ARL-TR-8978 ● JUNE 2020

DEVCOM ARMY RESEARCH LABORATORY

# Descriptions and Procedures for Running the Meteorological Error Determination (MED) Scripts and Programs

by J Cogan

## NOTICES

### Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

ARL-TR-8978 ● JUNE 2020


# Descriptions and Procedures for Running the Meteorological Error Determination (MED) Scripts and Programs

**J Cogan**
***Computational and Information Sciences Directorate, CCDC Army Research Laboratory***

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE *(DD-MM-YYYY)* | 2. REPORT TYPE | 3. DATES COVERED (From - To) |
|---|---|---|
| June 2020 | Technical Report | 15 October 2019–12 May 2020 |

**4. TITLE AND SUBTITLE**
Descriptions and Procedures for Running the Meteorological Error Determination (MED) Scripts and Programs

**5a. CONTRACT NUMBER**

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
J Cogan

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
CCDC Army Research Laboratory
ATTN: FCDD-RLC-E
Adelphi, MD 20783-1138

**8. PERFORMING ORGANIZATION REPORT NUMBER**
ARL-TR-8978

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

**10. SPONSOR/MONITOR'S ACRONYM(S)**

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
This report briefly describes the several scripts and programs used to generate vertical profiles with a user-defined vertical structure in terms of height or pressure from model and radiosonde observation (RAOB) data, compute the differences between the respective model- and RAOB-based profiles, and produce tables of several statistical measures such as mean and standard deviation by variable and height or pressure level. The basic version of this software was developed earlier, but was modified for the purpose of producing statistics for meteorological error determination. This set of scripts and programs greatly reduces the time needed for analysis of model accuracy relative to RAOBs, as compared with previous methods that used, for example, spreadsheets to compute differences and statistical values. The report details the scripts and programs that compose the new version of the software and presents samples of the output.

**15. SUBJECT TERMS**
model evaluation, meteorological error determination, model accuracy statistics, user-defined vertical structure, model-derived soundings

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | J Cogan |
| Unclassified | Unclassified | Unclassified | UU | 45 | 19b. TELEPHONE NUMBER (Include area code) (301) 394-2304 |

## Contents

## List of Figures



## List of Tables

# 1. Introduction

This report briefly describes the several scripts and programs used to generate vertical profiles with a user-defined vertical structure in terms of height or pressure from model and radiosonde observation (RAOB) data, compute the differences between the respective model- and RAOB-based profiles, and produce tables of several statistical measures such as mean and standard deviation by variable and height or pressure level. While the basic version of this software was developed earlier (Cogan 2018), it was modified for the purpose of producing statistics that may be used in the development of a new method for meteorological (MET) error determination (MED), or new MED. This set of scripts and programs greatly reduces the time needed for analysis of model accuracy relative to RAOBs, as compared to previous methods that used, for example, spreadsheets to compute differences and statistical values.

More detailed descriptions of some of the several programs or scripts, as well as input and output samples, may be found in the respective technical reports and notes. The published reports are listed in the References section of this report. The Bash scripts call one or more Python 3 scripts and/or C programs. The C programs and the underlying algorithms are described in Cogan (2015), which is more related to specific applications, and a journal article (Cogan 2017), which may be applied more generally. The more general version of the code currently resides on the US Army Combat Capabilities Development Command (CCDC) Army Research Laboratory's (ARL's) open GitHub site, https://github.com/usarmyresearchlab, and has the project name ARL_MET-profile_Converter.

A Bash script was written to simplify the procedure for extracting vertical profiles from either Global Forecast System (GFS) or Global Air Land Weather Exploitation Model (GALWEM) output. The Bash script is mostly self-explanatory: Type the name of the script and enter the requested information. The Bash script also allows the user to convert the GFS- or GALWEM-derived vertical profiles into profiles at user-defined height or pressure levels and for layers defined by those levels. For comparison purposes, another Bash script with the appropriate external C programs processes RAOB text files from the University of Wyoming's Weather Web (WWeb) or the NOAA archive site. The several scripts and programs are briefly described here and Appendix A provides a list of those scripts and programs. These scripts and programs have been modified, as appropriate, specifically to generate statistics for model evaluation.

## 2. Extraction of Vertical Profiles from RAOB and GFS and GALWEM Output Files

Producing vertical profiles of MET variables from RAOB data is accomplished using a Bash script that includes calls to C programs (see Cogan [2017] for a description of an earlier version of the C programs). The procedure for extracting vertical "soundings" from GRIB2 model output files and the process to generate vertical profiles of MET variables at user-defined levels and layers are first described for a Bash script that combines separate Python 3 scripts and C programs. Then, the report briefly discusses individually called scripts and programs. Cogan (2018) describes an earlier version of the software, along with flowcharts and input and output samples.

### 2.1 Obtaining RAOB Files

RAOB files are readily available from two archive sites. One is the University of Wyoming's WWeb (https://weather.uwyo.edu/upperair/uamap.shtml). Soundings in text and other formats are available from numerous World Meteorological Organization (WMO) sites around the world, and weather maps may also be found for regions that together cover nearly the entire planet. The other site is NOAA's Earth System Research Laboratory (ESRL) Global Systems Division (GSD) archive for RAOB data (https://ruc.noaa.gov/raobs/), which contains data for WMO sites worldwide but in a less readable text format than that from the University of Wyoming site. The format of the archived data is described at https://ruc.noaa.gov/raobs/fsl_format-new.html. Nevertheless, the text is much easier to read than the standard WMO format for RAOBs. San Francisco State University's website describes the WMO code (http://tornado.sfsu.edu/geosciences/classes/m400/Lab2_files/RadiosondeCode.html). As with the WMO format, all variables (height, pressure, temperature, dewpoint, wind direction, and wind speed) appear at the standard levels (1000, 925, 850, 700, and so on in hectopascals [hPa]), but "special" levels have height, pressure, wind direction, and wind speed (wind data level) or height, pressure, temperature, and dewpoint (thermal data level). The number of special levels is most often different for different soundings and most often the number is different for wind and thermal data within the same sounding. For further details on NOAA's data archives. see https://www.ncdc.noaa.gov/data-access/weather-balloon/integrated-global-radiosonde-archive, which has links to several data archives.

For either source of RAOB data, the requested soundings appear in text form on the website and may be copied to a file on the user's computer versus using software

to extract the files "directly". Also, both contain RAOB data from the most recently received (hours after the nominal time) to several decades in the past (for some RAOB stations, the University of Wyoming has data back to the 1970s and NOAA back to the 1940s).

The University of Wyoming WWeb site obtains the input for their soundings from the NOAA archive and interpolates them, so that all levels have all the aforementioned wind and thermal variables plus a few other computed quantities, such as mixing ratio and potential temperature. Occasionally, a sounding from the NOAA archive will have special levels or standard and special levels with the same pressure and the same or slightly different heights. In this case, processing at the WWeb keeps the same pressures and retains or interpolates the values of the other variables. Also, when the pressures for successive data lines are the same in the WWeb sounding, the height of the first data line may be higher than that of the second. For example, lines 70 and 71 of an 82-line sounding have a pressure of 41 hPa, yet line 70 has a height of 21800 m and line 71 has a height of 21770 m. The computed output sounding with a user-defined vertical structure ends as if the first of the repeated lines was the last input data line from the WWeb sounding. Using the aforementioned example, the software would take line 70 with a pressure of 41 hPa and a height of 21800 m as the last input data line and ignore successive data lines.

A short Python 3 script may be used to check for "duplicate" data lines, and if found, copy the text file containing the sounding extracted from the WWeb site to a new version without the extra data lines. If no duplicates are found, the script ends without copying the initial text file. The script also checks for missing data fields within each data line. If it finds one or more lines without all the variables, the initial file is copied to a new version without the line(s) with missing data. The script is executed by typing its name on the command line as follows:

```
python3 raobcheck.py
```

If the user permissions allow execution, *./raobcheck.py* will work as well. The script queries the user for the name of the file to be checked. Currently, the copy, if needed, has the same name as the input file with "R" added to indicate a revised version. For example, LMN_2020041712 would be modified and copied to LMN_2020041712R. Since further processing is not dependent on the exact name, the addition of the R will have no effect on the other scripts or programs.

For soundings from the NOAA archive, repeated data lines appear to have no noticeable effect with respect to the conversion to a sounding with a user-defined structure. That may be a consequence of somewhat different processing of soundings from NOAA and WWeb, where wind and thermal (e.g., temperature and

dewpoint) are input into separate arrays and later follow separate paths in the processing. For example, a special level for wind may follow a special level for temperature and dewpoint, even though both have the same pressures (to the nearest 0.1 hPa). Several test runs with the relevant C programs (Section 2.5) suggested a very small difference (e.g., virtual temperature differences of some hundredths of kelvins) in soundings with a user-defined pressure structure and none for those with a user-defined height structure. Nevertheless, a very recent version of a Python 3 script for obtaining multiple soundings from the NOAA archive includes code to find consecutive data lines with duplicated pressures and remove those that have successive decreasing heights. A follow-on document will describe that script and other ongoing upgrades.

Details on converting a RAOB (or model output profile) into a sounding with user-defined height or pressure vertical coordinates may be found in Cogan (2017), and the latest versions of the relevant C programs, as of the preparation of this report, are summarized in Section 2.5.

## 2.2 Obtaining the GRIB2 Model Output Files

For the purpose of generating statistics for a new MED, only GFS output files from the National Center for Environmental Prediction (NCEP) or GALWEM output files from the Air Force 557th Weather Wing (557 WW, formerly the Air Force Weather Agency [AFWA]) may be used as input at this time. For the initial version of a new MED, only GFS output files were used, since GALWEM data are not archived and "age out" after a few days. Recent operational 0.25° GFS output may be downloaded from the relevant NOAA website, https://www.nco.ncep.noaa.gov/pmb/products/gfs/, if no older than about 9 to 10 days. If the appropriate software is installed on the computer, the *wget* command with the appropriate path and filename may be used to obtain the files or the files may be downloaded via the website. Otherwise, those data may be obtained via the National Center for Atmospheric Research (NCAR) (https://rda.ucar.edu/datasets/ds084.1/#!access) for GFS files starting from about a day after being produced at NCEP. However, NCAR requires the user to register and then provides a user password. The user selects the data on the appropriate website and a short Python script is generated that the user can download and run on the local computer to obtain the data files. This script may be modified with respect to the requested GFS output filenames so that files that are different from the initial request may be obtained "directly" without manually accessing the website via a browser. The name of the Python script is *download_ds084.1.py*. The user types the script name (e.g., ./download_ds084.1.py if permissions are set for execution) and the files are downloaded directly to the user's computer.

