INTEGRATED OCEAN OBSERVING SYSTEM

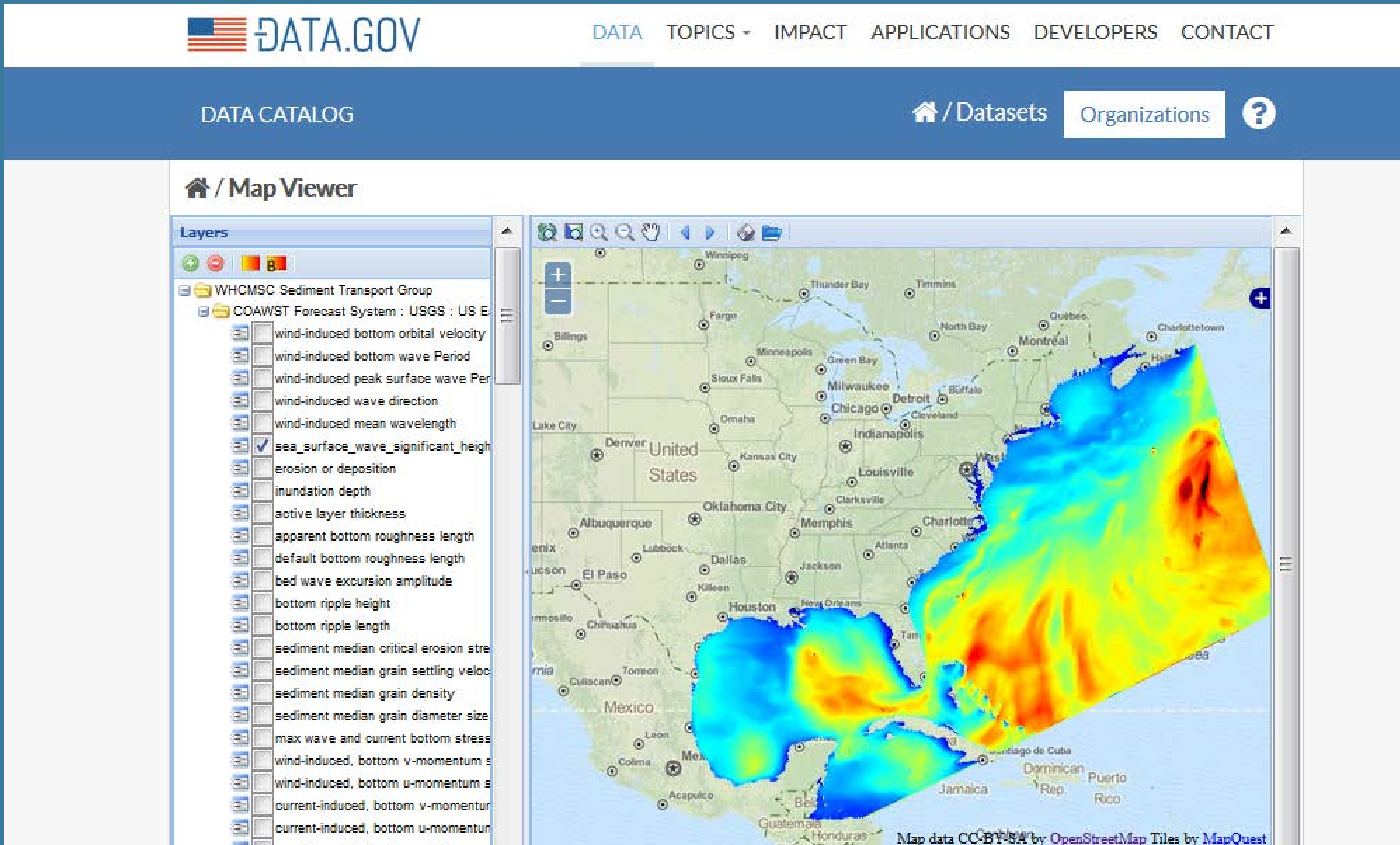
nganju@gam: /usgs/data0/bbleh/spring2012

```
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$ ls
his_bbleh_0001.nc  his_bbleh_0019.nc  his_bbleh_0037.nc  his_bbleh_0055.nc
his_bbleh_0002.nc  his_bbleh_0020.nc  his_bbleh_0038.nc  his_bbleh_0056.nc
his_bbleh_0003.nc  his_bbleh_0021.nc  his_bbleh_0039.nc  his_bbleh_0057.nc
his_bbleh_0004.nc  his_bbleh_0022.nc  his_bbleh_0040.nc  his_bbleh_0058.nc
his_bbleh_0005.nc  his_bbleh_0023.nc  his_bbleh_0041.nc  his_bbleh_0059.nc
his_bbleh_0006.nc  his_bbleh_0024.nc  his_bbleh_0042.nc  his_bbleh_0060.nc
his_bbleh_0007.nc  his_bbleh_0025.nc  his_bbleh_0043.nc  his_bbleh_0061.nc
his_bbleh_0008.nc  his_bbleh_0026.nc  his_bbleh_0044.nc  his_bbleh_0062.nc
his_bbleh_0009.nc  his_bbleh_0027.nc  his_bbleh_0045.nc  his_bbleh_0063.nc
his_bbleh_0010.nc  his_bbleh_0028.nc  his_bbleh_0046.nc  his_bbleh_0064.nc
his_bbleh_0011.nc  his_bbleh_0029.nc  his_bbleh_0047.nc  his_bbleh_0065.nc
his_bbleh_0012.nc  his_bbleh_0030.nc  his_bbleh_0048.nc  his_bbleh_0066.nc
his_bbleh_0013.nc  his_bbleh_0031.nc  his_bbleh_0049.nc  his_bbleh_0067.nc
his_bbleh_0014.nc  his_bbleh_0032.nc  his_bbleh_0050.nc  his_bbleh_0068.nc
his_bbleh_0015.nc  his_bbleh_0033.nc  his_bbleh_0051.nc  his_bbleh_0069.nc
his_bbleh_0016.nc  his_bbleh_0034.nc  his_bbleh_0052.nc  his_bbleh_0070.nc
his_bbleh_0017.nc  his_bbleh_0035.nc  his_bbleh_0053.nc
his_bbleh_0018.nc  his_bbleh_0036.nc  his_bbleh_0054.nc
nganju@gam:/usgs/data0/bbleh/spring2012$
```

nganju@gam: /usgs/data0/bbleh/spring2012

```
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$
nganju@gam:/usgs/data0/bbleh/spring2012$ ls
00_dir_roms.ncml    his_bbleh_0018.nc  his_bbleh_0036.nc  his_bbleh_0054.nc
his_bbleh_0001.nc  his_bbleh_0019.nc  his_bbleh_0037.nc  his_bbleh_0055.nc
his_bbleh_0002.nc  his_bbleh_0020.nc  his_bbleh_0038.nc  his_bbleh_0056.nc
his_bbleh_0003.nc  his_bbleh_0021.nc  his_bbleh_0039.nc  his_bbleh_0057.nc
his_bbleh_0004.nc  his_bbleh_0022.nc  his_bbleh_0040.nc  his_bbleh_0058.nc
his_bbleh_0005.nc  his_bbleh_0023.nc  his_bbleh_0041.nc  his_bbleh_0059.nc
his_bbleh_0006.nc  his_bbleh_0024.nc  his_bbleh_0042.nc  his_bbleh_0060.nc
his_bbleh_0007.nc  his_bbleh_0025.nc  his_bbleh_0043.nc  his_bbleh_0061.nc
his_bbleh_0008.nc  his_bbleh_0026.nc  his_bbleh_0044.nc  his_bbleh_0062.nc
his_bbleh_0009.nc  his_bbleh_0027.nc  his_bbleh_0045.nc  his_bbleh_0063.nc
his_bbleh_0010.nc  his_bbleh_0028.nc  his_bbleh_0046.nc  his_bbleh_0064.nc
his_bbleh_0011.nc  his_bbleh_0029.nc  his_bbleh_0047.nc  his_bbleh_0065.nc
his_bbleh_0012.nc  his_bbleh_0030.nc  his_bbleh_0048.nc  his_bbleh_0066.nc
his_bbleh_0013.nc  his_bbleh_0031.nc  his_bbleh_0049.nc  his_bbleh_0067.nc
his_bbleh_0014.nc  his_bbleh_0032.nc  his_bbleh_0050.nc  his_bbleh_0068.nc
his_bbleh_0015.nc  his_bbleh_0033.nc  his_bbleh_0051.nc  his_bbleh_0069.nc
his_bbleh_0016.nc  his_bbleh_0034.nc  his_bbleh_0052.nc  his_bbleh_0070.nc
his_bbleh_0017.nc  his_bbleh_0035.nc  his_bbleh_0053.nc
nganju@gam:/usgs/data0/bbleh/spring2012$
```

