GeoFish - integration of fisheries models with GIS

Tiffany C. Vance, Sharon Mesick and Christopher W. Moore
NOAA/NMFS, Seattle, Washington; NOAA/PMEL, Seattle, Washington;
NOAA/NCDDC, Stennis Space Center, Mississippi

Abstract

In GeoFish, we have provided a prototype of integrating a GIS with oceanographic and fisheries models. Scientists are able to use a graphical interface to display datasets, select the data to be used, set the weights for factors in the model and execute the model. The results are returned to the GIS for display and spatial analysis. The project creates a framework for linking to other types of fisheries, oceanographic, and ecosystem models. We plan to visualize and analyze the results of two models; a model of ocean circulation and a model of larval dispersion based upon the fields produced by the circulation model.

Introduction

Models provide a way to understand and predict the behavior of natural systems. Ocean-atmosphere models such as the Regional Ocean Modeling System (ROMS) are intended to provide a multi-purpose, multi-disciplinary oceanic modeling tool. Fisheries models such as the particle tracking model used in the Chesapeake Bay Oyster Larvae Tracker (CBOLT) and individual-based models provide ways to study simplified versions of complex ecosystems. While models provide greater understanding, the results can be very large, multi-dimensional and hard to understand. Ideally, visualization and analysis of the output from spatial ecological models could be easily provided by a geographic information system (GIS). In the past, these two have not been well integrated for scientific uses. Lack of true integration hinders the ability of managers and scientists to create interactive, GIS-based models for management and research. However, GIS packages are now providing programming constructs, by exposing code and objects, to allow closer coupling of core GIS functionality and analytical/modeling tools.

In creating GeoFish (Figure 1), we provide a prototype of how one might integrate a GIS with a number of oceanographic and fisheries models. With this tool, scientists and managers are able to use a graphical interface to display datasets, select the data to be used in a scenario, set the weights for factors in the model and execute the model on a compute server. The results are returned to the GIS for display and spatial analysis. The project creates a framework for linking to other types of back-end fisheries, oceanographic, and ecosystem models written in a variety of programming languages. Current prototype applications include two examples - one an oceanographic model and the other an example of particle tracking for oyster larvae. The first involves setting parameters for a regional ocean modeling system (ROMS) model and displaying results draped over a three-dimensional globe. The second allows the setting of sources for dispersion of oyster larvae in Chesapeake Bay and the display of the resulting dispersion tracks in two and three-dimensions.

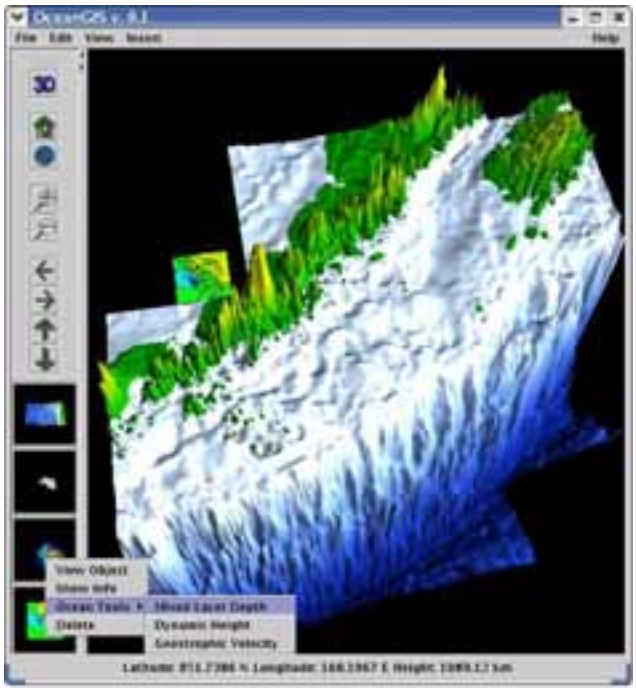

Figure 1 GeoFish interface for CTD data

Background

Optimally, integration of models would be tightly coupled so that setup and display would be done using a familiar GIS-based interface, while the computationally intensive model execution would be done on a powerful processor. In this scenario the processor could be scaled as the model became more complex without affecting the tools to setup and display the model. GIS-based models are unable to handle high computational loads and communication between PC-based GIS applications and a UNIX/Linux-based models has been problematic. Java APIs for GIS functions, such as reprojecting data and calculating spatial statistics, are now available, and can be used for reformatting input to models and for analyzing the output of models. Java can also be used to link to advanced tools for visualization and display to allow for visual exploration of the results of model runs.

