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 paradigm implicitly assumes every potential end user has the resources to copy and process a potentially massive amount of data, which is not always the case. Even if an analyst has access sufficient resources, tools operating on local systems increase the project costs in terms of bandwidth usage, time and storage. Additionally, coupling decentralized storage and local processing 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 conclusions. Normalizing these results introduces a potential point of error and likewise increases the costs of analysis in terms of time and accuracy. The local analysis paradigm violates the ethos of distributed data storage as the goal of a federated project is to minimize data redundency. Therefore, data processing and visualization software must adapt to the imposed distributed memory model to likewise to realize the same benefits.

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 replication and reformatting of information. 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 and promotes the separation of concerns software and project management practice. SCI-WMS may be deployed by a member of a much larger project who's sole responsibility is providing a visualization platform, promoting quality through specialization. Additionally, SCI-WMS saves costs for 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

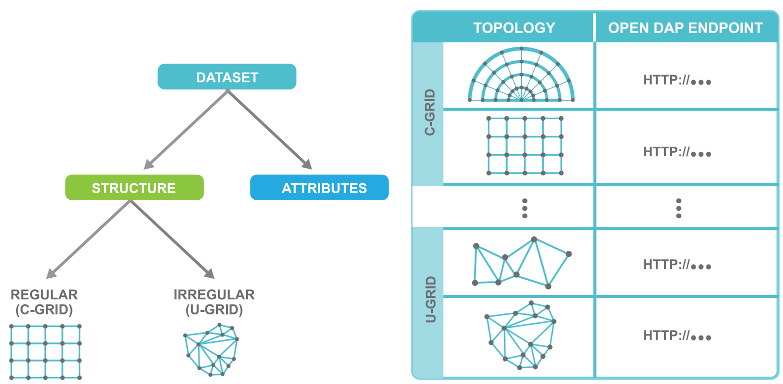
SCI-WMS is an open-source implementation of the Open Geospatial Consortium's (OGC) Web Map Service (WMS) standard which specifies an HTTP interface for generating rasterized visualizations of geospatial data [2]. More specifically, WMS defines a REST API [6] which responds to standardized HTTP GET requests from a web client. A typical WMS request specifies a data layer and region of interest with optional meta data such as rendering style parameters. A WMS response may include additional meta data providing information about data registered with the service or a visualization (commonly a PNG or other standard image format) of a particular data layer for display by the front end web client.

SCI-WMS is implemented in Python using the Django [7] web framework and standard cross-platform numerical software, NumPy [8] and Matplotlib [9] 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 [10], to maintain pace with the latest geospatial software and standards developments including unstructured grid support and CF-UGRID Compliance [11].

While the OGC WMS specification standardizes the client-server communication protocol, WMS implementations vary dramatically in how a particular system fulfills the WMS requests and responses. Vital to the efficiency of SCI-WMS is the abstraction of a registered dataset into structure (also known as topology) and attributes as shown in figure 1a and detailed in subsequent sections. SCI-WMS maintains a local topology 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. As 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 numerical attributes needed to fulfill each request prior to retrieving the appropriate data, which are typically external to the server hosting SCI-WMS and accessed via HTTP. Furthermore, by classifying a topology as regular or irregular, efficient algorithms and data structures are exploited to optimize the computation of attribute subsets.

2.2 Data Model

Visualization tools often decompose data into structure and attributes [12]. Any continuous function to be represented by a digital computer must be measured at a discrete set of samples while rendering numerical data typically requires knowledge of the values between samples to produce 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 continue the abstraction of structure into topology and geometry [13], however in the context of this research, topology is synonymous with structure. Figure 1a outlines the data model adopted by SCI-WMS. A dataset is composed of attributes with associated structure which is further classified as a regular or irregular, 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

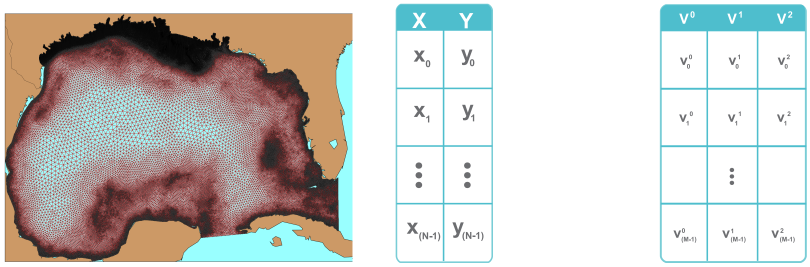
c-grid topologies refer geo-referenced locations with connectivity relations that can be defined analytically, e.g., rectilinear or curvilinear grids. Storing c-grid topologies amount to storing the closed-form expression. Algorithms for processing c-grids such as finding nearest neighbors or computing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

2.4 u-grid Topolgy

An unstructured (u-grid) topology is defined as a topology that is not regular, i.e., does not admit a closed-form representation of sample locations and connectivity. Unstructured topologies offer the highest level of flexibility from a visualization and modeling standpoint as higher sampling frequencies may be used in regions of interest while sparsely sampling low-impact areas to conserve computational resources. As such, many modern CF prediction and modeling algorithms compute attributes associated with unstructured topologies.