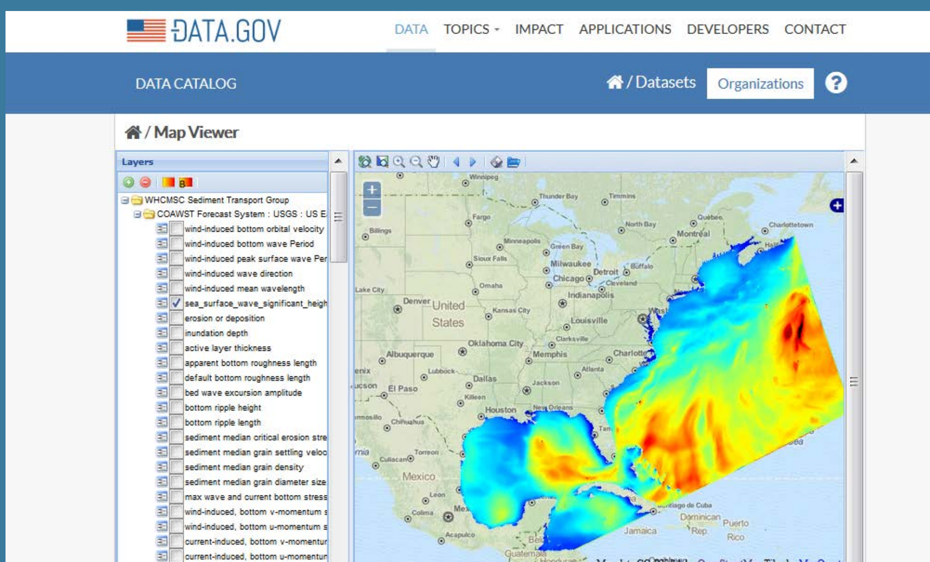
```xml
<attribute name="title" type="string"
 value="COAWST Hindcast:Barnegat Bay:ADCIRC tides,Real rivers,Plume,Lowpass Espresso bdry,NAM,n
<attribute name="summary" type="String"
 value="Barnegat Bay run driven by ADCIRC tides on the boundaries, realistic river forcing with

<attribute name="project" type="String" value="CMG_Portal"/>
<attribute name="Conventions" value="CF-1.4, SGRID-0.1"/>

<variable name="zeta">
 <attribute name="standard_name" type="String" value="water_surface_height_above_reference_datum"/>
 <attribute name="grid" value="grid"/>
 <attribute name="location" value="face"/>
</variable>
<variable name="grid" type="int">
 <attribute name="cf_role" value="grid_topology"/>
 <attribute name="topology_dimension" type="int" value="2"/>
 <attribute name="node_dimensions" value="xi_psi eta_psi"/>
 <attribute name="face_dimensions"
  value="xi_rho: xi_psi (padding: both) eta_rho: eta_psi (padding: both)"/>
 <attribute name="edge1_dimensions" value="xi_u: xi_psi eta_u: eta_psi (padding: both)"/>
 <attribute name="edge2_dimensions" value="xi_v: xi_psi (padding: both) eta_v: eta_psi"/>
 <attribute name="node_coordinates" value="lon_psi lat_psi"/>
 <attribute name="face_coordinates" value="lon_rho lat_rho"/>
 <attribute name="edge1_coordinates" value="lon_u lat_u"/>
 <attribute name="edge2_coordinates" value="lon_v lat_v"/>
 <attribute name="vertical_dimensions" value="s_rho: s_w (padding: none)"/>
</variable>

<!--aggregation specification (here aggregate files like "his_bbleh_0001.nc, his_bbleh_0002.nc, ..."
-->
<aggregation dimName="ocean_time" type="joinExisting">
 <scan location="." regExp=".*his_bb.*_[0-9]{4}\.nc$"/>
</aggregation>
</netcdf>
```

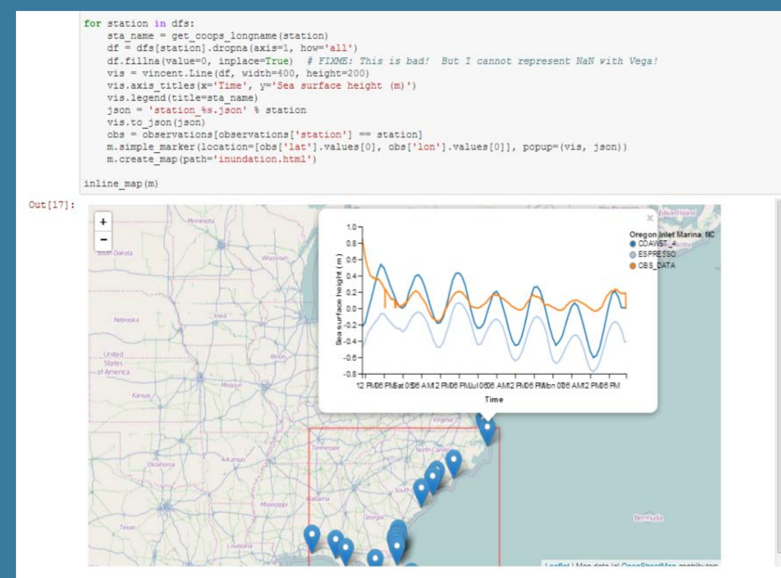

DATA.GOV
DATA
TOPICS
IMPACT
APPLICATIONS
DEVELOPERS
CONTACT
DATA CATALOG
/ Datasets
Organizations
/ Map Viewer
Layers
WHCMSC Sediment Transport Group
COAWST Forecast System : USGS : US E
wind-induced bottom orbital velocity
wind-induced bottom wave Period
wind-induced peak surface wave Per
wind-induced wave direction
wind-induced mean wavelength
sea_surface_wave_significant_heigh
erosion or deposition
inundation depth
active layer thickness
apparent bottom roughness length
default bottom roughness length
bed wave excursion amplitude
bottom ripple height
bottom ripple length
sediment median critical erosion stre
sediment median grain settling veloc
sediment median grain density
sediment median grain diameter size
max wave and current bottom stress
wind-induced, bottom v-momentum s
wind-induced, bottom u-momentum s
current-induced, bottom v-momentur
current-induced, bottom u-momentur
United States
Mexico
Map data CC-BY-SA by OpenStreetMap Tiles by MapQuest

INTEGRATED OCEAN OBSERVING SYSTEM

Exploiting IOOS: A Distributed, Standards-Based Framework and Software Stack for Searching, Accessing, Analyzing and Visualizing Met-Ocean Data

Rich Signell (USGS-CMG)
Filipe Fernandes (SECOORA)
Kyle Wilcox (Axiom Data Science)
Andrew Yan (USGS-CIDA)



