/data_processing/on29.htm

Another method of obtaining little_r observations is to download observations from the Meteorological Assimilation Data Ingest System (MADIS; https://madis.noaa.gov/) and convert them to little_r format using the MADIS2LITTLER tool provided by NCAR (http://www2.mmm.ucar.edu/wrf/users/wrfda/download/madis.html). Note that to allow single-level above-surface observations to be properly dealt with by OBSGRID, MADIS2LITTLER must be modified to mark such observations as soundings (in module_output.F, subroutine write_littler_onelvl must be modified to set is_sound = .TRUE.).

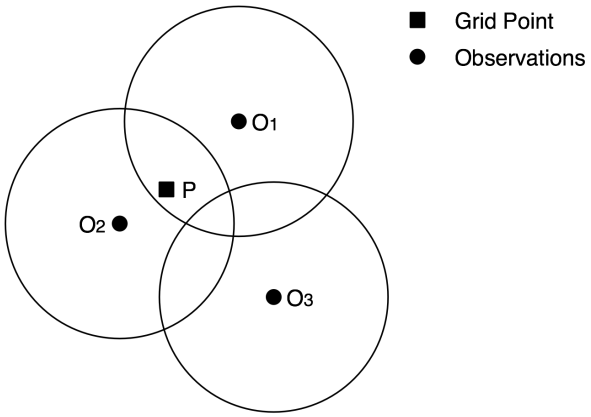
Objective Analysis techniques in OBSGRID

Cressman Scheme

Three of the four objective analysis techniques used in OBSGRID are based on the Cressman scheme, in which several successive scans nudge a first-guess field toward the neighboring observed values.

The standard Cressman scheme assigns to each observation a circular radius of influence, R. The first-guess field at each grid point, P, is adjusted by taking into account all the observations that influence P.

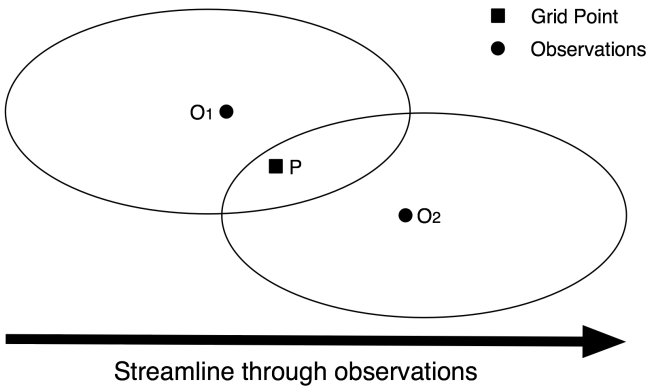
The differences between the first-guess field and the observations are calculated, and a distance-weighted average of these difference values is added to the value of the first-guess at P. Once all grid points have been adjusted, the adjusted field is used as the first guess for another adjustment cycle. Subsequent passes each use a smaller radius of influence.



Observations O_1 and O_2 influence grid point P, O_3 does not.

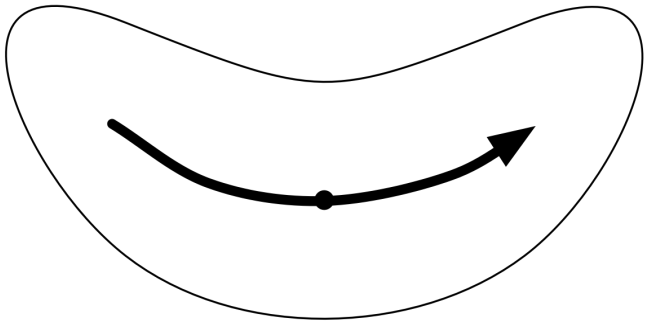
Ellipse Scheme

In analyses of wind and relative humidity (fields strongly deformed by the wind) at pressure levels, the circles from the standard Cressman scheme are elongated into ellipses, oriented along the flow. The stronger the wind, the greater the eccentricity of the ellipses. This scheme reduces to the circular Cressman scheme under low-wind conditions.



Banana Scheme

In analyses of wind and relative humidity at pressure levels, the circles from the standard Cressman scheme are elongated in the direction of the flow, and curved along the streamlines. The result is a banana shape. This scheme reduces to the Ellipse scheme under straight-flow conditions, and the standard Cressman scheme under low-wind conditions.



Multiquadric scheme

The Multiquadric scheme uses hyperboloid radial basis functions to perform the objective analysis. Details of the multiquadric technique may be found in Nuss and Titley, 1994: "Use of multiquadric interpolation for meteorological objective analysis." *Mon . Wea . Rev* ., 122, 1611-1631. Use this scheme with caution, as it can produce some odd results in areas where only a few observations are available.

Quality Control for Observations

A critical component of OBSGRID is the screening for bad observations. Many of these QC checks are optional in OBSGRID.

Quality Control on Individual Reports

- Gross Error Checks (same values, pressure decreases with height, etc.)
- Remove spikes from temperature and wind profiles.
- Adjust temperature profiles to remove superadiabatic layers.
- No comparisons to other reports or to the first-guess field.

The ERRMAX test

The ERRMAX quality-control check is optional, but highly recommended.

- Limited user control over data removal. The user may set thresholds, which vary the tolerance of the error check.
- Observations are compared to the first-guess field.
- If the difference value (obs - first-guess) exceeds a certain threshold, the observation is discarded.
- Threshold varies depending on the field, level, and time of day.
- Works well with a good first-guess field.

The Buddy test

The Buddy check is optional, but highly recommended.

- Limited user control over data removal. The user may set weighting factors, which vary the tolerance of the error check.
- Observations are compared to both the first guess and neighboring observations.
- If the difference value of an observation (obs - first-guess) varies significantly from the distance-weighted average of the difference values of neighboring observations, the observation is discarded.
- Works well in regions with good data density.