Historically, there is a large quantity of CF data associated with structured topologies. As storage and processing hardware has become more accessible, complex unstructured representations have become more prevalent. However, most existing visualization technologies in the CF community can render only simpler, structured datasets. SCI-WMS is one of the first visualization services to support rendering data with unstructured topologies.

Any NCML file exposing 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 plane or space with sample locations, the vertices of the topology, exposed as an array of coordinates in the appropriate ambient space. Vertex connectivity is expressed as a $M \times D$ array where each element is an index into the vertex list. The dimension of a topology (equal to $D-1$) defines the atomic spatial element created by the connectivity list. In 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 the volume enclosed by a set of faces called ***volumes***. For example, figure 2b visualizes a list of a sample locations as an array of coordinates in the real plane. Topology connectivity is specified by a list of indicies into the vertex array interpreted anticlockwise. Figure 2c visualizes a 2D connectivity list, defining a triangulation in the plane, for the vertex array in figure 2b where each entry indexes an element of the vertex list: $v_n^{\cdot} \in \{0, \ldots, N-1\}$. Each row of the connectivity array defines a triangle: 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.



(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.

Figure 2a shows a u-grid 2D topology, a triangulation in the plane, used by a climate model for simulating hurricane conditions in the Gulf of Mexico. Vertices represent locations where attributes are estimated and the connectivity is shown as red lines connecting the vertices throughout the region of interest. The visualization was produced by accessing the CF-UGRID compliant NcML file. The u-grid topology allows the modelers to save storage and computational resources by sparsely sampling 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, u-grid topologies require explicit enumeration of sample locations and connectivity, requiring spatially-aware data structures for optimal storage and processing for performant visualization algorithms. To this end, SCI-WMS maintains a local topology cache, storing u-grid topologies as R-tree [14] data structures on disk local to the deployment server for fast access along with the connectivity array. The R-tree is created when the dataset is first registered with the SCI-WMS service. If a change in the underlying data is detected at an endpoint associated with a topology cache, the R-tree is rebuilt.

2.5 Attributes

Attributes are defined as any numerical quantity associated with a topology, regardless of the classification of the underlying structure as c-grid or u-grid. For example, an atmospheric model may estimate wind direction at the vertices of a triangulated 2D topology or may compute air temperature at the centroid of cell volumes specified by a 3D topology. Attributes have their own dimensionality which is not correlated with the dimensionality of the topology. An attribute specifying temperature or sea-surface-height would consist of scalar values while wind directions are vectors, and tensor attributes are also possible.

2.6 Distributed Memory Model

SCI-WMS stores the data topology locally and fetches numerical attributes hosted externally from the federation as needed for rendering. Given a request for the visualization of a attributes pertaining to a region of interest, the visualization pipeline consists of first computing the sample locations withing the region of interest, using the implicit representation for c-grid and R-tree for u-grid topologies, then fetching the external attributes via OGC web-services. For rendering, the sample connectivity within the area of interest is constructed from the connectivity array which is used for interpolation.

The local topology cache and external attribute mechanism defines a distributed memory model for datasets registered with SCI-WMS as visualized in figure 3. The grid on the left, depicts a regular grid stored locally to SCI-WMS. A possible region of interest is highlighted by the dashed red square. Attributes are stored externally by multiple institutions and are exposed to SCI-WMS in such a way as it can be visualized using the table on the left of the figure. Different attribute layers are represented by the rows of the table while columns correspond to memory locations of attribute values associated with the topology. In this example, attributes associated with positions within the region of interest correspond to memory locations highlighted by the red cells in the table. Given the request for a rendering of an attribute layer(s), SCI-WMS dispatches a request for externally hosted numerical data necessary to render the region of interest.