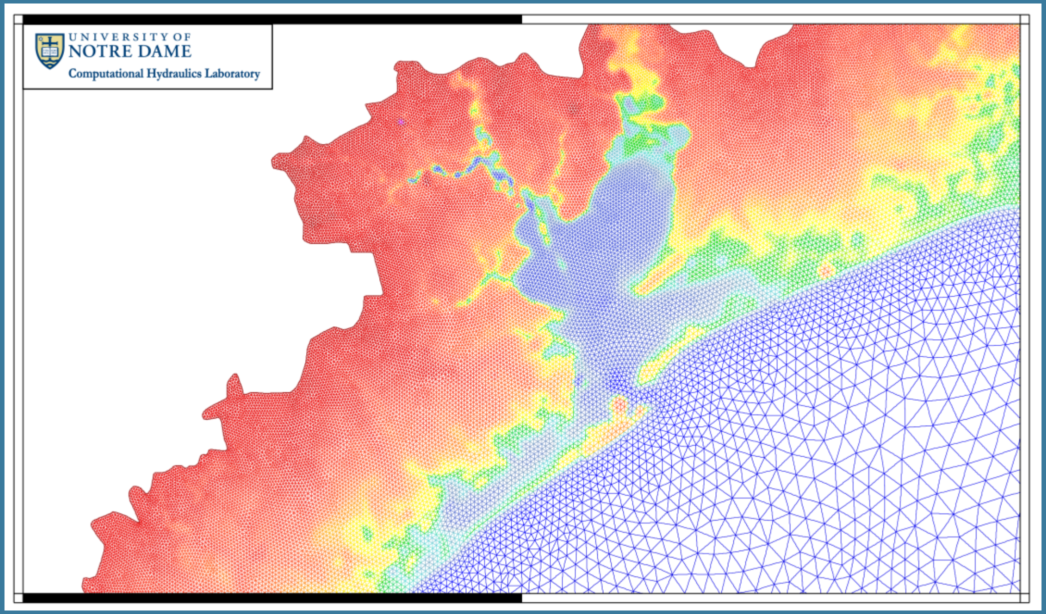


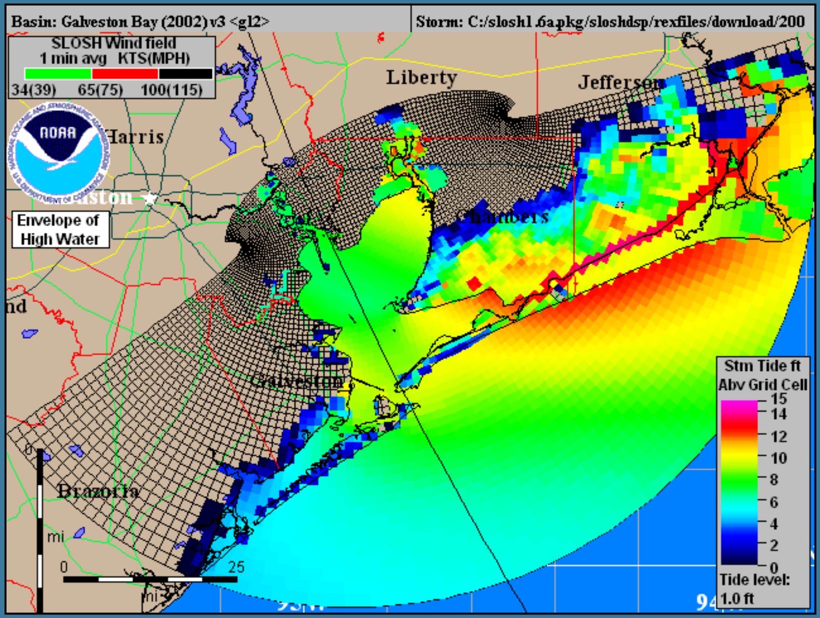
Objectives

- Set up a standards-based framework for easy and efficient access to insitu and ocean model data
- Provide a high-level search and browse web interface for program datasets, for scientists, end users and program managers
- Contribute to a growing standardized data search, access and use infrastructure that supports all geoscience

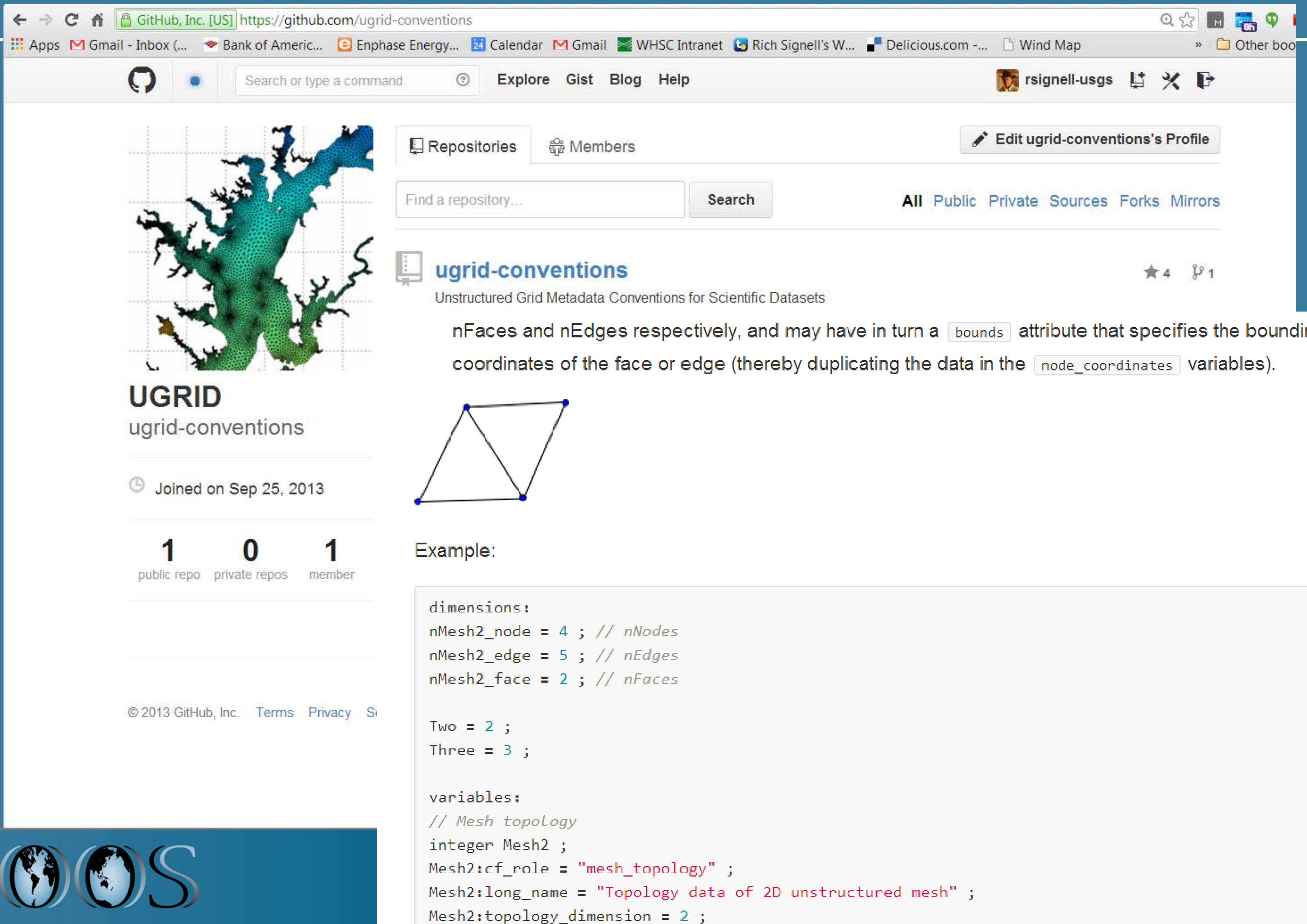
Why not just use ERDDAP?