Additional Observations

Input of additional observations, or modification of existing (*and erroneous*) observations, can be a useful tool at the objective analysis stage.

In OBSGRID, additional observations are provided to the program the same way (*in the same wrf_obs / little_r format*) as standard observations. Additional observations must be in the same file as the rest of the observations. Existing (*erroneous*) observations can be modified easily, as the observations input format is ASCII text. Identifying an observation report as "bogus" simply means that it is assumed to be good data, but no quality control is performed for that report.

Surface FDDA option

The surface FDDA option creates additional analysis files for the surface only, usually with a smaller time interval between analyses (*i.e., more frequently*) than the full upper-air analyses. The purpose of these surface analysis files is for later use in WRF with the surface analysis nudging option.

The LAGTEM option controls how the first-guess field is created for surface analysis files. Typically, the surface and upper-air first-guess (*analysis times*) is available at twelve-hour or six-hour intervals, while the surface analysis interval may be 3 hours (*10800 seconds*). So at analysis

times, the available surface first-guess is used. If LAGTEM is set to .**FALSE**., the surface first-guess at other times will be temporally interpolated from the first-guess at the analysis times. If LAGTEM is set to .**TRUE**., the surface first guess at other times is the objective analysis from the previous time.

Objective Analysis on Model Nests

OBSGRID has the capability to perform the objective analysis on a nest. This is done manually with a separate OBSGRID process, performed on met_em_d0x files for the particular nest. Often, however, such a step is unnecessary; it complicates matters for the user and may introduce errors into the forecast. At other times, extra information available to the user, or extra detail that objective analysis may provide on a nest, makes objective analysis on a nest a good option.

The main reason to do objective analysis on a nest is if you have observations available with horizontal resolution somewhat greater than the resolution of your coarse domain. There may also be circumstances in which the representation of terrain on a nest allows for better use of surface observations (*i.e., the model terrain better matches the real terrain elevation of the observation*).

The main problem introduced by doing objective analysis on a nest is inconsistency in initial conditions between the coarse domain and the nest. Observations that fall just outside a nest will be used in the analysis of the coarse domain, but discarded in the analysis of the nest. With different observations used right at a nest boundary, one can get very different analyses.

How to run OBSGRID

Get the source code

The source code can be obtained from: https://github.com/wrf-model/OBSGRID, or from https://www2.mmm.ucar.edu/wrf/users/download/get_sources_pproc_util.html. Once the tar file is unzipped (gunzip OBSGRID.TAR.gz), and untared (tar -xf OBSGRID.TAR), it will create an OBSGRID/ directory.

```
cd OBSGRID
```

Generate the executable

The only library that is required to build the WRF model is netCDF. The user can find the source code, precompiled binaries, and documentation at the UNIDATA home page (http://www.unidata.ucar.edu/software/netcdf/).

To successfully compile the utilities `plot_level.exe` and `plot_sounding.exe`, NCAR Graphics needs to be installed on your system. These routines are not necessary to run OBSGRID, but are useful for displaying observations. Since version 3.7.0 NCL scripts are available and therefore these two utilities are no longer needed to plot the data.

To configure, type:

```
./configure
```

Choose one of the configure options, then compile.

```
./compile
```

If successful, this will create the executable `obsgrid.exe`. Executables `plot_level.exe` and `plot_sounding.exe`, will be created if NCAR Graphics is installed.

Prepare the observations files

Preparing observational files is a user responsibility. Some data are available from NCAR's RDA web site. Data from the early 1970's are in ON29 format, while data from 1999 to present are in NCEP BUFR format. Help using these datasets are available. For more information see the section **Source of Observations** on page 7-3 of this Users' Guide.

A program is also available for reformatting observations from the GTS stream (*unsupported*). This can be found in OBSGRID/util, and is called *gts_cleaner.f.* The code expects to find one observational input file per analysis time. Each file should contain both surface and upper-air data *(if available).*

Edit the namelist for your specific case

The most critical information you'll be changing most often is the start date, end date, and file names.

Pay particularly careful attention to the file name settings. Mistakes in observation file names can go unnoticed because OBSGRID will happily process the wrong files, and if there are no data in the (*wrongly-specified*) file for a particular time, OBSGRID will happily provide you with an analysis of no observations.

Run the program

Run the program by invoking the command:

```
./obsgrid.exe >& obsgrid.out
```

Check the `obsgrid.out` file for information and runtime errors.

Check your output

Examine the `obsgrid.out` file for error messages or warning messages. The program should have created the files called `metoa_em*.` Additional output files containing information about observations found, used and discarded will probably be created, as well.

Important things to check include the number of observations found for your objective analysis, and the number of observations used at various levels. This can alert you to possible problems in specifying observation files or time intervals. This information is included in the printout file.

You may also want to experiment with a couple of simple plot utility programs, discussed below.

There are a number of additional output files, which you might find useful. These are discussed below.

Output Files

The OBSGRID program generates some ASCII/netCDF files to detail the actions taken on observations through a time cycle of the program. In support of users wishing to plot the observations used for each variable (at each level, at each time), a file is created with this information. Primarily, the ASCII/netCDF files are for consumption by the developers for diagnostic purposes. The main output of the OBSGRID program is the gridded, pressure-level data set to be passed to the `real.exe` program (files `metoa_em`*).

In each of the files listed below, the text ".d*n*.YYYY-MM-DD_HH:mm:ss.tttt" allows each time period that is processed by OBSGRID to output a separate file. The only unusual information in the date string is the final four letters "tttt" which is the decimal time to ten thousandths of a second. These files will be dependent on the domain being processed.

metoa_em*

These are the final analysis files at surface and pressure levels. Generating this file is the primary goal of running OBSGRID.

