Position Paper for the ESRI Ocean Summit: Ocean Model Data Interoperability for ArcGIS Users

Richard P. Signell, US Geological Survey, Woods Hole Coastal and Marine Science Center
384 Woods Hole Rd, Woods Hole, MA 02543
rsignell@usgs.gov - 508-457-2229

Data from ocean simulations and forecasts play an increasingly important role in coastal and marine science, emergency response, spatial planning, and environmental management. The majority of these data are available via the OPeNDAP web service, yet currently there are no OPeNDAP access tools for ArcGIS users. To better support the marine community, ESRI should seek to provide an effective interface to the OPeNDAP web service by: (1) enabling access by using existing Python modules; (2) taking advantage of CF Metadata Conventions to enable standard workflows; and (3) improving methods for bringing ocean data into native ArcGIS data structures.

The OPeNDAP service is popular for three main reasons: (1) it's free software, easy to install and configure, (2) it allows effective variable extraction and subsetting, and (3) it is capable of representing the data structures commonly used in oceanography. Oceanographic datasets are typically 4D (3D space + time) and often have curvilinear or triangular mesh coordinate systems in the horizontal plane and stretched terrain-following coordinates in the vertical plane. OPeNDAP is currently the only service that can serve these complex data structures effectively: the OGC Web Coverage Service is currently limited to uniformly-spaced data (e.g. raster data). OPeNDAP usage is growing across the geosciences, not just in oceanography.

One straightforward and effective way to enable OPeNDAP access in ArcGIS is to make use of the Python interface available in ArcGIS 10 and use one of the Python modules that support OPeNDAP access. One of the most popular modules is NetCDF4-Python, which conveniently allows a remote OPeNDAP dataset to be accessed just as if it was a local NetCDF file. Another plus is that it is built on top of the Unidata NetCDF C library, ensuring that further developments of NetCDF and OPeNDAP will be available to Python users. This creates a low-barrier to entry for both providers and consumers of ocean model data: providers can simply place their NetCDF3, NetCDF4, HDF4, HDF5, GRIB, or GRIB2 files on a file system and have their datasets served via OPeNDAP (thanks to powerful and easy to install OPeNDAP data servers), and consumers can work with these remote datasets as virtual NetCDF files, extracting just the data they need. The NetCDF4-Python module with OPeNDAP support can be a tricky module to build, but it is part of the Enthought Python Distribution, providing an easy way for ArcGIS users to use this functionality.

Efficient extraction is important, since these datasets are often massive. Daily forecast models in the US Integrated Ocean Observing System (IOOS) produce datasets in the range of 10-200 GigaBytes, and a recent 30 year ocean hindcast commissioned for the Massachusetts State Ocean Plan totaled 33 TeraBytes. The Fifth IPCC assessment (due out in 2014) will rely on an estimated 3.3 PetaBytes of available data. [1]

While OPeNDAP is flexible and efficient, there is very little required structure or metadata, which can make development of standardized workflows and applications challenging. The Climate and Forecast

(CF) Conventions have been developed so that common data structures (curvilinear grids, unstructured grids, swaths, trajectories, collections of profiles) are treated uniformly, fostering interoperable use. Currently the only methods that take advantage of these CF Conventions are contained in Unidata's NetCDF-Java Library, which has allowed a number of Java applications to be developed that rely on NetCDF-Java to deliver common operations (e.g. ability to subset 3D temperature data from an ocean model within a specified bounding box and temporal range, returning georeferenced data in the native model data structure). What we need now is a CF library for Python, so that ArcGIS users can have similar methods available to them. Without them, they will have to figure out on a case-by-case basis the complicated calculation of vertical coordinate computations, how to extract data in a specified geospatial range, and other needless operations. There are several ongoing efforts to enable CF library for Python users. Unidata's effort to develop a libCF library in C would be particularly useful in that many languages could then wrap the functionality (e.g. R, Julia, and others in addition to Python). ESRI could help accelerate this development with minimal developer and/or resource development.

Finally, it would be very useful if there was an easier way to get data from ocean models into ArcGIS data structures. Raster or uniformly-spaced data is easy to get from Python into Arc, thanks to the ESRI NumPyArrayToRaster function, but there is no equivalent function for TIN, and there is no direct support for structured, non-uniformly spaced data (e.g. curvilinear grid models). For triangular grid ocean model data, it is currently necessary to use the LandXMLtoTIN function to write binary data from Python to XML on disk, and then load into ArcGIS. The TIN construct does not allow the ability to have multiple fields, so the only choice is to write and load each data variable with the same longitude, latitude arrays and connectivity arrays, hugely inefficient.

With an OPeNDAP interface, a library for working with common data structures, and the ability to represent these structures internally, ArcGIS users would be able to work with ocean model (and other gridded data commonly used in the geosciences) much more easily, making research, planning and management in the ocean much more efficient and effective. Because each step of building the interface adds additional value, the interface could be rolled out immediately, with immediate value.

The first step, using a Python-based interface to OPeNDAP modeling data has already been used successfully. Using ArcGIS10 and NetCDF4-Python for OPeNDAP access, we created a collection of Python tools that allowed the Massachusetts Coastal Zone Management personnel to directly access their 33 TeraByte triangular mesh raw ocean output via OPeNDAP, compute their own products, and ingest them to ArcGIS in a variety of forms. Depending on what type of additional analysis they wanted to do using the ArcGIS environment, at the end of the processing chain, they could: (1) preserve the data on the triangular grid by creating a TIN via LandXMLtoTIN; (2) interpolate to a regular grid using SCIPY routines and create a raster using NumPyArrayToRaster; or (3) convert the products to Shapefiles, using PYSHP. They can now respond dynamically to emerging coastal planning, management and emergency response needs, instead of anticipating the GIS products required.