- Two reasons:
- 1. Unstructured grid models
- 2. Curvilinear grid models





UGRID Conventions on GitHub



SGRID Conventions: github/sgrid

```
        v:grid = "grid" ; // SGRID attribute
        v:location = "edge2" ; // SGRID attribute
    float zeta(ocean_time, eta_rho, xi_rho) ;
        zeta:long_name = "free-surface" ;
        zeta:units = "meter" ;
        zeta:time = "ocean_time" ;
        zeta:coordinates = "lat_rho lon_rho" ;
        zeta:grid = "grid" ; // SGRID attribute
        zeta:location = "face" ; // SGRID attribute
// SGRID variable
    int grid ;
        grid:cf_role = grid_topology
        grid:topology_dimension = 2 ;
        grid:node_dimensions = "xi_psi eta_psi" ;
        grid:face_dimensions = "xi_rho: xi_psi (padding: both) eta_rho: eta_psi (padding: both)" ;
        grid:edge1_dimensions = "xi_u: xi_psi eta_u: eta_psi (padding: both)" ;
        grid:edge2_dimensions = "xi_v: xi_psi (padding: both) eta_v: eta_psi" ;
        grid:node_coordinates = "lon_psi lat_psi" ;
        grid:face_coordinates = "lon_rho lat_rho" ;
        grid:edge1_coordinates = "lon_u lat_u" ;
        grid:edge2_coordinates = "lon_v lat_v" ;
        grid:vertical_dimensions = "s_rho: s_w (padding: none)" ;

// global attributes:
        :Conventions = "CF-1.0" ;
        :title = "ROMS/TOMS 2.2 - Adria02 Uber Run" ;
}
```

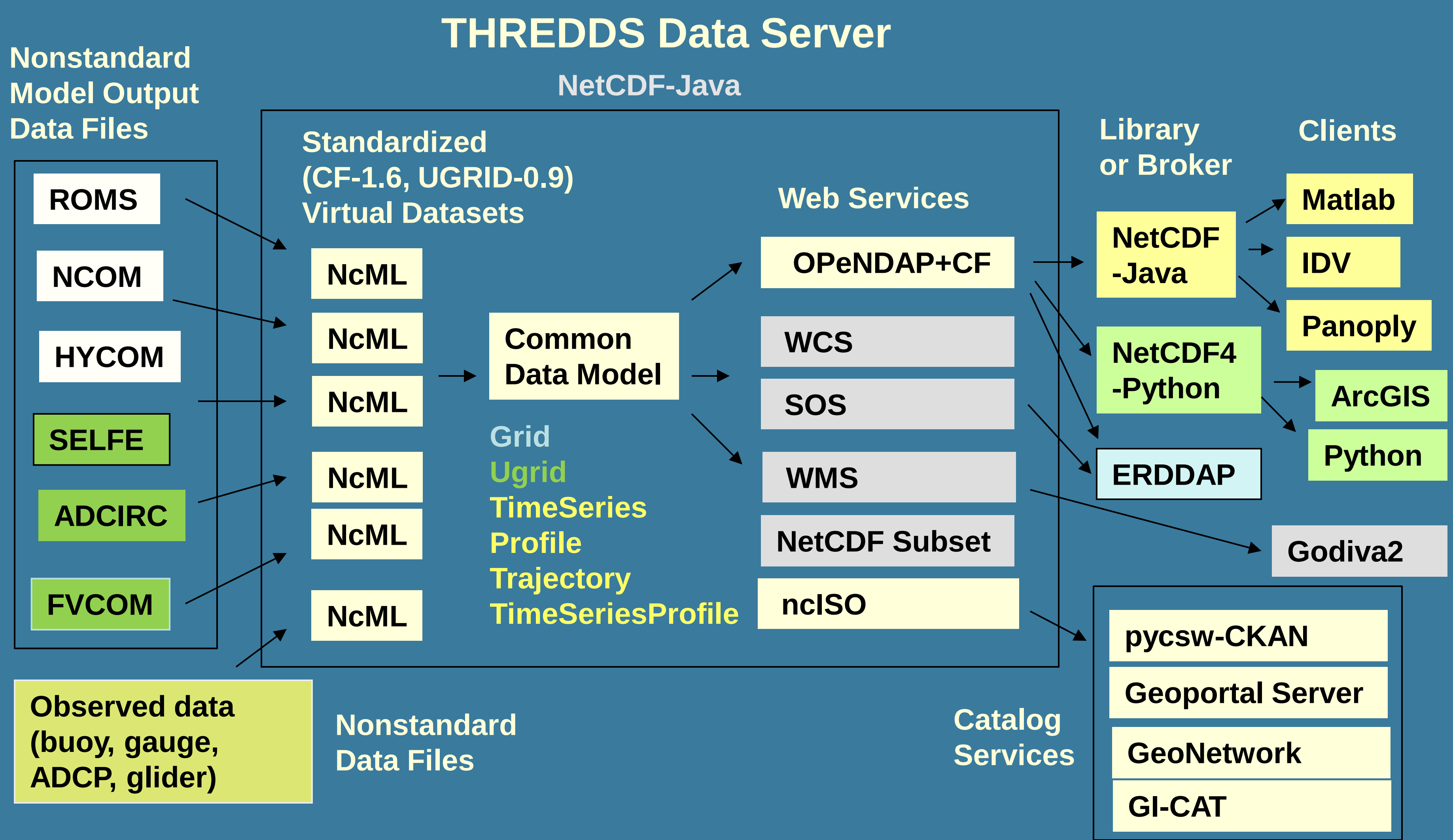
WRF (ARW version)

The WRF-ARW also uses a C-grid. In this case the resulting data is described as a 3D grid topology. It could also be interpreted as a 2D layered model just like Delft3D and ROMS (or those models could also result in 3D grid topologies). A layered approach is more consistent with the UGRID approach that distinguishes between the unstructured mesh in the horizontal and the structured (layered) grid in the vertical, but the 3D grid offers a more symmetric treatment of the staggered U, V and W components.

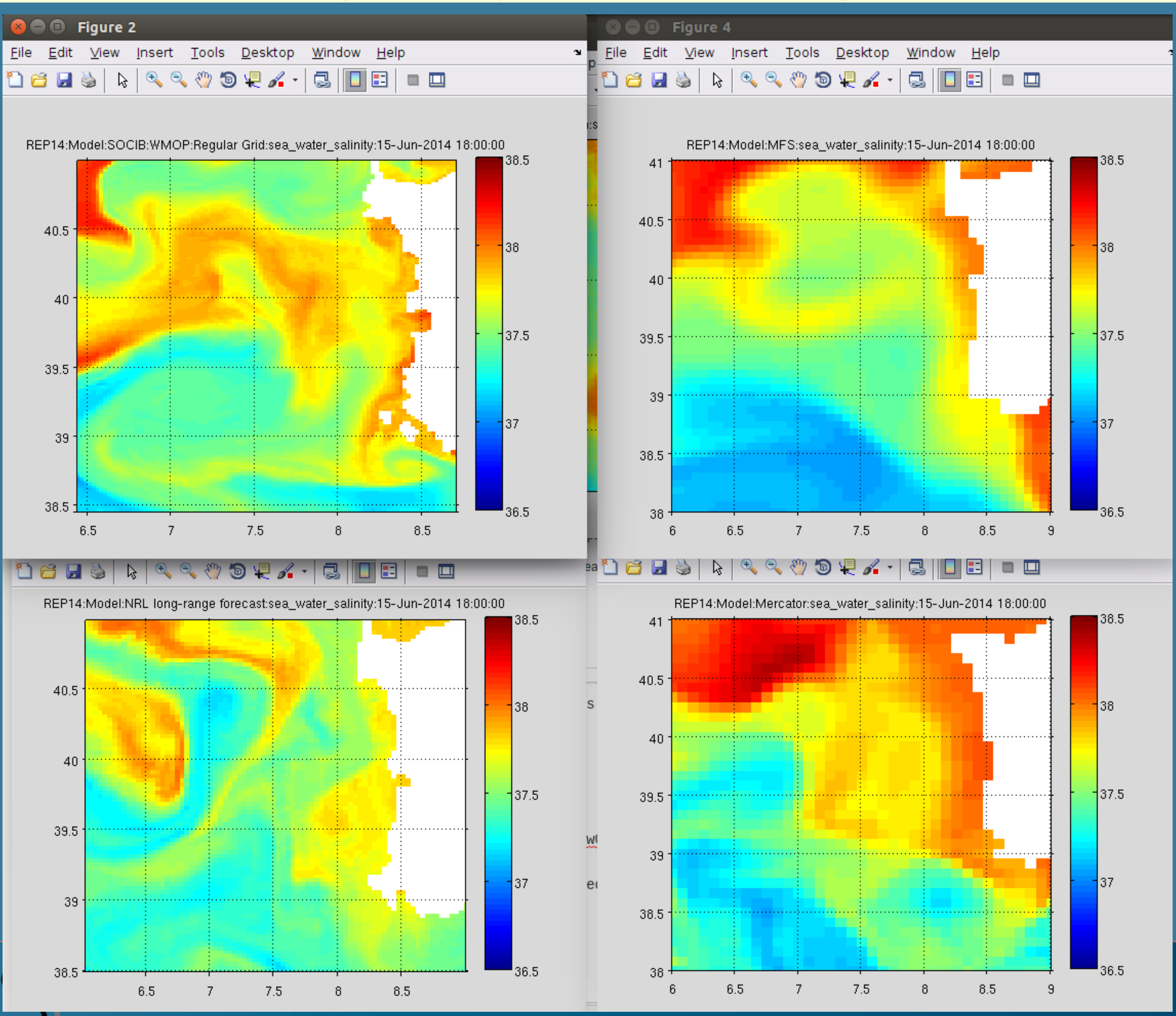
It might be interesting to verify the result for WRF-NMM since that model uses an E-grid, but I couldn't find an example file.

```
netcdf wrfout_v2_Lambert {
dimensions:
    Time = UNLIMITED ; // (13 currently)
```