## 2.3 Generation of Vertical Profiles with a User-Defined Vertical Structure from RAOB Soundings

The generation of vertical profiles with a user-defined vertical structure from a RAOB sounding (in text format) is performed using a Bash script, *raoball.sh*. The word "all" or "full" in this and other scripts described later in this report indicate that the profiles with the user-defined vertical structure have a full set of variables that may be a part of the new MED. They include the height (or pressure level) of each data line, wind direction, wind speed, u and v wind components, vector wind, temperature, virtual temperature, pressure (or height), and relative humidity.

The procedure to run this Bash script is to type its name:

./raoball.sh

The script first asks for the filename followed by the file type, which can be either w (from the University of Wyoming archive site) or n (from the NOAA RAOB archive site). It then asks for the path of the input file. This path is copied to the file *input_parameters* that has the input and output paths for the called C programs. It then asks if the output path is the same as the input path, and if not, then asks for the output path. The path must end with a backward slash mark (e.g., *userdata/*). If the C executables are in the same directory as the input or output file, "./" is sufficient for the input or output file, respectively. The script finally queries the user for the vertical coordinate of the output file: height (h), pressure (p), or both (b). The output files have characters added to the input name as follows: for height-based profiles, "_USRLVL" for data at the levels as defined by *usrhgt_lvls* and "_USRMSG" for data for the layers bounded by those levels (except the surface values, which are taken from the "surface" data of the input file); and for pressure-based profiles, "_USRLVL_P" for data at levels defined by *usrprs_lvls* and "_USRMSG_P" for layers bounded by those pressure levels (except surface data taken from the "surface" data of the input file). The files *usrhgt_lvls* and *usrprs_lvls* must be in the same directory as the respective C executable or the program will end with an error message. At the end of the included "while" loop, the script asks if the user wants to process another RAOB input file. If yes, the script returns to ask the name of the input file; if no, the script ends. Note that the initial input pathname is used for all successive input RAOB files. Section 2.5 has additional details on generating a profile with a user-defined vertical structure. Table 1 presents queries, responses, and meanings and/or brief explanations for *raoball.sh*. Figure B-1 in Appendix B is similar to the one in Appendix B of Cogan (2018), but reflects the recent modifications.

**Table 1 Queries, responses, and meanings and/or brief explanations for *raoball.sh.* The queries in the table are shortened versions of what appear when running the script.**

| Query | Response | Meaning and/or explanation |
|---|---|---|
| Enter input RAOB filename. | Filename | Filename only, without pathname. |
| Is the file from the U of Wyoming or NOAA website? | w or n | Enter w for the University of Wyoming WWeb or n for the NOAA ESRL GSD database for RAOBs. If another character, the script will exit. |
| Enter input pathname. | Pathname | Use full path or relative to directory with the called C executable. Short cuts such as "./" or "../" will work. This path is used for all input RAOBs entered for this run of the script. |
| Is output path same as for input? | y for yes, other character for no | If yes, it sets output path = input path. If no, then one has to enter output path (next query if needed). |
| Enter output path (only if not the same as for input) | Pathname | Use full path or relative to directory with the called C executable. Short cuts such as "./" or "../" will work. Only asked if not the same as the input path. |
| For output profiles enter character for vertical coordinate. | h for height, p for pressure, or b for both | Determines vertical coordinate or basis of output profiles with a user-defined vertical structure in height (above ground level [AGL]) or pressure. If both, then separate height and pressure profiles are generated. The height and/or pressure levels are input via the usrhgt_lvls and/or usrprs_lvls files. If a character other than h, p, or b is entered the script ends with an error message |
| Is there another RAOB to process? | y for yes, other character for no | If yes returns to the first query via a while loop. If no, the script ends. |

## 2.4 Generation of Vertical Profiles with a User-Defined Vertical Structure from NWP Output Files

This Bash script calls Python 3 scripts and C programs that allow the user to optionally run the complete set of programs or skip the generation of the profiles with a user-defined vertical structure. Here vertical structure refers to the definition of heights or pressure levels for the vertical coordinate. In the current version, the user is prompted to input several input parameters such as the input filename and location's latitude and longitude. As noted in Section 2.3, the word "full" or "all" in the script name indicates that the profiles with the user-defined vertical structure have a full set of variables that may be a part of the new MED.

The script is run by entering its name:

./wgrb2all.sh

The first item entered is the name of the input file and the second is the user-selected name of the output file. Afterward, the script appends characters (i.e., _out) to produce the name of the profile extracted from the model output with data lines at model pressure levels. Other items include the longitude and latitude of the desired profile. The number of grid points in x and y directions of the output grid and their spacing are set in the script. The grid has a spacing or interval of 0.00001° (the default value, which is roughly 1 m), and the number of x and y grid points is two each for an essentially vertical profile. The location longitude and latitude are in decimal degrees (e.g., 45.26) and are within the bounds –180 < longitude < 180 and –90 < latitude < 90, respectively. Those parameters are used as input to wgrib2 commands that produce a very small GRIB2 file having a 2 × 2 horizontal grid and extract a profile from that small file. The wgrib2 procedures are described starting in Section 2.4.1. In addition, the script asks for the input and output directories with their paths (can be the same) for the external C programs that generate profiles with user-specified height or pressure levels and layers. The C programs have the same requirements as noted in Section 2.3 (e.g., require two files that have the height or pressure levels that define the vertical structure of the output profiles, *usrhgt_lvls* and *usrprs_lvls* for height and pressure levels, respectively).

At the end of the included "while" loop, the script asks if another site will be processed for the same input GRIB2 file. If yes, the script returns to the query for the output filename; if no, the script ends.

The filename of the output profile from the called wgrib2 commands and Python 3 script has the input filename plus "_out" appended at the end. The filename of the output with a user-defined vertical structure has additional appended characters as described in Section 2.3 on *raoball.sh* (e.g., "_USRLVL" appended at the end of "gfs_OUN_2020030700_out" for an output filename of "gfs_OUN_2020030700 _out_USRLVL"). Also, the user has the option of skipping the computation of user-defined profiles, or generating them based on height (h), pressure (p), or both height and pressure (b). Table 2 presents the queries, responses, and meanings and/or brief explanations for *wgrb2all.sh*. Figure B-2 in Appendix B is similar to the one in Appendix B of Cogan (2018), but also contains the recent modifications.

**Table 2 Queries, responses, and meanings and/or brief explanations for *wgrb2all.sh*. The queries in the table are shortened versions of what appears when running the script.**

| Query | Response | Meaning and/or explanation |
|---|---|---|
| Enter input GRIB2 filename. | Filename | This version of the script processes GRIB2 output from GFS and GALWEM (also GRIB2 output from WRF when available). The same input file is used for all user-selected sites for this run. |
| Enter the pathname. | Pathname | Use full path or relative to directory with the called C executable. Short cuts such as "./" or "../" will work. This path is used for all sites to be processed for the selected GRIB2 input file. |
| Enter name of output file without "_out" added. | Filename | The "_out" extension is added to the output file from the included Python 3 script. That file has the more readable tabular format. |
| Enter the longitude and latitude of the site in decimal degrees. | Two numbers beginning with the longitude | The longitude and latitude are in decimal degrees (e.g., 45.26) and are within the bounds –180 < longitude < 180 and –90 < latitude < 90, respectively, where a minus sign indicates west longitude and south latitude. |
| Generate profiles with user-defined structure? | Enter y or yes for yes, other character or string for no | If yes, then script will call C program(s) to produce profiles with a user-defined structure in height (AGL) and/or pressure. If no, then the output profile is the selected filename + _out (e.g., Site_2020041512_out). |
| Output path the same as input path? | Enter y or yes for yes, other character or string for no | If yes, then set output path = input path. If no, then the script will ask for output path. |
| Enter output path (only if not the same as input path). | Pathname | Use full path or relative to directory with the called C executable. Short cuts such as "./" or "../" will work. Only requested if not the same as the input path. |
| For output profiles enter character for vertical coordinate. | h for height, p for pressure, or b for both | Determines vertical coordinate or basis of output profiles with a user-defined vertical structure in height (AGL) or pressure. If both, then both height- and pressure-based profiles are generated. The height and/or pressure levels are input via the usrhgt_lvls and/or usrprs_lvls files. If a character other than h, p, or b is entered, the script ends with an error message. |
| Is there another location within the current GFS or GALWEM domain? | y for yes, other character for no | If yes, then return to query for name of output file (third query in this table) via a while loop. If no, the script ends. |

### 2.4.1 Wgrib2 Commands

The Bash script *wgrb2all.sh* calls two wgrib2 commands. First, a 2 × 2 grid point GRIB2 (.grb2) file is created from the larger global or regional GRIB2 file. As used here, *wgrib2* with *–new_grid* will generate a smaller grid interpolated from the fields of the parent grid (http://www.cpc.noaa.gov/products/wesley/wgrib2/new

_grid.html). Note that a "bug" in the standard version will not allow the bounds of the horizontal domain to be less than 1° longitude and latitude. A special version was created using information provided by the developers, which eliminated the issue. However, that still may not be part of the standard software. The horizontal grid points of the smaller (new) grid are interpolated using bilinear interpolation of the larger grid data, unless otherwise specified (e.g., change to nearest neighbor). The command line for GFS or GALWEM output is as follows with uppercase denoting generic names such as for input or output files (e.g., OUTPUT_FILE):

> wgrib2 INPUT_GRIB2 -set_grib_type same -new_grid_winds earth -new_grid latlon LON:X DIRECTION POINTS:DX LAT:Y DIRECTION POINTS:DY SMALLER_GRIB2

where INPUT_GRIB2 is the input GRIB2 file, SMALLER_GRIB2 is the smaller output GRIB2 file, LON and LAT are the user-entered longitude and latitude in decimal degrees, X and Y DIRECTION POINTS refer to the number of grid points in the x- and y-directions, and DX and DY refer to the distance between grid points in the respective directions in units of longitude and latitude, respectively.

