SCI-WMS: A Python based Web Mapping Service for Visualizing Geospatial Data

Brandon A. Mayer, Brian McKenna, Alexander Crosby and Kelly Knee

Abstract SCI-WMS [1] is an open-source python implementation of an OGC WMS service [2] for qualitatively assessing society-critical atmospheric and oceanagraphic applications including forecasting, risk assessment, model comparison and algorithmic/parameter selection. The modular cross-platform implementation of SCI-WMS allows the service to keep pace with the rapid developments in the geospatial data science community and to produce visualizations for numerous types of model outputs with transparent support for both structured and unstructured geo-referenced topologies. This article outlines the implementation and technology stack for visualizing geospatial Climatological and Forecasting (CF) data using SCI-WMS and details the deployment of SCI-WMS for visualizing model data and simulations within the scope of the U.S. IOOS Coastal and Ocean Modeling Testbed project [3].

1 Motivation

Due to the explosion in the amount of atmospheric, oceanagraphic, climate and weather data, either recorded *in-situ* or generated by modeling, inference or prediction algorithms, it is no longer feasible for a single institution to host and maintain a centralized datastore. Modern data management has been shifting hosting and maintenance responsibilities of large datasets to multiple participating institutions unified by a catalogue service to provide a single, unified view of the distributed data to end users and analysts.

Brandon A. Mayer
Brown University, 182 Hope Street Providence RI 02906 e-mail: brandon_mayer@brown.edu

Brian McKenna, Kelly Knee
RPS-ASA e-mail: {BMcKenna,KKnee}@asascience.com

Alexander Crosby
5 River Rd. Suite 1 Cos Cob, CT 06807 e-mail: alexc@oceanweather.com

A common practice is for data producing organizations to host their own data which is exposed to the catalogue service via a particular communication protocol. The institution responsible for maintaining the catalogue then provides a unified view of the aggregated dataset to end users by compiling meta data associated with the participating organizations and registered data. Such federated datasets may span petabytes of geospatial information, be composed of millions of files in different formats generated and hosted by vastly different systems located across the globe. Additionally, the catalogue can grow or shrink as new datasets or participants are added and removed from the federation. By interacting with the catalogue, the end user (such as an analyst) can navigate and search through the aggregated meta data and interact with particular data of interest, agnostic to distributed nature of the database.

While a decentralized approach to data management offers many advantages such as robustness to failure (a failure at any one organization only effects the data associated with that organization), data reduction and analysis tools have been slower to adapt to the new framework. For example, there are an abundance of applications for visualizing cartographic data on a single computer or clusters. However, many such programs are designed to deal with data saved in a particular *local* file format, requiring analysts to download a local copy datasets, potentially reformat the data into the appropriate container before processing and analysis may begin. This increases the projects cost in terms of bandwidth usage, time and storage. Additionally, coupling decentralized hosting and local analytical workflows introduces the risk that different analysts working with identical local copies of data obtained from the same federation may use different local programs to generate incompatible visualizations or comparisons. Normalizing these results introduces another potential point of error and likewise increases the costs of analysis in terms of time and accuracy. Furthermore, the local analysis paradigm violates the ethos of distributed data storage. The goal of a federated datastore is to minimize data redundency. Therefore, data processing and visualization software must adapt to the imposed distributed memory model to likewise minimize redundency.

SCI-WMS is a web service designed to solve many of the aforementioned problems. By maintaining a list of web accessible endpoints, SCI-WMS is able to transparently produce consistent visualizations of federated data. While SCI-WMS implements the OGC WMS protocol, it is augmented with services for interacting with standard meta-data catalogues such as CSW [4], allowing SCI-WMS to autonomously track dynamic federations.

SCI-WMS uses NcML (NetCDF Markup Language) to implement a data abstraction layer. This allows data hosting agencies to maintain their own environments and file formats which are exposed to SCI-WMS without writing custom i/o software or replicating and reformating data file to a standard format. Additionally, SCI-WMS is CF-Compliant [5], offering consistent views of endpoints which adhering to the CF-Metadata conventions embedded in the NcML file.

SCI-WMS embraces distributed database principles while saving costs to data analysis projects by providing a simple interface for end users to generate consistent visualizations of federated data. Perhaps more importantly, the introduction of a single web-based visualization service for distributed datasets ensures qualitative assessments and conclusions are made on equal footing regardless of analyst or data origin and is one of the first services for visualizing geospatial data associated with an unstructured topology.