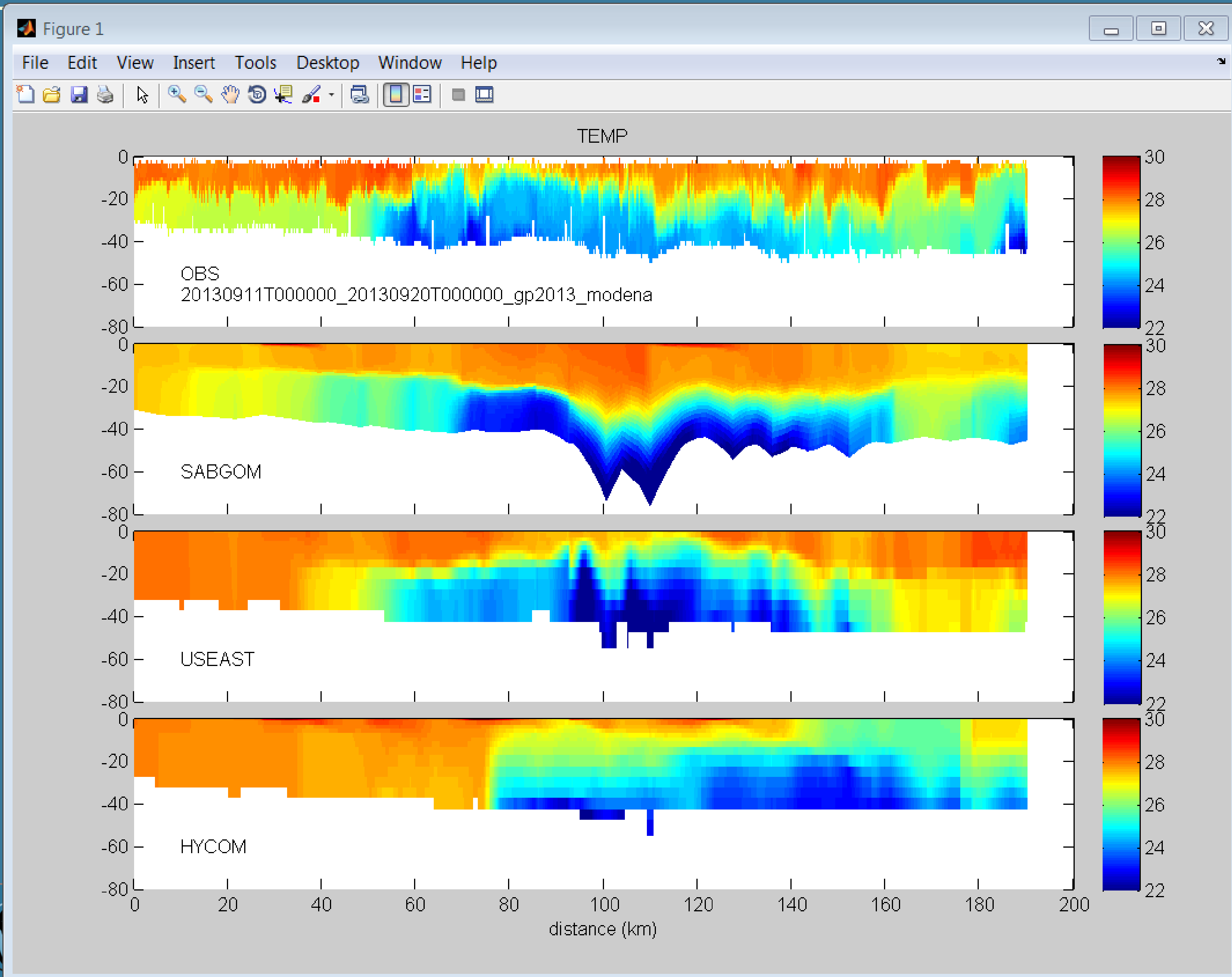
IOOS Model Data Interoperability Design



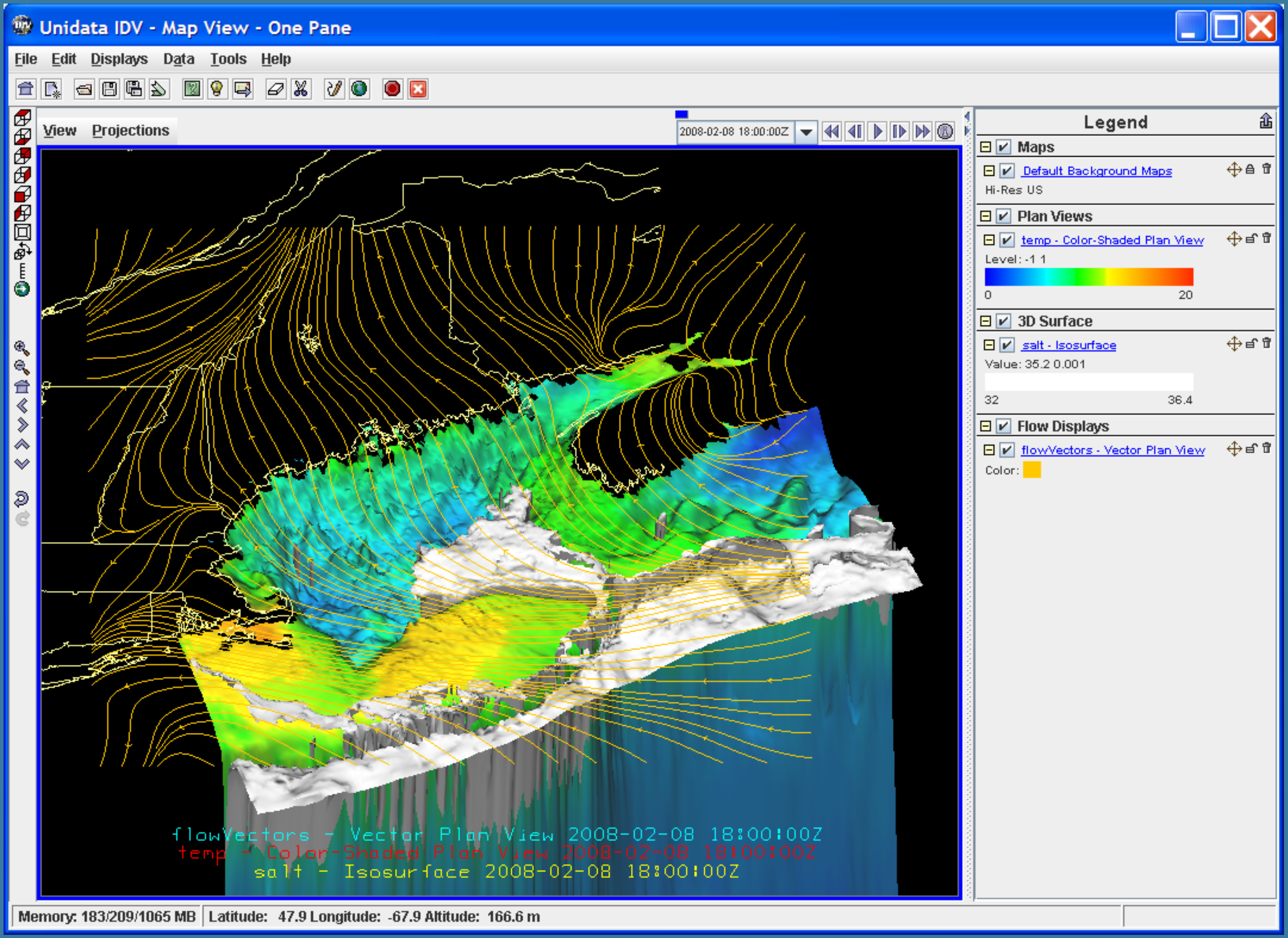
Interoperable Model Comparison in Matlab (using nctoolbox)