In the command line, "*same*" for *-set_grib_type* results in another GRIB2 file and "*earth*" for *-new_grid_winds* leads to winds relative to the Earth versus to the grid or undefined. Winds relative to Earth produces values that may be compared to measurements such as from a RAOB. The parameter "*latlon*" for *-new_grid* results in a new grid interpolated from the parent (old) grid, where the listed longitude and latitude are those for the new grid's lower-left corner (approximately the southwest corner).

An example for a location near the US East Coast is the following:

> wgrib2 gfs.t00z.pgrb2.0p25.f012 -set_grib_type same -new_grid_winds earth -new_grid latlon -76.12:2:0.00001 39.10:2:0.00001 small_file.grb2.

The file small_file.grb2 has a 2 × 2 horizontal grid with a 0.00001° grid interval, which translates to roughly a 1-m separation. Essentially the result is a vertical profile for the selected longitude and latitude. Information on these and many other arguments used for *wgrib2* may be found on the NOAA website http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2/long_cmd_list.html.

Next, extract a sounding from the small grib2 file. That process is accomplished using another *wgrib2* command:

> wgrib2 SMALLER_GRIB2 -v -s -lon LON LAN,

where SMALLER_GRIB2 is the output from the previous command, -v refers to verbose output (includes the data values), -s refers to simple inventory (listing of

the output variables and parameters), and -lon produces data for the nearest grid point to the stated longitude and latitude. With a grid separation of approximately 1 m, the values are essentially at the stated coordinates. This *wgrib2* process produces a list of the variables, data values, and other information for the output grid point and prints it on the screen. To save the output, redirect the data to a separate file. Using the aforementioned example, one uses the following:

```
wgrib2 small_file.grb2 -v -s -lon -76.41 39.10 > SiteA_profile.
```

The output is very verbose and contains additional variables not used here. The following few lines out of a total of over 400 indicate the complexity of the output. Each line of output begins with the line number:

```
62:12810:d=2017081500:ABSV Absolute Vorticity [1/s]:50 mb:6 hour
fcst::lon=9.930000,lat=52.810000,i=1,ix=1,iy=1,val=0.000109774
63:13020:d=2017081500:O3MR Ozone Mixing Ratio [kg/kg]:50 mb:6 hour
fcst::lon=9.930000,lat=52.810000,i=1,ix=1,iy=1,val=3.67529e-06
64:13230:d=2017081500:HGT Geopotential Height [gpm]:70 mb:6 hour
fcst::lon=9.930000,lat=52.810000,i=1,ix=1,iy=1,val=18817.7
```

### 2.4.2 Convert the Output File into a User-Friendly Form

The output from the *wgrib2* process of the previous section is very wordy and not readily useable for additional processing, such as to create a vertical profile with a height structure similar to a computer MET message (METCM). A Python 3 script, *gg_wg2.py*, converts the output into a readily readable and useable form. The first two letters of the filename refer to GFS (g) and GALWEM (g) and wg2 to wgrib2 output as the input to the script. It reads in the aforementioned output file from *wgrib2*, extracts appropriate information, and lists the extracted information in a readable tabular form.

To run the script for either a GFS or a GALWEM file, type

```
python3 gg_wg2.py INPUT_FILE
```

For example,

```
python3 gg_wg2.py gfs_GYX_2020030700
```

which produces the output file gfs_GYX_2020030700 _out. Table 3 shows part of this output file including the header information.

**Table 3 Part of the output from *gg_wg2.py* showing the header and first several data lines, where P is pressure, Hgt is height (mean sea level [MSL]), Tmp is temperature, RH is relative humidity, and U and V are the horizontal wind components. Units are as shown in the table.**

| 12-hour forecast after model start at: 2020030700<br>Latitude: 43.900 Longitude: –70.250 | | | | | |
|---|---|---|---|---|---|
| P<br>(hPa) | Hgt<br>(m) | Tmp<br>(K) | RH<br>(%) | U<br>(m/s) | V<br>(m/s) |
| 1008.7 | 70.4 | 267.00 | 50.9 | 0.00 | –7.28 |
| 1000.0 | 138.3 | 266.07 | 51.0 | –0.49 | –11.40 |
| 975.0 | 334.8 | 264.16 | 54.9 | –1.15 | –15.40 |
| 950.0 | 535.1 | 262.40 | 58.7 | –1.95 | –17.37 |
| 925.0 | 739.4 | 260.92 | 57.9 | –3.41 | –19.23 |
| 900.0 | 948.6 | 261.02 | 37.1 | –7.44 | –21.82 |
| 850.0 | 1389.7 | 265.41 | 6.7 | –13.96 | –21.40 |
| 800.0 | 1860.5 | 264.32 | 4.0 | –13.03 | –18.31 |

The GFS output files from NOAA starting on 12 Coordinated Universal Time (UTC) 11 May 2016 extend up to 1.0 hPa, but before then the last data line ended at 10 hPa. The aforementioned methods work for both. Also, recent GALWEM files reach up to over a 50-km altitude. The pressure is to the nearest hPa so that the value for, say, 54 km is listed as 0 hPa. To try to make the top level a little more realistic, a pressure listed as 0 is arbitrarily changed to 0.5 hPa. However, for nearly all terrestrial applications (not space weather), the minimum pressure is very likely to be greater than or equal to 1 hPa.

## 2.5 Sounding with User-Defined Height or Pressure Levels and Layers

The ARL_MET-profile_Converter programs (located at the following GitHub site: https://github.com/usarmyresearchlab/ARL_MET-profile_Converter) may be used to produce soundings of user-defined levels or layers from RAOBs or model output vertical profiles. Cogan (2017), and its included references, describes the set of C programs and its application, and presents samples of output. The software produces output for height levels and layers or pressure levels and layers. The heights are AGL. Different versions produce output for two different sets of variables. This set of software was modified so that all likely MED variables were present in each of the output soundings with a user-defined height- or pressure-based structure. In addition, the programs for processing data from the NOAA site were modified so as to ignore data lines where dewpoint was missing when temperature was included. This situation is relatively rare and data lines within a very large majority of soundings have values for both variables. The current programs for converting data from RAOBs have four versions to account for the differences in soundings from the NOAA archive and the University of Wyoming WWeb archive, as well as output with either height or pressure as the vertical

coordinate (height- or pressure-based). The programs used with GFS and GALWEM data have only two versions, since the sounding format for both sources is the same.

The input and output directories are defined in the *input_parameters* file, which is in the same directory as the C program executable. For height-based output, the file *usrhgt_lvls* contains the output levels, which are the boundary levels of the layers starting with the surface through the highest level listed. It is also in the same directory as the C program. Note that the first data line (may be listed as line 0) in the layer output has values for the surface only. For the pressure-based output, the parameter file is *usrprs_lvls* and is in the same directory as the C program. As with the height-based output, the first line (line 0) has values for the surface only. The number 2 at the end of the executable name indicates modified versions for the MED analysis.

For RAOBs extracted from the WWeb archive, the program is run as follows for height-based output:

./convertwyo2 INPUT_FILE

and for pressure-based output

./convertprswyo2 INPUT_FILE

where INPUT_FILE is the generic name for an input file (e.g., ETGB_20200415).

For RAOBs extracted from the NOAA archive, the program is run in a similar manner as follows:

./convertnoaa2 INPUT_FILE

and for pressure-based output

./convertprsnoaa2 INPUT_FILE

For profiles extracted from GFS or GALWEM data, the program is run using the following command line for height-based output:

./convertgfs2 INPUT_FILE

and for pressure-based output

./convertprsgfs2 INPUT_FILE

For a height-based example, the command line could read

./convertgfs2 ETGB_2020041506_out

The output files for all versions have the following extensions appended to the input filename: _USRLVL and _USRMSG for height-based level and layer profiles, respectively, and _USRLVL_P and _USRMSG_P for pressure-based level and layer profiles, respectively. For example,

./convertprsgfs2 ETGB_2017081506_out

produces the level and layer output files ETGB_2017081506_out_USRLVL_P and ETGB_2017081506_out_USRMSG_P, respectively. Also, some operating systems may not use the "./" before the executable name.

The C programs discussed previously produce wind-related output in the form of wind speed, wind direction, u wind component, and v wind component. The thermal and moisture variables are temperature, virtual temperature, and relative humidity. Table 4 shows a section of pressure-based output with a user-defined structure from a vertical profile extracted from a GFS output file. The header information and several data lines are shown.

**Table 4 Output from *convertprsgfs2* showing the header and first several data lines. A wind directly from the north may be listed as either 0.0 or 360.0. The units are as listed. The -999.0 values for Ceiling and Visibility are fillers at this time, but may have measured or computed values in future applications. The spacing of the columns including that of the column labels was changed from the original so that all items for each row appear on one line in the table.**

**USER-DEFINED PRESSURE LEVEL OUTPUT (MODEL)**
**Date: 2020030700 Time: 12 Latitude: 43.900 Longitude: -70.250**
**Elevation: 70.40 Ceiling: -999.0 Visibility: -999.0**

| Level | Pressure (hPa) | Height (m) | Wind_Dir (°) | Wind_Speed (m/s) | U_comp (m/s) | V_comp (m/s) | Virt_Temp (K) | Temperature (K) | Rel_Humidity (%) |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 1008.7 | 0.0 | 360.0 | 7.28 | 0.00 | –7.28 | 267.20 | 267.00 | 50.90 |
| 1 | 1000.0 | 67.9 | 2.5 | 11.41 | –0.49 | –11.40 | 266.25 | 266.07 | 51.00 |
| 2 | 990.0 | 145.9 | 3.3 | 13.01 | –0.75 | –12.99 | 265.49 | 265.31 | 52.55 |
| 3 | 975.0 | 264.4 | 4.3 | 15.44 | –1.15 | –15.40 | 264.33 | 264.16 | 54.90 |
| 4 | 950.0 | 464.7 | 6.4 | 17.48 | –1.95 | –17.37 | 262.57 | 262.40 | 58.70 |
| 5 | 925.0 | 669.0 | 10.1 | 19.53 | –3.41 | –19.23 | 261.07 | 260.92 | 57.90 |
| 6 | 900.0 | 878.2 | 18.8 | 23.05 | –7.44 | –21.82 | 261.12 | 261.02 | 37.10 |

The C source code, compile files, and executables are currently located in a separate subdirectory. If changes are made to the source code, compile in the separate directory and only copy the executable to the main directory where the Python 3 and Bash script files are located. In that way, the number of files within the main directory is kept to a more manageable size.