2 SCI-WMS

2.1 Overview

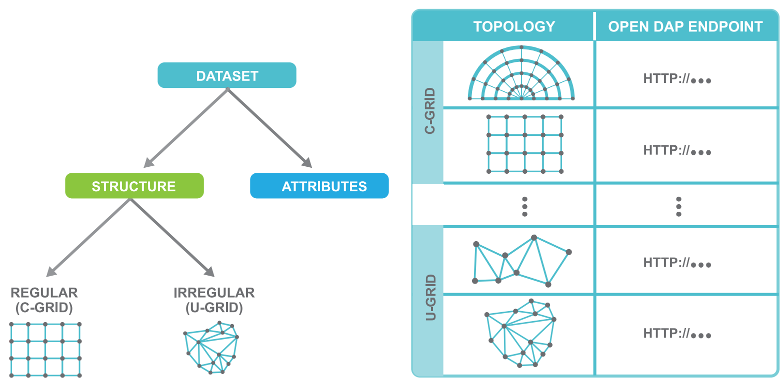
SCI-WMS is an open-source implementation of the Open Geospatial Consortium's (OGC) Web Map Service (WMS) standard which specifies an HTTP interface for generating rasterized visualizations of geospatial data [2]. More specifically, WMS defines a RESTful API which responds to standardized HTTP GET requests from a web client. Typically, a WMS request specifies a data layer and region of interest with optional meta data such as rendering style parameters. A WMS response may be either meta data providing information about datasets registered with the service or a rasterized visualization (typically a png or other another image format) for display by the front end client.

SCI-WMS is implemented in Python using the Django [6] web framework and standard cross-platform numerical software, NumPy [7] and Matplotlib [8] for generating visual content. Additionally, the open-source python implementation provides a cross-platform WMS solution which can leverage the suite of tools developed by the geospatial data analysis community, such as pyugrid [9], to maintain pace with the latest geospatial software and standards developments including unstructured grid support and CF-UGRID Compliance [10].

While the OGC WMS specification standardizes the client-server communication protocol, WMS are left free to design and implement the system that will fulfill WMS requests and responses. Vital to the efficiency of SCI-WMS is the abstraction of a registered dataset into structure and attributes as shown in figure **??** and detailed in subsequent sections. SCI-WMS maintains a local cache for efficient storage and processing of spatial neighborhoods and subsets with respect to data structure. For storage efficiency, attributes are not replicated locally but referenced via OGC compliant web-services and a database of structure-endpoint pairs is maintained as visualized in figure 1b. Because geospatial WMS requests are commonly restricted to a subset of the Earth's surface, SCI-WMS uses the topology cache to compute the subset of model data needed to fulfill each request prior to retrieving the appropriate attributes (which are external to the SCI-WMS server and accessed via HTTP). Furthermore, by classifying topologies as either regular or irregular, efficient algorithms and data structures are exploited to optimize the computation of relevant model data subsets.

2.2 Dataset Abstractions

Visualization tools often decompose data into structure and attributes [11]. Due to the discrete nature of digital computers, any continuous function to be estimated must be sampled and measured at a discrete set of points. However, rendering a visualization typically requires knowledge of the values between the samples to produce a perceptually continuous images from arbitrary viewpoints. Structure encapsulates both the locations and connectivity relations onto which attributes are superimposed where connectivity serves to constrain the interpolation problem. Note that some authors further decompose structure further into topology and geometry [12], however, in the context of this research, topology is synonymous with the structure abstraction. Figure 1a outlines the data model abstraction adopted by SCI-WMS. A dataset is composed of attributes and an associated structure, further classified as a regular or irregular topology, known as c-grid and u-grid topologies in SCI-WMS documentation.



(a) (b)

Fig. 1: (a) Decomposition hierarchy of the data model. A dataset submitted to SCI-WMS is decomposed into attributes and structure which is further classified as a regular (c-grid) or irregular (u-grid) topologies. (b) SCI-WMS topology and endpoint data store. Topolgies are stored locally in implicit form for c-grid or binary R-Tree databases for u-grid topologies.

2.3 c-grid Topology

c-grid topologies refer geo-referenced locations and geometries that can be defined analytically, e.g., rectilinear or curvilinear grids. Storing c-grid topologies amount to storing the closed form formula. Algorithms for processing c-grids such as finding nearest neighbors or finding points that fall withing a polygonal subset are computed directly using the implicit c-grid representation, incurring computational overhead.

