

# Conventions for netCDF

Unidata TDS Training Workshop  
25-26 October 2018

# What is netCDF?

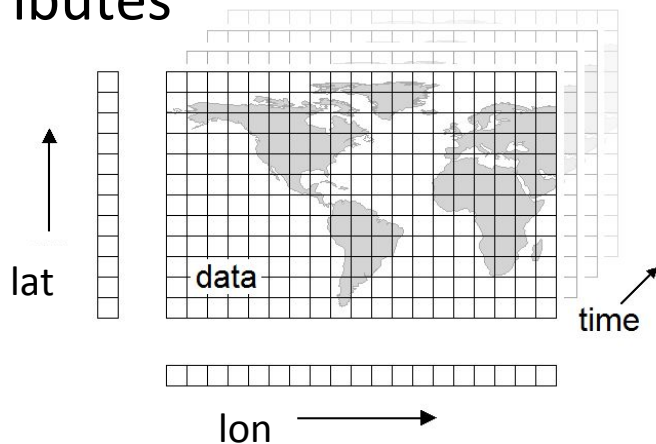
- Binary data format
- Software libraries / APIs
  - C (Fortran, C++, Python, ...) and Java
  - Upon which many data management, analysis, and visualization tools have been built
- Data model
  - Conceptual model of the data
  - Independent of data format details
  - Independent of programming language

# What is netCDF?

- Classic netCDF data model
  - Multidimensional arrays of data values
    - Which share dimensions
  - Attributes

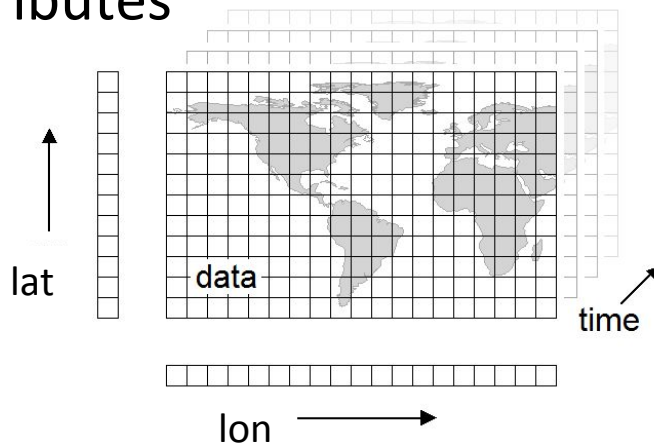
# What is netCDF?

- Classic netCDF data model
  - Multidimensional arrays of data values
    - Which share dimensions
  - Attributes



# What is netCDF?

- Classic netCDF data model
  - Multidimensional arrays of data values
    - Which share dimensions
  - Attributes



```
netcdf mydataset {  
  dimensions:  
    lat = 12 ; lon = 19 ; time = 4 ;  
  variables:  
    float lat(lat) ;  
    float lon(lon) ;  
    float temp(time, lat, lon) ;  
    float rh(time, lat, lon) ;  
  attributes:  
    :Conventions = "CF-1.6";  
}
```

# What is netCDF?

- Classic netCDF data model
  - Multidimensional arrays of data values
    - Which share dimensions
  - Attributes
- Enhanced netCDF data model
  - Adds hierarchical groups
    - Organize and group dimensions and variables
  - Adds structures
  - Only available in netCDF-4 (based on HDF5)

# The Same?

```
netcdf mydataset {  
  dimensions:  
    lon = 19 ;  
    lat = 12 ;  
    time = 4 ;  
  variables:  
    float lat(lat) ;  
    float lon(lon) ;  
    float time(time) ;  
    float temp(time, lat, lon) ;  
    float rh(time, lat, lon) ;  
}
```

```
netcdf yourdataset {  
  dimensions:  
    longitude = 19 ;  
    latitude = 12 ;  
    time = 4 ;  
  variables:  
    float latitude(latitude) ;  
    float longitude(longitude) ;  
    float time(time) ;  
    float temperature(time, latitude, longitude) ;  
    float rh(time, lat, lon) ;  
}
```

