Supplemental Info for AMS Student Conference 2016
Tools of the Trade: Big Data
Kevin Tyle, Dept of Atm & Env Sci., UAlbany

I. Meteorological Data Formats

As with any type of data, meteorological data falls into one of two types: **text** and **binary**. Actually, it turns out that the former is just a special case of the latter.

Text:

When we talk about *text-based* data, we really mean that it is of a format that is *human-readable* (with apologies to all of the non-human life forms that may actually be able to read, that is :D). The most common standard for representing text-based data is a format called ASCII. There are plenty of examples of meteorological data in this format:

1. NWS Area Forecast Discussion

```
FXUS61 KALY 071802
AFDALY

AREA FORECAST DISCUSSION
NATIONAL WEATHER SERVICE ALBANY NY
102 PM EST THU JAN 7 2016

SYNOPSIS...
HIGH PRESSURE WILL REMAIN IN CONTROL OF OUR WEATHER THROUGH
FRIDAY. A COMPLEX LOW PRESSURE SYSTEM WILL APPROACH AND MOVE
ACROSS THE REGION THIS WEEKEND BRINGING UNSETTLED WEATHER TO
THE AREA. LIGHT MIXED PRECIPITATION IS POSSIBLE FRIDAY NIGHT INTO
SATURDAY...THEN TEMPERATURES WILL WARM UP ENOUGH FOR RAIN.
```

2. METAR Bulletin

```
SAUS70 KWBC 071400
METAR
KANQ 071355Z AUTO 00000KT 10SM OVC090 M02/M06 A3017 RMK=
KFLY 071358Z AUTO 28004KT 10SM BKN055 BKN065 M02/M06 A2972 RMK AO2
T10241061=
```

Note that with the latter, although the *format* is ***ASCII***, this doesn't mean that the substance of the file cannot be in an encoded format.

Binary:

Remember, computers really just process an arbitrary amount of **0** and **1**'s. Even the above-mentioned text format is just a special, standardized case (**ASCII**) of binary data. As with ASCII, having established *standards* for how the data is formatted is essential in order for the data to be analyzed and/or visualized. The **World Meteorological Organization** (**WMO**) establishes standards for certain types of meteorological data; below we'll discuss two such examples: one typically used for **point** data, and the other for **gridded**.

1. ***BUFR*** *(point-based observations)*
This data format is formally-named The Binary Universal Form for the Representation of meteorological data.
Take a look at the the WMO documentation of BUFR.

2. ***GRIB*** *(data on a grid)*
This data format is formally-named GRIB (GRIdded Binary or General Regularly-distributed Information in Binary form. It is most typically used with output from numerical weather prediction models, such as the GFS, NAM, or ECMWF. There are two *versions* of data in **GRIB** format. One, *GRIB-1*, uses no compression. The second, *GRIB-2*, packs the bulk of its data utilizing compression software. As meteorological data of all kinds, but particularly NWP output, have increased in size, the vast majority of data produced by operational centers is in the latter, compressed, format.

Take a look at the WMO documentation of GRIB.

Both of these data formats include *header* information before the actual "meat and potatoes" of the meteorological information itself appears in the file. Such header information includes details such as:

1. Valid date of the product
2. Originating forecast center (e.g., NCEP, ECMWF)
3. Size of the entire file
4. Tables used to map products into meteorologically-relevant fields (e.g. 500 mb geopoential height)

Point (4) above is very important. ***Without the correct set of tables, there is no way to tell for sure what meteorological field you are looking at! Tables constantly change, and as is typical with understaffed bureaucratic agencies, coordination of such changes is not practiced as consistently as it should!***

Fortunately, there is another data format that is increasingly used to represent meteorological data, which addresses the problems with table-driven formats such as **BUFR** and **GRIB!**

3*. NetCDF*
This data format, NetCDF (Network Common Data Form), was developed and is maintained by The Unidata program at the University Corporation for Atmospheric Research in Boulder, Colorado. It incorporates two principles:
a. *Machine independence*: i.e., the data can be read from and written to regardless of computer hardware.
b. *Self-describing*: all the information, or *metadata* which are necessary to identify a product is contained within the **NetCDF** file itself. This eliminates the need for external tables!