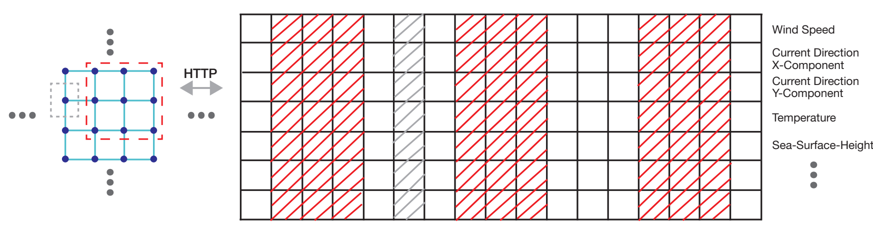


Fig. 3: The SCI-WMS distributed memory model. The grid on the left shows a regular grid stored locally to the SCI-WMS server and the table to the right visualizes the attribute memory locations hosted by an external web-service. While attributes may be heterogeneous, the external attribute table visualization offers a convenient first order conceptualization while the exact relationship between topology and attributes is specified via NcML meta data.

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

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]. A crucial component for the success of the U.S. IOOS COMT mission is a web accessable tool for quickly visualizing and assessing a diverse set of coastal modeling data. 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.

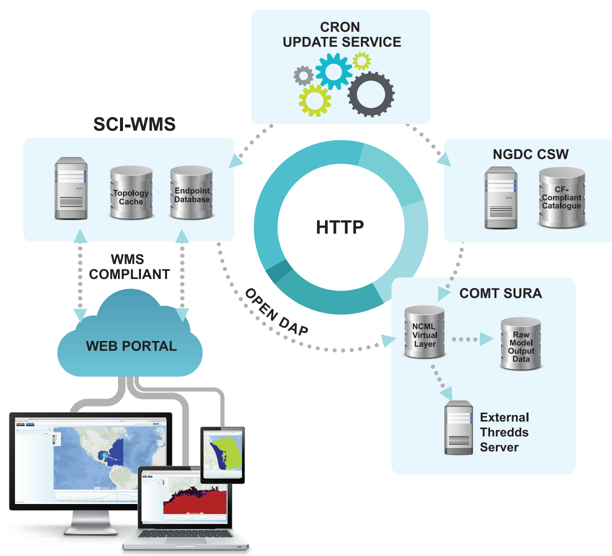


Fig. 4: Overview of the SCI-WMS deployment for the U.S. IOOS COMT project. SCI-WMS updates its topology and endpoint database via a nightly service which queries CF-Compliant datasets cataloged by NGDC. Model data is hosted on an external web server exposed by an NcML facade as a single NetCDF data structure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

Figure 4 outlines the cyberinfrastructre behind the deployment of SCI-WMS for the COMT project. The National Oceanic and Atmospheric Administration (NOAA) - National Geophysical Data Center (NGDC) geoportal indexes public geophysical datasets and provides an OGC CSW service to query datasets by their metadata attributes. SCI-WMS queries the NGDC Geoportal at regular intervals updating the local topology cache and structure-endpoint database (figure 1b) with new or modified datasets. Raw coastal data is hosted by the Southeastern Universities Research Association (SURA) on a dedicated server for the COMT project [15]. 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 [16], OPeNDAP [17] accessible object. 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.

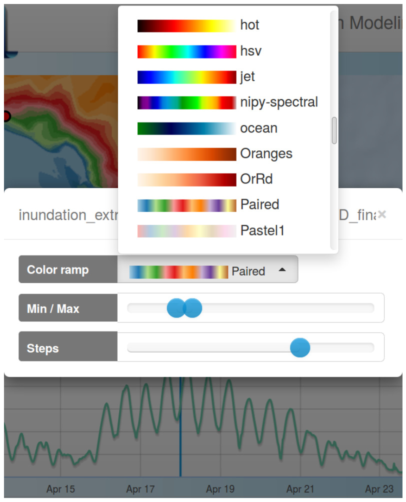
Currently, SCI-WMS is used to visualize data from the first phase groups of IOOS COMT program: *estuarine hypoxia, shelf hypoxia and coastal inundation* [3]. For each modeling group, SCI-WMS successfully generates consistent visualizations of data generated by ADCIRC [18], FVCOM [19], SELFE [20] and SLOSH [21] coastal modeling algorithms and serves as a use-case for how SCI-WMS can be leveraged as a scalable solution for delivering visualizations of scientific data to a diverse community.