# Community conventions

- Various community agreed upon attribute conventions have developed over the years
  - NUG, COARDS, NCAR-RAF, ...
  - CF (Climate & Forecast) conventions
    - Gridded data has long been the focus of CF
    - Now moving into observational data
  - ACDD (Attribute Convention for Data Discovery)
    - Originally based on Dublin Core and others
    - Current focus on aligning with ISO 19115



# Goals for CF Conventions

- Locate data in space–time and as a function of other independent variables
- Identify data sufficiently to enable users of data from different sources to decide what is comparable

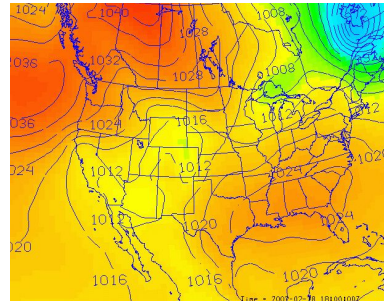
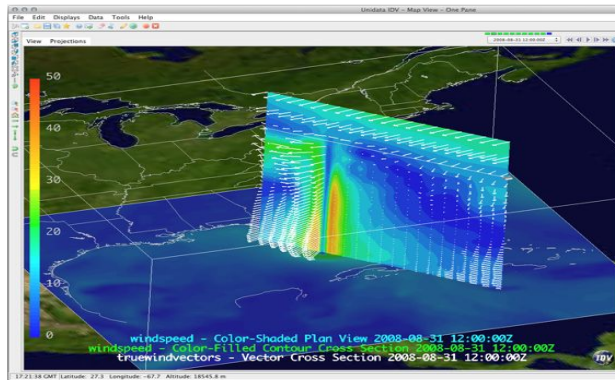
# CF Conventions

```
netcdf mydataset {  
  dimensions:  
    lat = 12 ;   lon = 19 ;   time = 4 ;  
  variables:  
    float lat(lat) ;  
      lat:units = "degrees_north" ;  
      lat:standard_name = "latitude" ;  
    float lon(lon) ;  
      lon:units = "degrees_east" ;  
      lon:standard_name = "longitude" ;  
  ...  
}
```

```
...  
  float temp(time, lat, lon) ;  
    temp:units = "Celsius" ;  
    temp:standard_name = "surface_temperature" ;  
  float rh(time, lat, lon) ;  
    rh:units = "percent" ;  
    rh:standard_name = "relative_humidity" ;  
  attributes:  
    :Conventions = "CF-1.6";  
}
```

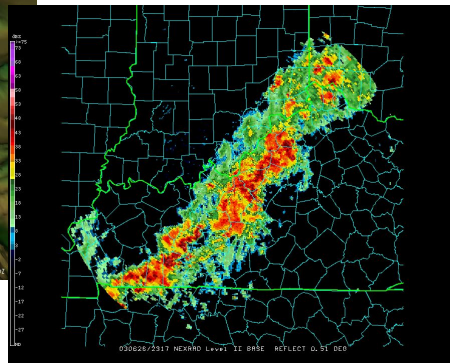
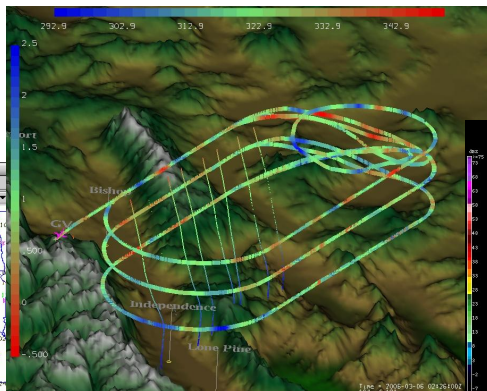
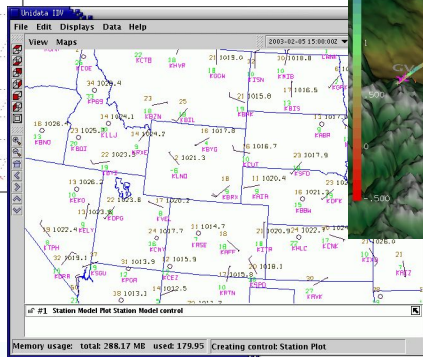
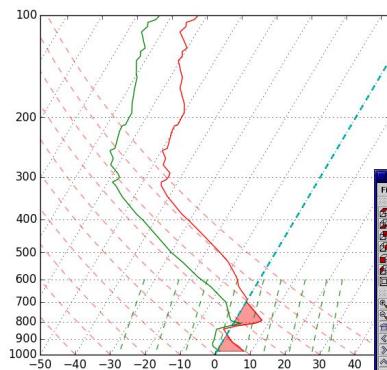
# CF conventions