NetCDF can be used for any type of meteorological (and other scientific) data, such as point observations, gridded datasets, and remotely-sensed data (e.g., satellite and radar data). It is the default data type for output grids produced by the open-source A*dvanced Research Weather Research and Forecasting Model* ***(ARW WRF)***.

NetCDF is currently at version 4. Version 3 is also commonly used. The more recent version avails itself of the Hierarchical Data Format, version 5 software library and data model. Among other things, **NetCDF-4** files support data compression.

Despite the many advantages of **NetCDF** over its **WMO**-sanctioned brethren, **GRIB** and **BUFR** remain the standard data format by operational meteorological centers such as NCEP, the NWS, and ECMWF … bureaucracy is tough to overcome!

Let's briefly discuss a few other common data formats seen in our field:

5. ***GEMPAK***: The venerable meteorological data analysis and visualization package uses its own format, simply described as *GEMPAK Data Management File* format. While there are several well-supported means of converting files from **GEMPAK** to **NetCDF**, the converse does not apply. However, **GEMPAK** provides its own decoders to convert from **GRIB** to **GEMPAK** format.

6. ***GINI*** *and* ***MCIDAS AREA****:* These formats are typically used for satellite imagery (either from the *imager* or *sounder* of **GOES** satellites)

7, ***NEXRAD:*** Data from **NEXRAD** sites are typically sent in two classes, ***NEXRAD-2*** and ***NEXRAD-3****.* The former represents one file for each site's individual volume scan, which comes out every 5-6 or 10-11 minutes, depending on whether the site is operating in ***precip*** or ***clear-air*** mode. The latter covers a class of several post-processed products, e.g. ***base reflectivity, storm-relative velocity,*** and ***velocity-azimuthal display***.

II. Meteorological Datasets

Datasets can simply be thought of as a collection of files of a particular ***datatype*** (a dataset just consists of one datatype, although datasets based on multiple datatypes can of course be hosted by a single entity).

The **Big Data** wave of course has flooded our field for many years now, and the number of datasets, ***repositories*** which host said sets, and their ***size and volume*** continue to grow. The distinction between ***datasets*** and ***data repositories*** warrants both being in their separate section here, but for brevity, let's defer this separation.

Here are some typical datasets and their approximate sizes:

1. One day's worth of (ASCII-format) METAR bulletins, worldwide: **35 MB**
2. One day's worth of (GRIB2) ½ degree GFS NWP output, (4x/day): **500 MB**
3. One 84-hour 20 km CONUS WRF NWP (hourly output, NetCDF-4): **20 GB**
4. Climate Forecast System Reanalysis (**CFSR)** (6-hourly output, GRIB2, 1979-present): **60 TB**
5. Climate Model Intercomparison Project, version 5 (**CMIP5**): **3.3 PB**

III. Analysis and Visualization of Meteorological Data

Armed with a programming language and a decent graphics card, anyone (including you) can design a data analysis and visualization system to meet your needs! However, most of us do not have the luxury to do so; fortunately, there are a wide variety of excellent (and **FREE**) software packages out there:

1. GEMPAK
2. AWIPS II
3. IDV and its "cousin", McIDAS-V
4. NCL
5. GrADS
6. ParaView
7. VAPOR

More generally, increasingly there are a number of programs and software libraries based on the **Python** language:

8. SHARPpy
9. MetPy
10. pyart

There are also Python libraries that support **NetCDF** and more generally other packages for structuring and analyzing large datasets, e.g. pandas :

11. netCDF4-python
12. xarray
13. NetCDF Operators (NCO)

Visualization in Python mostly leverages matplotlib. Mapping typically relies on the basemap package, also a part of **matplotlib**.