These files can now be used in place of the *met_em** files from WPS to generate initial and boundary conditions for WRF. To use these files when running `real.exe` you can do one of two things:

1. Rename or link the *metoa_em** files back to *met_em**. This way `real.exe` will read the files automatically.
2. Use the *auxinput1_inname* namelist option in WRF's `namelist.input` file to overwrite the default filename `real.exe` uses. To do this, add the following to the *&time_control* section of the WRF `namelist.input` file before running `real.exe` *(use the exact syntax as below – do not substitute the <domain> and <date> for actual numbers)*:

   ```
   auxinput1_inname = "metoa_em.d<domain>.<date>"
   ```

wrfsfdda_d*n*

Use of the surface FDDA option in OBSGRID creates a file called `wrfsfdda_dn`. This file contains the surface analyses at INTF4D intervals, analyses of T, TH, U, V, RH, QV, PSFC, PMSL, and a count of observations within 250 km of each grid point.

Due to the input requirements of the WRF model, data at the current time (_OLD) and data for the next time (_NEW) are supplied at each time interval. *Due to this requirement, users must take care to specify the same interval in the WRF fdda section for surface nudging as the interval used in OBSGRID to create the wrfsfdda_dn file*. This also means that the user may need to have data available for OBSGRID to create a surface analysis beyond the last analysis actually used by WRF surface analysis nudging. With a positive value for the length of rampdown, even though the _OLD field at the beginning of the rampdown will be nudged throughout the rampdown, WRF still requires a _NEW field at the beginning of the rampdown period.

OBS_DOMAIN*dxx*

These files can be used in WRF for observational nudging. The format of this file is slightly different from the standard *wrf_obs / little_r* format. *See the Observation Nudging User's Guide or Chapter 5 of this User's Guide for details on observational nudging.*

The "*d*" in the file name represents the domain number. The "*xx*" is just a sequential number.

These files contain a list of all of the observations available for use by the OBSGRID program.

- The observations have been sorted and the duplicates have been removed.
- Observations outside of the analysis region have been removed.

- Observations with no information have been removed.
- All reports for each separate location *(different levels, but at the same time)* have been combined to form a single report.
- Data that has had the "discard" flag internally set *(data which will not be sent to the quality control or objective analysis portions of the code)* are not listed in this output.
- The data have gone through an expensive test to determine if the report is within the analysis region, and the data have been given various quality control flags. Unless a blatant error in the data is detected *(such as a negative sea-level pressure),* the observation data are not typically modified, but only assigned quality control flags.
- Data with qc flags higher than a specified value *(user controlled, via the namelist),* will be set to missing data.

The WRF observational nudging code requires that all observational data are available in a single file called OBS_DOMAINd01 *(where d is the domain number),* whereas OBSGRID creates one file per time. Therefore, to use these files in WRF, they should first be concatenated to a single file. A script *(run_cat_obs_files.csh)* is provided for this purpose. By running this script, the original OBS_DOMAINd01 files will be moved to OBS_DOMAINd01_sav, and a new OBS_DOMAINd01 file (containing all the observations for all times) will be created. This new file can be used directly in the WRF observational nudging code.

qc_obs_raw.d*n*.YYYY-MM-DD_HH:mm:ss.tttt(.nc)

This file contains a listing of all of the observations available for use by the OBSGRID program.

- The observations have been sorted and the duplicates have been removed.
- Observations outside of the analysis region have been removed.
- Observations with no information have been removed.
- All reports for each separate location *(different levels, but at the same time)* have been combined to form a single report.
- Data that has had the "discard" flag internally set *(data which will not be sent to the quality control or objective analysis portions of the code)* are not listed in this output.
- The data have gone through an expensive test to determine if the report is within the analysis region, and the data have been given various quality control flags. Unless a blatant error in the data is detected *(such as a negative sea-level pressure),* the observation data are not typically modified, but only assigned quality control flags.
- Two files are available, both containing identical information. One is the older ASCII format, while the other is in netCDF format.
- *The data in the ASCII file can be used as input to the plotting utility* `plot_sounding.exe`
- *The netCDF file can be used to plot both station data (util/station.ncl) and sounding data (util/sounding.ncl). This is available since version 3.7 and is the recommended option.*

qc_obs_used.d*n*.YYYY-MM-DD_HH:mm:ss.tttt(.nc)

These files are similar to the above "raw" files, and can be used in the same way. But in this case it contains the data used by the OBSGRID program, which are also the data saved to the OBS_DOMAINdxx files.

qc_obs_used_earth_relative.dn.YYYY-MM-DD_HH:mm:ss.tttt(.nc)

These files are identical to the above "qc_obs_used" files except that the winds are in an earth-relative framework rather than a model-relative framework. The non-netCDF version of these files can be used as input for the Model Evaluation Tools (MET; http://www.dtcenter.org/met/users/).

plotobs_out.d*n*.YYYY-MM-DD_HH:mm:ss.tttt

This file lists data by variable and by level, where each observation that has gone into the objective analysis is grouped with all of the associated observations for plotting or some other diagnostic purpose. The first line of this file is the necessary FORTRAN format required to input the data. There are titles over the data columns to aid in the information identification. Below are a few lines from a typical file. *This data can be used as input to the plotting utility* `plot_level.exe`. *But since version 3.7, it is recommended to use the* `station.ncl` *script that uses the data in the new netCDF data files.*

```
( 3x,a8,3x,i6,3x,i5,3x,a8,3x,2(g13.6,3x),2(f7.2,3x),i7 )
Number of Observations 00001214
Variable Press  Obs    Station Obs        Obs-1st   X         Y         QC
Name     Level  Number ID      Value      Guess     Location  Location  Value
U        1001   1      CYYT    6.39806    4.67690   161.51    122.96    0
U        1001   2      CWRA    2.04794    0.891641  162.04    120.03    0
U        1001   3      CWVA    1.30433   -1.80660   159.54    125.52    0
U        1001   4      CWAR    1.20569    1.07567   159.53    121.07    0
U        1001   5      CYQX    0.470500  -2.10306   156.58    125.17    0
U        1001   6      CWDO    0.789376  -3.03728   155.34    127.02    0
U        1001   7      CWDS    0.846182   2.14755   157.37    118.95    0
```