- Historically dealt with gridded data



# CF conventions

- Historically dealt with gridded data
- Starting to deal with observational data

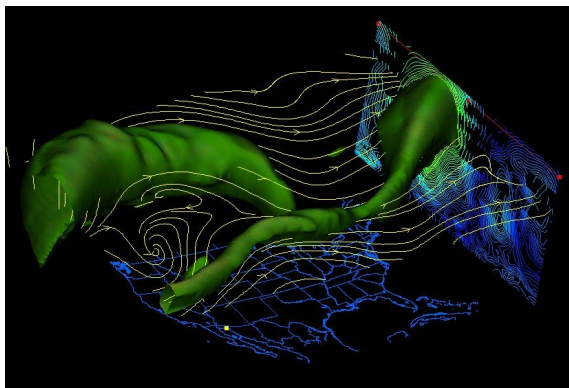


# CF Conventions

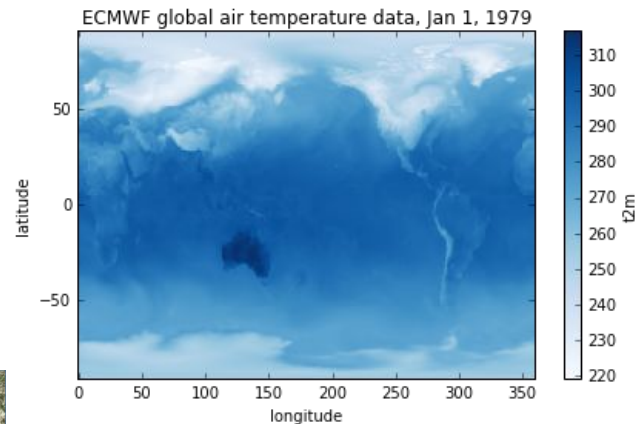
- Currently supported data types:
  - Gridded data
  - Timeseries, soundings, aircraft tracks
  - Unstructured grids (e.g., triangular mesh)
  - CF-Radial: radial data for radar and lidar
- Data types accepted into CF:
  - Timeseries for a polyline or polygon (aka Geometries)
- Under development:
  - Satellite swath data
  - Data uncertainty
  - Linked Data with netCDF

# Gridded Data

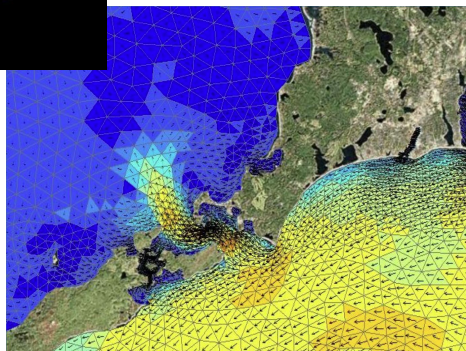
Visualization of wind using the IDV:  
streamlines and speed isosurface