## 3. Vertical Profile Differences and Statistics

The following Bash scripts contain Python 3 scripts that compute 1) the differences by variable and data line (i.e., each line represents a height or pressure level or

layer) between user-defined vertical profiles from any two sources and calculate basic statistics for those differences by variable over all data lines, and 2) compute statistics by variable and data line over multiple tables of the differences. An example could be the root mean square (RMS) error (RMSE) (or RMS difference) of temperature at 1500 m (or 850 hPa) over, say, 20 tables of differences. For this example, the sample size would be 20 for temperature at 1500 m (or 850 hPa).

## 3.1 Computation of Differences

The Bash script *prof_compareall.sh* computes the differences between soundings from two different sources. It can process one or more sounding pairs from those sources where one pair consists of one sounding from each of both sources with the same vertical structure. The script asks for the input filenames and a character that indicates whether the files are height or pressure based. For pressure-based input soundings, the script also asks if the user wants to manually enter a baseline pressure. The baseline pressure limits the output to pressures at and lower (pressure levels at or higher) than the baseline. If yes, then the pressure in hPa is entered. The baseline pressure should not exceed the lowest surface pressure found in the input files. The highest pressure from *usrprs_lvls* common to all input soundings is selected if the user-selected value is higher than the minimum surface pressure. To run the program, enter the name of the script:

```
./prof_compareall.sh
```

The default output filename is diff_profiles; the user has the option of choosing another filename. If the default name is chosen, the script asks if the run is the first one. If yes, the previous version of the default output file, if any, is removed. The script iteratively processes one or more pairs of soundings where each successive output table of differences (one table per pair) is appended to the previous ones within the same file. Each time through the loop, the user is queried if the iteration that just finished was the last one. If yes, the script ends, and if no, another pair of input files are processed. Table 5 presents the queries, responses, and meanings and/or brief explanations. Figure B-3 in Appendix B is similar to the one in Appendix B of Cogan (2018), but also contains recent modifications.

**Table 5 Queries, responses, and meanings and/or brief explanations for *profcompareall.sh*. The queries in the table are shortened versions of what appears when running the script.**

| Query | Response | Meaning/Explanation |
|---|---|---|
| Input files based on height or pressure? | h or p | Vertical coordinate of the input has height (h) or pressure (p). If a character other than h or p is entered, the program exits with an error message. |
| Manually enter highest pressure for output file (only if pressure selected in first query)? | y for yes, other character for no | Only if pressure (p) is entered in first query, choose y for manual entry of baseline pressure for output file via next query. If no, the software will find the value of the highest common pressure. |
| Enter baseline pressure (only if manual entry). | Pressure in hPa | Baseline pressure should be at or lower than the lowest surface pressure in all the compared soundings. If a higher pressure is entered, the script chooses the highest common pressure in usrprs_lvls. |
| Use default name for output file? | y for yes, other character for no | If yes, use default name, that is, diff_profiles. If no, the user will enter output filename. |
| Is this the first run using the default output filename (only if use default name)? | y for yes, other character for no | Only asked if the default name is chosen for output file. If yes, then the previous version if any is deleted. If no, then new output is appended to the existing default output file. |
| Enter output filename (only if not using default). | Output filename | Enter the output filename if not using the default name. |
| At this point enter a while loop. | NA | Successive tables of differences appended to the same file until the loop is ended. |
| Enter first input file. | Filename of first of the initial and any successive pairs of input files. | If the first file is from source A (e.g., model output), then it should be from source A for the entire loop. |
| Enter second input file. | Filename of second of the initial and any successive pairs of input files. | If the second file is from source B (e.g., RAOB), then should be from source B for the entire loop. |
| Is this the last iteration of data input? | y for yes, other character for no | If yes, then the loop ends, and if the default filename was not chosen, the output is sent to the file with the user-chosen output filename. |

In addition to the table of differences, statistics for each column of values for each variable are calculated and presented as a smaller separate table following the respective difference table. Table 6 shows a sample taken from one difference table along with the column statistics.

**Table 6 Sample of a difference table and column statistics**

Compared output from mebdata/gfs_YMEH_2020030300_out_USRLVL and mebdata/YMHB_2020030300_USRLVL
Difference values for listed variables in diff_profiles
Number of data levels 36

| Level | Height (m) | Pressure (hPa) | Wind_Dir (degrees) | Wind_Speed (m/s) | U_wind (m/s) | V_wind (m/s) | Vect_Wind (m/s) | Temperature (K) | Virt_Temp (K) | Rel_Humidity (%) | Hgt(not_diff) (m) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.0 | -9.7 | -37.40 | -1.53 | -1.14 | 1.69 | 2.04 | -0.90 | -0.92 | 1.70 | 0.0 |
| 1 | 0.0 | -9.7 | -55.20 | -3.76 | -1.93 | 4.55 | 4.94 | -1.32 | -1.35 | 2.89 | 100.0 |
| 2 | 0.0 | -9.3 | -49.00 | -3.98 | -2.48 | 4.48 | 5.12 | -0.78 | -0.80 | 1.84 | 250.0 |
| 3 | 0.0 | -9.1 | -30.30 | -2.52 | -2.30 | 2.27 | 3.23 | -0.74 | -0.76 | 1.03 | 500.0 |
| 4 | 0.0 | -8.9 | -8.60 | -1.72 | -1.80 | 0.29 | 1.82 | -0.71 | -0.73 | 2.43 | 750.0 |
| 5 | 0.0 | -8.8 | -18.40 | -2.29 | -2.57 | 1.20 | 2.84 | -0.71 | -0.73 | 1.23 | 1000.0 |
| 6 | 0.0 | -8.3 | -41.40 | -1.25 | -3.45 | 5.73 | 6.69 | 2.55 | 2.34 | -36.65 | 1500.0 |
| . | | | | | | | | | | | |
| . | | | | | | | | | | | |
| . | | | | | | | | | | | |
| 33 | 0.0 | -0.2 | 13.30 | -8.61 | 8.23 | -3.15 | 8.81 | -0.01 | -0.01 | 0.17 | 28000.0 |
| 34 | 0.0 | -0.2 | 55.60 | -8.00 | 7.98 | 6.68 | 10.41 | -0.32 | -0.32 | 0.15 | 29000.0 |
| 35 | 0.0 | -0.2 | 80.60 | -6.99 | 2.85 | 10.06 | 10.46 | -0.61 | -0.61 | 0.13 | 30000.0 |
| Column Statistics | | | | | | | | | | | |
| Mean | 0.0 | -3.78 | -5.35 | -10.81 | -7.70 | -1.90 | 11.84 | -0.22 | -0.23 | 3.73 | |
| Median | 0.0 | -3.02 | -0.95 | -8.54 | -6.42 | -2.70 | 10.43 | -0.60 | -0.60 | 1.37 | |
| Std Dev | 0.0 | 3.38 | 35.19 | 7.76 | 9.18 | 7.27 | 7.60 | 1.28 | 1.26 | 10.21 | |
| RMSE | 0.0 | 5.04 | 35.11 | 13.24 | 11.89 | 7.42 | 14.01 | 1.28 | 1.27 | 10.73 | |

The included Python 3 scripts may be run separately, that is, prof_comparefull_height.py, prof_compare_baseprs.py, or prof_comparefull _pressure.py. The Python 3 script for height-based input is executed by entering the script name followed by the two input filenames and two characters:

python3 prof_comparefull_height.py INPUT_FILE1 INPUT_FILE2 X -v

where INPUT_FILE1 and INPUT_FILE2 are the input files from which the differences are computed, and X is a character that tells the program to append (a) the output or write (w) to its own file. In the version for MED, X is always set to "a" (append). The two files must have the same height intervals (use the same *usrhgt_lvls* file noted previously to set up the height structure). However, the number of height or pressure levels can be different. The script will only compute differences up to the highest data level of the shortest profile. Note that highest pressure level refers to the level with the lowest pressure. The "-v" is optional and is the short form for the argument --verbosity (either form may be used) that if "true" means something will be displayed on the screen, in this case, the command line input (e.g., the filenames). Although the parameter -v is not used in the current version of the Bash script, it may be useful for testing if running the script separately. For information on usage of the script, type -h (long form is --help) after the script name (thereafter the script will end). For details and some examples of the use of -v and -h, see https://docs.python.org/3/howto/argparse.html#id1 and the included links. An example of use of the script is the following:

python3 prof_comparefull_height.py gfs_OUN_2018052512_USRLVL
R_2018052512_USRLVL a -v

where each difference value is from the GFS-derived, height-based sounding less that from the RAOB sounding, and the output goes to either the default file (*diff_profiles*) or a file with a name chosen by the user. In this example, -v is used and the names of the input files and the parameter a (append) are printed to screen with an explanation of their meaning.