Plot Utilities

The OBSGRID package provides two utility programs for plotting observations. These programs are called `plot_soundings.exe` and `plot_levels.exe`. These optional programs use NCAR Graphics to build, which is often problematic. Two new NCL scripts are provided instead, `sounding.ncl` and `station.ncl`. Using these as opposed to the Fortran code are recommended.

sounding.ncl / plot_soundings.exe

The script `util/sounding.ncl` plots soundings. This script generates soundings from the netCDF files qc_obs_raw.d*n*.YYYY-MM-DD_HH:mm:ss.tttt.nc and qc_obs_used.d*n*.YYYY-MM-DD_HH:mm:ss.tttt.nc. Only data that are on the requested analysis levels are processed.

By default the script will plot the data from **all** the "qc_obs_used" files in the directory. This can be customized through the use of command line setting. For example:

ncl ./util/sounding.ncl 'qcOBS="raw"'
    will plot data from the "qc_obs_raw" files
ncl util/sounding.ncl YYYY=2010 MM=6
    will plot data from the "qc_obs_used" files for June 2010

Available command line options are:

qcOBS	Dataset to use. Options are "raw" or "used". Default is "used"
YYYY	Integer year to plot. Default is all available years.
MM	Integer month to plot. Default is all available months.
DD	Integer day to plot. Default is all available days.
HH	Integer hour to plot. Default is all available hours.
outTYPE	Output type. Default is plotting to the screen, i.e., "x11". Other options are "pdf" or "ps". The script creates the following output files(s): `qc_obs_<qcOBS>.sounding.<date>.<outTYPE>` for instance: qc_obs_used.sounding.2010-03-06_09.pdf

The older program `plot_soundings.exe` also plots soundings. This program generates soundings from the qc_obs_raw.d*n*.YYYY-MM-DD_HH:mm:ss.tttt and qc_obs_used.d*n*.YYYY-MM-DD_HH:mm:ss.tttt data files. Only data that are on the requested analysis levels are processed. The program uses information from `&record1, &record2` and `&plot_sounding` in the `namelist.oa` file to generate the required output. The program creates output file(s): `sounding_<file_type>_<date>.cgm`

plot_level.exe

The script `util/station.ncl` creates station plots for each analysis level. These plots contain both observations that have passed all QC tests and observations that have failed the QC tests. Observations that have failed the QC tests are plotted in various colors according to which test failed. This script generates soundings from the netCDF files qc_obs_raw.d*n*.YYYY-MM-DD_HH:mm:ss.tttt.nc and qc_obs_used.d*n*.YYYY-MM-DD_HH:mm:ss.tttt.nc.

By default the script will plot the data from **all** the "qc_obs_used" files in the directory. This can be customized through the use of command line setting. For example:

ncl ./util/station.ncl 'qcOBS="raw"'
will plot data from the "qc_obs_raw" files
ncl util/station.ncl YYYY=2010 MM=6
will plot data from the "qc_obs_used" files for June 2010

Available command line options are:

qcOBS	Dataset to use. Options are "raw" or "used". Default is "used"
YYYY	Integer year to plot. Default is all available years.
MM	Integer month to plot. Default is all available months.
DD	Integer day to plot. Default is all available days.
HH	Integer hour to plot. Default is all available hours.
outTYPE	Output type. Default is plotting to the screen, i.e., "x11". Other options are "pdf" or "ps". The script creates the following output files(s): `qc_obs_<qcOBS>.station.<date>.<outTYPE>` for instance: qc_obs_used.station.2010-03-06_09.pdf

The older program `plot_level.exe` creates station plots for each analysis level. These plots contain both observations that have passed all QC tests and observations that have failed the QC tests. Observations that have failed the QC tests are plotted in various colors according to which test failed. The program uses information from `&record1` and `&record2` in the `namelist.oa` file to generate plots from the observations in the file `plotobs_out.dn.YYYY-MM-DD_HH:mm:ss.tttt`. The program creates the file(s): `levels_<date>.cgm`.

Observations Format

To make the best use of the OBSGRID program, it is important for users to understand the *wrf_obs/little_r* Observations Format.

Observations are conceptually organized in terms of reports. A report consists of a single observation or set of observations associated with a single latitude/longitude coordinate.

Examples

- a surface station report including observations of temperature, pressure, humidity, and winds.
- an upper-air station's sounding report with temperature, humidity, and wind observations at many height or pressure levels.

- an aircraft report of temperature at a specific lat/lon/height.
- a satellite-derived wind observation at a specific lat/lon/height.

Each report in the *wrf_obs/little_r* Observations Format consists of at least four records:

- A *report header record*
- one or more *data records*
- an *end data record*
- an *end report record* .

The *report header record* is a 600-character-long record (*much of which is unused and needs only dummy values*) that contains certain information about the station and the report as a whole (location, station id, station type, station elevation, etc.). The report header record is described fully in the following table. Shaded items in the table are unused:

