ARL-TR-8589 ● Dec 2018

US Army Research Laboratory

Extraction and Comparison of Vertical Profiles from Global and Mesoscale Models

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-8589 ● DEC 2018

US Army Research Laboratory

Extraction and Comparison of Vertical Profiles from Global and Mesoscale Models

by J Cogan
Computational and Information Sciences Directorate, ARL

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* December 2018	**2. REPORT TYPE** Technical Report	**3. DATES COVERED (From - To)**

4. TITLE AND SUBTITLE Extraction and Comparison of Vertical Profiles from Global and Mesoscale Models	**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) US Army Research Laboratory ATTN: RDRL-CIE 2800 Powder Mill Road Adelphi, MD 20783-1138	**8. PERFORMING ORGANIZATION REPORT NUMBER** ARL-TR-8589
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

Evaluation of numerical weather prediction models commonly considers the surface and near surface layers where more observations with greater detail are often available. For larger-scale models, publically available comparisons mostly consist of certain standard levels at set time intervals. Comprehensive and sophisticated statistical packages are available that can generate a variety of statistical measures, although they mostly consider the surface, near surface, and a limited number of higher levels, mostly within or near the boundary layer or at standard pressure levels. Evaluation of vertical soundings from model output at user-defined intervals vertically through the atmosphere up or near to the uppermost model output level would allow a more complete evaluation of model performance. An ability to readily compare profiles from two or more sources of data, such as a model and a radiosonde, will assist evaluation over those levels or layers. This report describes a series of methods for extraction of vertical soundings from the output of several representative models, including the Global Forecast System, the Global Air Land Weather Exploitation Model, and the Weather Research and Forecast model, as well as methods to evaluate single or multiple soundings from model output.

15. SUBJECT TERMS

meteorological model evaluation, user-defined profiles, extraction of vertical profiles, sounding statistics, sounding comparisons

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

Contents

List of Figures

List of Tables

1. Introduction

Evaluation of numerical weather prediction (NWP) models on finer scales commonly considers the surface and near surface layers where a larger number of observations are available, often in greater detail. For larger-scale models, such as those run at weather centers, publically available comparisons mostly consider data at certain standard levels at set time intervals. The National Center for Environmental Prediction (NCEP; http://www.emc.ncep.noaa.gov/gmb/STATS_vsdb/) and the European Centre for Medium-Range Weather Forecasting (ECMWF; https://www.ecmwf.int/en/forecasts/charts/catalogue/) have readily accessible verification websites. Worldwide comparisons are available for deterministic forecasts at the ECMWF (http://apps.ecmwf.int/wmolcdnv/) and for ensemble forecasts at the Japan Meteorological Agency (JMA; http://epsv.kishou.go.jp/ EPSv/). Those comparisons generally apply only to surface and standard pressure levels. Comprehensive and sophisticated statistical packages are available that can generate a variety of statistical measures, although those also mostly consider the surface, near surface, and a limited number of higher levels. However, Model Evaluation Tools (MET) users can select a specific vertical level for analysis. For example, the Weather Research and Forecasting model (WRF) Developmental Test Center (DTC) has used the MET and a Unified Post Processing system for analysis of output and statistical evaluation. Similar tools are available at other centers such as NCEP. The DTC websites (http://www.dtcenter.org/met/users/ and http://www.dtcenter.org/upp/users/) contain links to extensive user's guides, other documentation, and tutorials.

NWP model output is frequently compared with radiosonde observations where the entire sounding is often referred to with the acronym for radiosonde observation (RAOB). Ingleby et al. (2018) have up-to-date information on available RAOB data as well as data from other sources such as aircraft. The aforementioned verification websites, which are maintained at major weather centers, mostly compare model output for levels above the surface with data from RAOBs. Many field experiments and tests involve comparisons of output from finer scale (e.g., mesoscale) models such as the WRF model (herein WRF) with RAOB data (e.g., Dutsch 2012; Kilpelainen et al. 2012; Cuevas et al. 2011). Skamarock, et al. (2008) describe details of the WRF, and although there have been upgrades since then, the basics as described remain much the same. Details on available output products from the National Oceanic and Atmospheric Agency (NOAA), and more specifically on the Global Forecast System (GFS) as used at NOAA, may be found at http://www.nco.ncep.noaa.gov/pmb/products/ and http://www.nco.ncep.noaa.gov/pmb/products/gfs/#GFS, respectively, and via included links. The Global Air

Land Weather Exploitation Model (GALWEM) has no similar readily available website. However, GALWEM is a version of the UK Met Office's Unified Model and the Met Office has a public website that includes links that give overviews of the model and its application (https://www.metoffice.gov.uk/ research/modelling-systems/unified-model).

This report presents a set of computer scripts and programs that allow the user to extract vertical profiles of meteorological variables from GFS and GALWEM output in GRIB2 format and convert them into profiles with a height or pressure level and layer structure as defined by the user. A related program also extracts vertical profiles from WRF output and converts them into profiles at user-defined levels and layers. RAOB soundings can also be converted into forms having user-defined height or pressure levels and layers. Those profiles may include wind speed and direction plus virtual temperature or u and v wind components plus relative humidity (RH). Both forms have temperature and pressure (or height) for the respective height (or pressure)-based profiles. An additional set of scripts generates tables of differences between any two profiles with the same user-defined structure and produces basic statistics over multiple tables by variable and by height or pressure level or layer.

2. Extraction and User Definition of Vertical Profiles from GFS, GALWEM, and WRF Output

The procedure for extraction of vertical profiles of meteorological variables ("soundings") from GFS and GALWEM (GRIB2) and WRF (netCDF) output files, as well as the process for generation of vertical profiles at user-defined levels and layers, is described for separate Python 3 scripts and C programs, and for combined sets of scripts and programs using Bash scripts. Sections 2.1–2.4 describe the means to extract vertical profiles from GRIB2 output files (GFS, GALWEM) and Section 2.5 presents methods to largely automate the process through the use of Bash scripts, including one for processing of WRF output files. Appendix A contains a list of the programs and scripts with very brief descriptions of their functions. It also contains attached listings of the Python 3 scripts (files with extension .py) and Bash scripts (extension .sh). The C program source codes (extension .c) are described elsewhere (Cogan 2017).

2.1 Create a Small GRIB2 File from the Input File

The extraction of a sounding (i.e., vertical profiles of meteorological variables from one geographical location for one model time) from a GRIB2 output file employs the wgrib2 program developed at NOAA, which is described in some detail at http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2 and included links. The first step in the process is to create a very small GRIB2 (.grb2) file from the larger global or regional GRIB2 file. As used here, wgrib2 with *–new_grid* in the command line will generate a smaller grid whose grid points are interpolated from the grid points of the parent grid. The values at the horizontal grid points of the smaller (new) grid are computed by bilinear interpolation from the data at the points of the larger grid, unless the user specifies another method (e.g., change to nearest neighbor).

The command line for GFS and standard 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(LON) LAT:Y DIRECTION POINTS:DY(LAT) SMALLER_GRIB2
```

where INPUT_GRIB2 is the input GRIB2 file, SMALLER_GRIB2 is the smaller output GRIB2 file, LON and LAT are longitude and latitude in decimal degrees specifying the lower-left corner of the output grid, X and Y DIRECTION POINTS refer to the number of grid points in the x- and y-directions in the output grid, and DX and DY refer to the distance between grid points in the output grid in the respective directions in units of longitude and latitude. The distance between grid points can be very small (e.g., 0.0001°).

In the command line, "same" for *–set-grib_type* results in a GRIB2 output file with the same GRIB packing as the input file and "earth" for *–new_grid_winds* leads to winds relative to the Earth versus to the grid or undefined. Use of "latlon" for *–new_grid* results in a new grid interpolated from the parent (old) grid, where the listed latitude and longitude are those for the new grid's lower-left (approximately southwest) corner.

The following is an example using a GFS input file for a location near the US East Coast:

```
wgrib2 gfs_4_20170404_000_012.grb2 –set_grib_type same –new_grid_winds earth –new_grid
latlon -76.12:2:0.0001 39.10:2:0.0001 small_file.grb2
```

As is apparent in the example, the file small_file.grb2 has a two-by-two horizontal grid 0.0001° apart, which translates to a separation on the order of 10 m (for latitude ~11.2 m and for longitude ~11.2 m within a few degrees of the equator decreasing to ~5.6 m near 60°). Information on these and many other arguments used for wgrib2 may be found via http://www.cpc.ncep.noaa.gov/products/wesley/wgrib2/long_cmd_list.html.

However, this wgrib2 command will not work for GALWEM files of the type used for input to the artillery meteorological system, the Profiler Virtual Module (PVM). Those GALWEM files have a different format with respect to the horizontal wind components. Instead of the u and v data for a given data level being contiguous, all the u and all the v data lines are in separate sections. As a consequence, additional code is required to manipulate the input file and sort the u and v component values so that they are in the proper order for processing by a similar wgrib2 command. This more complicated procedure uses the Unix/Linux sed command along with a Linux sort process. Details and examples may be found at http://www.cpc.noaa.gov/products/wesley/wgrib2/new_grid.html. Information on sed may be found at https://www.gnu.org/software/sed/manual/sed.html#Command_002dLine-Options. The wgrib2 and other commands are as follows:

```
wgrib2 GALWEM_PVM_INPUT |sed -e 's/:UGRD:/:UGRDa:/' -e 's/:VGRD:/:VGRDb:/' | \
sort -t: -k3,3 -k5,8 -k4,4 | \
wgrib2 GALWEM_PVM_INPUT -i -new_grid_winds earth -new_grid latlon LON:X DIRECTION
            POINTS:DX(LON)  LAT:Y DIRECTION POINTS:DY(LAT)  SMALLER_GRIB2
```

where many of the parameters for the wgrib2 commands are the same as before. UGRD and VGRD (UGRDa and VGRDb) are the identifiers for the u and v wind components in the input (modified input) file. GALWEM_PVM_INPUT is a generic name for a GALWEM output file that is compatible with the PVM. The sort command is standard in Linux. The parameter –i means read STDIN to get the inventory (output from previous commands).

The following is an example for a location near the center of the US East Coast using a GALWEM file downloaded from the relevant website for input to the artillery meteorological system:

```
wgrib2 GAL_test_NA_PVM.gr2 |sed -e 's/:UGRD:/:UGRDa:/' -e 's/:VGRD:/:VGRDb:/' | \
sort -t: -k3,3 -k5,8 -k4,4 | \
wgrib2 GAL_test_NA_MET.gr2 -i -new_grid_winds earth -new_grid latlon -77.3:2:0.0001
   38.55:2:0.0001 small_test.grb2
```

This procedure produces a very small grid with the dimensions noted previously for the unmodified GRIB2 files (GFS and GALWEM), that is, 0.0001° or a distance on the order of 10 m between grid points.

2.2 Extract Vertical Profile Data

Next, we extract a sounding from the small GRIB2 file. This process applies to both GFS and GALWEM, and is accomplished using another wgrib2 procedure:

```
wgrib2 SMALLER_GRIB2 –v –s –lon LON LAT,
```

where SMALLER_GRIB2 is the output from the procedure of the previous section, *–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 (LON) and latitude (LAT). With a grid separation of approximately 10 m, the values are essentially for 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 as shown in the example:

```
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. Table 1, which shows a few lines out of a total of over 400, indicates the complexity of the output. Each line of output begins with the line number. Note that each line is numbered followed by information on date and time (d=2018022812 in yyyymmddhh format, where yyyy is the four-digit year, and mm, dd, and hh are the two-digit month, day, and hour, respectively). The date and time field is followed by the variable label, variable name, units, and so on. Longitude (lon) is in degrees east and is always a positive number, from 0 to 360 (e.g., 90W or –90 is expressed as 270). The number (after val=) at the end is the value of the variable.