Visualization using python:  
xarray and matplotlib



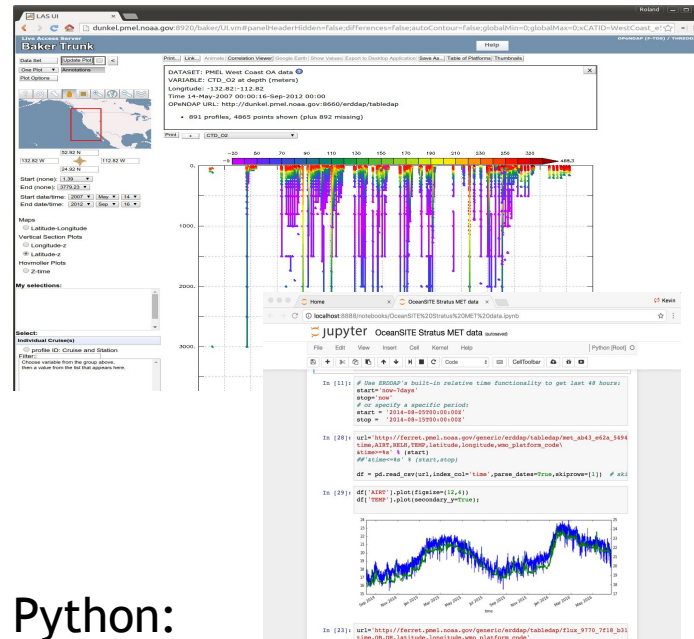
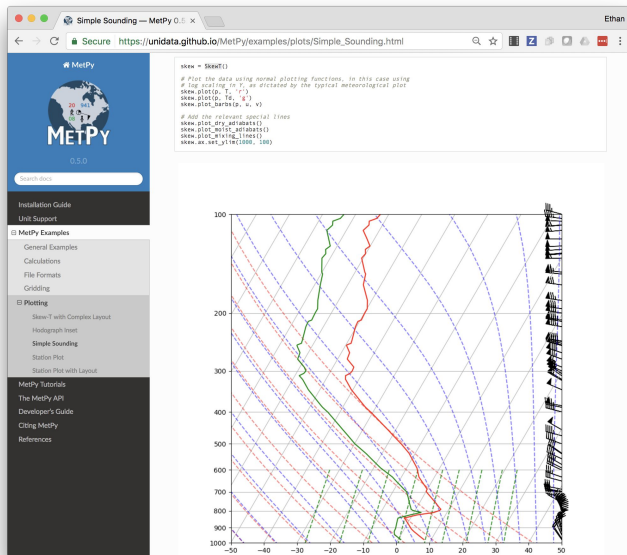
Visualization of  
triangular grid data



# Timeseries and Soundings

# Ferret and LAS

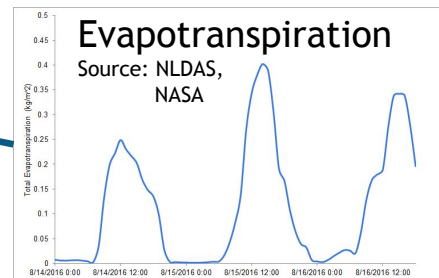
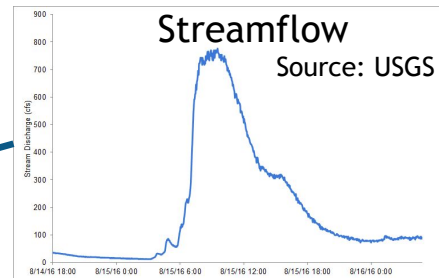
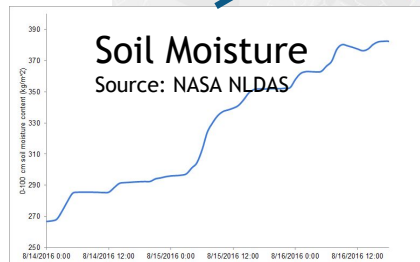
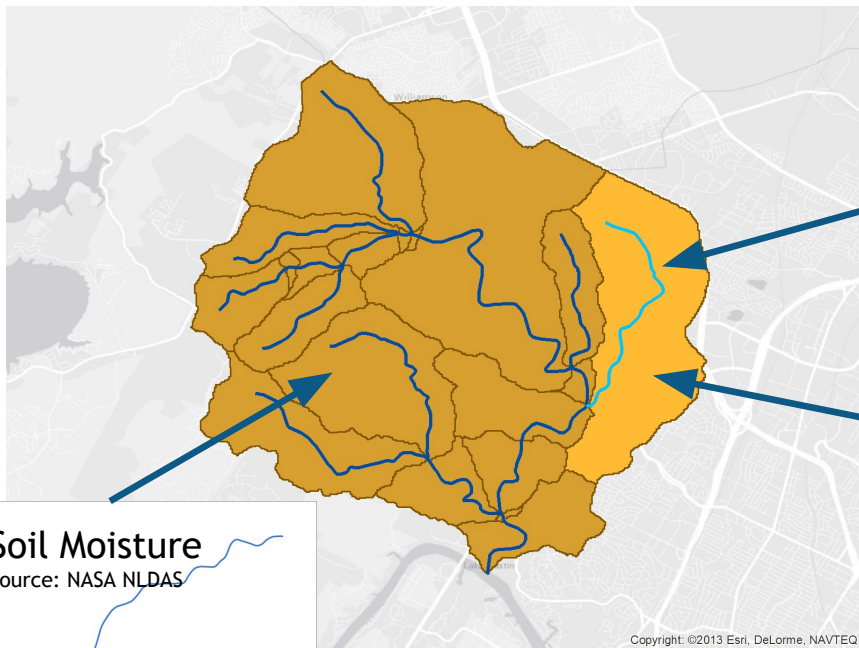
# Python: MetPy Library