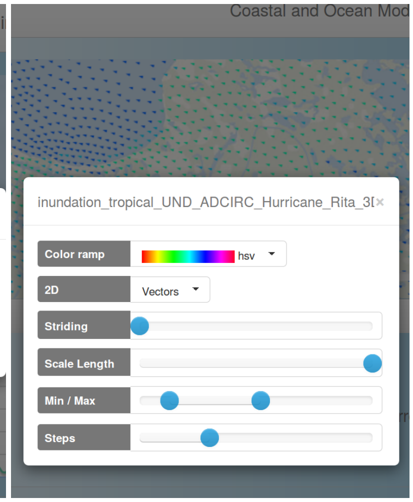
SCI-WMS currently supports contour and filled-contour visualizations styles for scalar attributes while 2D flow fields can be shown as arrows or barbs for vector valued attributes. Figure 6 shows a web portal utilizing the SCI-WMS backend to compare ADCIRC model output for Hurricane Ike with water levels observed by NOAA stations and figure 7 visualizes current direction and speed in the Chesapeake Bay area. Figure 8 renders the sea surface wave height computed along the Atlantic coast of South America, the Gulf of Mexico, up to Canada. The topology in this example is unstructured (u-grid), a triangulation containing over 5 million vertices (sample locations). Attributes are fetched from the appropriate external server as needed, rendered, cached for performance, but ultimately discarded after processing to minimize storage redundancy.

A custom web portal[1] for interacting with the SCI-WMS visualization service was developed the COMT IOOS project to provide researchers an intuitive interface for generating visualizations of observational and modeling data. Figure 5a shows the user interface for manipulating color themes of a scalar filled-contour plot while figure 5b shows the interface for manipulating the visualization parameters of a vector flow field.

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.

[1] http://testbedwww.sura.org/explorer/





(a) Colormaps available for scalar attributes. (b) Parameters for visualizing vector fields.

Fig. 5: User interface for manipulating rendering style parameters to be sent to SCI-WMS via HTTP. The images behind the UI controls are the rendering returned by SCI-WMS.

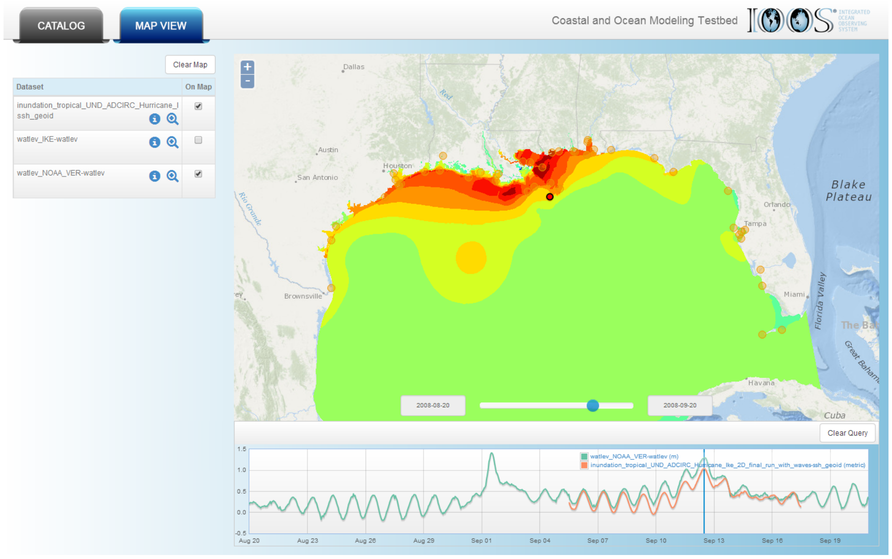


Fig. 6: Comparison of ADCIRC (unstructured topology) model results with observed water levels in the Northern Gulf of Mexico for Hurricane Ike. Verified observed water levels are from NOAA's Station 8760922 (red dot on map). The map shows modeled water levels (in meters above the geoid) at the peak of 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.

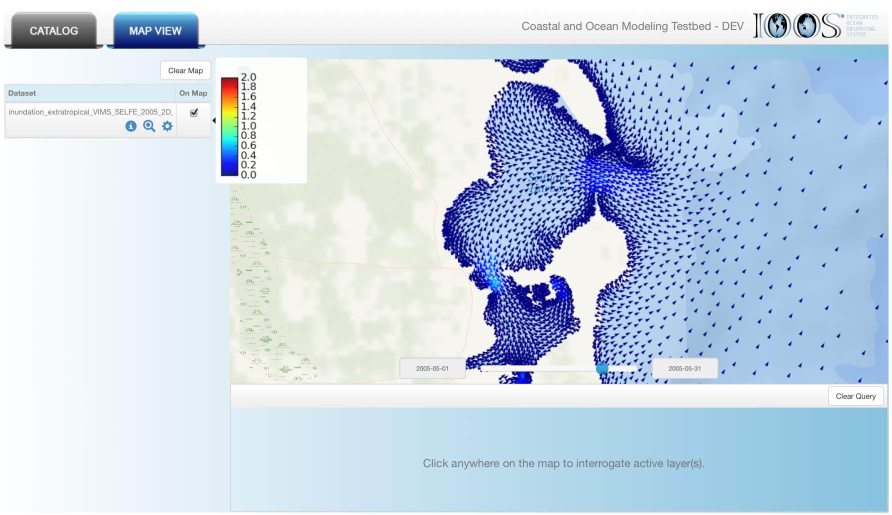


Fig. 7: SELFE model of current direction and speed in the Chesapeake Bay area.