Table 1 Sample of output from the wgrib2 process for extraction of a sounding from a GRIB2 file. Only 9 lines of a total of greater than 400 lines are shown. Many lines have variables that are not used for the user-defined meteorological soundings (e.g., icing severity).

```
141:29276:d=2018022812:HGT Geopotential Height [gpm]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=6348.67
142:29486:d=2018022812:TMP Temperature [K]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=247.584
143:29696:d=2018022812:RH Relative Humidity [%]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=26.6564
144:29906:d=2018022812:VVEL Vertical Velocity (Pressure) [Pa/s]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=0.0119045
145:30116:d=2018022812:UGRD U-Component of Wind [m/s]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=30.6803
146:30326:d=2018022812:VGRD V-Component of Wind [m/s]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=-6.00326
147:30536:d=2018022812:ABSV Absolute Vorticity [1/s]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=0.000162359
148:30746:d=2018022812:CLWMR Cloud Mixing Ratio [kg/kg]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=0
149:30956:d=2018022812:ICSEV Icing severity [non-dim]:450 mb:15 hour
fcst::lon=283.880000,lat=39.970000,i=1,ix=1,iy=1,val=0
```

2.3 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 meteorological message (METCM). A Python 3 program (often called a Python 3 script) converts the output into a readily useable and readable form. It reads in the aforementioned output file, extracts appropriate information, and lists the extracted information in a readable tabular form.

To run the script for the wgrib2 output from either a GFS or a GALWEM file, type the program and input filenames:

```
python3 gg_wg2.py INPUT_FILE,
```

where INPUT_FILE is a generic name for the file created by wgrib2 (Section 2.2) and gg in the program name refers to ability to process either GFS or GALWEM derived output, for example,

```
python3 gg_wg2.py GAL_test,
```

which produces the output file GAL_test_out. Table 2 shows part of the output file GAL_test_out including header information.

Table 2 Sample of output from the script gg_wg2.py. The header and the first 11 data lines are shown. P is pressure, Hgt is height above mean sea level (MSL), Tmp is temperature, RH is relative humidity, and U and V are the horizontal components of the wind (relative to Earth). The units are as indicated.

```
6 hour forecast after model start at: 2018032800
Latitude:  37.000   Longitude:  -77.000

P (hPa)    Hgt (m) Tmp (K)    RH (%)  U (m/s) V (m/s)
 1019.9      25.0   277.00     94.6    1.00    2.20
 1013.0      81.0   277.32     93.5    2.20    3.90
 1000.0     186.0   277.46     89.2    4.50    6.10
  975.0     392.0   278.53     68.5    8.90    6.40
  950.0     605.0   279.32     73.7   10.70    4.10
  925.0     825.0   282.17     68.9   11.80    1.60
  900.0    1053.0   284.84     61.6   12.30   -0.60
  875.0    1289.0   283.63     68.1   12.60   -2.40
  850.0    1530.0   282.19     78.4   11.80   -3.40
  825.0    1776.0   280.58     87.3   10.40   -2.80
  800.0    2029.0   278.98     84.7    9.60   -1.80
```

Note that GFS output files from NOAA starting on 12 Coordinated Universal Time (UTC) 11 May 2016 extend up to 1.0 hectoPascal (hPa), but before then the data lines ended at 10 hPa. The aforementioned methods work for both. Also, data lines in recent GALWEM files reach up to over a 50-km altitude. Pressure in the

GALWEM output files is to the nearest hPa so that the value at the highest level for, say, 53 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. Based on the US standard atmosphere (1976) (https://www. engineeringtoolbox.com/standard-atmosphere-d_604.html), the pressure at 50 km is about 0.80 hPa and at 60 km is about 0.22 hPa. Using the standard hypsometric formula to compute the pressure at the highest output level (~53.1–53.2 km) for three soundings led to pressures of 0.5±0.1 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.

2.4 Compute Sounding with User-Defined Height or Pressure Levels and Layers

ARL_MET-profile_Converter programs (https://github.com/usarmyresearchlab), or similar ones available on request, may be used to produce a table of user-defined levels or layers from the output sounding of Section 2.3. My previous work (Cogan 2015, 2017) and included references describe these C programs and their application, and present samples of output. One version produces output for height levels and layers above ground level, and a second provides output for pressure levels and layers. The user provides a text file that contains the height or pressure levels, respectively, which also serve as the vertical boundaries of the included layers. A second set of programs was added later to output profiles of horizontal wind components (u, v) instead of wind speed and direction. The second set replaces the profile of virtual temperature with a profile of RH. Both sets of programs are run in the same way.

The complete input and output directories are defined in the input_parameters file, which is in the same directory as the C program executable. The complete path starts from the base directory (e.g., /data/programs/input/) versus the relative path (e.g., input/). The / mark at the end of the path name is required. For height-based output, the file usrhgt_lvls contains the heights, which are also the bounding levels for layer computations, 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 (listed as line 0) in the layer output has values for the surface only. Consecutive lines have weighted mean layer values. For the pressure-based output, the parameter file is usrprs_lvls, also 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. Both GFS- and GALWEM-derived soundings are processed using the same executable (convertgfs) since the format of the input files is the same.

The program is run using the following command line for height based output:

```
./convertgfs INPUT_FILE
```

and for pressure based output:

```
./convertprsgfs INPUT_FILE
```

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

```
./convertgfs ETGB_2017081506_out
```

Note that some operating systems may not use the "./" before the executable name.

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

For example,

```
./convertprsgfs ETGB_2017081506_out
```

produces the level and layer output files ETGB_2017081506_out_USRLVL_P and ETGB_2017081506_out_USRMSG_P, respectively.

For output with columns of u and v wind components and RH use programs having the same names, but with "1" appended to the end. For example, a sounding derived from GFS output could use the following:

```
./convertprsgfs1 ETCB_2917081506_out
```

The output files have the number 1 appended to the end of the filenames. For example ETGB_2017081506_out_USRLVL_P1 and ETGB_2017081506_out_USRMSG_P1 for the level and layer pressure-based output, respectively. For height-based output, the files would end in USRLVL1 and USRMSG1. Tables 3–5 present examples of an extracted sounding, a user-defined sounding based on height with wind speed and direction, and a similar sounding based on pressure with u and v wind components.

Table 3 Vertical profile (sounding) extracted from a GFS output file (0.25° horizontal grid) for the 12-h forecast starting at 00 UTC on 17 August 2018 for the National Weather Service site at Dulles Airport, Virginia. The first 10 data lines are shown. The variables and units are as in Table 2.

```
12 hour  forecast after model start at: 2018081700
Latitude:  38.980   Longitude:  -77.460

 P (hPa)  Hgt (m) Tmp (K)   RH (%)  U (m/s) V (m/s)
 1007.1     97.1  297.62    77.8    0.25    1.01
 1000.0    158.6  296.79    76.7    0.41    1.50
  975.0    380.0  296.20    74.7    2.16    2.30
  950.0    608.0  298.23    59.8    5.01    3.55
  925.0    842.2  297.38    56.5    7.16    3.83
  900.0   1081.7  295.78    56.0    7.74    3.60
  850.0   1576.4  291.87    59.4    7.44    2.69
  800.0   2093.4  287.64    64.3    7.03    1.75
  750.0   2635.4  283.74    67.6    7.98    2.36
  700.0   3207.2  280.33    65.7    9.67    3.31
```

Table 4 The sounding of Table 3 converted to a user-defined height structure. Virtual temperature is computed for all levels, and then all variables are interpolated to the listed heights. Wind direction and wind speed (Wind_Dir and Wind_Speed, respectively) and virtual temperature (Virt_Temp) are displayed. The number shown for the date has the date and time of the "0-h" forecast of the model and the time has the model forecast hour (i.e., 12). Elevation is in m, Time in h, and Latitude and Longitude are in degrees. The ceiling and visibility in the header are for potential future use and the values shown indicate missing data. At this time, wind speed is in knots. The first nine data lines (0–8) are shown.

```
USER DEFINED LEVEL OUTPUT (MODEL)

Date: 2018081700   Time: 12  Latitude: 38.980   Longitude: -77.460
Elevation:   97.10     Ceiling:  -999.0     Visibility:  -999.0

Level   Height  Wind_Dir  Wind_Speed  Virt_Temp  Pressure  Temperature
          (m)   (degrees)    (kn)        (K)       (hPa)       (K)

  0       0.0    193.9       2.02      300.33    1007.10     297.62
  1     200.0    217.0       4.87      298.91     984.30     296.42
  2     500.0    234.4      11.65      300.45     951.15     298.13
  3    1000.0    245.2      16.55      297.60     898.41     295.66
  4    1500.0    250.3      15.32      293.38     847.95     291.70
  5    2000.0    256.0      14.10      289.07     799.65     287.61
  6    2500.0    253.7      16.03      285.28     753.46     284.02
  7    3000.0    251.5      19.15      282.05     709.40     280.99
  8    3500.0    251.4      21.56      279.00     667.49     278.04
```