Report header format

Variable	Fortran I/O Format	Description
latitude	F20.5	station latitude (north positive)
longitude	F20.5	station longitude (east positive)
id	A40	ID of station
name	A40	Name of station
platform	A40	Description of the measurement device
source	A40	GTS, NCAR/ADP, BOGUS, etc.
elevation	F20.5	station elevation (m)
num_vld_fld	I10	Number of valid fields in the report
num_error	I10	Number of errors encountered during the decoding of this observation
num_warning	I10	Number of warnings encountered during decoding of this observation.
seq_num	I10	Sequence number of this observation
num_dups	I10	Number of duplicates found for this observation
is_sound	L10	T/F Above-surface or surface (i.e., all non-surface observations should use T, even above-surface single-level obs)
bogus	L10	T/F bogus report or normal one
discard	L10	T/F Duplicate and discarded (or merged) report.
sut	I10	Seconds since 0000 UTC 1 January 1970

julian	I10	Day of the year
date_char	A20	YYYYMMDDHHmmss
slp, qc	F13.5, I7	Sea-level pressure (Pa) and a QC flag
ref_pres, qc	F13.5, I7	Reference pressure level (for thickness) (Pa) and a QC flag
ground_t, qc	F13.5, I7	Ground Temperature (T) and QC flag
sst, qc	F13.5, I7	Sea-Surface Temperature (K) and QC
psfc, qc	F13.5, I7	Surface pressure (Pa) and QC
precip, qc	F13.5, I7	Precipitation Accumulation and QC
t_max, qc	F13.5, I7	Daily maximum T (K) and QC
t_min, qc	F13.5, I7	Daily minimum T (K) and QC
t_min_night, qc	F13.5, I7	Overnight minimum T (K) and QC
p_tend03, qc	F13.5, I7	3-hour pressure change (Pa) and QC
p_tend24, qc	F13.5, I7	24-hour pressure change (Pa) and QC
cloud_cvr, qc	F13.5, I7	Total cloud cover (oktas) and QC
ceiling, qc	F13.5, I7	Height (m) of cloud base and QC

Following the report header record are the *data records*. These data records contain the observations of pressure, height, temperature, dewpoint, wind speed, and wind direction. There are a number of other fields in the data record that are not used on input. Each data record contains data for a single level of the report. For report types that have multiple levels (*e.g., upper-air station sounding reports*), each pressure or height level has its own data record. For report types with a single level (*such as surface station reports or a satellite wind observation*), the report will have a single data record. The data record contents and format are summarized in the following table

Format of data records		
Variable	**Fortran I/O**	**Description**

	Format	
pressure, qc	F13.5, I7	Pressure (Pa) of observation, and QC
height, qc	F13.5, I7	Height (m MSL) of observation, and QC
temperature, qc	F13.5, I7	Temperature (K) and QC
dew_point, qc	F13.5, I7	Dewpoint (K) and QC
speed, qc	F13.5, I7	Wind speed (m/s) and QC
direction, qc	F13.5, I7	Wind direction (degrees) and QC
u, qc	F13.5, I7	u component of wind (m/s), and QC
v, qc	F13.5, I7	v component of wind (m/s), and QC
rh, qc	F13.5, I7	Relative Humidity (%) and QC
thickness, qc	F13.5, I7	Thickness (m), and QC

The *end data record* is simply a data record with pressure and height fields both set to -777777.

After all the data records and the end data record, an *end report record* must appear. The end report record is simply three integers, which really aren't all that important.

Format of end_report records		
Variable	**Fortran I/O Format**	**Description**
num_vld_fld	I7	Number of valid fields in the report
num_error	I7	Number of errors encountered during the decoding of the report
num_warning	I7	Number of warnings encountered during the decoding the report

QCFlags

In the observation files, most of the meteorological data fields also have space for an additional integer quality-control flag. The quality-control values are of the form 2n, where n takes on positive integer values. This allows the various quality control flags to be additive, yet permits the decomposition of the total sum into constituent components. Following are the current quality control flags that are applied to observations:

```
pressure interpolated from first-guess height      = 2 **  1 =      2
pressure int. from std. atmos. and 1st-guess height= 2 **  3 =      8
temperature and dew point both = 0                 = 2 **  4 =     16
wind speed and direction both = 0                  = 2 **  5 =     32
```

```
wind speed negative                                  = 2 **  6 =     64
wind direction < 0 or > 360                          = 2 **  7 =    128
level vertically interpolated                        = 2 **  8 =    256
value vertically extrapolated from single level      = 2 **  9 =    512
sign of temperature reversed                         = 2 ** 10 =   1024
superadiabatic level detected                        = 2 ** 11 =   2048
vertical spike in wind speed or direction            = 2 ** 12 =   4096
convective adjustment applied to temperature field = 2 ** 13 =   8192
no neighboring observations for buddy check          = 2 ** 14 =  16384
-----------------------------------------------------------------------
data outside normal analysis time and not QC-ed      = 2 ** 15 =  32768
-----------------------------------------------------------------------
fails error maximum test                             = 2 ** 16 =  65536
fails buddy test                                     = 2 ** 17 = 131072
observation outside of domain detected by QC         = 2 ** 18 = 262144
```

OBSGRID Namelist

The OBSGRID namelist file is called "`namelist.oa`", and must be in the directory from which OBSGRID is run. The namelist consists of nine namelist records, named "record1" through "record9", each having a loosely related area of content. Each namelist record, which extends over several lines in the `namelist.oa` file, begins with "&record<#>" (where <#> is the namelist record number) and ends with a slash "/".

The namelist record `&plot_sounding` is only used by the corresponding utility.

Namelist record1

The data in namelist record1 define the analysis times to process:

Namelist Variable	Value	Description
`start_year`	`2000`	`4-digit year of the starting time to process`
`start_month`	`01`	`2-digit month of the starting time to process`
`start_day`	`24`	`2-digit day of the starting time to process`
`start_hour`	`12`	`2-digit hour of the starting time to process`
`end_year`	`2000`	`4-digit year of the ending time to process`
`end_month`	`01`	`2-digit month of the ending time to process`
`end_day`	`25`	`2-digit day of the ending time to process`
`end_hour`	`12`	`2-digit hour of the ending time to process`

interval	21600	Time interval (s) between consecutive times to process

Namelist record2

The data in record2 define the model grid and names of the input files:

