Write and read data, and perform diagnostic/graphical analyses

Data formats commonly used in atmospheric science

ASCII/Text format

- Easy to read and write.
- Can directly edit.
- File sizes are large, slow to read and write, not suitable for large datasets.

Binary

- Different on big endian and little endian computers.
- Language (e.g., C, Fortran) dependent.
- Not self-descriptive. Without knowing exactly how the data is written, almost impossible to read.
- No internal compression.

GRIB/GRIB2

- World Meteorological Organization (WMO) standard for writing and distributing gridded NWP model data.
- Portable and compact with GRIB2 that supports internal compression.
- Complex encoding, poor documentations, difficult to read without existing subroutines/libraries.

BUFR

- A WMO standard format for encoding observational data.
- BUFR belongs to the category of table-driven code forms, where the meaning of data elements is determined by referring to a set of tables that are kept and maintained separately from the message itself.
- Not surprisingly, it is difficult to read also.

Data formats (borrowing from a slide)

Text, ASCII:

- METARS, meteograms, Wyoming upper air sonde profiles, Aeronet
- TXT, DAT, CSV, TSV, JSON,YAML, HTTP, XML, KML, ICARTT: NASA ESPO

Pros: Can open in almost anything, even a text editor or web browser but too large files are difficult or slow to open.

Cons: inefficient for read/write and storage. Different OSes treat non-printables differently

Excel, *.xls or *.csv

- IMPROVE field experiment data
- Column-ordered data, convertible to ASCII
-

Pros: mostly easy to import.
Cons: requires proprietary software to read. Limited complexity of data set.

HDF, netCDF, GIF – scientific data formats

- ARM, NASA EOS, MPLNet, ECMWF
- Cross-platform binary files
-

Pros: Cross-platform, efficient read/write and archival. Self-documentable
Cons: Requires libraries to open and access within a dev. environment.

NetCDF (Network Common Data Form) is a set of data formats and associated software developed by Unidata (unidata.ucar.edu), an organization under UCAR (the governing body of NCAR), initially to support data needs by the atmospheric science community, especially the modeling community. It has now become a general scientific data format use by man disciplines.

It is the most widely used data format by many community atmospheric modeling systems, including WRF and NCAR climate system models.

GRIB/GRIB2 is still the (WMO) standard data format used by most operational NWP models (e.g., GFS), there are conversion tools to convert GRIB format data to NetCDF format.

According to

https://www.unidata.ucar.edu/software/netcdf/

NetCDF (Network Common Data Form) is a set of software libraries and machine-independent data formats that support the creation, access, and sharing of array-oriented scientific data. It is also a community standard for sharing scientific data. The Unidata Program Center supports and maintains netCDF programming interfaces for C, C++, Java, and Fortran. Programming interfaces are also available for Python, IDL, MATLAB, R, Ruby, and Perl.

Data in netCDF format is:

- **Self-Describing.** A netCDF file includes information about the data it contains.
- **Portable.** A netCDF file can be accessed by computers with different ways of storing integers, characters, and floating-point numbers.
- **Scalable.** Small subsets of large datasets in various formats may be accessed efficiently through netCDF interfaces, even from remote servers.
- **Appendable.** Data may be appended to a properly structured netCDF file without copying the dataset or redefining its structure.
- **Sharable.** One writer and multiple readers may simultaneously access the same netCDF file.
- **Archivable.** Access to all earlier forms of netCDF data will be supported by current and future versions of the software.
- **Efficient.** Supports internal data compression. Support parallel I/O for on large parallel supercomputers.

How to deal with data in various formats? - Using existing software

Non-commercial software

- Python, pyart (Python ARM Radar Toolkit)
- R (GNU version of "S" statistical computing/graphics package)
- Unidata packages: AWIPS, GEMPAK, IDV, McIDAS, MetPy
- Ncview – a netCDF visual brower (http://meteora.ucsd.edu/~pierce/ncview_home_page.html) that is very convenient to quickly plot/visualize netCDF data.
- Vapor (https://www.vapor.ucar.edu/) – a 3D visualization software that supports netCDf.
- Github, SourceForge, Community-supported sites

Commercial software

- **Mathematica (OU educational license)**
- **Matlab (OU educational license)**
- **SPSS (OU educational license)**
- IDL
- IGOR
- OriginPro (https://www.originlab.com/)

Writing codes of your own to read and write data

In this class, we will focus on reading and writing NetCDF files using Python. Documentations for reading and writing NetCDF files using other languages can be found at https://www.unidata.ucar.edu/software/netcdf/.

There are different packages/modules available for reading/writing NetCDF format data. We will mainly look at the most commonly used netCDF4 and the xarray/rioarray package.

http://unidata.github.io/netcdf4-python/ is the Unidata netcdf4-python website that contains tutorials.

OU Library's Jupyter Lab that we will use in this class.

Go to http://jupyter.lib.ou.edu/, log in using your OU's 4+4 ID. The first time you log in, you will go through CILogon and may be asked to choose University of Oklahoma as the organization.

Dr. Mark Laufersweiler from OU library will provide guest lectures and tutorials on using the platform.

Let's try to download some simple example Python programs from

https://www.unidata.ucar.edu/software/netcdf/examples/programs/

and try to run within this environment.