# Python: Jupyter Notebook

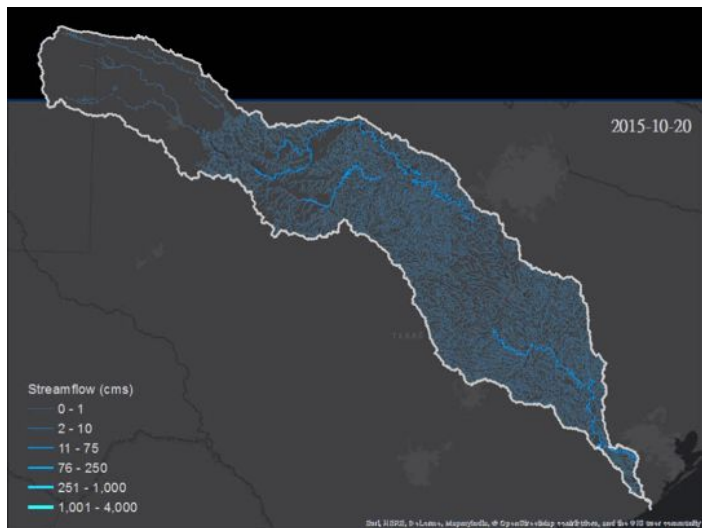


# Geometries





# Geometries (polylines & polygons)



## Included

Type	
Point	
LineString	
Polygon	

Type	
MultiPoint	
MultiLineString	
MultiPolygon	

[https://en.wikipedia.org/wiki/Well-known\\_text](https://en.wikipedia.org/wiki/Well-known_text)

## Compatible With

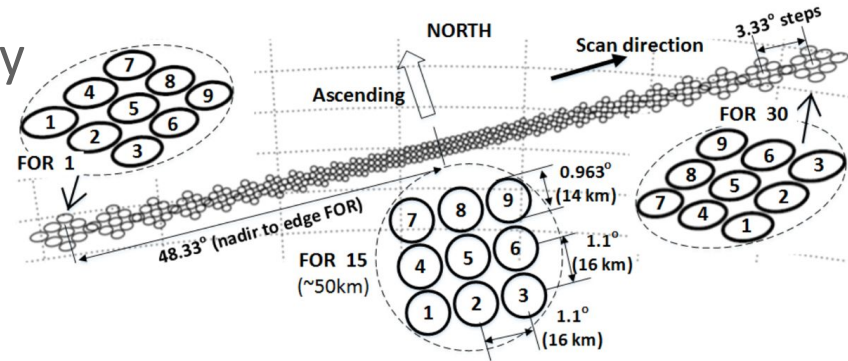
- Well-known Text geometry primitives
- OGC Simple Features
- GeoJSON
- Shapefile
- Various geospatial databases