compare_secoora_model_sections.m

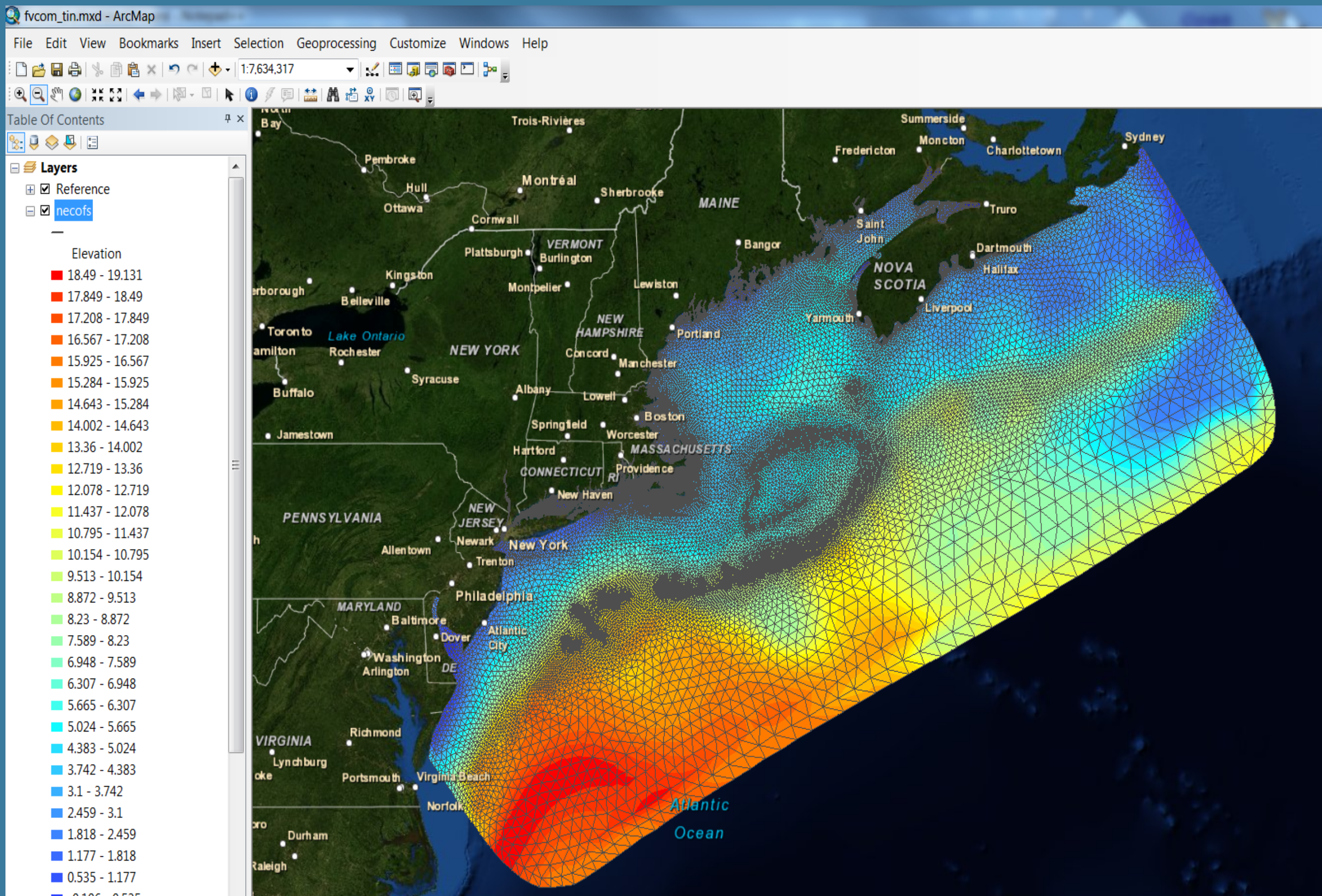


3D visualization of data with IDV

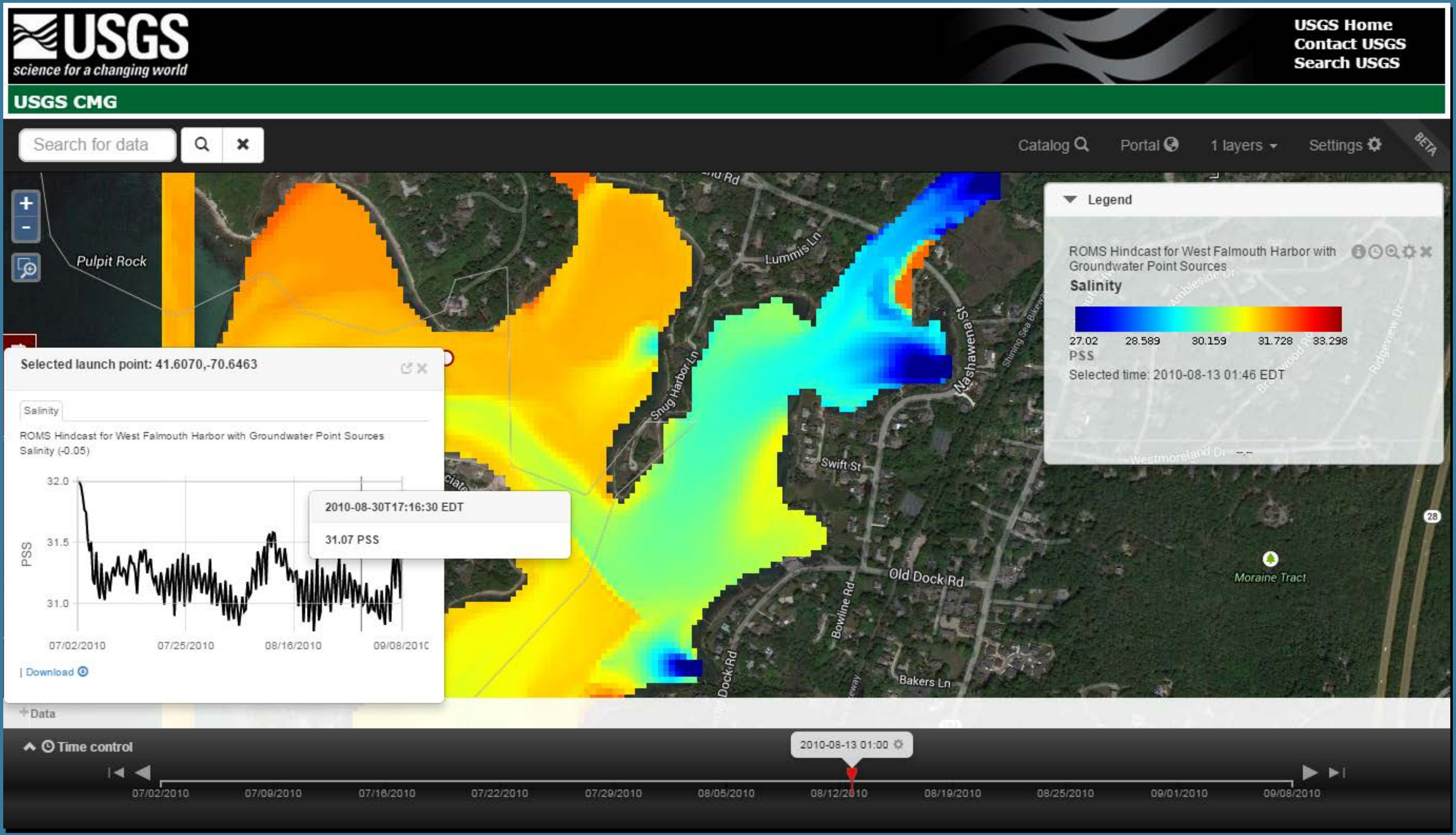


IOOS

NECOFS Access in ArcGIS (using the dap2arc python toolbox)