There are two Python 3 scripts for pressure, one with and one without a user-entered maximum allowable pressure (lowest pressure level). Since the pressure at the surface can vary from day to day for the same location, and vary considerably between multiple locations, the user can either enter a pressure value that is equal to or less than the lowest surface pressure level (highest surface pressure) or let the program select the lowest pressure level (highest pressure). The procedure for running the Python 3 script with user input of the highest pressure is as follows:

```
python3 prof_compare_baseprs.py INPUT_FILE1 INPUT_FILE2 PPP X -v
```

where INPUT_FILE1 and INPUT_FILE2 are the input files from which the differences are computed, PPP is the lowest pressure level (highest pressure) in hPa (or millibars), and X is a character that tells the script to append (a) the output or write (w) to its own file (always set to "a" for MED). The parameter -v is optional, and if used, additional information is printed to the screen, as noted previously. The value of PPP should be no higher than the lowest surface pressure for both input soundings for a single comparison or all input soundings for two or more appended comparisons. The need to have a common minimum pressure level for all input soundings limits the comparisons to the common levels within the usrprs_lvls file. Different surface heights above MSL does not affect the comparisons with respect to height since all are AGL, that is, start at 0 m. The output file from the Python 3 script always goes to *diff_profiles*, which is the default filename, although the user has the option of selecting a different filename via the Bash script.

An example is the following:

```
python3 prof_compare_baseprs.py gfs_OUN_2018052512_USRLVL_P
R_2018052512_USRLVL_P 928 a
```

where the input files beginning with gfs and R (RAOB) files are based on pressure levels, 928 is the user-defined baseline pressure in hPa, and a indicates the output is appended. The verbosity parameter (-v) is not used in this example.

The second script to compute differences using pressure-based input files lets the software select the highest pressure (lowest pressure level). For this version, the command line is as follows:

```
python3 prof_comparefull_pressure.py INPUT_FILE1 INPUT_FILE2 X -v
```

where, as before, INPUT_FILE1 and INPUT_FILE2 are the input files from which the differences are computed; X is a character that tells the program to append (a) the output or write (w) to its own file (always set to "a" for MED); and -v is the optional verbosity argument, as noted previously. An example is as follows:

```
python3 prof_comparefull_pressure.py gfs_OUN_2018052512_USRMSG
_P R_2018052512_USRMSG_P a -v
```

where both input files have pressure-based layers and the output is appended to *diff_profiles*. Here -v is used and the names of the input files and the parameter a (append) are printed to screen with an explanation of their meaning.

A short Python 3 script, *duplichk.py*, is used to detect duplicated files in the file with appended tables of differences. It compares the path and filenames in the first header line of each appended table and if a duplicate is found prints those names and the number of duplicates on the screen. To run this script, enter its name:

```
python3 duplichk.py
```

The script queries the user for the name of the file to be checked. For example, if the file *diff_profiles_test* is to be checked the user enters its name after the query "Enter name of file to check for duplication:". If the same difference table is duplicated one or more times, the input filenames for that table are printed and will be repeated for each duplication (e.g., if duplicated three times, the same line is repeated three times). Table 7 has an example of the output to screen from *duplichk.py*.

**Table 7 Sample of output to screen from *duplichk.py* including the request for the input file. One table was duplicated. The filenames include the path if not in the same directory and are in the first header line of the respective difference table.**

```
Enter name of file to check for duplication: diff_test

Starting search for duplicate cases (1 case = 1 site at 1 date/time).

Difference tables duplicated
        Table ID
  meddata/gfs_GYX_2020030300_out_USRLVL and
meddata/GYX_2020030300N_USRLVL

Number of duplicates:  1

END OF DUPLICATION CHECK
```

## 3.2 Statistics by Variable and Data Line

The Bash script *profstatfull.sh* is used to generate basic statistics of the differences by variable and data line over all the appended difference tables in *diff_profiles* (or other filename for the appended output). The statistics computed at this time are mean, median, standard deviation, and RMSE. Here error refers to the difference relative to the "truth" sounding, which is often a RAOB, but could be, for example, a sounding derived from another configuration of the model. In addition, the number of samples used in the computation of the aforementioned statistics for each data line appears in the last column of each of the output statistics tables. The script asks for the input filename (appended file from *prof_compareall.sh*), a character that indicates whether the files are height or pressure based, and a string that tells the script whether or not to convert input differences into their absolute values. This script in turn calls the Python 3 program *profstatfull.py*. The default output filename is *profiles_stats* or *profile_stats_absval*, if absolute values are used. To run the Bash script, type the name

```
./profstatfull.sh
```

and enter the information as requested. Note that the previous copy of the output file is deleted if the user indicates it is the first use of the output file so as to avoid appending multiple tables for each statistic. Table 8 presents the queries, responses, and meanings and/or brief explanations and Fig. B-4 presents an overview of the script in the form of a flowchart. It is similar to the respective one in Appendix B of Cogan (2018), but also contains recent modifications.

**Table 8 Queries, responses, and meanings and/or brief explanations for *profstatfull.sh.* The queries in the table are shortened versions of what appears when running the script.**

| Query | Response | Meaning/explanation |
|---|---|---|
| Compute statistics of absolute differences? | y for yes, other character for no. | Determines whether statistics will use absolute values of differences or differences as listed. |
| Append to previous output file? | y for yes, other character for no. | If no, default output filename deleted if it exists. If one wants to append to file with a user-defined name, enter the name later when asks for a user-defined filename. |
| Enter input filename. | Name of input file | Output file from prof_compareall.sh. |
| Difference values based on height or pressure. | h for height or p for pressure | What type of profiles: vertical coordinate height (h) or pressure (p). Must enter h or p. |
| Use the default output filename? | y for yes, other character for no | If no, then the user will enter a filename after the statistics are computed. Default name is profile_stats except if absolute values of differences selected then name is profile_stats_absval. |
| Enter output filename (if do not use default). | Name of output file. | Only used if the default name is not used. |

Table 9 presents a sample of the output from *profstatfull.sh.*

**Table 9 Sample of output from *profstatfull.sh* for profiles that have height (AGL) as the vertical coordinate. Only the first eight data lines are shown for each statistics table (lines 0–7). Since height is the vertical coordinate, the mean and so on of the differences of height are 0. Additional columns provide the height levels and the number of samples (one sample for each input table that has data for that line and height). The variables are the same as in Table 6 and the units are as shown.**

Statistics for profile differences by variable and data line.

Mean values

| Level | Height (m) | Pressure (hPa) | Wind_Dir (degrees) | Wind_Speed (m/s) | U_wind (m/s) | V_wind (m/s) | Vect_Wind (m/s) | Temperature (K) | Virt_Temp (K) | Rel_Humidity (%) | Hgt(not_diff) (m) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | -0.37 | -2.21 | -2.71 | 1.93 | -0.40 | 4.14 | -3.54 | -3.52 | 9.90 | 0.0 | 10 |
| 1 | 0 | -0.52 | -10.34 | -2.96 | 2.10 | -0.18 | 6.06 | -2.81 | -2.79 | 8.39 | 100.0 | 10 |
| 2 | 0 | -0.67 | -9.50 | -2.90 | 2.32 | -0.87 | 7.53 | -1.90 | -1.87 | 5.13 | 250.0 | 10 |
| 3 | 0 | -0.55 | -17.00 | -1.77 | 2.38 | -1.56 | 7.81 | -1.06 | -1.04 | 3.06 | 500.0 | 10 |
| 4 | 0 | -0.39 | 13.44 | -3.07 | 1.22 | -3.62 | 9.20 | -0.45 | -0.50 | -5.54 | 750.0 | 10 |
| 5 | 0 | -0.38 | 15.91 | -3.73 | -0.50 | -4.06 | 9.87 | 0.29 | 0.22 | -10.51 | 1000.0 | 10 |
| 6 | 0 | -0.39 | 5.35 | -6.70 | -4.52 | -4.30 | 11.90 | 0.33 | 0.27 | -12.13 | 1500.0 | 10 |
| 7 | 0 | -0.24 | 1.04 | -9.66 | -7.88 | -4.78 | 12.91 | -0.13 | -0.13 | -6.00 | 2000.0 | 10 |

Median values

| Level | Height (m) | Pressure (hPa) | Wind_Dir (degrees) | Wind_Speed (m/s) | U_wind (m/s) | V_wind (m/s) | Vect_Wind (m/s) | Temperature (K) | Virt_Temp (K) | Rel_Humidity (%) | Hgt(not_diff) (m) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | -2.40 | -1.95 | -2.17 | 2.87 | -0.13 | 4.33 | -1.94 | -1.90 | 4.60 | 0.0 | 10 |
| 1 | 0 | -2.60 | -10.50 | -2.27 | 2.78 | 0.69 | 5.41 | -1.15 | -1.14 | 5.62 | 100.0 | 10 |
| 2 | 0 | -2.75 | -14.75 | -3.15 | 1.26 | 1.42 | 7.00 | -0.53 | -0.55 | 3.19 | 250.0 | 10 |
| 3 | 0 | -1.75 | -21.55 | -2.40 | 0.49 | 0.73 | 8.41 | -0.23 | -0.18 | 3.29 | 500.0 | 10 |
| 4 | 0 | -0.65 | -9.10 | -2.07 | -1.35 | -2.36 | 10.24 | -0.04 | -0.14 | -4.10 | 750.0 | 10 |
| 5 | 0 | -0.50 | -11.80 | -4.39 | -1.59 | -3.00 | 8.20 | 0.02 | 0.05 | -7.16 | 1000.0 | 10 |
| 6 | 0 | -0.45 | -1.00 | -6.27 | -2.23 | -3.68 | 8.13 | 0.01 | -0.16 | -5.96 | 1500.0 | 10 |
| 7 | 0 | -0.20 | -2.25 | -8.83 | -6.04 | -5.63 | 12.45 | -0.44 | -0.40 | 3.50 | 2000.0 | 10 |