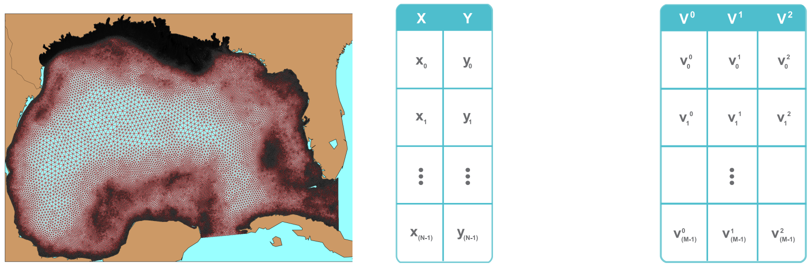
2.4 u-grid Topolgy

A u-grid topology is defined as a topology that is not regular, i.e., do not admit a closed form representation. u-grid topologies offer the highest level of flexibility from a visualization and modeling standpoint as a higher sampling frequency may be used regions of interest and less samples allocated for low-impact areas.

Any NCML file which exposes the topology of an externally hosted dataset according to the CF-UGRID sepcification can be processed by SCI-WMS. According the CF-UGRID standard, a topology is always embedded on the real line, in the real plane or in space with the locations of a u-grid topology's vertices exposed as an array of coordinates in the model's ambient space. The connectivity of vertices is a $M \times D$ where each element of the array is an index into the vertex array. The dimension of a topology defines the atomic spatial element created by the connectivity list.. In the CF-UGRID terminology, a topology with dimension 0 defines a set of disconnected points (no connectivity) called ***nodes***, a 1D topology consists of lines or curved boundaries known as ***edges***, a 2D topology is a set of planes or surfaces enclosed by a set of edges (e.g. triangulation) called ***faces***, and a 3D topology specifies a volume enclosed by a set of faces called volumes ***volumes***.

For example, figure 2b visualizes a list of a topology's vertex locations as an array of coordinates in the real plane. Topolgy connectivity is specified by a list of indicies into the vertex array interpreted anticlockwise. Figure 2c visualizes a 2D connectivity list for the vertex array in figure 2b where each entry is an indexes an element of the vertex list: $v_n^{\cdot} \in \{0,\ldots,N-1\}$. The rows specify triangles in an anticlockwise manner: triangle m is specified by linking the coordinates associated with vertex lists by connecting v_m^0 to v_m^1 to v_m^2 and back to v_m^0.

For example, figure 2a shows a u-grid 2D topology used by a climate model for simulating hurricane conditions in the Gulf of Mexico. Vertices represent locations where attributes computed and the connectivity is shown as the red lines connecting the vertices. The u-grid topology allows the modelers to save storage and computational resources by sparsely modeling areas where accuracy can be sacrificed, such as the center of the Gulf, while allowing for higher sampling rates where accuracy is paramount, such as densely populated coastal areas. However as shown, u-grid topologies require explicit enumeration of verticies and connectivity, and as such, require spatially-aware data structures for optimal storage and processing.



(a) (b) (c)

Fig. 2: Examples of an u-grid topology and CF-UGRIDcompliant vertex and connectivity arrays. (a) The location of N vertices in the real plane. (b) A 2D topology with M triangles. Each triangle represented as a row in the table referencing locations as indices into the vertex array.

2.5 Local Topology Cache

Attributes are defined as numerical quantities that are associated with the topology. For example, an atmospheric model may estimate wind direction at the vertices of a triangulated 2D topology or may specify air temperature at the centroid of cell volumes specified by a 3D topology. Additionally, attributes maintain their own dimensionallity. An attribute specifying temperature or sea-surface-height for example would consist of an array of scalars while wind directions are vector valued and tensor valued attributes are also possible.

2.6 Distributed Memory Model

Following these conventions, SCI-WMS decomposes an externally hosted dataset into a structure (topology), defined as a geo-referenced spatial set of locations and connections, and the underlying data attribute layer.

