Week 5: Odds and Ends

OpeNDaP
Accessing OpeNDaP servers with Ferret
Numeric Python
F2PY: Calling Fortran modules from Python

OpeNDAP (DODS)

- is a framework that simplifies all aspects of scientific data networking.
- Provides a standard network framework between local datasets and remote data users.
- Working on OpenGIS standards to interoperate with GIS desktop applications

OpeNDaP Sites:

- http://gcmd.gsfc.nasa.gov/Data/portals/dods/
- http://nomads.ncdc.noaa.gov/
- http://dataportal.ucar.edu
- http://cola8.iges.org:9191/dods/

Accessing OpeNDaP from Ferret

- go dods_winds_demo
- go dods_demo
- use "http://www.cdc.noaa.gov/cgi-bin/nph-dods/Datas"
- show data
- shade/L=1 precip

Numeric Python

- Python library that allows the efficient processing of arrays
- Arrays must be composed of values of the same type, either ints, floats, floats32, chars, etc...
- NetCDF variables are Numeric arrays in python

Creating a Numeric Array

```
import Numeric
# Create with values
a = Numeric.array([0,1,2,3,4,5])
# Create using a helper, results in same value as above
a = Numeric.arange(5)
# Create an array full of zeroes
a = Numeric.zeros(5)
```

```
#Sample Python Code
import Numeric

a = Numeric.array([0,1,2,3,4,5,6,7,8,9])
print Numeric.sum(a)
# prints 45
print Numeric.average(a)
# prints 4.5

# Add 10 to each number of array
array2 = a + 10
# Take sin() of each value
array3 = Numeric.sin(a)
# Accumulate an array
array4 = Numeric.add.accumulate(a)
#  Mask
array5 = Numeric.where(a > 1, 1, 0)
```

F2PY: Calling Fortran modules from Python

```
$ cat hello.f
C File hello.f
      subroutine foo (a)
      integer a
      print*, "Hello from Fortran!"
      print*, "a=",a
      end
$ /usr/local/python/bin/f2py -c -m hello hello.f
$ /usr/local/python/bin/python
>>> import hello
>>> hello.foo(4)
```

Scientific Python: ArrayIO

```
from Scientific.IO.ArrayIO import *


t = readFloatArray(“myfile.dat”)
#
# t is now a Numeric array, which we can go lots of things!
```

Practical Example 1

- We have three files
  - One file with an array of values
  - Two files with lats and lons in them
- We want to combine these files into a NetCDF file. Here's the python code to do it!

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

# Python Imports
from Scientific.IO.ArrayIO import *
import Numeric
from Scientific.IO.NetCDF import *

# Load up the precip data from file
precip = readFloatArray("precip.dat")
# Get the shape of this precip array
shp = Numeric.shape(precip)

# Read lat and lon values from file
latitudes = readFloatArray("lats.dat")
longitudes = readFloatArray("lons.dat")

# Create NetCDF file
nc = NetCDFFile("precip.nc", 'w'  )
# Create two dimensions with the size of the precip data
nc.createDimension("x", shp[1])
nc.createDimension("y", shp[0])

# Create three variables
lat = nc.createVariable("latitude", Numeric.Float, ("y", "x") )
lon = nc.createVariable("longitude", Numeric.Float, ("y", "x") )
prec = nc.createVariable("precip", Numeric.Float, ("y", "x") )

# Assign in.
lat.assignValue(latitudes)
lon.assignValue(longitudes)
prec.assignValue(precip)

# Close file, we are done!
nc.close()
```

For next time?

- Compare NCEP stage4 precip data against MM5 forecasted precipitation...