Standard Deviation values

| Level | Height (m) | Pressure (hPa) | Wind_Dir (degrees) | Wind_Speed (m/s) | U_wind (m/s) | V_wind (m/s) | Vect_Wind (m/s) | Temperature (K) | Virt_Temp (K) | Rel_Humidity (%) | Hgt(not_diff) (m) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 7.51 | 37.53 | 2.28 | 2.50 | 3.21 | 1.81 | 4.22 | 4.20 | 15.65 | 0.0 | 10 |
| 1 | 0 | 7.42 | 39.10 | 3.42 | 4.09 | 4.72 | 2.59 | 3.65 | 3.66 | 11.33 | 100.0 | 10 |
| 2 | 0 | 7.01 | 41.12 | 4.58 | 5.47 | 5.77 | 3.56 | 3.17 | 3.20 | 9.53 | 250.0 | 10 |
| 3 | 0 | 6.77 | 47.89 | 5.49 | 6.35 | 5.71 | 4.46 | 2.79 | 2.85 | 4.01 | 500.0 | 10 |
| 4 | 0 | 6.26 | 57.94 | 6.71 | 6.28 | 6.98 | 4.25 | 2.66 | 2.71 | 8.15 | 750.0 | 10 |
| 5 | 0 | 6.05 | 59.61 | 7.87 | 7.28 | 7.46 | 5.29 | 2.09 | 2.14 | 17.25 | 1000.0 | 10 |
| 6 | 0 | 5.60 | 50.78 | 10.25 | 8.40 | 9.39 | 7.48 | 1.19 | 1.18 | 22.15 | 1500.0 | 10 |
| 7 | 0 | 5.19 | 34.77 | 8.02 | 7.42 | 8.35 | 6.56 | 1.44 | 1.39 | 26.60 | 2000.0 | 10 |

Root Mean Square Error values

| Level | Height (m) | Pressure (hPa) | Wind_Dir (degrees) | Wind_Speed (m/s) | U_wind (m/s) | V_wind (m/s) | Vect_Wind (m/s) | Temperature (K) | Virt_Temp (K) | Rel_Humidity (%) | Hgt(not_diff) (m) | Samples |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 7.52 | 37.59 | 3.54 | 3.16 | 3.24 | 4.52 | 5.51 | 5.48 | 18.51 | 0.0 | 10 |
| 1 | 0 | 7.44 | 40.44 | 4.52 | 4.60 | 4.72 | 6.59 | 4.61 | 4.60 | 14.10 | 100.0 | 10 |
| 2 | 0 | 7.05 | 42.20 | 5.42 | 5.94 | 5.83 | 8.32 | 3.69 | 3.71 | 10.82 | 250.0 | 10 |
| 3 | 0 | 6.80 | 50.82 | 5.77 | 6.78 | 5.92 | 9.00 | 2.98 | 3.03 | 5.05 | 500.0 | 10 |
| 4 | 0 | 6.27 | 59.48 | 7.38 | 6.40 | 7.86 | 10.14 | 2.69 | 2.76 | 9.85 | 750.0 | 10 |
| 5 | 0 | 6.06 | 61.70 | 8.71 | 7.30 | 8.49 | 11.20 | 2.11 | 2.16 | 20.20 | 1000.0 | 10 |
| 6 | 0 | 5.61 | 51.06 | 12.25 | 9.54 | 10.32 | 14.06 | 1.23 | 1.21 | 25.25 | 1500.0 | 10 |
| 7 | 0 | 5.20 | 34.79 | 12.55 | 10.82 | 9.62 | 14.48 | 1.45 | 1.39 | 27.26 | 2000.0 | 10 |

The Python 3 script, *profstatfull.py*, called by *profstatfull.sh* may be run separately as follows:

python3 profstatfull.py INPUT_FILE X Y -v

where INPUT_FILE is the input file that holds the appended difference tables (default name is *diff_profiles*), X is either h or p for height- or pressure-based input, respectively, and Y indicates whether to convert the differences in the input tables to absolute values (for absolute values, Y = y for yes). The optional argument -v operates as described in Section 3.1. The output filename currently is *profile_stats*, except that for absolute value output, the name is *profile_stats_absval*. Since each table for each statistic such as mean or standard deviation is appended to the same file, a new run will continue to append, leaving a file with multiple sets of tables for the four statistics. The user should delete the file *profile_stats* or *profile_stats_absval* (or other name if not using the default) before starting a new

run to produce a set of tables as a separate file. To save separate files for different runs, copy the completed set to another file (e.g., profile_stats01) and then delete *profile_stats* or *profile_stats_absval*. The Bash script will delete the file if the user indicates it is the first use of the output file.

### 3.3 Overall Processing Sequence

This section briefly details the overall sequence of processing, from obtaining sounding data to generating the statistics tables. The first main effort is to obtain data from a RAOB archive site (NOAA or University of Wyoming) and one of the providers of model output either "real time" (e.g., NCEP or 557th Weather Wing) or archive (e.g., NCAR) as presented in Sections 2.1 and 2.2, respectively. The next stage is to run the relevant scripts to extract the vertical profiles from the model output and then generate "soundings" with user-defined height or pressure levels and layers as described in Sections 2.3 (RAOB) and 2.4 (model output). At this point, the vertical profiles from RAOB and model output should have the same user-defined vertical structure. In the following stage, a Bash script and the included Python 3 scripts produce vertical profiles of the differences between the coincident model and RAOB files in tabular form as described in Section 3.1. The resultant tables of differences for a number of input file pairs are appended in a single file that becomes the input for a Bash script that in turn calls a Python 3 script that generates tables of statistics as discussed in Section 3.2 with a sample in Table 9.

Figure B-5 in Appendix B presents a "high level" outline of the sequence of in the form of a flowchart.

## 4. Conclusion

Preparing a new MED requires the collection of model and comparison RAOB data (or observations from other sources such as aircraft), the comparison of co-located MET profiles from the model and the comparison data sets, and the generation of appropriate statistics. The scripts and programs within this report help to fulfill the data-processing needs for a new MED. Most of these scripts and programs were developed earlier for other applications and modified for the specific purpose of developing a new MED. The overall set of programs is flexible, in that any change to the height (or pressure) levels or layers for a new MED may be accomplished by revision of a single parameter file that contains the height (or pressure) levels. The software may be used for other applications, such as the comparison between two models or two model configurations, or the comparison between data between two types of observing systems such as from RAOBs and aircraft. Ongoing efforts

include enabling the Bash scripts to run in a type of batch mode where several files can be input using a list of filenames within a text file versus entry via queries to the user. Another is to enable the direct download of RAOB soundings versus manually copying from a screen display.

## 5. References

Cogan J. A generalized method for vertical profiles of mean layer values of meteorological variables. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2015 Sep. Report No.: ARL-TR-7434.

Cogan J. Evaluation of model-generated vertical profiles of meteorological variables: method and initial results. Meteorol Appl. 2017;24:219–229.

Cogan JL. Extraction and comparison of vertical profiles from global and mesoscale models. Aberdeen Proving Ground (MD): Army Research Laboratory (US); 2018 Dec. Report No.: ARL-TR-8589.

## Additional Resources

The following provide the various resources cited in the report:

- Details on the NOAA data archives may be found at https://www.ncdc.noaa.gov/data-access/weather-balloon/integrated-global-radiosonde-archive, which has links to several data archives.
- Older GFS data may be obtained via the NCAR archive: https://rda.ucar.edu/datasets/ds084.1/#!access. In addition, NCAR requires the user to register and then NCAR supplies a user password for access.
- Operational 0.25° GFS output may be downloaded from the NOAA website, https://www.nco.ncep.noaa.gov/pmb/products/gfs/, if no older than about 9 to 10 days.
- The basic civilian versions of the C code reside on the CCDC Army Research Laboratory (ARL) open GitHub site, https://github.com/usarmyresearchlab/ARL_MET-profile_Converter. ARL_MET-profile_Converter is the project name.
- The NOAA ESRL GSD archive site for RAOB data (https://ruc.noaa.gov/raobs/) contains data for sites worldwide, but in a less user-friendly text format than at the University of Wyoming site. It contains data similar to the standard form for WMO soundings with standard and special data levels, but in a more readable text format.
- The University of Wyoming WWeb is located at https://weather.uwyo.edu/upperair/uamap.shtml. Soundings in text and other formats are available from numerous sites around the world. Weather maps for the most recent few months may be found for regions that together cover nearly the entire planet.

# Appendix A. Scripts and Programs

This appendix lists the several programs and scripts that produce vertical profiles of meteorological variables (soundings) from numerical weather prediction (NWP) model output and radiosonde observations (RAOBs).

## A.1. Bash Scripts

The following are the Bash scripts:

- *wgrb2all.sh* extracts one or more "soundings" from a Global Forecast System (GFS) or Global Air Land Weather Exploitation Model (GALWEM) output file and converts them into soundings with user-defined height or pressure levels and layers. More than one sounding may be extracted from an input GRIB2 model output file without rerunning the script.
- *raoball.sh* takes a RAOB from the University of Wyoming's weather website (WWeb) or from the NOAA archive website and converts it to forms with user height or pressure levels and layers. More than one RAOB may be processed without rerunning the script.
- *prof_compareall.sh* computes the differences between height- or pressure-based soundings from different sources (e.g., between two models or a model and a RAOB) and generates a table of differences by height or pressure level or layer. It includes the Python 3 scripts *prof_comparefull_height.py* for 1) height levels or layers and 2) *prof_comparefull_pressure.py* and *prof_compare_baseprs.py* for pressure levels or layers (script or user selects the highest pressure, respectively). Basic statistics are computed for each variable over all heights. Output may be appended to a single file for later processing by profstatfull.sh.
- *profstatfull.sh* computes basic statistics for a set of appended difference tables generated by *prof_compareall.sh*. The statistics (e.g., mean and standard deviation) are for each variable at each level or layer over all the tables. For example, the mean value of temperature for the 1000- to 1500-m layer over all the appended difference tables. Also, it can compute the statistics for the absolute values of the differences. It includes the Python 3 script *profstatfull.py*.

## A.2 Python 3 Scripts (or Programs)

The following are the Python 3 scripts:

- *duplichk.py* checks the file of appended difference tables for duplicate tables.