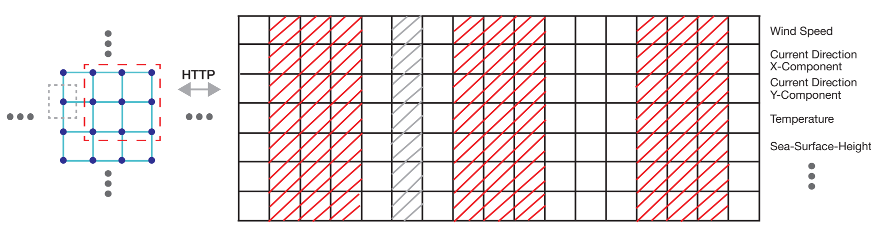


Fig. 3: SCI-WMS distributed memory model.

For example, atmospheric and oceanagrphic models typically define a fixed topology covering a particular spatial extent of the earth. A model will extimate attributes of interest such as sea-surface-height, wind or current magnitudes and directions. The topology of the model encapsulates the positions and connectivity of the dataset for which attributes are associated. Visualizations of datasets are typically restricted so some region of interest, a subset of the available topology and a single attribute such as current direction. It is therefore paramount to the efficiency of visualization software to represent topologies in such a way as to optimize topology storage and reduction algorithms to facilitate efficient attribute retrieval.

To this end, when an dataset endpoint is submitted to SCI-WMS, the topology of the underlying endpoint is stored locally to SCI-WMS and a database of topology-endpoint associations are maintained as visualized in Figure 1b.

3 Deploying SCI-WMS for the U.S. IOOS COMT Testbed

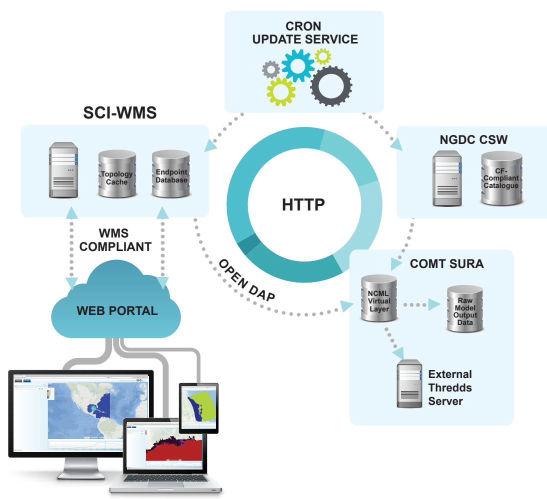


Fig. 4: Overview of the SCI-WMS deployment for the U.S. IOOS COMT project. SCI-WMS updates its topology and endpoint database via a nightly service which queries CF-Compliant datasets cataloged by NGDC. Model data is hosted on an external web server exposed by an NcML facade as a single NetCDF data structure accessable to SCI-WMS via OPeNDAP. SCI-WMS responds to http clients interfacing through a custom built web portal.

While SCI-WMS is a general software solution for geospatial visualization, it is a key component in realizing the U.S. IOOS COMT mission, facilitating qualitative model comparisons and aggregation through a unified visualization framework. Figure 4 outlines the cyberinfrastructre behind the deployment of SCI-WMS for the COMT project.

The U.S. Integrated Ocean Observing System (IOOS) Coastal and Ocean Modeling Testbed (COMT) was formed to unify otherwise disparate entities in government, academia and industry to leverage the proliferation of oceanagraphic data and modeling techniques to combat natural and man-made coastal stressors by accelerating the turnaround from research and development to operational application of society-critical applications including: forecasting, model comparison, model skill assessment, and algorithmic/parameterization improvements [3]. Key to the U.S. IOOS COMT mission is an extensible and universally available tool for quickly visualizing and assessing a diverse set of coastal modeling data. SCI-WMS is a general OGC WMS solution for serving rasterized visualizations of geospatial data which has been deployed for the COMT project to provide visualizations of a wide range of scientific data.

The NOAA[1]–NGDC[2] Geoportal indexes public geophysical datasets and provides an OGC Catalogue Web Service (CSW) to query datasets by their metadata attributes. SCI-WMS queries the NGDC Geoportal at regular intervals updating both the topologies and OPeNDAP links for all new or modified datasets.

Raw coastal data is hosted by the Southeastern Universities Research Association (SURA) on a dedicated server for the COMT project [13]. Each data set may consist of multiple files in different formats, and may be the result of very different models run by various institutions with disparate computing resources. However, accompanying each dataset is an NcML virtual layer which exposes each dataset as a single NetCDF object which may be accessed via OPeNDAP[3]. Furthermore, the NcML facade presents a consistent set of meta information in accordance to CF-Conventions [5] providing services like SCI-WMS access to the raw data through a uniform interface.