Table 5 The sounding of Table 3 converted to a user-defined structure of pressure layers. The values of the variables shown are weighted mean values for the listed layers as indicated by the pressure midpoints except for the surface (level 0), which has the surface values. The u and v wind components (U-wind and V-wind) and relative humidity (Rel-Humidity) are displayed. The pressure (P) value shown is for the midpoint pressure of the layer (average of boundary values) and the height value is the midpoint height (Midpt) of the layer, except for "layer" 0, which has the surface values. The level and layer values of the variables other than P are computed relative to ln (P). Also, the pressures at the height midpoints may be somewhat different from the midpoint pressures displayed since pressure is not linear with height. The first 16 data lines (0–15) are shown.

```
USER DEFINED PRESSURE LAYER OUTPUT (MODEL)

Date: 2018081700  Time: 12   Latitude: 38.980   Longitude: -77.460
Elevation:   97.10     Ceiling:  -999.0     Visibility:  -999.0

Layer Pres-Midpt  Hgt-Midpt  U-wind  V-wind  Rel-Humidity Temperature
        (hPa)        (m)     (m/s)   (m/s)        (%)          (K)

  0     1007.1         0.0     0.25    1.01      77.80       297.62
  1     1003.5        30.8     0.33    1.25      77.25       297.21
  2      995.0       105.4     0.76    1.66      76.30       296.67
  3      982.5       216.1     1.63    2.06      75.30       296.38
  4      962.5       396.9     3.59    2.92      67.25       297.22
  5      937.5       628.0     6.09    3.69      58.15       297.80
  6      912.5       864.8     7.45    3.72      56.25       296.58
  7      887.5      1106.5     7.67    3.38      56.84       294.82
  8      862.5      1353.9     7.52    2.92      58.54       292.86
  9      837.5      1606.6     7.34    2.46      60.61       290.83
 10      812.5      1865.1     7.13    1.99      63.06       288.71
 11      787.5      2129.6     7.26    1.90      65.11       286.68
 12      762.5      2400.6     7.74    2.21      66.76       284.73
 13      737.5      2678.8     8.40    2.59      67.13       282.90
 14      712.5      2964.7     9.24    3.07      66.18       281.20
 15      687.5      3258.9     9.99    3.40      67.17       279.46
```

2.5 Combining Python 3 Scripts and C programs using Bash Scripts

The several scripts and programs may be combined via Bash scripts that also allow the user to enter needed parameters such as filenames and whether the output is to go to a separate file or is appended to a file for later processing. Appendix B contains flowcharts for the several Bash scripts as well as a generalized chart for the complete set of Bash scripts when used as a single package from extraction of soundings to generation of tables of basic statistics.

2.5.1 Processing GFS or GALWEM Output Files

A Bash script combines the Python 3 and C programs for processing GFS or GALWEM output, and allows the user to optionally run the complete set of programs or skip the last part that generates soundings with user-defined structures and only save the sounding extracted from the GFS or GALWEM output (e.g., Table 3). In the current version, the user is prompted to input several input parameters such as the input filename and location's latitude and longitude.

The script for processing GRIB2 output is run by entering its name:

```
./wgrb2.sh
```

The first item entered is the name of the input file (GFS or GALWEM), the second is whether the input is the artillery meteorological system compatible GALWEM file, and the third is the name of the output file without extensions. The output filenames follow the same convention as for Section 2.3, that is, "_out" is added to the user-defined filename. The soundings produced with a user-defined height or pressure structure have filenames that follow the convention of Section 2.4, that is, extended with, for example, "_USRLVL", "USRMSG_P, "USRLVL_P1", and so on. Other items requested of the user include the bounding latitudes and longitudes, which, as noted previously, can have a spacing of 0.0001° or about 10 m, whether output should include wind speed and direction or u and v wind components, and the input and output directories (can be the same) for the external C programs for generation of profiles for user-specified height or pressure levels and layers. The user can choose default values for some of the other parameters or they may be entered as appropriate. The location latitude and longitude are in decimal degrees (e.g., 45.26) and have to be within the bounds $-90° <$ latitude $<90°$ and $-180° \leq$ longitude $\leq 180°$, respectively. The user has the options of skipping the computation of user-defined profiles and saving only the extracted profile as noted previously, or generating them based on height, pressure, or both height and pressure in separate files.

2.5.2 Processing WRF Output Files

A separate Bash script combines a National Center for Atmospheric Research Common Language (NCL) script (Reen 2017) and the C programs for processing WRF output, and allows the user to optionally run the complete set of programs or skip the generation of user-defined vertical profiles and only save the extracted sounding. The data lines in the extracted sounding are at model output pressure levels. The user is prompted to input several input parameters such as the input filename and location's latitude and longitude. Other parameters needed for the NCL program include a string containing the model date and time of the output file in a specified format (i.e., yyyy-mm-dd_hh:mm:ss, where, for example, a specific date and time could be 2018-08-25_12:00:00).

The script is run by entering its name:

```
./wrf_extract.sh
```

The user is prompted for several options not provided via the other Bash scripts. They include, for example, whether to use latitude and longitude or to use x and y grid indices. Normally latitude and longitude are chosen. Another option is the method for interpolating, or not, between grid points for the specified location. The methods include nearest grid point (no interpolation), inverse distance weighting, and bilinear. For consistency with the wgrib2 default method, select bilinear. The included C programs generate soundings at user-defined height or pressure levels and layers defined by those levels. Parameters for the C program include the complete input and output paths for the input and output files, respectively, and the user-defined list of heights or pressure levels. The user is prompted for the paths, but the user-defined heights or pressures are in a parameter file as noted in Section 2.4 (e.g., "usrhgt_lvls"). A complete path starts from the base directory. For example, a user running a program from the program_files directory would enter /data/user/program_files/input/, not the relative path input/. As with soundings extracted from a GRIB2 output file, the filenames follow the same convention as for Section 2.3, that is, "_out" is added to the user-defined filename. The same holds for soundings produced with a user-defined height or pressure structure, that is, they have filename extensions that follow the convention of Section 2.4 (e.g., "_USRMSG1", "USRMSG_P, "USRMSG_P1").

2.5.3 Processing RAOB Data

The Bash script raob.sh produces a sounding for user-defined height or pressure levels and layers from RAOB data using the included C programs. The output soundings may contain either wind speed, wind direction, and virtual temperature (T_v) or u and v wind components and RH.

The script is run by entering its name:

```
./raob.sh
```

The script prompts the user for input information such as input filename; source of the data, which at this time is either the U of Wyoming weather website or the NOAA archive website (http://weather.uwyo.edu/upperair/sounding.html or https://ruc.noaa.gov/raobs/, respectively); whether output should have wind speed and direction (and T_v) or u and v wind components (and RH); and the full input and output paths for the C program input and output files. The format of the output soundings and the naming convention are the same as presented in Section 2.4.

3. Statistics of Differences Between Vertical Profiles

The following Bash scripts run Python 3 scripts that compute 1) the differences by variable and data line (i.e., height or pressure level or layer) between user-defined vertical profiles from any two sources and calculate basic statistics by variable over all data lines, and 2) statistics by variable and data line over multiple tables of the differences. An example could be the root-mean-square error (RMSE) (or root-mean-square difference) of temperature at 1500 m (or 850 hPa) over, say, 20 tables of differences, where each table is the result of the respective comparison between two soundings. For this example, the sample size would be 20 for temperature at 1500 m (or 850 hPa). Flowcharts for the following Bash scripts are found in Appendix B along with the Bash scripts of Section 2.5 plus the chart for the entire package from extraction of soundings to generation of statistics.

3.1 Computation of Differences

The Bash script prof_compare.sh computes the differences between two user-defined soundings that were generated using wgrb2.sh, wrf_extract.sh, or raob.sh. The height or pressure structure for both must be based on the same usrhgt_lvls or usrprs_lvls parameter file (see Section 2.4). The script asks for the input filenames; a character that indicates whether the input files are height or pressure based; another character for whether 1) wind speed and direction and T_v or 2) u and v wind components and RH should be in the output file; and if the output is to be written as a single file for each comparison or appended to a file that may contain multiple comparisons. For pressure-based input soundings, the script also asks if the user wants to manually enter a baseline pressure that normally is a user-input pressure from the usrprs_lvls list of pressure levels that equals or is less than the highest pressure (lowest pressure level) that is expected to be encountered in the input files. If yes, then the pressure in hectopascals is entered. If the user-entered value is higher than the highest common pressure for both input soundings, then the highest common value is selected. If no, the script uses the highest pressure common to both soundings. To run the program, enter the name of the script:

```
./prof_compare.sh
```

The output filename for a single file is currently Diff:INPUT_FILE1-INPUT_FILE2 indicating that values from the second input file are subtracted from the respective ones from the first file. An example could be Diff:WRF1_ 2018052512_USRMSG-R_2018052512_USRMSG for an output “sounding” of layer values from a WRF integration compared with a coincident RAOB. The output file includes the mean, median, standard deviation (SD), and RMSE of the values for each variable over all heights or pressures (i.e., over the entire column). For this report, the term "error" in RMSE refers to the difference between respective values from the two input soundings. If appended, the output goes to a file named diff_profiles (default name).

The included Python 3 scripts may be run separately, that is, prof_compare_height.py, prof_compare_baseprs.py, or prof_compare_pressure.py. The script for height-based input is run by entering the script name followed by the two input filenames and two characters, plus an optional character:

```
python3 prof_compare_height.py   INPUT_FILE1 INPUT_FILE2 Y W -v
```

where INPUT_FILE1 and INPUT_FILE2 are the input files from which the differences are computed, Y is a character that tells the program to append (a) the output or write (w) to its own file, and W is a character that directs the script to compute and output differences in vector wind speed (v) instead of relative humidity (any other character) from input soundings with horizontal (u, v) wind components and RH. If the input soundings have wind speed and direction (and T_v) the script ignores W without producing an error message. The character for W may be omitted, but in that case the script defaults to keeping RH and not computing vector wind speed differences. The two files must have the same height intervals (used the same usrhgt_lvls file noted in Section 2.4 to set up the height structure). The –v is optional and represents the argument –verbosity that if "true" means something will be displayed on the screen, in this case, the command-line input (e.g., the filenames). For information on usage of the script, type –h (--help) after the script name (thereafter the script will end). Details and some examples of the use of –v and –h may be found at, for example, https://docs.python.org/3/howto/rgparse.html#id1 and included links. The following is an example of use of the script:

```
python3 prof_compare_height.py WRF1_2018052512_USRLVL   R_2018052512_USRLVL w n -v
```

where each difference value is from the WRF-derived, height-based sounding less that from the RAOB sounding, the output goes to a separate file named Diff: WRF1_2018052512_USRLVL-R_2018052512_USRLVL, and since the input files have wind speed and direction, the character with respect to vector wind speed is ignored. If the output is appended (Y = a), it goes to the file diff_profiles. If a character other than an "a" or "w" is entered, an error message is printed and the script exits. In this example, –v is used and the names of the input files and the characters w (write to a single file) and n (do not replace RH differences) 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 represents the lowest user-defined pressure level (highest user-defined pressure) or let the program select the lowest pressure level (highest pressure). The procedure for running the script with user input of the highest pressure is as follows:

```
python3 prof_compare_baseprs.py INPUT_FILE1 INPUT_FILE2 PPP Y W -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 hectopascals (or mb), Y is a character that tells the script to append (a) the output or write (w) to its own file, and W is a character to direct the script to compute and output vector wind speed differences (v) instead of RH differences (any other character) as with prof_compare_height.py when the input files contain the horizontal wind components. The argument –v is optional, and if used additional information is printed to the screen, as noted previously. The value of PPP normally should be the lowest user chosen pressure level (highest user-chosen pressure) in the file usrprs_lvls (Section 2.4) that is common to both input soundings for a single comparison, or all input soundings for two or more appended comparisons. As before, if the input soundings have wind speed and direction and T_v, the script ignores W. The parameter W does not determine whether or not the output has differences in the horizontal wind components or wind speed and direction; that is determined by the type of input.

The following is an example:

