PyNGL & PyNIO
Geoscience Visualization &
Data IO Modules

SciPy ‘08

Dave Brown
National Center for Atmospheric Research
Boulder, CO
Topics

• What are PyNGL and PyNIO?
• Quick summary of PyNGL graphics
• PyNIO interface to NumPy
• Data model
• New features
• Current status - new open source beta
• Future directions
Brief History
NCAR
Graphics
HLUs
NCL
NIO
PyNGL
PyNIO

2000-present: 30+
NCL Workshops

1995: NCL Conference

1992: NCARG Conference

Original Fortran Late 60s

GKS/CGM/Fortran 77/UNIX conversion 1980s

PyNIO and PyNGL July 2006 (Numeric 24.x/NumPy 1.0)

PyNIO 2005

PyNIO 2005

PyNGL 2004

PyHLU 2003

GSUN - Late 90s

NCL Mid 90s

C interface Mid 90s

HLUs Mid 90s

NCAR Graphics

Late 60s

GKS/CGM/Fortran 77/UNIX conversion

1980s

NCAR Graphics

1992: NCARG Conference

1995: NCL Conference

1992: NCARG Conference

1995: NCL Conference

2000-present: 30+
NCL Workshops

2000-present: 30+
NCL Workshops

2000-present: 30+
NCL Workshops

2000-present: 30+
NCL Workshops

2000-present: 30+
NCL Workshops
PyNGL

• **Python NCL Graphics Library**
• NCL (NCAR Command Language) is a visualization and data-processing scripting language
• PyNGL is Python interface to the same core graphics as NCL.
• Contours, XY plots, vectors, streamlines, geographical maps, many map projections, overlays, primitives, etc.
• Regular and irregular rectangular grids; triangular meshes for scalar fields.
• New very accurate high-res global map boundary data set (includes provincial/state boundaries for China, India, Brazil, Australia, Canada, Mexico, US + US counties)
Currents at 600 m depth POP displaced pole grid

vcGlyphStyle = 'curlyvector'
Texas Mesonet real time WRF forecast animation

http://mesonet.tamu.edu/scoop-cgi/ogc/wrf_ncl
PyNIO

• A multi-format IO module
• Modelled after Konrad Hinson’s Scientific.IO.NetCDF
• A unified NetCDF-like view of all formats
• Reads:
  – NetCDF, NetCDF 4 Classic, GRIB 1 & 2, HDF 4 (SDS), HDFEOS 2 (Grid and Swath)
• Writes:
  – NetCDF, HDF 4
• Now a stand-alone package separate from PyNGL
import Nio
f = Nio.open_file("nio-ex5.nc")
print f

Signature:

open_file(filepath, mode='r', options=None,
          history='', format='')

Regular selection:

tmp = f.variables['var']
a = tmp[3,::,1,40:80:2,:20]
Nio file: nio-ex5.nc

global attributes:
dimensions:
  lon = 120
  lat = 61
  lev = 9
  time = 6
variables:
  float lon [ lon ]
    long_name : longitude
    units : degrees_east
    axis : X
  float lat [ lat ]
    long_name : latitude
    units : degrees_north
    axis : Y
  float lev [ lev ]
    long_name : Isobaric surface
    units : Pa
    axis : Z
  integer time [ time ]
    long_name : Forecast offset from initial time
    units : hours since 11/15/2006 12:00
    axis : T
  float tmp [ time, lev, lat, lon ]
    long_name : Temperature
    units : K
    _FillValue : -999
  float hgt [ time, lev, lat, lon ]
    long_name : Geopotential height
    units : gpm
    _FillValue : -999
MaskedArray support

- **MaskedArrayMode options**
  - MaskedIfFillAtt (default)
  - MaskedAlways
  - MaskedNever
  - MaskedIfFillAttAndValue
  - MaskedExplicit
    - ExplicitFillValues, MaskBelowValue, MaskAboveValue
Extended Selection

• Thanks to Juerg Schmidli
• Coordinate and index space selection using named dimensions
• Reorder dimensions (transpose) using dimension names
• Scalar, vector, and slice selection
• Basic bi-linear interpolation built-in to the selection mechanism
• Syntax is a whitespace-separated string inside square brackets
## Extended selection examples

Given:

```python
tmp = f.variables['tmp']
print f.variables['lev']
[ 1000.  5000.  15000.  30000.  45000.  60000.  75000.  90000.  97500.]
```

```python
tmp['time|i0 lev|1000,100000 lat|60 lon|100:120']
# Positional syntax:
tmp['i0 1000,100000 60 100:120']
# Reordering dimensions:
tmp['time|i0 lat|60 lon|100,120 lev|:']
# Inserting variables in the selection string
tmp['time|i0 lat|60 lon|%f,%f lev|:' % (minlon,maxlon)]
# Interpolating to 10 equally spaced levels:
tmp['time|i0 lat|60 lon|100,120 lev|0:100k:10ki']
# Reversing the levels:
tmp['time|i0 lat|60 lon|100,120 lev|100k:0:-10ki']
# Selection using an auxiliary multidimensional coordinate variable:
tmp['time|0,3 lev|hgt|1500,3000 lat|50,60 lon|237:252']
```
Current status

• PyNGL & PyNIO betas 1.3.0b1 released 8/18
• OSI-compliant open source license
• Binaries available for:
  – Various flavors of Linux, Mac OS X,
  – Other Unix systems, 32 and 64 bit
• Source distribution available
• Extensive documentation
• PyNGL and PyNIO now separate downloads
• http://www.pyngl.ucar.edu/download.html
Future plans

• More IO formats and functionality:
  – NetCDF 4, HDF/HDFEOS 5
  – Support for groups, structures, etc.
  – Datasets aggregated from multiple files (NcML)
• A new display model for PyNGL based on Cairo.
• Vector plotting for triangular mesh data.
Documentation

• PyNGL
  – http://www.pyngl.ucar.edu
  – Tutorial, examples with line-by-line descriptions, function reference

• PyNIO
  – http://www.pyngl.ucar.edu/Nio.html
  – Reference