Currently, SCI-WMS is used to visualize data from the first phase groups of IOOS COMT program: *estuarine hypoxia, shelf hypoxia and coastal inundation* [3]. For each modeling group, SCI-WMS successfully generates visualizations from ADCIRC, FVCOM, SELFE and SLOSH models and serves as a use-case of how SCI-WMS can be leveraged as a scalable solution for delivering consistent visualizations of scientific data to a diverse community.

---

[1] National Oceanic and Atmospheric Administration

[2] National Geophysical Data Center

[3] http://www.opendap.org/

4 Results

Figure 5 shows a web portal utilizing the SCI-WMS backend to compare ADCIRC model output for Hurricane Ike with water levels observed by NOAA stations. Ongoing development is in progress for SCI-WMS to support emerging geophysical datasets such as ensemble model output and to provide clear visual support for the assessment and quantification of model skill and performance metrics.

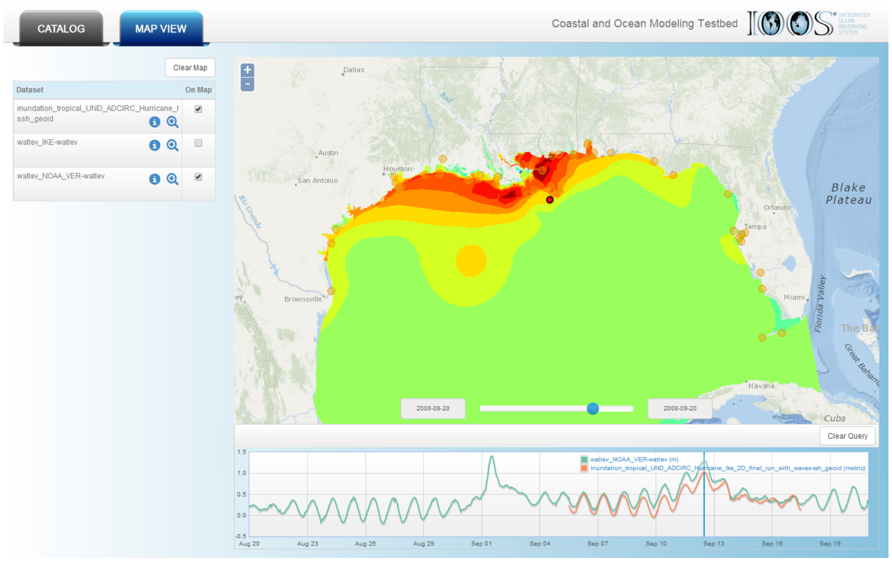


Fig. 5: Comparison of ADCIRC (unstructured topology) model results with observed water levels in the Northern Gulf of Mexico for Hurricane Ike. Verified observed water levels are from NOAA's Station 8760922 (red dot on map). The map shows modeled water levels (in meters above the geoid) at the peak of the storm in southern Louisiana. The time series plot shows both the modeled (green) and observed (orange) water levels. The vertical blue line in the time series plot corresponds to the current time of the map.

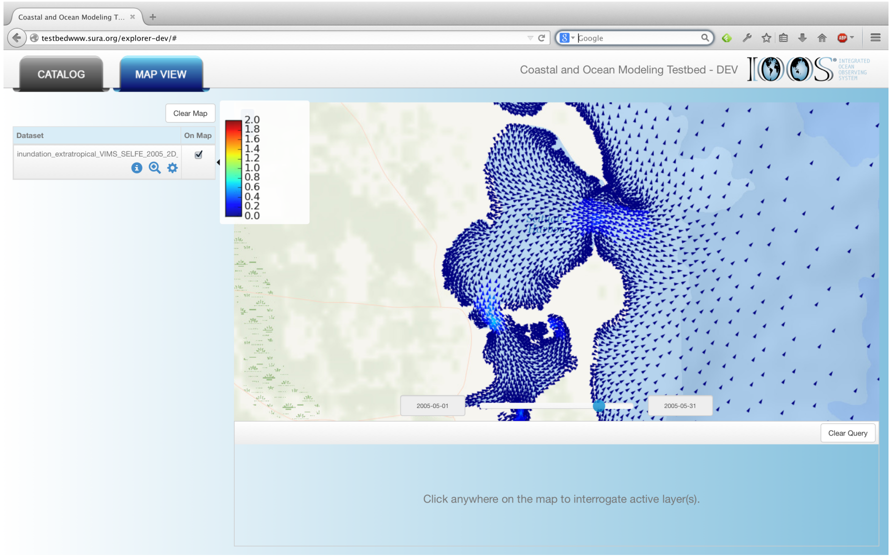


Fig. 6: SELFE model of current direction and speed in the Chesapeake Bay area.