```
python3 prof_compare_baseprs.py WRF1_2018052512_USRLVL_P R_2018052512_USRLVL_P 925 w n
```

where the input files beginning with WRF1 and R (RAOB) files are based on pressure levels, 925 is the highest common user-defined pressure in hectopascals selected from the file usrprs_lvls, w indicates the output is written to a single file and not appended, and although n means do not compute and output vector wind speed differences it is ignored since the input files have wind speed and direction. 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_compare_pressure.py INPUT_FILE1 INPUT_FILE2 Y W -v
```

where, as before, INPUT_FILE1 and INPUT_FILE2 are the input files from which the differences are computed, Y is a character that tells the program to append (a) the output or write (w) to its own file, W indicates whether or not to compute and output vector wind speed differences, and –v is the optional verbosity argument, as noted previously. An example is as follows:

```
python3 prof_compare_pressure.py WRF1_2018052512_USRMSG1 R_2018052512_USRMSG1 a v -v
```

where both input files have height based layers, the output is appended to diff_profiles, and since the input soundings have wind components the last parameter (v) means that vector wind speed differences replace RH differences in the output. Here –v is used and the names of the input files and the parameters a (append) and v (compute and output vector wind speed differences) are printed to screen with an explanation of their meaning.

As noted in Section 2.4, the output from the C programs may have wind components and RH instead of wind speed, wind direction, and T_v. However, both the height- and pressure-based programs handle either type of input file. The Bash script prof_compare.sh directs the input to the correct version of the Python 3 script for height or pressure depending on user responses to the several requests for information. To avoid appending difference tables from both height- and pressure-based versions, or wind components and wind speed and direction output, the user should copy the “final” set of difference tables to a file with a different name (e.g., copy diff_profiles to diff_profiles_hgt-values). The Bash script will ask whether or not the user is starting a new set. If yes, then the current append file is deleted if it has the same name. Table 6 presents a sample difference table computed from files based on pressure levels along with the statistics for each column of differences. Note that if the parameter W was set to v, the column with Rel_Humidity (RH difference) would be replaced by a column with Vector_W-spd (vector wind speed difference). Table 7 shows an example with Vector_W-spd output.

Table 6 A difference table based on pressure level input files, plus basic statistics, for the input files listed in the first header line. The differences were between soundings based on pressure levels that contained wind components and RH. The first 20 and last 5 data lines are shown, plus the column statistics.

```
Compared output from fintest/BNA/BNA_GFS_2018-05-08_12_out_USRLVL_P1 and
fintest/BNA/R_BNA_2018050812_USRLVL_P1
Difference values for listed variables in diff_profiles
Number of data levels   68
 Level   Pressure  Height  U-wind  V-wind  Rel-Humidity   Temperature Prs(not_diff)
         (hPa)      (m)     (m/s)     (m/s)       (%)          (K)         (hPa)
    0     0.0      20.1      2.57     -0.96      9.26       -1.13       950.00
    1     0.0      19.4      0.94     -1.60     -1.00       -0.73       925.00
    2     0.0      18.6      0.72      0.00     10.46       -0.98       900.00
    3     0.0      18.5     -0.80      0.45      0.23       -0.40       875.00
    4     0.0      18.1     -2.45      0.08    -12.20        0.27       850.00
    5     0.0      18.2     -2.03     -0.24    -27.74        1.14       825.00
    6     0.0      19.3     -0.51     -1.02      1.38        0.74       800.00
    7     0.0      18.7     -0.70     -0.24      9.94       -0.14       775.00
    8     0.0      18.8     -0.99      0.30      4.80       -0.26       750.00
    9     0.0      18.6      0.00     -1.19     -1.38        0.00       725.00
   10     0.0      19.3      1.42     -2.06      0.60        0.24       700.00
   11     0.0      19.0      1.01     -1.10     -1.01       -0.11       675.00
   12     0.0      22.0      0.71     -0.12     -3.81       -0.04       650.00
   13     0.0      20.9      0.05      1.56    -12.65        0.66       625.00
   14     0.0      22.6      0.00      2.55    -10.49        0.39       600.00
   15     0.0      23.1      1.78      2.40      5.05       -0.34       575.00
   16     0.0      26.3      2.94      3.12      1.67       -0.28       550.00
   17     0.0      24.8      0.40      2.43      1.65       -0.33       525.00
   18     0.0      23.2     -1.36      2.32      1.20       -0.36       500.00
   19     0.0      21.1      3.06      1.21      1.76       -0.46       475.00
.
.
.
.

   63     0.0       6.7      6.92      1.79      0.30       -0.86        30.00
   64     0.0       9.2      3.44     -3.22      0.03        0.38        25.00
   65     0.0      -0.1     -0.85      6.60     -0.30       -1.26        20.00
   66     0.0       5.1      0.45     -0.37     -0.47        0.65        15.00
   67     0.0      -9.3      0.45      3.18     -0.70       -0.36        10.00

Column Statistics
Mean      0.0      16.0     -0.07      0.45     11.34       -0.01
Median    0.0      18.3      0.03      0.64      1.79       -0.04
Std Dev   0.0       6.8      2.49      2.21     18.86        0.90
RMSE      0.0      17.3      2.47      2.24     21.89        0.89
```

Table 7 A difference table based on height layer input files, plus basic statistics, for the input files listed in the first header line. The differences were between soundings from Tucson (TUS), Arizona, based on height levels that contained wind components and RH, and where the column for RH differences was replaced with one for vector wind speed differences (Vector_W-spd). A WRF derived sounding was compared with that from the coincident RAOB. The first 12 and last 3 data lines are shown, plus the column statistics.

```
Compared output from fintest/TUS/TUS_WRF_2016042900_0km_USRMSG1 and
fintest/TUS/R_TUS_2016042900_USRMSG1
Difference values for listed variables in diff_profiles
Number of data levels   26
Layer   Height   Pressure  U-wind   V-wind  Vector_W-spd Temperature Hgt(not_diff)
         (m)       (hPa)     (m/s)     (m/s)      (m/s)        (K)          (m)
   0     0.0        3.0      5.33      0.85       5.40      -0.66         0.00
   1     0.0        2.7      2.55      2.37       3.48      -1.03       100.00
   2     0.0        2.6      1.44      2.83       3.18      -0.65       350.00
   3     0.0        2.5      2.08      1.48       2.55      -0.82       750.00
   4     0.0        2.2      1.84     -1.67       2.48      -0.90      1250.00
   5     0.0        1.9      2.27     -2.07       3.07      -1.22      1750.00
   6     0.0        1.4      1.64      0.30       1.67      -1.05      2250.00
   7     0.0        1.1      0.22     -1.30       1.32      -0.31      2750.00
   8     0.0        0.9      3.58     -1.87       4.04       0.37      3250.00
   9     0.0        1.0      7.20     -1.63       7.38       1.34      3750.00
  10     0.0        1.1      7.53     -0.85       7.58       2.12      4250.00
  11     0.0        1.3      8.64     -1.39       8.75       2.50      4750.00
.
.
.
.
  23     0.0        0.4      1.44     -4.74       4.95      -0.72     16500.00
  24     0.0        0.2      4.44     -3.06       5.39      -1.89     17500.00
  25     0.0        0.0     -3.00     -2.70       4.04      -1.27     18500.00

Column Statistics
Mean     0.0        1.38     2.47     -1.95       5.23      -0.02
Median   0.0        1.34     2.17     -1.65       5.17      -0.48
Std Dev  0.0        0.84     3.39      3.66       2.60       1.40
RMSE     0.0        1.60     4.14      4.08       5.81       1.37
```

The 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 as follows:

```
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:". Table 8 shows an example for TUS on 29 April 2016 at 00 UTC.

Table 8 Sample screen display from duplichk.py. The first line is the query to the user and the entered name of the file to be checked. In this example one file was duplicated one time. If the same file is duplicated more than once, then the line after "Table ID" will be repeated (e.g., if duplicated three times, the same line is repeated three times).

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

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

 Difference tables duplicated
      Table ID
/data/pyfiles/fintest/TUS/TUS_WRF_2016042900_100N_USRLVL-
/data/pyfiles/fintest/TUS/R_TUS_2016042900_USRLVL

Number of duplicates:  1

END OF DUPLICATION CHECK
```

3.2 Statistics by Variable and Data Line

The Bash script profstats.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, SD, 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_compare.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 profstats.py. The output is a file named profile_stats or profile_stats_abs if absolute values are used. To run the Bash script, type the name as follows:

```
./profstats.sh
```

and enter the information as requested. Note that the previous copy of the output file is deleted if the user indicates it's the first use of the output file so as to avoid appending multiple tables for each statistic.

The Python 3 script may be run separately as follows:

```
python3 profstats.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 = abs). The optional argument –v operates as described in Section 3.1. The current program processes both forms of output, that is, differences in wind components and RH (or vector wind speed) or differences in wind speed and direction and T_v. The output filename currently is profile_stats, except that for absolute value output the name is profile_stats_abs. 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_abs (or other name if not 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_abs. The Bash script will delete the file if the user indicates it is the first use of the output file. Table 9 shows a sample of the output arrays or tables derived from a diff_profiles file produced via prof_compare.sh.

Table 9 Tables of basic statistics (mean, median, SD, and RMSE) for the difference tables of the diff_profiles append file. For brevity, only the first 10 data lines are shown for each statistic, except that the last table (RMSE) has some additional lines that show the change from 6 samples to 2.

```
Statistics for profile differences by variable and data line.