Namelist Variable	Value	Description
grid_id	1	ID of domain to process
obs_filename	CHARACTER	Root file name (*may include directory information*) of the observational files. All input files must have the format obs_filename:<YYYY-MM-DD_HH>. One file required for each time period. If a wrfsfdda is being created, then similar input data files are required for each surface fdda time.
remove_data_above_qc_flag	200000	Data with qc flags higher than this will not be output to the OBS_DOMAIN*dxx* files. Default is to output all data. Use 65536 to remove data that failed the buddy and error max tests. To also exclude data outside analysis times that could not be QC-ed use **32768** *(recommended).* This does not affect the data used in the OA process.

remove_unverified_data	.FALSE.	By setting this parameter to **.TRUE.** *(recommended)* any observations that could not be QC'd due to having a pressure insufficiently close to an analysis level will be removed from the *OBS_DOMAINdxx* files. Obs QC'd by adjusting them to a nearby analysis level or by comparing them to an analysis level within a user-specified tolerance will be included in the OBS_DOMAINdxx files. See use_p_tolerance_one_lev in &record4.
trim_domain	.FALSE.	Set to .TRUE. if this domain must be cut down on output
trim_value	5	Value by which the domain will be cut down in each direction

The *met_em** files which are being processed must be available in the OBSGRID/ directory.

The *obs_filename* and interval settings can get confusing, and deserve some additional explanation. Use of the *obs_filename* files is related to the times and time interval set in namelist &record1, and to the F4D options set in namelist &record8. The *obs_filename* files are used for the analyses of the full 3D dataset, both at upper levels and the surface. They are also used when F4D=.TRUE.; that is, if surface analyses are being created for surface FDDA nudging. The *obs_filename* files should contain all observations (upper-air and surface) to be used for a particular analysis at a particular time.

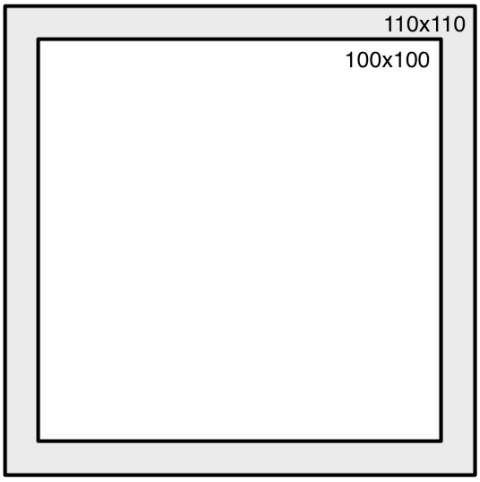
Ideally there should be an *obs_filename* for each time period for which an objective analysis is desired. Time periods are processed sequentially from the starting date to the ending date by the time interval, all specified in namelist &record1. All observational files must have a date associated with them. If a file is not found, the code will process as if this file contains zero observations, and then continue to the next time period.

If the F4D option is selected, the *obs_filename* files are similarly processed for surface analyses, this time with the time interval as specified by INTF4D.

If a user wishes to include observations from outside the model domain of interest, geogrid.exe (WPS) needs to be run over a slightly larger domain than the domain of interest. Setting

trim_domain to .TRUE. will cut all 4 directions of the input domain down by the number of grid points set in trim_value.

In the example below, the domain of interest is the inner white domain with a total of 100x100 grid points. geogrid.exe has been run for the outer domain (110x110 grid points). By setting the trim_value to 5, the output domain will be trimmed by 5 grid points in each direction, resulting in the white 100x100 grid point domain.



Namelist record3

The data in the &record3 concern space allocated within the program for observations. These are values that should not frequently need to be modified:

Namelist Variable	Value	Description
max_number_of_obs	10000	Anticipated maximum number of reports per time period
fatal_if_exceed_max_obs	.TRUE.	T/F flag allows the user to decide the severity of not having enough space to store all of the available observation

Namelist record4

The data in &record4 set the quality control options. There are four specific tests that may be activated by the user: An error max test; a buddy test; removal of spike, and; the removal of super-adiabatic lapse rates. For some of these tests, the user has control over the tolerances, as well.

Namelist Variable	Value	Description
qc_psfc	.FALSE.	Execute error max and buddy check tests for surface pressure observations (temporarily converted to sea level pressure to run QC)
Error Max Test: For this test there is a threshold for each variable. These values are scaled for time of day, surface characteristics and vertical level.		
qc_test_error_max	.TRUE.	Check the difference between the first-guess and the observation
max_error_t	10	Maximum allowable temperature difference (K)
max_error_uv	13	Maximum allowable horizontal wind component difference (m/s)
max_error_z	*8*	*Not used*
max_error_rh	50	Maximum allowable relative humidity difference (%)
max_error_p	600	Maximum allowable sea-level pressure difference (Pa
max_error_dewpoint	20	Maximum allowable dewpoint difference (K)
Buddy Check Test: For this test there is a threshold for each variable. These values are similar to standard deviations.		
qc_test_buddy	.TRUE.	Check the difference between a single observation and neighboring observations
max_buddy_t	8	Maximum allowable temperature difference (K)
max_buddy_uv	8	Maximum allowable horizontal wind component difference (m/s)
max_buddy_z	*8*	*Not used*
max_buddy_rh	40	Maximum allowable relative humidity difference (%)
max_buddy_p	800	Maximum allowable sea-level pressure difference (Pa)
max_buddy_dewpoint	20	Maximum allowable dewpoint difference (K)

buddy_weight	1.0	Value by which the buddy thresholds are scaled
Spike removal		
qc_test_vert_consistency	.FALSE.	Check for vertical spikes in temperature, dew point, wind speed and wind direction
Removal of super-adiabatic lapse rates		
qc_test_convective_adj	.FALSE.	Remove any super-adiabatic lapse rate in a sounding by conservation of dry static energy
For satellite and aircraft observations, data are often horizontally spaced with only a single vertical level. The following entries determine how such data are dealt with and are described in more detail below the table.		
use_p_tolerance_one_lev	.FALSE.	Should single-level above-surface observations be directly QC'd against nearby levels (.TRUE.) or extended to nearby levels (.FALSE.)
max_p_tolerance_one_lev_qc	700	Pressure tolerance within which QC can be applied directly (Pa)
max_p_extend_t	1300	Pressure difference (Pa) through which a single temperature report may be extended
max_p_extend_w	1300	Pressure difference (Pa) through which a single wind report may be extended

Dewpoint quality control:
Note that the dewpoint error max check and buddy check are using the same moisture field as the relative humidity checks. The dewpoint checks are to allow for an additional level of quality control on the moisture fields and may be helpful for dry observations where RH differences may be small but dewpoint differences are much larger. The maximum dewpoint thresholds are scaled based on the observed dewpoint to increase the threshold for dry conditions where larger dewpoint variations are expected. If the user does not wish to use dewpoint error checks, simply set the thresholds to very large values.