- *gg_wg2.py* converts a sounding from a GFS or a GALWEM output file, extracted via wgrib2, into a user-friendly format.
- *raobcheck.py* checks for consecutive data lines in a RAOB sounding from the University of Wyoming WWeb site.
- The *prof_comparefull_*.py* scripts process files having values for all the likely input variables (e.g., wind speed and direction, u and v wind components, temperature, relative humidity [RH]). Specific scripts in this group follow.
- *prof_comparefull_height.py* computes the differences between two height-based soundings from different sources (e.g., between two models or a model and a RAOB) and generates a table of differences by height level or layer. Basic statistics are computed for each variable over all heights. Output may be appended to a single file for later processing by *profstatfull.py.*
- *prof_comparefull_pressure.py* computes the differences between pressure-based soundings from different sources (e.g., between two models or a model and a RAOB) and generates a table of differences by pressure level or layer. Basic statistics are computed for each variable over all pressures. Output may be appended to a single file for later processing by *profstatfull.py*.
- *prof_comparefull_baseprs.py* also computes the differences between pressure-based soundings from different sources (e.g., between two models or a model and a RAOB) and generates a table of differences by pressure level or layer. However, the user chooses a lowest pressure level (highest pressure) that is entered as a command line argument versus having the program determine that value. Basic statistics are computed for each variable over all pressures. Output may be appended to a single file for later processing by *profstatfull.py*.
- *profstatfull.py* computes basic statistics for a set of appended difference tables generated by one of the *prof_comparefull_*.py* scripts. The statistics (e.g., mean, standard deviation) are for each variable at each level or layer over all the tables. For example, the mean value of temperature for the 1000- to 1500-m layer over all the appended difference tables. It can process difference tables that have either height or pressure as the vertical coordinate. Also, it can compute the statistics for the absolute values of the differences.

## A.3 C Programs

The C programs combined two earlier versions, almost entirely by modifying the output function. The programs now output wind speed, wind direction, u wind component, v wind component, temperature, virtual temperature, and RH. The number 2 at the end of the filenames identifies the slightly revised programs (e.g., *convertgfs2* vs. *convertgfs*). The *input_parameters* file is produced using the *wgrb2all.sh* script, but also may be generated manually if running one of the C programs by itself. In that case, the user has to ensure that the complete input and output paths are in the *input_parameters* file with the path ending with a "/" (e.g., /home/user/output_data/):

- *convertgfs2* converts a sounding derived from GFS or GALWEM output into a form having user-defined height levels and layers defined by those levels.
- *convertprsgfs2* converts a sounding derived from GFS or GALWEM output into a form having user-defined pressure levels and layers defined by those levels.
- *convertnoaa2* converts a RAOB from the NOAA archive into a form having user-defined height levels and layers defined by those levels.
- *convertprsnoaa2* converts a RAOB from the NOAA archive into a form having user-defined pressure levels and layers defined by those levels.
- *convertwyo2* converts a RAOB from the University of Wyoming WWeb into a form having user-defined height levels and layers defined by those levels.
- *convertprswyo2* converts a RAOB from the University of Wyoming WWeb into a form having user-defined pressure levels and layers defined by those levels.

Notes:

1) The versions that output all the aforementioned variables are denoted with the number 2 at the end of the filename (e.g., convertprswyo2).
2) The C programs require certain parameter files. One named input_parameters is generated via the wgrb2all.sh script. Two others provide the vertical structure of the height or pressure level and layer output. They are usrhgt_lvls and usrprs_lvls, respectively. These parameter files may be modified by the user for specific applications. The programs (height or pressure-based) will not run if the respective parameter files are missing.

## Appendix B. Bash Script Flowcharts

Appendix B contains flowcharts of the several Bash scripts that include queries to the user and the main processes. They are similar to the related charts in Appendix B in Cogan (2018), but include the recent modifications noted in this report.

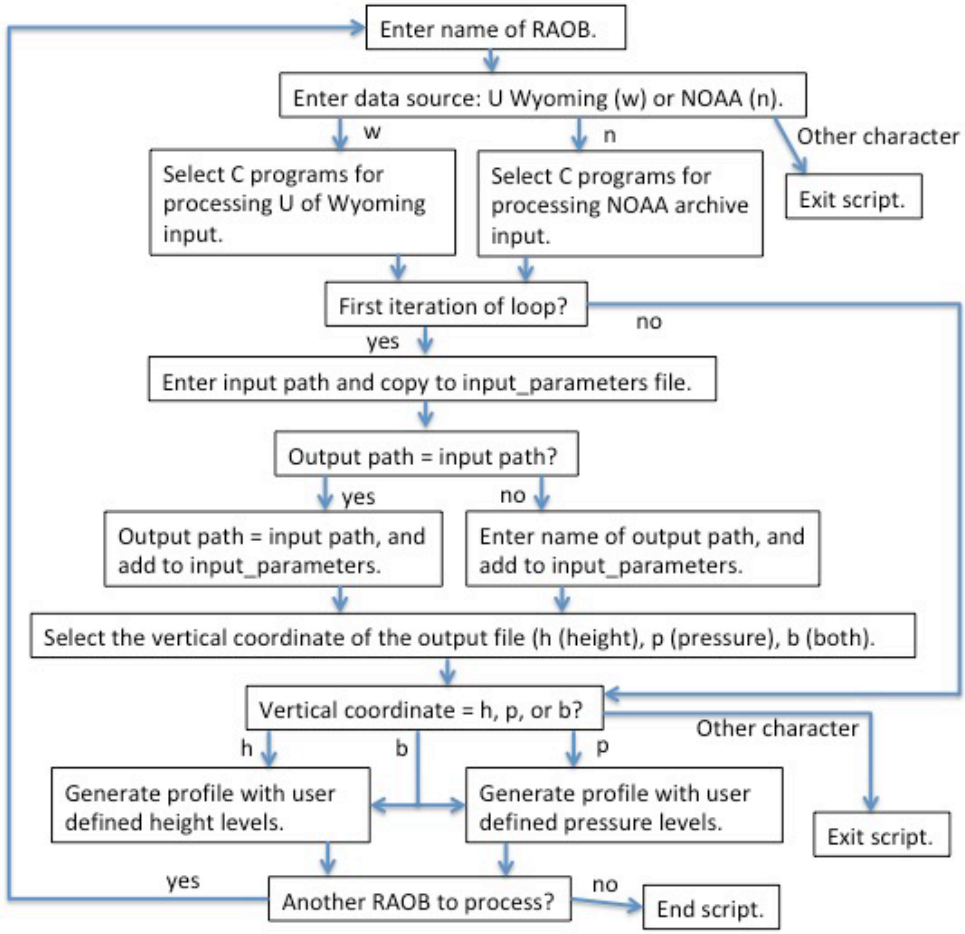


**Figure B-1 Flowchart of *raoball.*sh that is similar to that for *raob.sh* in Appendix B in Cogan (2018), but includes the recent modifications. One primary change is the ability to process more than one radiosonde observation (RAOB) without rerunning the script.**

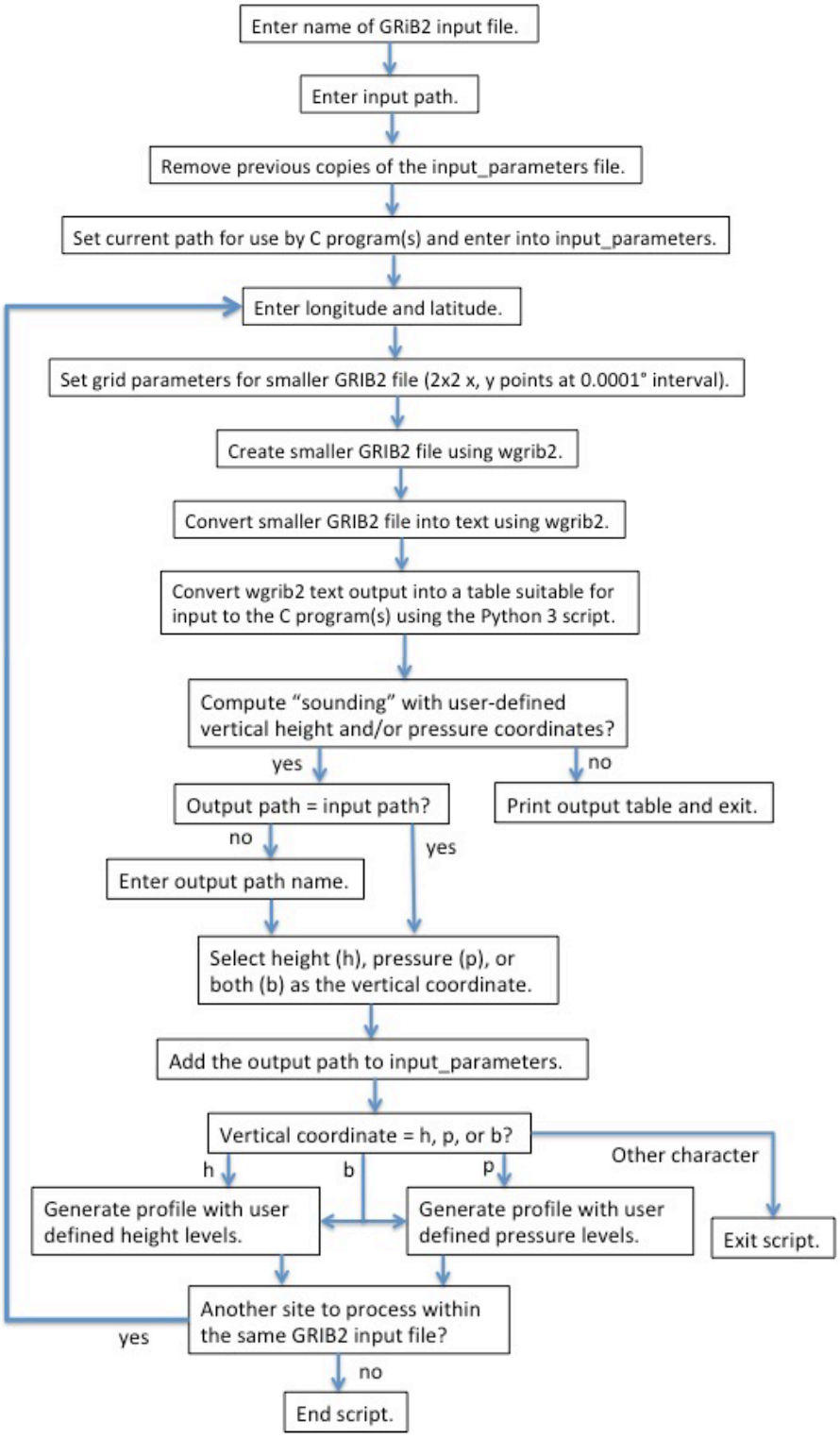


**Figure B-2 Flowchart of *wgrb2all.sh* similar to the one for *wgrb2.sh* in Cogan (2018), but contains the recent modifications. One primary change is the ability to process soundings for more than one site within the domain of the input file without rerunning the script.**

