Week 3: NetCDF

What is NetCDF?

- Self Describing – All data and meta-data is encapsulated in one file.
- Machine Independent – Data files work on most any platform.
- Direct Access – Efficiently read subsets of larger datasets.
- Appendable – You can quickly add data to old files
- Sharable – One writing process and many reading processes can occur at once.

Who is using NetCDF?

- A better question is who doesn't use it.
- NCDC archives all of their data in NetCDF format.
- FSL heavily uses it
- WRF model
- You should too! :)

NetCDF basics

- DIMENSION
    - An integer value that describes the length
- VARIABLE
    - Values defined along dimensions
- ATTRIBUTE
    - Meta data for variables.
- COORDINATE VARIABLE
    - A variable with the same name as the dimension it is defined along.

NetCDF Data Types

- char (8 bit)
- byte (8 bit)
- short int (16 bit)
- int (32 bit)
- float or real (32 bit) IEEE floating point
- double (32 bit) IEEE floating point

NetCDF Best Practices

- File names should end with “.nc”
- Take care to define your data model beforehand.
- Create a .cdl file first, to avoid creating the file in code
- Add as many attributes as necessary
- Always define the 'units' and 'long_name' attribute

Creating a NetCDF File

- Write a .cdl file to describe your NetCDF file and then use the 'ncgen' command to generate a NetCDF file.
- Create the NetCDF file from your program. This should be generally avoided.
- Copy the .cdl from a previously existing NetCDF file and use its CDL for the new file.

Basic CDL

```
netcdf test {
dimensions:
        recNum = UNLIMITED ;
variables:
        float temperature(recNum) ;
        float latitude(recNum) ;
        float longitude(recNum) ;
}
```

You would generate a netcdf file with ncgen -o test.nc test.cdl

Adding attributes

```
netcdf test {
dimensions:
        recNum = UNLIMITED ;
variables:
        float temperature(recNum) ;
                temperature:long_name = "temperature" ;
                temperature:units = "kelvin" ;
        float latitude(recNum) ;
                latitude:long_name = "latitude" ;
                latitude:units = "degree_north" ;
        float longitude(recNum) ;
                longitude:long_name = "longitude" ;
                longitude:units = "degree_east" ;
}
```

Creating a NetCDF file from Python

```
#!/usr/local/python/bin/python

from Scientific.IO import NetCDF

nc = NetCDF.NetCDFFile(“test.nc”, 'w')

nc.createDimension('recNum', None)

tmpk = nc.createVariable('temperature', Numeric.Float, ('recNum',) )
tmpk.long_name = 'Temperature'
tmpk.units = 'Kelvin'

nc.close()
```

Python NetCDF Interface

```
#!/usr/local/python/bin/python

from Scientific.IO import NetCDF

nc = NetCDF.NetCDFFile(“test.nc”, 'a')

recNum = nc.dimensions[“recNum”]

tmpk = nc.variables[“temperature”]
tmpk[0] = 273.01
tmpk[1] = 300.00

nc.close()
```

Assignment for Next Time

- Take the comma delimited file from the first week and generate a netcdf file of it.