Quality control of single-level above-surface observations:

Option 1: `use_p_tolerance_one_lev = .FALSE.`:

For single-level above-surface observations marked as 'FM-88 SATOB' or 'FM-97 AIREP', the observations are adjusted to the nearest pressure level. If the observation's pressure is within max_p_extend_t Pa of the nearest first-guess level, the temperature of the observation is adjusted to the first-guess level using a standard lapse rate, otherwise the temperature is marked as missing. If the observation's pressure is within max_p_extend_w Pa of the nearest first-guess level, the winds are used without adjustment. The dewpoint is marked as missing regardless of the pressure of the observation. The pressure of the observation is changed to be the pressure of the pressure level against which it is being quality controlled.
If a single-level above-surface observation is marked as anything other than 'FM-88 SATOB' or 'FM-97 AIREP', it appears that it will not be quality controlled unless its pressure happens to exactly match one of the pressure levels in the first guess field. Note that max_p_tolerance_one_lev_qc is ignored if use_p_tolerance_one_lev = .FALSE.

Option 2: use_p_tolerance_one_lev = .TRUE.:
For all single-level above-surface observations, the observations will be quality controlled as long as the closest first-guess field is within max_p_tolerance_one_lev_qc Pa of the observation. In order to allow all single-level above-surface observations to be close enough to a first-guess pressure level that quality control directly comparing the closest pressure level to the observation is valid, the user may need to interpolate the first guess to additional pressure levels prior to ingestion into OBSGRID. OBSGRID will print out the pressure ranges for which error max quality control is not available (i.e., the pressures for which single-level above-surface observations will not be quality controlled). See max_p_tolerance_one_lev_oa in namelist record9 for the equivalent pressure tolerance for creating objective analyses. Note that max_p_extend_t and max_p_extend_w are ignored if use_p_tolerance_one_lev = .TRUE.

Namelist record5

The data in &record5 control the enormous amount of printout that may be produced by the OBSGRID program. These values are all logical flags, where TRUE will generate output and FALSE will turn off output.

```
print_obs_files ; print_found_obs ; print_header ;
print_analysis ;print_qc_vert ; print_qc_dry ;
print_error_max ; print_buddy ;print_oa
```

Namelist record7

The data in &record7 concern the use of the first-guess fields and surface FDDA analysis options. Always use the first guess.

Namelist Variable	Value	Description
use_first_guess	.TRUE.	Always use first guess (use_first_guess=.TRUE.)
f4d	.TRUE.	Turns on (.TRUE.) or off (.FALSE.) the creation of surface analysis files.
intf4d	10800	Time interval in seconds between surface analysis times
lagtem	.FALSE.	Use the previous time-period's final surface analysis for this time-period's first guess (lagtem=.TRUE.); or Use a temporal interpolation between upper-air times as the first guess for this surface analysis (lagtem = .FALSE.)

Namelist record8

The data in `&record8` concern the smoothing of the data after the objective analysis. Note, only the differences fields (*observation minus first-guess*) of the analyzed are smoothed, not the full fields.

Namelist Variable	Value	Description
smooth_type	1	1 = five point stencil of 1-2-1 smoothing; 2 = smoother-desmoother
smooth_sfc_wind	0	Number of smoothing passes for surface winds
smooth_sfc_temp	0	Number of smoothing passes for surface temperature
smooth_sfc_rh	0	Number of smoothing passes for surface relative humidity
smooth_sfc_slp	0	Number of smoothing passes for sea-level pressure
smooth_upper_wind	0	Number of smoothing passes for upper-air winds
smooth_upper_temp	0	Number of smoothing passes for upper-air temperature
smooth_upper_rh	0	Number of smoothing passes for upper-air relative humidity

Namelist record9

The data in `&record9` concern the objective analysis options. There is no user control to select the various Cressman extensions for the radius of influence (*circular, elliptical or banana*). If the Cressman option is selected, ellipse or banana extensions will be applied as the wind conditions warrant.

Namelist Variable	Value	Description
oa_type	"Cressman"	"MQD" for multiquadric; "Cressman" for the Cressman-type scheme, "None" for no analysis, this string is case sensitive
oa_3D_type	"Cressman"	Set upper-air scheme to "Cressman", regardless of the scheme used at the surface
oa_3D_option	0	How to switch between "MQD" and "Cressman" if not enough observations are available to perform "MQD"
mqd_minimum_num_obs	30	Minimum number of observations for MQD
mqd_maximum_num_obs	1000	Maximum number of observations for MQD
radius_influence	5,4,3,2	Radius of influence in grid units for Cressman scheme
radius_influence_sfc_mult	1.0	Multiply above-surface radius of influence by this value to get surface radius of influence
oa_min_switch	.TRUE.	T = switch to Cressman if too few observations for MQD; F = no analysis if too few observations
oa_max_switch	.TRUE.	T = switch to Cressman if too many observations for MQD; F = no analysis if too many observation
scale_cressman_rh_decreases	.FALSE.	T = decrease magnitude of drying in Cressman analysis; F = magnitude of

		drying in Cressman analysis unmodified
oa_psfc	.FALSE.	T = perform surface pressure objective analysis; F = surface pressure only adjusted by sea level pressure analysis
max_p_tolerance_one_lev_oa	700	Pressure tolerance within which single-level above-surface observations can be used in the objective analysis (Pa)

When oa_type is set to *Cressman*, then the *Cressman* scheme will be performed on all data.