Mean values
 Level  Pressure  Height  U-wind    V-wind  Rel-Humidity  Temperature  Prs(not_diff)  Samples
         (hPa)     (m)     (m/s)    (m/s)       (%)           (K)          (hPa)
   0       0       5.93     1.61    -0.58        3.74        -0.50        950.0          6
   1       0       5.47     0.63    -1.26       -5.43        -0.30        925.0          6
   2       0       5.00     0.99    -0.16        4.07        -0.54        900.0          6
   3       0       5.27     0.18     0.13       -0.32        -0.29        875.0          6
   4       0       4.70    -1.05     0.00       -9.23         0.13        850.0          6
   5       0       4.73    -1.08     0.20      -21.75         0.74        825.0          6
   6       0       5.57    -0.14    -0.04       -4.61         0.72        800.0          6
   7       0       5.27    -0.34     0.51        2.01         0.17        775.0          6
   8       0       5.60    -0.62     0.61        2.77        -0.01        750.0          6
   9       0       5.50    -0.01    -0.50       -0.23         0.03        725.0          6

Median values
  Level  Pressure  Height  U-wind   V-wind  Rel-Humidity  Temperature  Prs(not_diff)  Samples
         (hPa)     (m)     (m/s)    (m/s)       (%)           (K)          (hPa)
   0       0       9.55     2.42    -0.57        5.61        -0.34        950.0          6
   1       0       9.20     0.94    -1.61       -8.15        -0.20        925.0          6
   2       0       8.75     0.95    -0.24        1.51        -0.49        900.0          6
   3       0       9.25    -0.14     0.19       -0.47        -0.25        875.0          6
   4       0       8.40    -1.23    -0.04      -11.18         0.13        850.0          6
   5       0       9.05    -0.96     0.29      -24.14         0.89        825.0          6
   6       0       9.65    -0.21    -0.51       -0.28         0.97        800.0          6
   7       0       9.35    -0.51     0.56        3.61         0.26        775.0          6
   8       0       9.40    -0.92     0.76        2.71        -0.01        750.0          6
   9       0       9.30    -0.01    -0.76       -0.35         0.04        725.0          6
```

Table 9 Tables of basic statistics (mean, median, SD, and RMSE) for the difference tables of the diff_profiles append file. For brevity, only the first 10 data lines are shown for each statistic, except that the last table (RMSE) has some additional lines that show the change from 6 samples to 2 (continued).

```
Statistics for profile differences by variable and data line.

Standard Deviation values
 Level  Pressure  Height  U-wind    V-wind  Rel-Humidity  Temperature  Prs(not_diff)  Samples
         (hPa)     (m)     (m/s)    (m/s)        (%)          (K)         (hPa)
   0       0      10.28     1.36     0.25      14.21         0.44        950.0          6
   1       0      10.23     0.59     0.75      11.27         0.34        925.0          6
   2       0      10.11     1.22     0.71       4.36         0.24        900.0          6
   3       0      10.02     1.48     0.75       3.57         0.11        875.0          6
   4       0      10.00     1.05     0.09       4.95         0.08        850.0          6
   5       0      10.27     0.61     2.48      10.77         0.38        825.0          6
   6       0      10.24     1.13     0.95       7.44         0.66        800.0          6
   7       0      10.44     0.76     0.85       5.95         0.66        775.0          6
   8       0      10.38     0.52     0.62       1.23         0.70        750.0          6
   9       0      11.20     0.73     0.79       6.02         0.93        725.0          6