USGS CMG Portal



NetCDF Point Subset Service

https://wakari.io/sharing/bundle/rsignell/NCSS_pandas_demo

Gmail - Inbox (... Bank of Americ... Enphase Energy... Calendar WHSC Intranet Rich Signell's W... Emoji cheat she... IOOS Catalog R... Delicious.com -... Wind Map » Othe

Wakari NCSS_pandas_demo

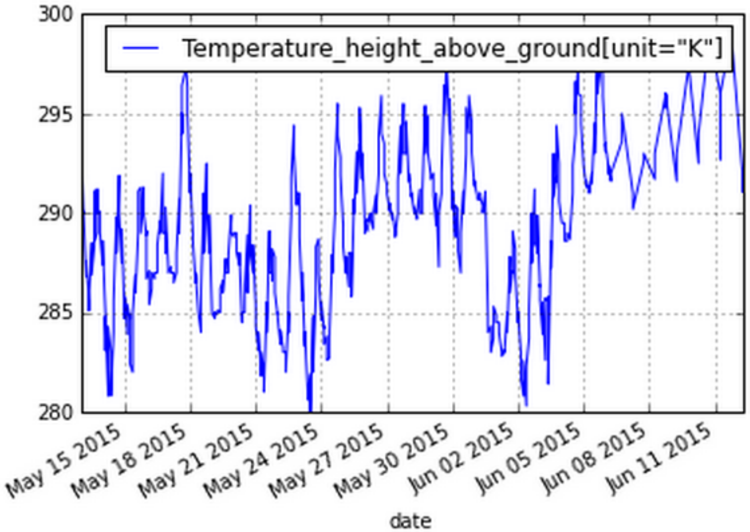
View Other Bundles by rsignell Download Entire Bundle Download This Notebook Run/Edit this Notebook

In [1]:
```
import pandas as pd
```

In [2]:
```
#GFS 0.5degree Global
url='http://thredds.ucar.edu/thredds/ncss/grib/NCEP/GFS/Global_0p5deg/Best?var=Temperature_height_above_ground&latitude=41.5&longitude=-71.5&time_start=2015-05-13T00%3A00%3A00Z&time_end=2015-06-12T06%3A00%3A00Z&vertCoord=&accept=csv'
```

In [3]:
```
df = pd.read_csv(url,index_col=0,parse_dates=True,usecols=[0,4])
```

In [4]:
```
df.plot();
```



IP[y]: Notebook   Iris test 2   Last saved: Mar 28 10:17 AM

File   Edit   View   Insert   Cell   Kernel   Help

Code   Cell Toolbar: None

In [12]:
```
# DAP URL: 30 year East Coast wave hindcast (Wave Watch 3 driven by CFSR Winds)
cubes = iris.load('http://geoport.whoi.edu/thredds/dodsC/fmrc/NCEP/ww3/cfsr/4m/best');
```

In [13]:
```
print cubes
```

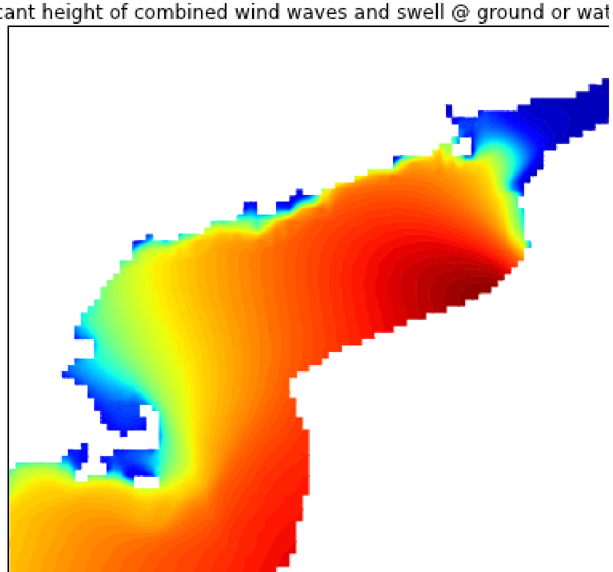
```
0: Significant height of combined wind waves and swell @ Ground or water surface / m (time: 90584; latitude: 481; longitude: 586)
1: u-component of wind @ Ground or water surface / m/s (time: 90096; latitude: 481; longitude: 586)
2: v-component of wind @ Ground or water surface / m/s (time: 90096; latitude: 481; longitude: 586)
3: Primary wave direction (degree true) @ Ground or water surface / unknown (time: 90584; latitude: 481; longitude: 586)
4: Primary wave mean period @ Ground or water surface / s (time: 90584; latitude: 481; longitude: 586)
```

In [14]:
```
hsig=cubes[0]
```

In [15]:
```
slice=hsig.extract(iris.Constraint(time=tval(hsig,'1989-05-07 21:00'),
    longitude=lambda cell: -71.5 < cell < -65.0,
    latitude=lambda cell: 39.5 < cell < 46.0))
```

In [16]:
```
# make the plot
figure(figsize=(10,10))
qplt.contourf(slice,100);
```