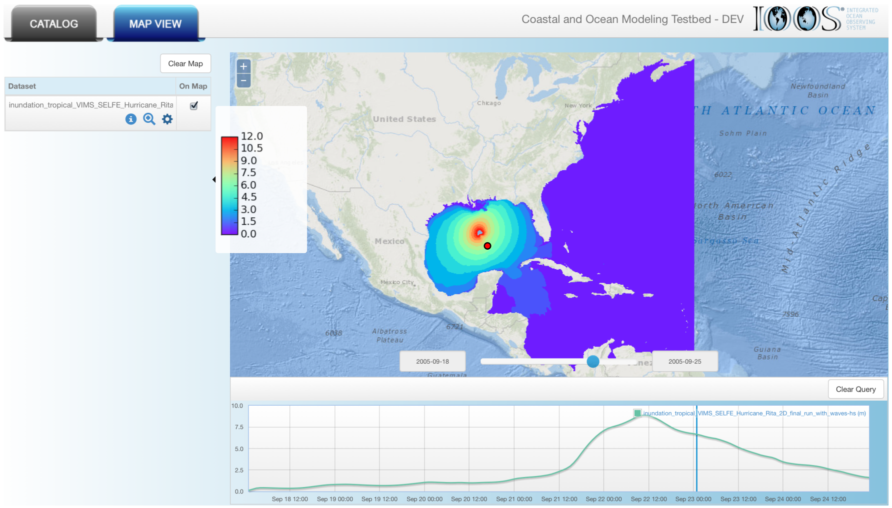


Fig. 8: Visualizing SELFE model of significant sea surface wave height along the eastern coast of the United States. The underlying topology is an unstructured grid with over 5 million nodes which SCI-WMS can handle in real time.

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. R. T. Fielding and R. N. Taylor, "Principled design of the modern web architecture," *ACM Trans. Internet Technol.*, vol. 2, pp. 115–150, May 2002.
7. Django [Computer Software]. Retrieved from https://djangoproject.com.
8. 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.
9. J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.
10. pyugrid [Computer Software]. Retrieved from https://github.com/pyugrid/pyugrid.
11. Climate & Forcast (CF) - UGRID Metadata Conventions, https://github.com/ugrid-conventions/ugrid-conventions/blob/v0.9.0/ugrid-conventions.md.
12. B. L. Will Schroeder, Ken Martin, *The Visualization Toolkit: An Object-Oriented Approach to 3D Graphics*. Kitware Inc., 4 ed., 2006.
13. K. J. Weiler, *Topological Structures For Geometric Modeling*. PhD thesis, Rensselear Polytechnic Institute, 1986.
14. A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data*, SIGMOD '84, (New York, NY, USA), pp. 47–57, ACM, 1984.
15. 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.
16. R. Rew and G. Davis, "Netcdf: an interface for scientific data access," *Computer Graphics and Applications, IEEE*, vol. 10, pp. 76–82, July 1990.
17. P. Cornillon, J. Gallagher, and T. Sgouros, "Opendap: Accessing data in a distributed, heterogeneous environment," *Data Science Journal*, vol. 2, pp. 164–174, 2003.
18. R. Luettich and J. Westernick, "Formulation and numerical implementation of the 2D/3D ADCIRC finite element model version 44.xx," tech. rep., University of North Carolina at Chapel Hill, 2004.
19. C. Chen, R. C. Beardsley, and G. Cowles, "An unstructured grid, finite-volume coastal ocean model (fvcom) system," *Oceanography*, vol. 19, pp. 78–89, 2006.
20. Y. Zhang and A. M. Baptista, "SELFE: A semi-implicit Eulerian-Lagrangian finite-element model for cross-scale ocean circulation," *Ocean Modelling*, vol. 21, pp. 71–96, 2008.
21. J. Chen, W. Shaffer, and A. Gilad, "SLOSH–A hurricane storm surge forecasting model," *Preprints, Oceans*, vol. 81, pp. 314–317, 1984.