Root Mean Square Error values
 Level  Pressure  Height  U-wind    V-wind  Rel-Humidity  Temperature  Prs(not_diff)  Samples
         (hPa)     (m)     (m/s)    (m/s)        (%)          (K)         (hPa)
   0       0      11.87     2.11     0.63      14.69         0.67        950.0          6
   1       0      11.60     0.86     1.47      12.51         0.45        925.0          6
   2       0      11.28     1.57     0.73       5.97         0.59        900.0          6
   3       0      11.32     1.49     0.77       3.58         0.31        875.0          6
   4       0      11.05     1.48     0.09      10.48         0.15        850.0          6
   5       0      11.31     1.24     2.48      24.27         0.83        825.0          6
   6       0      11.66     1.13     0.95       8.76         0.98        800.0          6
   7       0      11.69     0.83     0.99       6.28         0.68        775.0          6
   8       0      11.79     0.81     0.87       3.03         0.70        750.0          6
   9       0      12.47     0.73     0.94       6.03         0.93        725.0          6
.
.
.
  55       0      10.83     2.52     3.82       0.57         1.64         70.0          6
  56       0      11.40     2.55     3.10       0.73         1.87         65.0          6
  57       0      13.88     3.60     2.17       0.76         0.26         60.0          6
  58       0      11.69     3.74     5.50       1.12         0.67         55.0          6
  59       0       8.20     6.22     0.55       1.52         1.22         50.0          2
  60       0       9.53     6.14     0.81       1.23         1.46         45.0          2
  61       0      24.22     0.90     4.56       1.29         1.07         40.0          2
  62       0      21.59     1.47     2.37       0.83         1.31         35.0          2
  63       0       4.74     7.36     1.44       0.47         0.90         30.0          2
```

4. Conclusion

A set of Python 3 and Bash scripts are presented that together may be used to extract vertical profiles (aka soundings) of meteorological variables from meteorological model output files in GRIB2 (GFS and GALWEM) and netCDF (WRF) formats. The NOAA set of programs, known as wgrib2, is used in the extraction of vertical profiles from the GRIB2 files, and an NCL script provided by Reen (2017) is used in the extraction of profiles from the netCDF files. A set of C programs mostly developed earlier converts those profiles into soundings with user-defined height or pressure structures. Output from these programs may be in terms of height or pressure levels, or height or pressure layers defined by those levels. In turn, Python 3 scripts produce tables of the differences of the variables by height or pressure levels or layers. Another Python 3 script can generate basic statistics for each variable at each common height or pressure level or layer, if the output for similar tables is appended to a single file, where similar refers to the same user-defined structure though not necessarily the same number of levels or layers. Many of these Python 3 scripts and C programs are called by Bash scripts that combine several of the Python 3 scripts and C programs into a single entity.

The several scripts and programs may be found on ARL's GitLab Sitcore site and those may be made available on request. The listings of the Bash and Python 3 scripts are attached in Appendix A. The source code of many of the C programs that generate wind speed and direction, and so on, output at user-defined levels and layers are available via the US Army Research Laboratory open GitHub site (https://github.com/ usarmyresearchlab) under the repository name ARL_MET_profile_Converter; descriptions of the methods with input and output samples may be found in my previous work (Cogan 2015, 2017).

5. References

Cogan J. A generalized method for vertical profiles of mean layer values of meteorological variables. Adelphi Laboratory Center (MD): Army Research Laboratory (US); 2015. 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.

Cuevas O, Chacon A, Cure M. Radiosonde campaign in Paranal Observatory 2011: PWV measurement. Valparariso (Chile): AstroMeteorology group, Physics and Astronomy Dept, Universidad de Valparaiso; 2011.

Dutsch ML. Evaluation of the WRF model based on observations made by controlled meteorological balloons in the atmospheric boundary layer of Svalbard. Bergen (Norway): Meteorologisk Institutt, Bergen; 2012.

Ingleby B, Isaksen L, Kral T, Haiden T, Dahoui M. Improved use of atmospheric in situ data. In: ECMWF Newsletter. 2018;155:20–25. doi:10.21957/cf724bi05s.

Kilpelainen T, Vihma T, Mannienen M, Sjoblom A, Jakobson E, Palo T, Maturilli M. Modelling the vertical structure of the atmospheric boundary layer over Arctic fjords in Svalbard. Q J R Meteorol Soc. 2012;138:1867–1883.

Reen B. Army Research Laboratory (US), Adelphi Laboratory Center, MD. Personal communication, 2017.

Skamarock WC, Klemp JB, Dudhia J, Gill DO, Barker DM, Duda MG, Huang X-Y, Wang W, Powers JG. A description of the Advanced Research WRF Version 3. Boulder (CO): National Center for Atmospheric Sciences; 2008. Report No.: NCAR/TN-475+STR.

Additional Resources

In addition, the following websites provide information on aspects of model evaluation and processing:

- http://www.emc.ncep.noaa.gov/gmb/STATS_vsdb/, verification website of NCEP.
- https://www.ecmwf.int/en/forecasts/charts/catalogue/, verification website of the ECMWF.
- http://apps.ecmwf.int/wmolcdnv/ at the ECMWF for worldwide comparisons for deterministic forecasts.
- http://epsv.kishou.go.jp/EPSv/ at the JMA for comparisons of ensemble forecasts.
- http://www.dtcenter.org/met/users/ and http://www.dtcenter.org/upp/users/ contain links to extensive user's guides, tutorials, and other documentation at the WRF DTC for their Model Evaluation Tool and Unified Post Processing system for analysis of output and statistical evaluation. Similar tools are available via NCEP.

Appendix A. Programs and Scripts

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). The listings of the Bash and Python 3 scripts are attached alongside the respective script as text files (.txt extension). The National Center for Atmospheric Research Common Language (NCL) script also is attached. The basic C program listings (source files) may be found on the US Army Research Laboratory GitHub site (https://github.com/usarmyresearchlab) and are not repeated here. The changes noted to output, for example, u and v wind components versus wind speed and direction, are in the output routine only.

A-1 Bash Scripts

wgrb2.sh - extracts a "sounding" from a GFS or GALWEM output file and converts it to a sounding with user height or pressure levels and layers. Includes options for the user.

wrf_extract.sh - extracts a sounding from a WRF output file and converts it to a sounding with user height or pressure levels and layers. Includes options for the user. The included NCL program was written by B Reen as were parts of this Bash script.

raob.sh - takes a radiosonde sounding from the University of Wyoming's weather website or from the NOAA archive website and converts it to forms with user height or pressure levels and layers. Includes options for the user.