Significant height of combined wind waves and swell @ ground or wat



scitools.org.uk/iris/

Inbox (... Bank of Americ... Enphase Energy... Gmail Calendar » Other

Iris

Fork me on GitHub

Home | Download | Documentation | Community | Governance | Code | Scitools

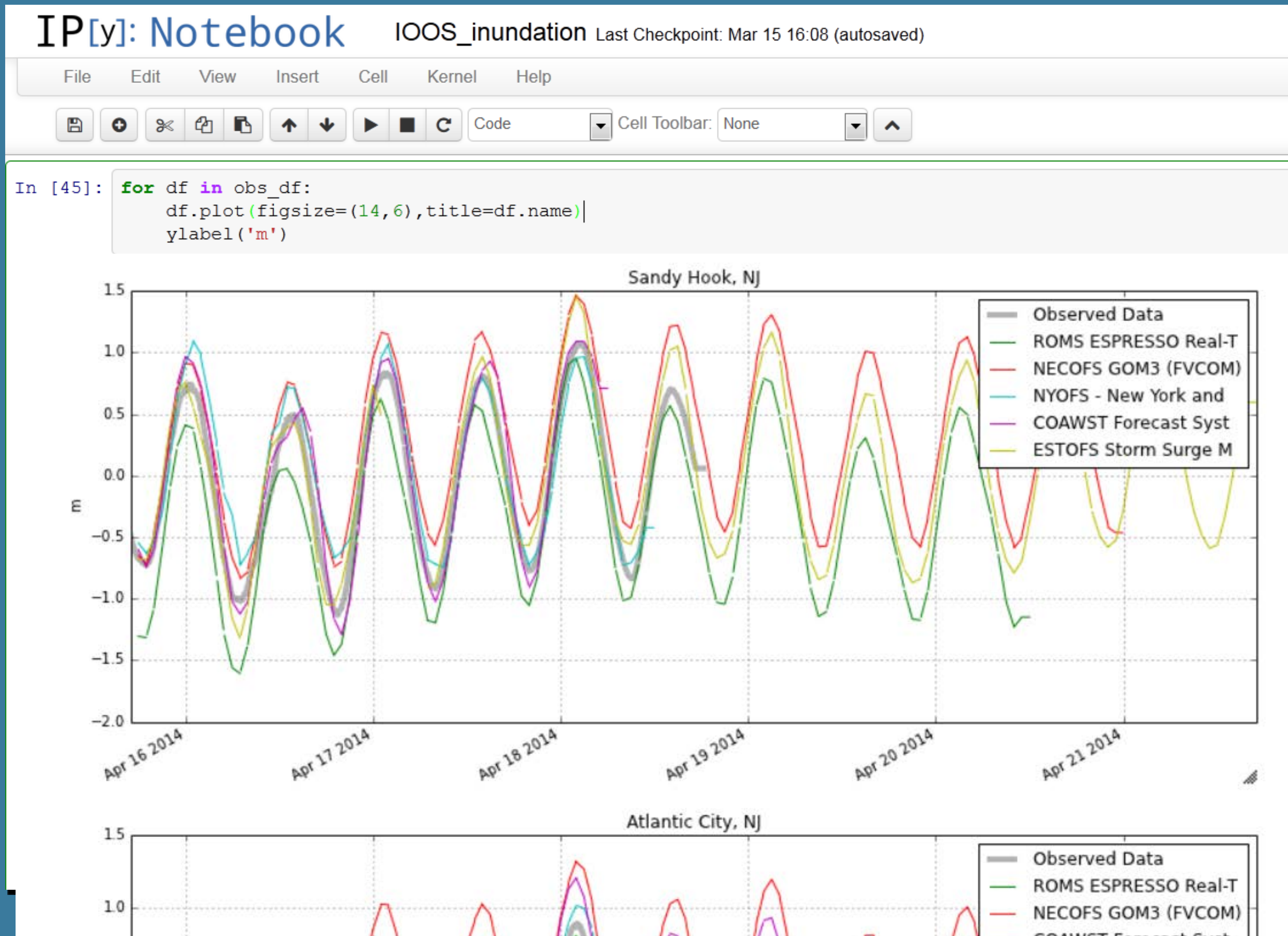
A Python library for Meteorology and Climatology

The Iris library implements a data model to create a data abstraction layer which isolates analysis and visualisation code from data format specifics. The data model we have chosen is the CF Data Model. The implementation of this model we have called an Iris Cube.

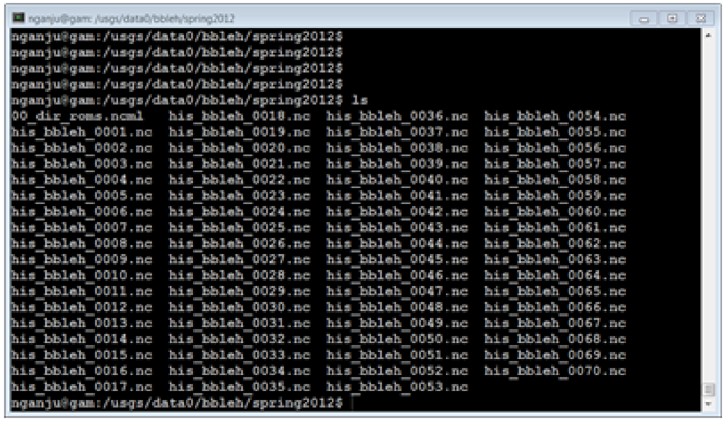
Iris currently supports read/write access to a range of data formats, including (CF-)netCDF, GRIB, and PP; fundamental data manipulation operations, such as arithmetic, interpolation, and statistics; and a range of integrated plotting options.

Iris is published under an LGPLv3 licence.

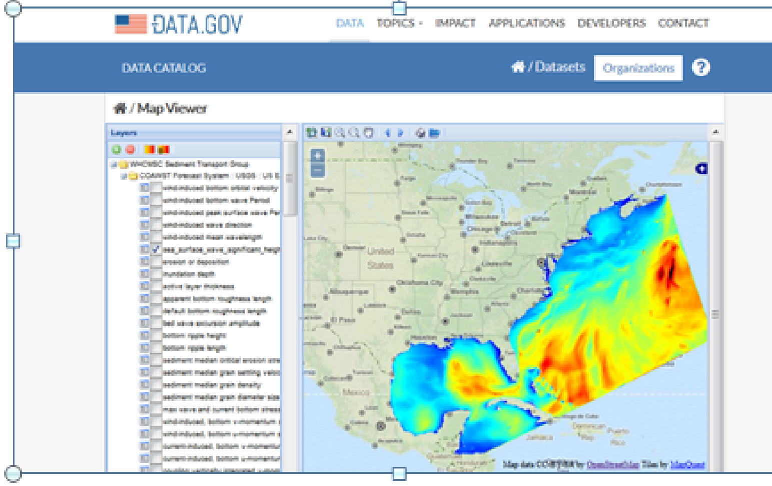
Automated model comparison



How do we get from a piles of files like this…



To catalog-driven portals and applications like this…



Provider:

1. Writes pile of netcdf files to a directory
2. Generates a yaml file describing the data and indicating whether data should be picked up by a specific portal
3. Run python script to generate NcML file that aggregates and makes dataset CF compliant and UGRID/SGRID compliant if appropriate

Application/User:

1. CSW query to discover datasets matching keywords and geospatial/temporal extents
2. Extract WMS and OPeNDAP service endpoints from the dataset metadata records
3. Create browse graphics via WMS map requests
4. Extract data via OPeNDAP data requests
5. Utilize CF/UGRID/SGRID conventions for ineroperability

Automated process:

1. Specified thredds servers are crawled on a schedule by a python script, which extracts ISO metadata
2. creates a WMS service by registering the dataset with sci-wms
3. Injecting the WMS endpoint into the ISO metadata
4. Datasets with matching “project” tags are picked up by the portal

Getting your model results connected

- Find someone with a THREDDS Data Server or install your own
- Drop your files in a directory, and add an NcML file that starts with “00_dir” (e.g. “00_dir_roms.ncml”) to aggregate, standardize and describe the dataset: Sample ROMS NcML file
- If you want your data to end up in the portal, add “CMG_Portal” to the “project” attribute: `<attribute name=“project”value=“CMG_Portal”/>`
- If you want your datasets to be discoverable, submit a PR on list of thredds catalogs being scanned on github
- Full instructions on the USGS-CMG Portal Github Wiki

A few problems… Packaging

- Ipython notebooks are a great way to document model skill assessment workflows (Filipe will talk about this)
- But python environment uses a lot of tricky packages. How to make this easy for folks?
- Conda and binstar to the rescue! (Filipe will talk about this)

A few problems… WMS

- ncWMS works great for CF compliant data
- Unstructured grids are not CF compliant.
- Staggered grids are not CF compliant.
- ncWMS doesn’t work for unstructured grid data (FVCOM, ADCIRC, SELFE), and doesn’t work for staggered grid velocities in models like ROMS, WRF and Delft3D
- sci-wms to the rescue, using UGRID conventions for unstructured grid (pyugrid), and SGRID conventions for staggered grid (pysgrid). (Kyle will talk about this)

Key Infrastructure Components

- Common data models for "feature types" (structured, staggered and unstructured grids, time series, profiles, swaths) (Unidata CDM, UGRID, SGRID)
- Standard web data services for delivering these common data model "feature types" (OPeNDAP/CF/UGRID/SGRID, WMS, SOS, WFS, ERDDAP/tabledap, ERDDAP/griddap)
- Standard catalog services for the metadata (OGC CSW, OpenSearch)
- Tools for easy delivery of data in standard services
- Tools for easy search, access and use of data in standard services (in all major environments: Python, ArcGIS, R, Matlab, JavaScript)

Infrastructure Benefits

- What are the benefits?
  - Less time wasted messing with data, more time spent on science
  - More skill assessment of models
  - More usage and more appropriate useage of model results
  - Faster feedback to modelers => improved models
  - Better science, better models =>better world