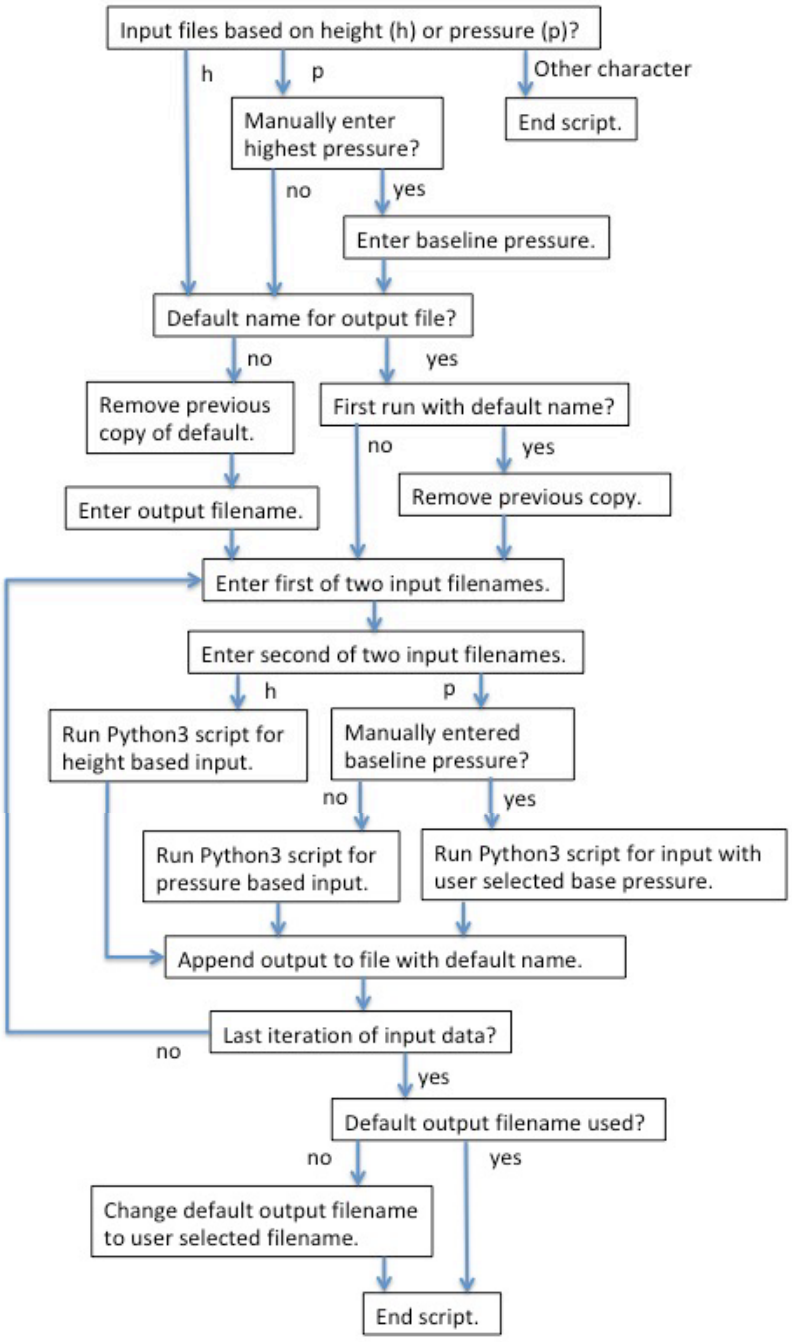


**Figure B-3 Flowchart of *prof_compareall.sh*. It is similar to the one for *prof_compare.sh* in Appendix B in Cogan (2018), but has the recent modifications.**

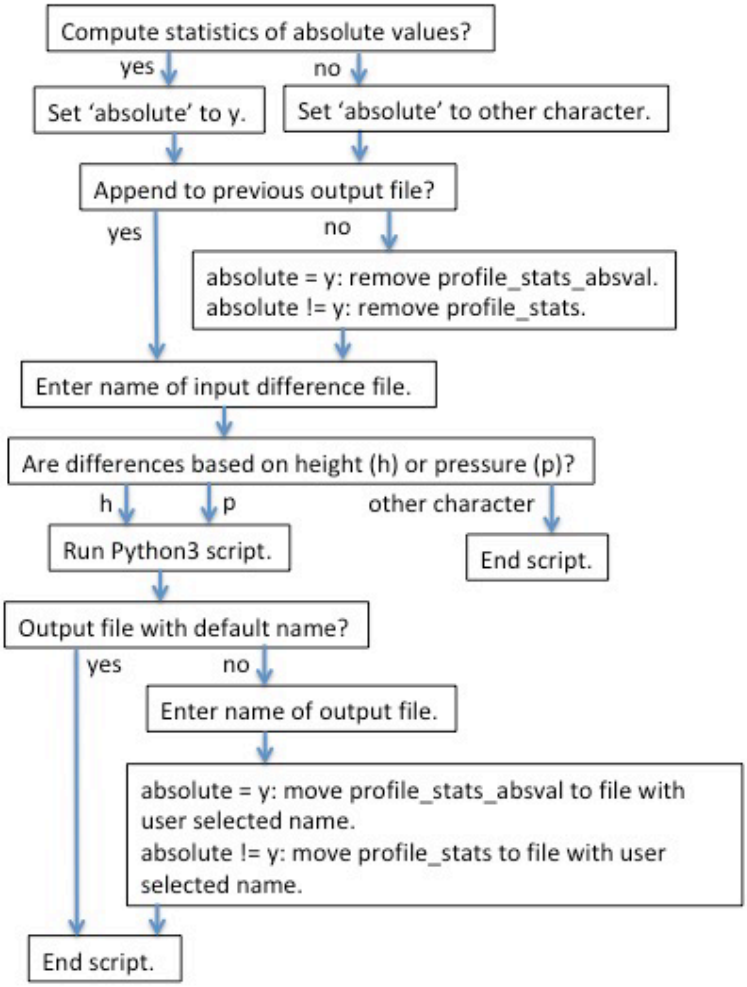


**Figure B-4 Flowchart of *profstatfull.sh*. While it is similar to the respective one for profstat.sh in Appendix B in Cogan (2018), it includes recent modifications.**

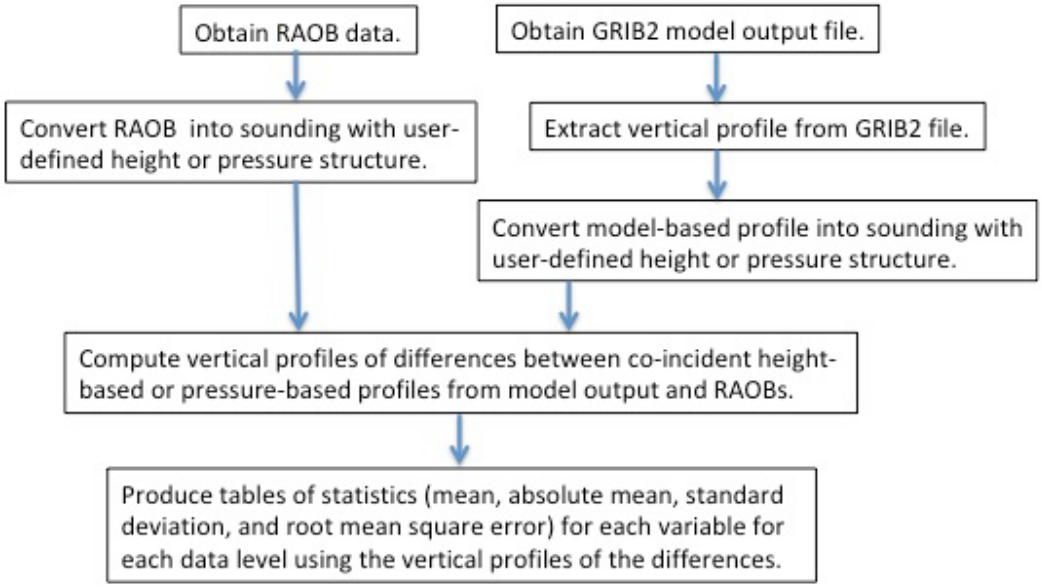


**Figure B-5 Overall sequence of the main processes described in this report. Height- or pressure-based refer to height or pressure as the vertical coordinate, respectively. Details on each step may be found in Figs. B-1 to B-4 and in the main text.**

## List of Symbols, Abbreviations, and Acronyms

| | |
|---|---|
| AFWA | Air Force Weather Agency |
| AGL | above ground level |
| ARL | Army Research Laboratory |
| CCDC | US Army Combat Capabilities Development Command |
| ESRL | Earth System Research Laboratory |
| GALWEM | Global Air Land Weather Exploitation Model |
| GFS | Global Forecast System |
| GSD | Global Systems Division |
| MED | meteorological error determination |
| MET | meteorological |
| METCM | computer meteorological message |
| MSL | mean sea level |
| NCL | NCAR Command Language |
| NCAR | National Center for Atmospheric Research |
| NCEP | National Center for Environmental Prediction |
| NOAA | National Oceanic and Atmospheric Administration |
| RAOB | radiosonde observation |
| RMS | root mean square |
| RMSE | RMS error |
| WMO | World Meteorological Organization |
| WWeb | University of Wyoming's Weather Web |

| | |
|---|---|
| 1 (PDF) | DEFENSE TECHNICAL INFORMATION CTR DTIC OCA |
| 1 (PDF) | CCDC ARL FCDD RLD CL TECH LIB |
| 1 (PDF) | CCDC ARL FCDD RLC E J COGAN |