prof_compare.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_compare_height.py. for(1) height levels and layers, and (2) prof_compare_pressure.py and prof_compare_baseprs.py for pressure levels and layers (program or user selects 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 profstats.sh

profstats.sh - computes basic statistics for a set of appended difference tables generated by prof_compare.sh. 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 1000-1500m layer over all the appended difference tables. It handles all versions of the output from prof_compare.sh. Also, it can compute the statistics for the absolute values of the differences. It includes the profstats.py script.

A-2 Python 3 Scripts (or Programs)

gg_wg2.py - extracts a sounding from a GFS or a GALWEM output file and writes it in a user-friendly format. At the maximum wind level and tropopause height RH is not in the input file. Consequently RH was assumed to be 50% at those levels.

prof_compare_height.py - computes the differences between 2 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. If the comparison is derived from soundings having wind components the user can choose to compute and output vector wind differences instead of RH differences. Basic statistics are computed for each variable over all heights. Output may be appended to a single file for later processing by profstats.py.

prof_compare_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. If the comparison is derived from soundings having wind components the user can choose to compute and output vector wind differences instead of RH differences. Basic statistics are computed for each variable over all pressures. Output may be appended to a single file for later processing by profstats.py.

prof_compare_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 vs. having the program determine that value. If the comparison is derived from soundings having wind components the user can choose to compute and output vector wind differences instead of RH differences. Basic statistics are computed for each variable over all pressures. Output may be appended to a single file for later processing by profstats.py.

profstats.py - computes basic statistics for a set of appended difference tables generated by prof_compare_*.py. 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 1000-1500m layer over all the appended difference tables. It handles both versions of the output from prof_compare_*.py. Also, it can compute the statistics for the absolute values of the differences. Here * indicates height, pressure, or baseprs.

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

A-3 NCL Script

extract_profile.ncl - written by B Reen (private communication) and is called from the Bash script wrf_extract.sh. It extracts a sounding from a WRF output file given the user input requested by wrf_extract.sh.

A-4 C Programs

The C programs have two versions, one for output of wind speed, wind direction, and virtual temperature, and a second for output of u and v wind components and relative humidity. The programs in the second set (e.g., wind components) are identified by the number 1 at the end of the filename (e.g., convertgfs1 vs. convertgfs). Only the first set is described here, since both use the same input and parameter files, and are run the same way. The input_parameters file is produced using the wgrb2.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/input-output_data/).

convertgfs - converts a sounding derived from GFS or GALWEM output into a form having user defined height levels and layers defined by those levels. (Use convertgfs1 for u, v and RH output.)

convertprsgfs - converts a sounding derived from GFS or GALWEM output into a form having user defined pressure levels and layers defined by those levels. (Use convertprsgfs1 for u, v and RH output.)

convertwrf - converts a sounding derived from WRF output into a form having user defined height levels and layers defined by those levels. (Use convertwrf1 for u, v and RH output.)

convertprswrf - converts a sounding derived from WRF output into a form having user defined pressure levels and layers defined by those levels. (Use convertprswrf1 for u, v and RH output.)

convertnoaa - converts a radiosonde observation sounding (RAOB) from the NOAA archive into a form having user defined height levels and layers defined by those levels. (Use convertnoaa1 for u, v and RH output.)

convertprsnoaa - converts a radiosonde observation sounding (RAOB) from the NOAA archive into a form having user defined pressure levels and layers defined by those levels. (Use convertprsnoaa1 for u, v and RH output.)

convertwyo - converts a radiosonde observation soundings (RAOB) from the U of Wyoming's weatherwebsite into a form having user defined height levels and layers defined by those levels. (Use convertwyo1 for u, v and RH output.)

convertprswyo - converts a radiosonde observation sounding (RAOB) from the U of Wyoming'sweather website into a form having user defined pressure levels and layers defined by those levels. (Use convertprswyo1 for u, v and RH output.)

Notes:

- The versions that output wind components (u, v) and relative humidity are denoted with the number 1 at the end of the filename (e.g., convertprswyo1).
- The C programs require certain parameter files. One named input_parameters is generated via the wgrb2.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. Those can be modified by the user for the needed height or pressure levels. The programs (height or pressure based) will not run if the respective parameter file is missing.

Appendix B. Flowcharts for the Bash Scripts

Approved for public release; distribution is unlimited.

This appendix contains a flowchart for the entire package of Bash scripts when applied from extraction of soundings through the generation of statistics (Fig. B-1), and for the several individual Bash scripts (Figs. B-2–B-6).

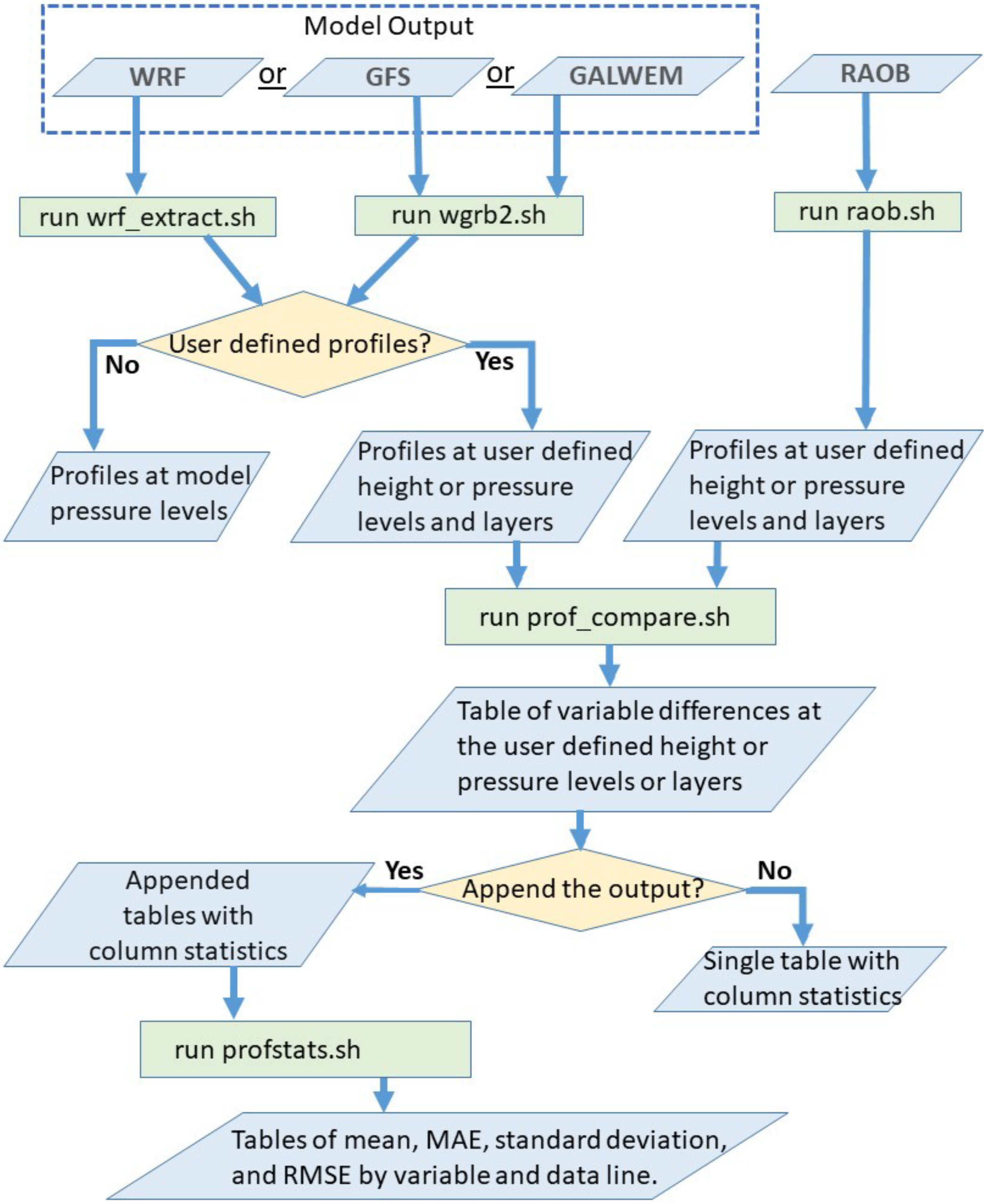

Model Output
WRF
or
GFS
or
GALWEM
RAOB
run wrf_extract.sh
run wgrb2.sh
run raob.sh
User defined profiles?
No
Yes
Profiles at model pressure levels
Profiles at user defined height or pressure levels and layers
Profiles at user defined height or pressure levels and layers
run prof_compare.sh
Table of variable differences at the user defined height or pressure levels or layers
Append the output?
Yes
No
Appended tables with column statistics
Single table with column statistics
run profstats.sh
Tables of mean, MAE, standard deviation, and RMSE by variable and data line.

Fig. B-1 Flowchart of the set of Bash scripts when used as a single package from extraction of soundings through generation of statistics for multiple sounding comparisons

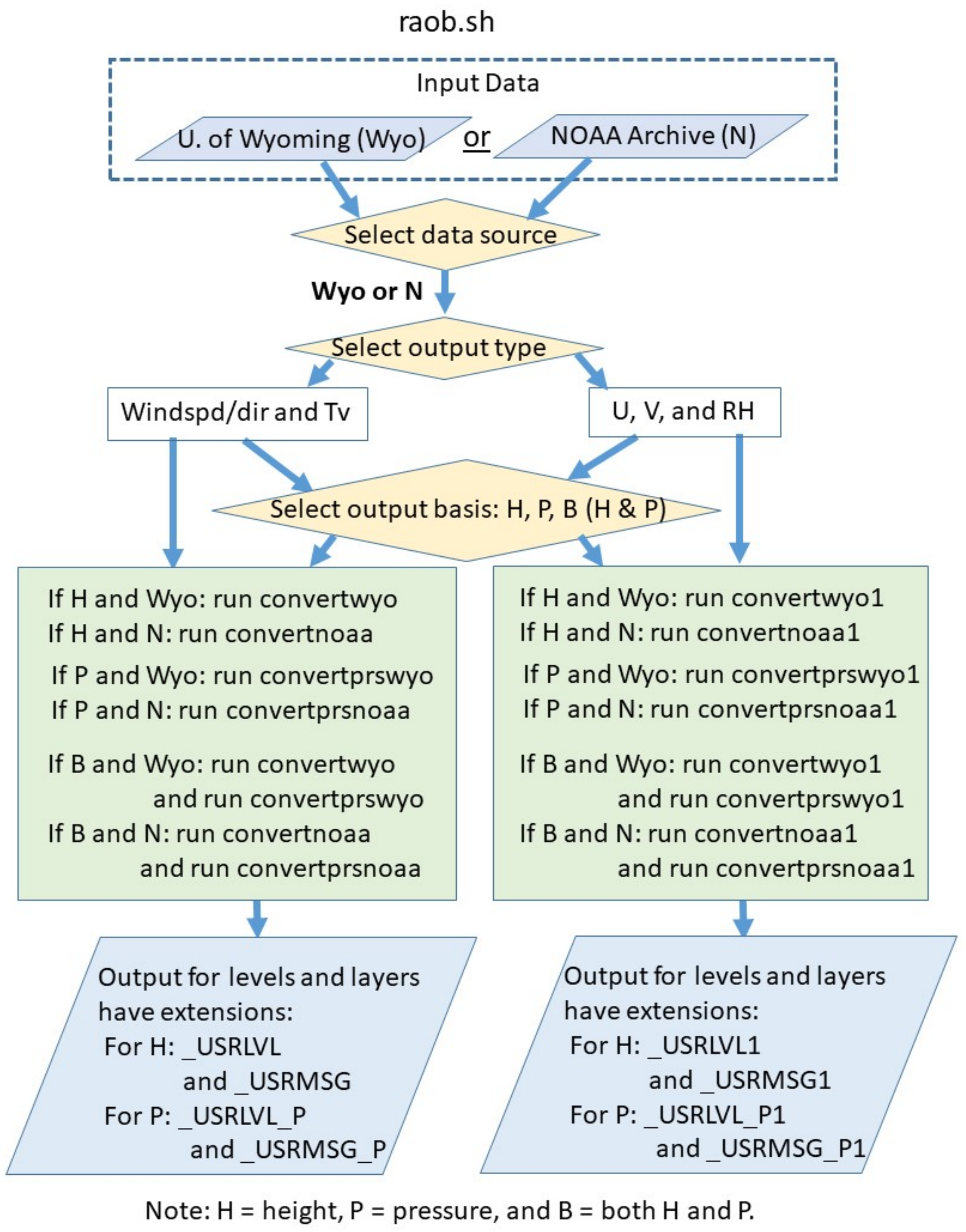

raob.sh
Input Data
U. of Wyoming (Wyo)
or
NOAA Archive (N)
Select data source
Wyo or N
Select output type
Windspd/dir and Tv
U, V, and RH
Select output basis: H, P, B (H & P)
If H and Wyo: run convertwyo
If H and N: run convertnoaa
If P and Wyo: run convertprswyo
If P and N: run convertprsnoaa
If B and Wyo: run convertwyo
and run convertprswyo
If B and N: run convertnoaa
and run convertprsnoaa
If H and Wyo: run convertwyo1
If H and N: run convertnoaa1
If P and Wyo: run convertprswyo1
If P and N: run convertprsnoaa1
If B and Wyo: run convertwyo1