When oa_type is set to *None,* then no objective analysis will be performed on any data.

When oa_type is set to *MQD*, there are a wide variety of options available that control when the code will revert back to the *Cressman* scheme.

- oa_max_switch ; mqd_maximum_num_obs
  The code will revert back to *Cressman* if the switch is set to true and the maximum number of observations is exceeded.
  This is to reduce the time the code runs and not for physical reasons.
  Recommended to leave switch set to true and just set the maximum number large.

- oa_min_switch ; mqd_minimum_num_obs
  The code will revert back to *Cressman* if the switch is set to true and there are too few observations. How and when the code reverts back to Cressman under these conditions are controlled by the oa_3D_option parameter.
  Recommended to leave switch set to true and start with the default minimum settings.

- oa_3D_type="Cressman"
  All upper-air levels will use the *Cressman* scheme, regardless of other settings.

  The surface will use *MQD* as long as there are enough observations to do so (mqd_maximum_num_obs ; mqd_minimum_num_obs), otherwise it will revert to the *Cressman* scheme.
  Note that if some time periods have enough observations and others do not, the code will only revert to *Cressman* for the times without sufficient observations.

- oa_3D_option
  There are three options (0,1,2). For all these options the surface will use *MQD* as long as

there are enough observations to do so (`mqd_maximum_num_obs ; mqd_minimum_num_obs`), otherwise it will revert to the *Cressman* scheme.
Note that if some time periods have enough observations and others do not, the code will only revert to *Cressman* for the times without sufficient observations.

The upper-air will react as follows:
0 (default): MQD is performed in the upper-air as long as there are enough observations to do so (`mqd_maximum_num_obs ; mqd_minimum_num_obs`). As soon as this is no longer the case, the code will STOP, with suggestions as to which parameters to set to run the code correctly.

1: The code will first check to see if, for a given time, all levels and variables in the upper-air have sufficient observations for the *MQD* scheme. If not, the code will revert to Cressman for that time period. Note that if some time periods have enough observations and others do not, the code will only revert to *Cressman* for the times without sufficient observations.

2: The code will check if sufficient observations are available per time, level, and variable for the *MQD* scheme. If not, the code will revert to the *Cressman* scheme for that particular time, level and variable. Note this can result in uncontrolled switching between *MQD* and *Cressman*. Therefore this option is not recommended.

`radius_influence`
There are three ways to set the radius of influence (RIN) for the *Cressman* scheme:

- Manually: Set the RIN and number of scans directly. E.g., 5,4,3,2, will result in 4 scans. The first will use 5 grid points for the RIN and the last, 2 points.
- Automatically 1: Set RIN to 0 and the code will calculate the RIN based on the domain size and an estimated observation density of 325 km. By default there will be 4 scans.
- Automatically 2: Set RIN to a negative number and the code will calculate the RIN based on the domain size and an estimated observation density of 325 km. The number of scans is controlled by the value of the set number. E.g, -5 will result in 5 scans.

`radius_influence_sfc_mult`
The RIN calculated as described above is multiplied by this value to determine the RIN for surface observations. This allows the finer scale structures observed at the surface to be retained. If this multiplication results in a RIN greater than 100 model grid points, then the RIN on the first scan is scaled to be 100 model grid points and all subsequent scans are scale by that same ratio. This is to prevent features from being washed out on fine-scale domains. In order to minimize “spots” on the solution, any scan with a RIN less than 4.5 model grid points is skipped. If this is set to 1.0 then the RIN for surface observations will match the RIN for above-surface observations.

`scale_cressman_rh_decreases`

This option is meant to mitigate overdrying that can occur when the need for drying diagnosed via an observation at one point is spread to another point where the first guess is already drier than the first guess at the location of the observation If this option is set to true then drying applied to a point where the first guess is drier than the first guess at the observation location is scaled by the ratio first guess relative humidity at the point the drying is being applied to divided by the first guess relative humidity at the location of the observation.

Note that this scaling is applied on each Cressman scan. See Reen et al. 2016 (http://dx.doi.org/10.1175/JAMC-D-14-0301.1) for further details.

`oa_psfc`
An objective analysis of surface pressure may allow Obsgrid surface analyses of other fields to be more effectively utilized in WRF if the first-guess surface pressure field is sufficiently coarse compared to the WRF domains (e.g., Reen 2015; http://www.arl.army.mil/arlreports/2015/ARL-TR-7447.pdf). This is because the surface pressure analysis may provide a better estimate of the pressure of the surface analyses and thus WRF is less likely to erroneously reject the surface analyses as being too distant from the actual surface. If there are an insufficient number of observations or if the first-guess surface pressure is not much coarser than WRF, this capability is less likely to add value.

`max_p_tolerance_one_lev_oa`
`If use_p_tolerance_one_lev = .TRUE.` in record4, then `max_p_tolerance_one_lev_oa` is the pressure tolerance (Pa) allowed between single-level above-surface observations and the pressure level they are being used in an objective analysis. If `use_p_tolerance_one_lev = .FALSE.` in record4, then `max_p_tolerance_one_lev_oa` is not used by OBSGRID.

Namelist plot_sounding

Only used for the utility plot_sounding.exe

Namelist Variable	Value	Description
`file_type`	`"raw"`	`File to read to produce the plots. Options are "raw" or "used"`
`read_metoa`	`.TRUE.`	`If set to .TRUE., the model domain information in the metoa_em files will be used to add location information on the plot.`