# Satellite Swath Data

Electromagnetic radiation collected from a specific direction into a solid angle and then measured at a number of intervals of the electromagnetic spectrum

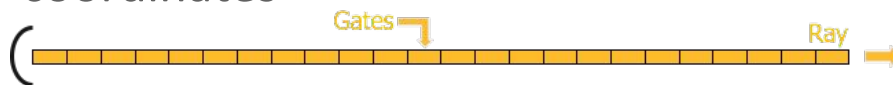
Data collected by instruments on satellites, airplanes, and unmanned aerial systems

Original Instrument Viewing Geometry

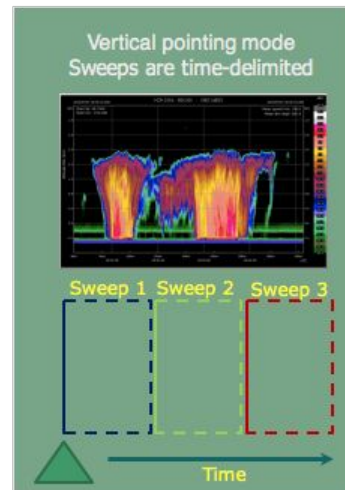
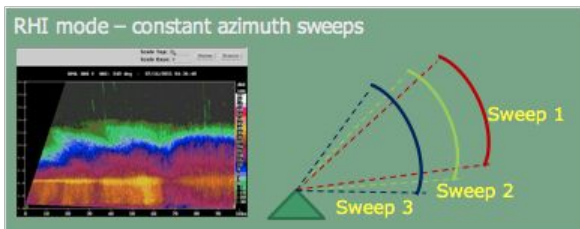
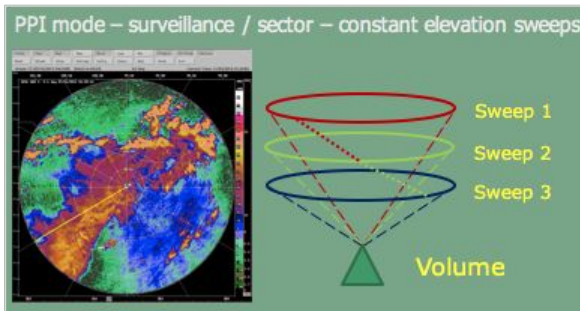


# CF-Radial

Represent data from pulsed instruments – RADARs and LIDARs – in their native polar coordinates



A collection of **GATES** forms a **RAY**.  
A collection of **RAYS** forms a **SWEEP**.  
A collection of **SWEEPS** forms a **VOLUME**.



# Describing the data

**Units:** mandatory for all data variables

**Standard name:** describe the physical quantity a variable represents

**Ancillary Variables:** metadata about individual data values

- E.g., standard error or data quality

**Valid minimum and maximum or Valid range:**

# Units

- Mandatory for all data variables
- Value is a `UDUnits` recognizable string
  - E.g., “degC”, “Pa”, “mbar”, “W m<sup>-2</sup>”, “kg/m<sup>2</sup>/s”,
- Date/Time: “hours since 2018-10-15”

# Standard Names

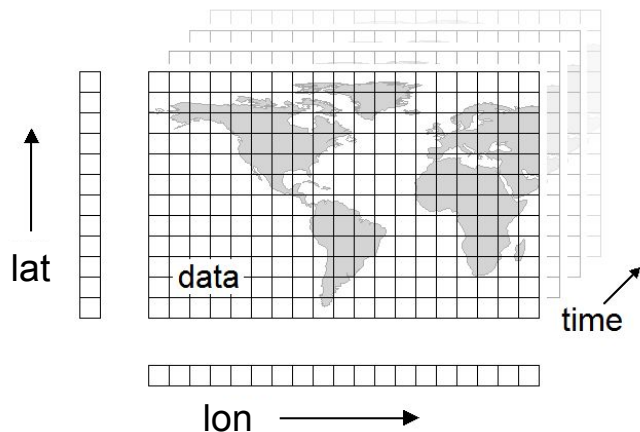
- Describe the physical quantity represented by a data variable
- CF Standard Names consist of
  - Name
  - Canonical Units
  - Definition
- Units must be consistent with standard name and any statistical processing e.g. variance