While not used as often as **Python** by atmospheric scientists, **R**, aka the R Project for Statistical Computing has particular strengths in its statistical packages, and also has some nifty visualization tools, e.g. ggplot2.

Although a programming language in its own right, **JavaScript** has some very nice visualizations, which leverage **HTML5**. Many of you may already have seen the excellent wind map animations via the Nullschool site. It uses **HTML5, NodeJS** and **JavaScript** for its entirely web-based interface. Even cooler is that all of its source code is available via the developer's GitHub page!

Let's not forget the tried-and-true, although pricey, data analysis and visualization packages, which you may have access to at your educational institution:

14. MATLAB
15. IDL

Finally, it is usually desirable to convert between one format and another (e.g., **GRIB** from/to **NetCDF**). Besides NCL, NCEP's wgrib (GRIB-1) wgrib2 (GRIB-2) software packages allow for conversion into NetCDF, as well as slicing/dicing/reconfiguring of GRIB files. Unidata's

NetCDF-Java software suite converts a variety of formats (including **GEMPAK** as well as **GRIB**) to NetCDF. **GEMPAK's dcgrib/dcgrib2** programs will convert **GRIB-1/GRIB-2** to **GEMPAK.** Unfortunately,**NetCDF** to **GEMPAK** does not have a one-stop solution to handle this conversion process.

IV. Data Hosting and Sharing

With a tip of the hat to the Beatles, "All the lovely data … where does it all come from?"

Prior to the advent of the World Wide Web in the 1990s, meteorological datasets were typically difficult to obtain in a timely manner. However, **NCAR** as well as what is now known as the **National Centers for Environmental Information** (formerly known as **NCDC**) served as data stewards and repository hosts. Data might be obtained in hardcopy form, or on magnetic tape (and in later years, optical media such as CD-ROM). In a university setting, data exchange might have used *sneaker-net* "technology" in order to transfer the data from person to person.

In the 1990s, web servers and FTP servers proved to be a much more speedy way to access data. Many of you may already be aware of NCEP's Model Analysis and Guidance web server, as well as their Realtime NCEP FTP Server. A disadvantage of using **HTTP** or **FTP** protocols is that this is typically an "all-or-nothing" way to transfer data. As datasets grew larger and larger, eventually it became clear that everyone downloading their own local copies of datasets was no longer feasible.

This prompted the development of a data access protocol, akin to the latter two above, which would eliminate the need for costly reproductions of datasets at any institution that had interest in it. This data access protocol, known as **DAP**, came about in the 1990s and is known as OPenDAP. It works as part of a client-server framework … where a remote site uses a *client* program to access data from a *server*.

As internet access speed impoved, particularly at larger institutions such as government and university centers, this 20-year old technology has increasingly become the way to go for the sharing of large datasets such as the **CFSR** and **CMIP5**. One of the most popular OPenDAP server packages is Unidata's THREDDS. **THREDDS** provides access to real-time and archived data and is easily deployed on a Linux system.

Other **OPenDAP** servers commonly used in the meteorological community include:

1. RAMADDA (very easy setup on any current OS; good integration with **IDV**)
2. GrADS Data Server, GDS.

Client software can easily incorporate the **OpenDAP** software library. Most all of the analysis and visualization packages discussed in section **III**, with the notable exception of **GEMPAK** have **OPenDAP** support.

Finally, access to the huge **CMIP5** climate simulation datasets is most easily done by using the Lawrence Livermore National Laboratories UV-CDAT toolset.

V. Closing Thoughts

Big Data is ubiquitous in our field and will only continue to grow in size and volume. Consider what is already here or soon-to-be upcoming:

1. 0.25 degree GFS Data
2. GOES-R
3. CMIP6

Many questions which in the past took months, if not years, to answer can now be just a Python script away. Think for example of the recently-hyped surge of warm air that reached the North Pole shortly after Christmas 2015. How often has the 2-meter air temperature at Santa's home exceeded the freezing point? The **CFSR**, accessible via NCAR/UCAR's Research Data Archive can readily be queried to answer this and similar questions.