and run convertprswyo1
If B and N: run convertnoaa1
and run convertprsnoaa1
Output for levels and layers
have extensions:
For H: _USRLVL
and _USRMSG
For P: _USRLVL_P
and _USRMSG_P
Output for levels and layers
have extensions:
For H: _USRLVL1
and _USRMSG1
For P: _USRLVL_P1
and _USRMSG_P1
Note: H = height, P = pressure, and B = both H and P.

Fig. B-2 Flowchart of the raob.sh Bash script for conversion of a radiosonde sounding from the University of Wyoming weather website or from the National Oceanic and Atmospheric Agency's (NOAA) radiosonde observation (RAOB) archive into user-defined height or pressure level and layer vertical profiles of meteorological variables. The output profiles include either wind speed and direction and virtual temperature (T_v) or horizontal wind components (u, v) and relative humidity (RH).

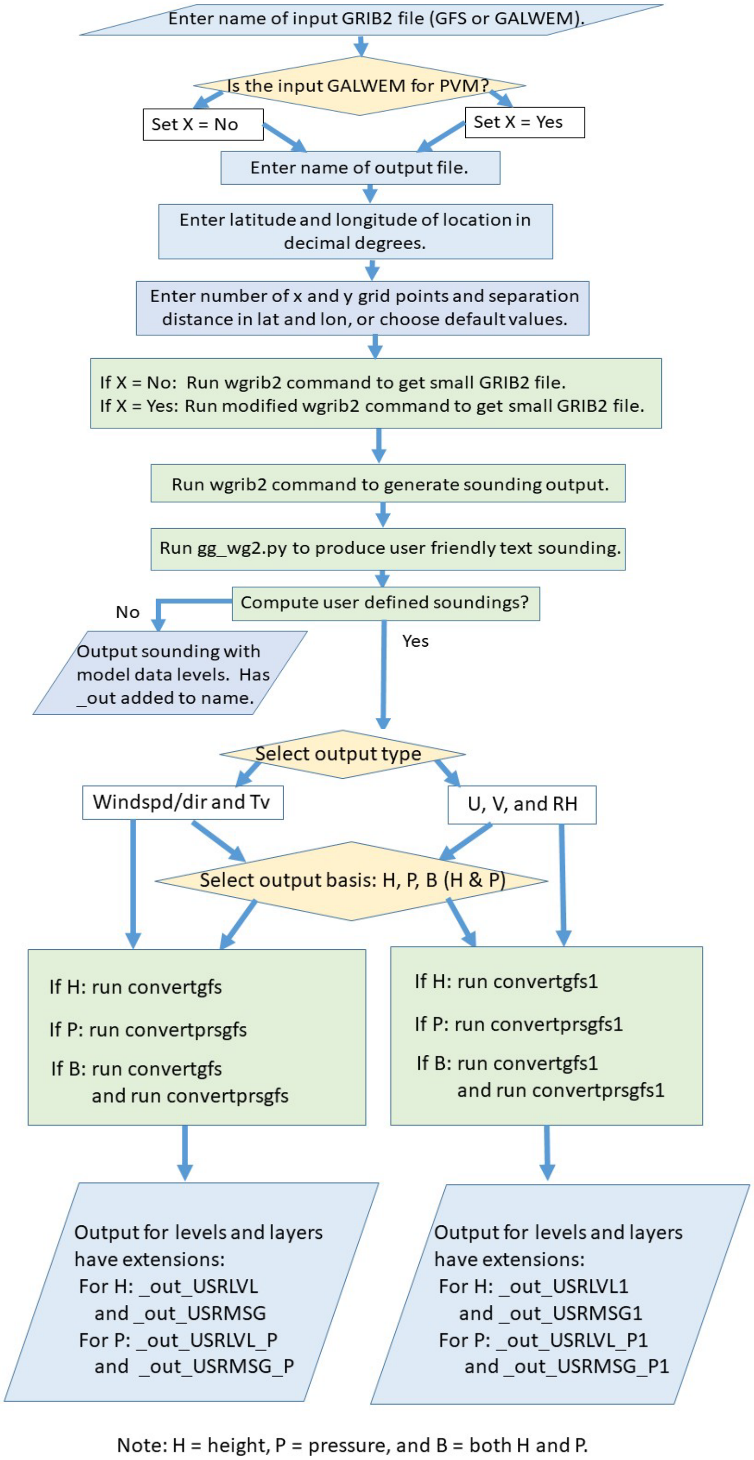

Enter name of input GRIB2 file (GFS or GALWEM).
Is the input GALWEM for PVM?
Set X = No
Set X = Yes
Enter name of output file.
Enter latitude and longitude of location in decimal degrees.
Enter number of x and y grid points and separation distance in lat and lon, or choose default values.
If X = No: Run wgrib2 command to get small GRIB2 file.
If X = Yes: Run modified wgrib2 command to get small GRIB2 file.
Run wgrib2 command to generate sounding output.
Run gg_wg2.py to produce user friendly text sounding.
Compute user defined soundings?
No
Output sounding with model data levels. Has _out added to name.
Yes
Select output type
Windspd/dir and Tv
U, V, and RH
Select output basis: H, P, B (H & P)
If H: run convertgfs
If P: run convertprsgfs
If B: run convertgfs and run convertprsgfs
If H: run convertgfs1
If P: run convertprsgfs1
If B: run convertgfs1 and run convertprsgfs1
Output for levels and layers have extensions:
For H: _out_USRLVL and _out_USRMSG
For P: _out_USRLVL_P and _out_USRMSG_P
Output for levels and layers have extensions:
For H: _out_USRLVL1 and _out_USRMSG1
For P: _out_USRLVL_P1 and _out_USRMSG_P1
Note: H = height, P = pressure, and B = both H and P.

Fig. B-3 Flowchart of the wgrb2.sh Bash script for extraction of a sounding from a GRIB2 output file (Global Forecast System [GFS] or Global Air Land Weather Exploitation Model [GALWEM]) and conversion into user-defined height or pressure level and layer vertical profiles. The output profiles include either wind speed and direction and virtual temperature (T_v) or horizontal wind components (u, v) and relative humidity (RH).

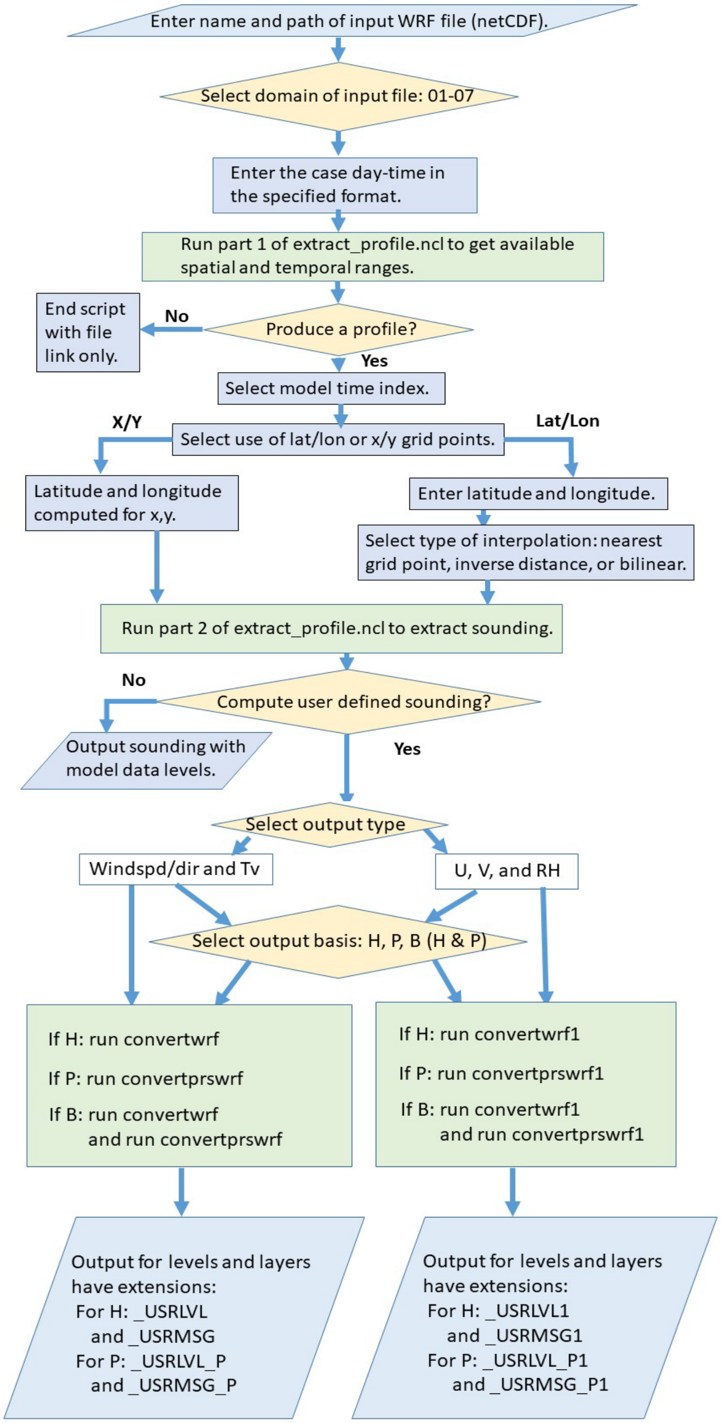
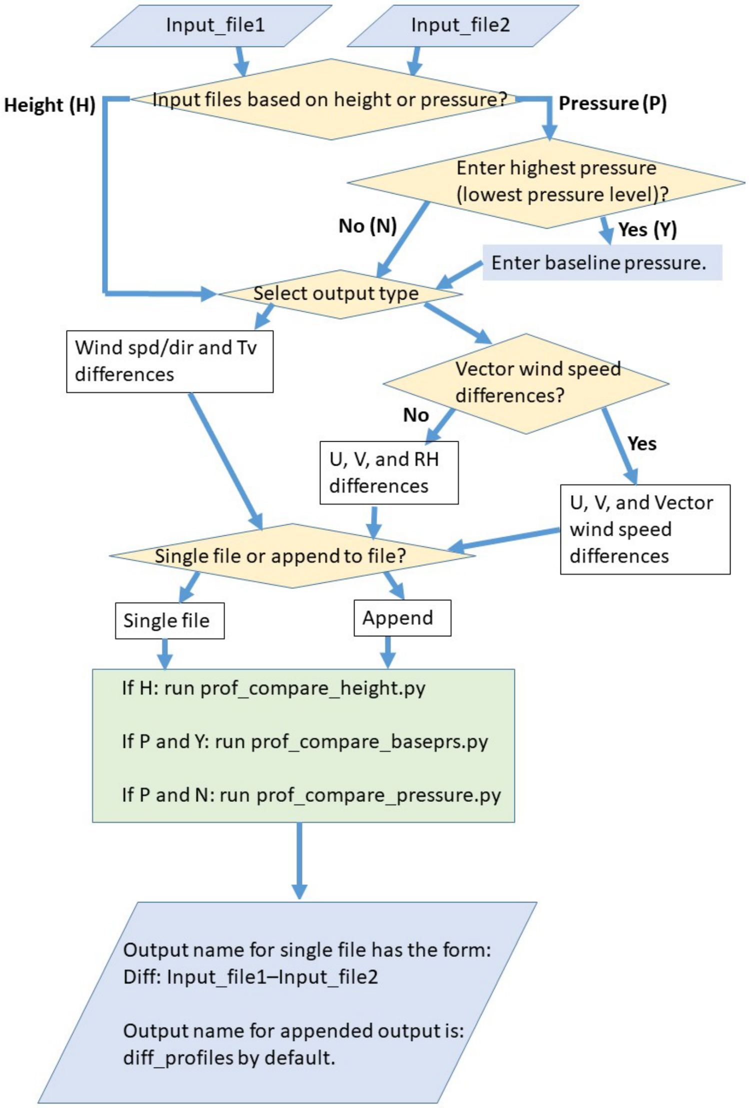

Enter name and path of input WRF file (netCDF).
Select domain of input file: 01-07
Enter the case day-time in the specified format.
Run part 1 of extract_profile.ncl to get available spatial and temporal ranges.
Produce a profile?
No
End script with file link only.
Yes
Select model time index.
X/Y
Select use of lat/lon or x/y grid points.
Lat/Lon
Latitude and longitude computed for x,y.
Enter latitude and longitude.
Select type of interpolation: nearest grid point, inverse distance, or bilinear.
Run part 2 of extract_profile.ncl to extract sounding.
No
Compute user defined sounding?
Output sounding with model data levels.
Yes
Select output type
Windspd/dir and Tv
U, V, and RH
Select output basis: H, P, B (H & P)
If H: run convertwrf
If P: run convertprswrf
If B: run convertwrf and run convertprswrf
If H: run convertwrf1
If P: run convertprswrf1
If B: run convertwrf1 and run convertprswrf1
Output for levels and layers have extensions:
For H: _USRLVL and _USRMSG
For P: _USRLVL_P and _USRMSG_P
Output for levels and layers have extensions:
For H: _USRLVL1 and _USRMSG1
For P: _USRLVL_P1 and _USRMSG_P1

Fig. B-4 Flowchart of the wrf_extract.sh Bash script for extraction of a sounding from a netCDF output file (WRF) and conversion into user-defined height or pressure level and layer vertical profiles of meteorological variables. The output profiles include either wind speed and direction and virtual temperature (T_v) or horizontal wind components (u, v) and relative humidity (RH).


Input_file1
Input_file2
Height (H)
Input files based on height or pressure?
Pressure (P)
Enter highest pressure
(lowest pressure level)?
No (N)
Yes (Y)
Enter baseline pressure.
Select output type
Wind spd/dir and Tv differences
Vector wind speed differences?
No
Yes
U, V, and RH differences
U, V, and Vector wind speed differences
Single file or append to file?
Single file
Append
If H: run prof_compare_height.py
If P and Y: run prof_compare_baseprs.py
If P and N: run prof_compare_pressure.py
Output name for single file has the form:
Diff: Input_file1–Input_file2
Output name for appended output is:
diff_profiles by default.

NOTE: operation of the script is independent of whether height or pressure based, selected wind and moisture variables, and whether or not the user enters the base (highest) pressure. The type of data in the output file is in the file name (single file) and in the header information within the file (single file and appended). The base pressure (user or script derived) is the pressure listed in the first data line.

Fig. B-5 Flowchart of the prof_compare.sh Bash script for comparison of two soundings from model and RAOB output or from two different models or model configurations. The output may include differences in either wind speed and direction and virtual temperature (T_v) or horizontal wind components (u, v) and relative humidity (RH). However, when the comparison is derived from soundings having wind components, the user can choose to compute and output vector wind differences instead of RH differences.

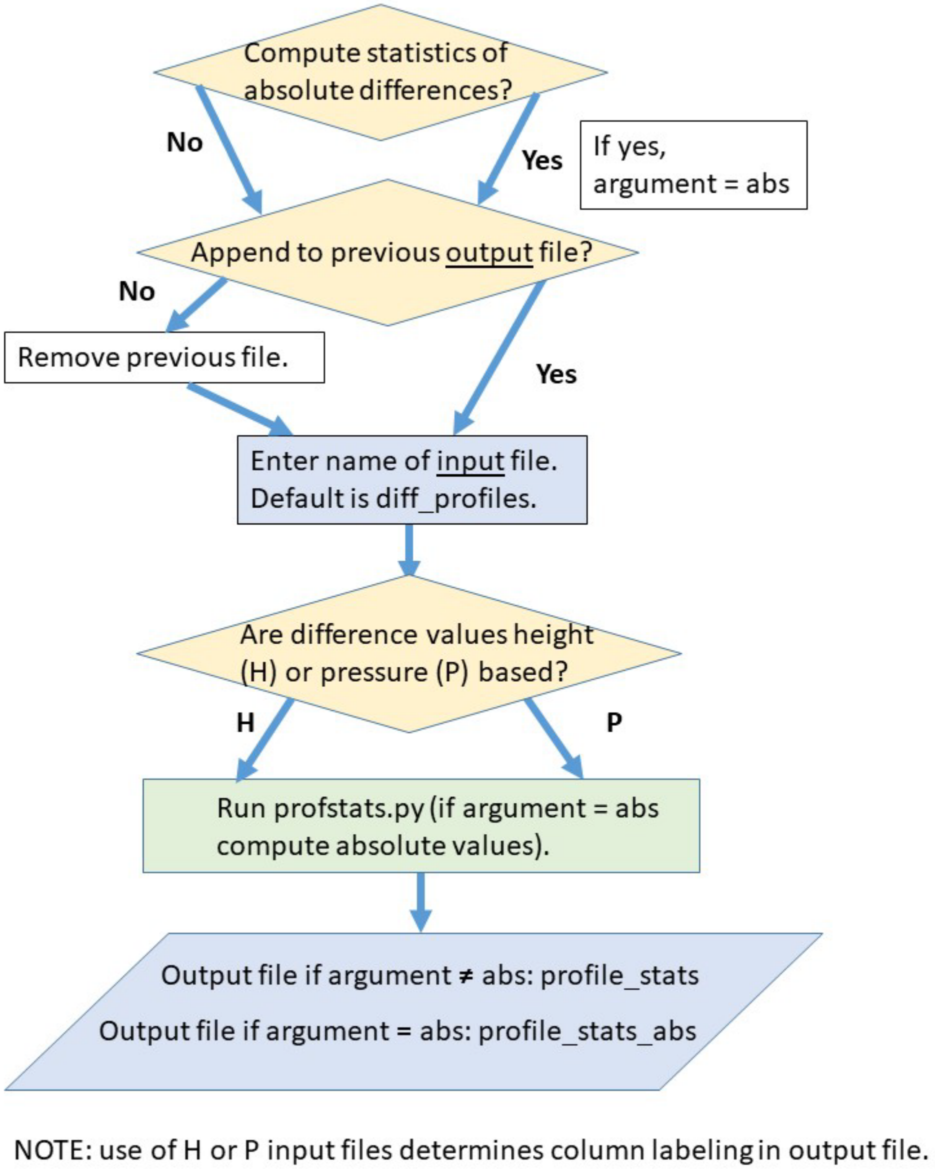

Compute statistics of absolute differences?
No
Yes
If yes, argument = abs
Append to previous output file?
No
Remove previous file.
Yes
Enter name of input file. Default is diff_profiles.
Are difference values height (H) or pressure (P) based?
H
P
Run profstats.py (if argument = abs compute absolute values).
Output file if argument ≠ abs: profile_stats
Output file if argument = abs: profile_stats_abs
NOTE: use of H or P input files determines column labeling in output file.

Fig. B-6 Flowchart of the profstats.sh Bash script for generation of tables of basic statistics such as mean and standard deviation by data line and variable. The user may choose to compute the statistics for the absolute values of the differences computed using prof_compare.sh.

List of Symbols, Abbreviations, and Acronyms

NCEP	National Center for Environmental Prediction
JMA	Japan Meteorological Agency
RAOB	radiosonde observation
DTC	Developmental Test Center
ECMWF	European Centre for Medium-Range Weather Forecasting
GALWEM	Global Air Land Weather Exploitation Model
GFS	Global Forecast System
LAT	latitude
LON	longitude
MET	Model Evaluation Tools
NCL	National Center for Atmospheric Research Common Language
NOAA	National Oceanic and Atmospheric Agency
NWP	numerical weather prediction
PVM	Profiler Virtual Module
RH	relative humidity
RMSE	root-mean-square error
SD	standard deviation
UTC	Coordinated Universal Time
WRF	Weather Research and Forecasting model

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
2 (PDF)	DIR ARL IMAL HRA RECORDS MGMT RDRL DCL TECH LIB
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
1 (PDF)	ARL RDRL CIE J COGAN