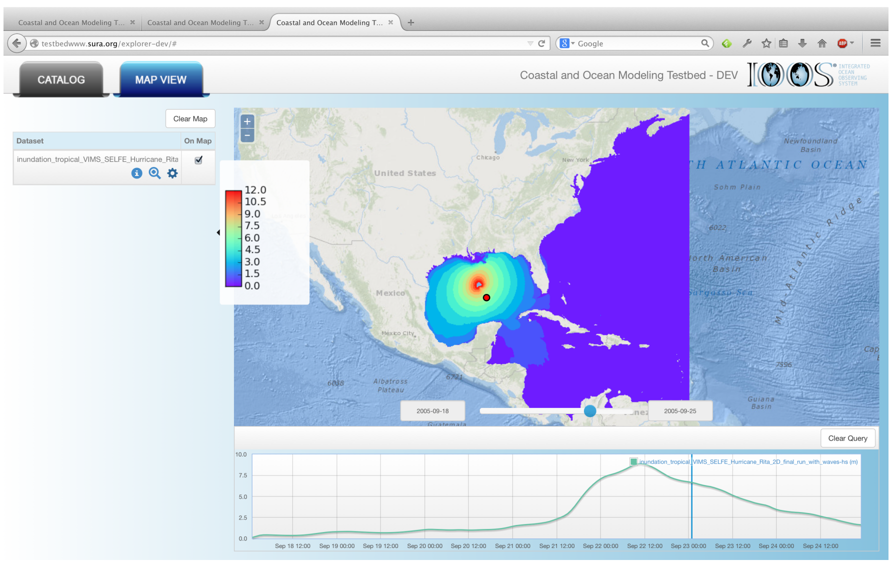


Fig. 7: Visualizing SELFE model of significant sea surface wave height along the eastern coast of the United States. The underlying topology is an unstructured grid with over 5 million nodes which SCI-WMS can handle in real time.

References

1. ASA-RPS https://github.com/asascience-open/sci-wms, "SCI-WMS."

2. Open Geospatial Consortium http://www.opengeospatial.org/standards/wms [Accessed: 2014-07-24], "OpenGIS Web Map Service."
3. R. A. Luettich, L. D. Wright, R. Signell, C. Friedrichs, M. Friedrichs, J. Harding, K. Fennel, E. Howlett, S. Graves, E. Smith, G. Crane, and R. Baltes, "Introduction to special section on the U.S. IOOS Coastal and Ocean Modeling Testbed," *Journal of Geophysical Research: Oceans*, vol. 118, no. 12, pp. 6319–6328, 2013.
4. Open Geospatial Consortium http://www.opengeospatial.org/standards/cat, "OpenGIS catalogue services specification."
5. Climate & Forcast (CF) Metadata Conventions, http://cfconventions.org/latest.html.
6. Django [Computer Software]. Retrieved from https://djangoproject.com.
7. S. V. D. Walt, S. C. Colbert, and G. Varoquaux, "The numpy array: A structure for efficient numerical computation," *Computing in Science & Engineering*, vol. 13, no. 2, pp. 22–30, 2011.
8. J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
9. pyugrid [Computer Software]. Retrieved from https://github.com/pyugrid/pyugrid.
10. Climate & Forcast (CF) - UGRID Metadata Conventions, https://github.com/ugrid-conventions/ugrid-conventions/blob/v0.9.0/ugrid-conventions.md.
11. B. L. Will Schroeder, Ken Martin, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware Inc., 4 ed., 2006.
12. K. J. Weiler, *Topological Structures For Geometric Modeling*. PhD thesis, Rensselear Polytechnic Institute, 1986.
13. R. A. Luettich, L. D. Wright, and S. Elizabeth, "SURA Final Report: A super-regional testbed to improve models of environmental processes on the U.S. atlantic and gulf of mexico coasts," tech. rep., SURA, May 2012.