\* Coming Soon

\*\* Starting development

# Dimensions and Coordinates

- Dimensions establish the index space of data variables
- Coordinates are the independent variables, data the dependent variables



```
netcdf mydataset {  
  dimensions:  
    lon = 19 ;  
    lat = 12 ;  
    time = 4 ;  
  variables:  
    float lat(lat) ;  
    float lon(lon) ;  
    float time(time) ;  
    float temp(time, lat, lon) ;  
    float rh(time, lat, lon) ;  
}
```

# Coordinate Systems

```
netcdf aggExisting.xml {  
  dimensions:  
    y = 228; x = 306; time = 41;  
  variables:  
    int Lambert_Conformal;  
      Lambert_Conformal:grid_mapping_name = "lambert_conformal_conic";  
      Lambert_Conformal:standard_parallel = 25.0;  
      Lambert_Conformal:longitude_of_central_meridian = 265.0;  
      Lambert_Conformal:latitude_of_projection_origin = 25.0;  
  
    double y(y); ... y:standard_name = "projection_y_coordinate" ;  
    double x(x); ... x:standard_name = "projection_x_coordinate" ;  
    double lat(y, x); ...  
    double lon(y, x); ...  
    int time(time); ...  
    float Temperature(time, y, x);  
      Temperature:units = "K";  
      Temperature:long_name = "Temperature @ surface";  
      Temperature:coordinates = "lat lon";  
      Temperature:grid_mapping = "Lambert_Conformal" ;  
}
```

:Conventions = "CF-1.4" ;



# Questions?

CF Conventions: <http://cfconventions.org>

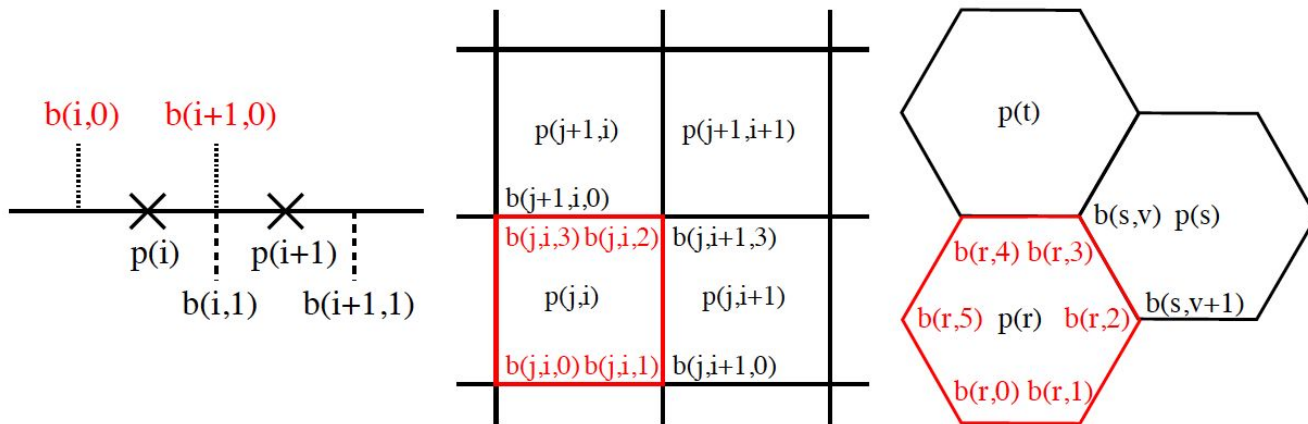
Ethan Davis: [edavis@ucar.edu](mailto:edavis@ucar.edu)

Unidata is one of the University Corporation for Atmospheric Research (UCAR)'s Community Programs (UCP), and is funded primarily by the National Science Foundation (Grant NSF-1344155).



# Cell Bounds

Extent over which a data value is valid



# Other stuff

- Cell Bounds
- Cell Methods
- Climatological statistics
-