Numerical models can be implemented within a GIS in a number of ways. They can be loosely coupled, with the GIS basically used to prepare data for use in a separate computational model; the model can be implemented using the functionality of the GIS, for example in calculating hillslopes and drainage patterns for hydrological model; the model and the GIS can be tightly computationally coupled, with the GIS used both for the input and for visualization of the output. Ideally, data could be exchanged directly and seamlessly between the model and a GIS. In this instance, the user would choose datasets, define model structures and select parameters for a scenario or model run within the GIS user interface. The model itself would combine spatial analytical tools from the GIS world with scientific modeling capabilities from the theoretical realm. Use of high end processors for the models would create an almost real-time interaction between the model back-end and the GIS front-end. Users would be able to describe a scenario, generate results and rerun the scenario with altered parameters in a timely and efficient manner. Results would be enhanced by the automatic generation of maps and geospatial displays.

Examples of GIS-centered models include modeling of seafloor habitat (Greene et al., 2005; Monaco et al. 2005), models of the spread of diseases (Cromley, 2003) and models of the aesthetics of landscapes (ESRI, 2005). Examples of hydrological models created within a GIS, taking advantage of

the native analytical functions of the GIS, include those built for vector data using the Arc Hydro Toolset (Arctur and Zeiler, 2004), or for raster data, such as TauDEM (Tarboton, 2005) and the Groundwater Modeling Tool for GRASS (Carrera-Hernandez and Gaskin, 2006).

Examples of standalone models that could be visualized in a GIS include weather models such as the Community Climate System Model (CCSM), Rapid Update Cycle (RUC) and ECMWF Re-Analysis models (Unidata, 2006). Tools such as RAMAS GIS use a GIS to organize data for input into a standalone habitat model (Akçakaya et al., 2004). The MODFLOW groundwater flow model can be visualized with the GRASS GIS (Brodie, 1998 and http://grass.gdf-hannover.de/twiki/bin/view/GRASS/JaimeCarrera).

Software tools used in GeoFish

GeoFish creates a Java-based framework for linking with back-end models that incorporate a variety of programming language-based models. A number of software tools are used to create this framework. These are the Visualization Toolkit (VTK), which is used to visualizing model output, the Java3D applications programming interface (API), also used for visualizing output, and GeoTools, used for reading in shapefiles. The final product is a standalone tool that can be downloaded [http://nctr-people.pmel.noaa.gov/cmoore/OceanGIS/] and used on a system running Java 1.5. No other software or licenses are needed.

The Java3D API extension is used to create visualizations [https://java3d.dev.java.net/]. It is designed as a high-level, platform independent 3-D graphics programming API, and is amenable to very high performance implementations across a range of platforms. This allows us to visualize the results on a variety of platforms including high performance systems such as grid compute servers. To optimize rendering, Java3D implementations are layered to take advantage of the native, low-level graphics API available on a given system. In particular, Java3D API implementations are available that utilize OpenGL, Direct3D, and QuickDraw3D. This means that Java3D rendering will be accelerated across the same wide range of systems that are supported by these low-level APIs.

We also make use of a second 3-D API called the Visualization Toolkit (VTK) [www.kitware.com] for creating the output visualizations. VTK is a cross-platform 3-D application programming interface built upon, and independent of, the native rendering library (OpenGL, etc). It exposes Java bindings (as well as Tcl and Python). It is written in C++ and includes similar scene-graph, lighting models, and graphic primitives as Java3D. VTK performs boolean operations on 3-D volumes (intersection, union), volume rendering, filtering, including convolution, FFT, Gaussian, Sobel filters, permutation, high- and low-pass Butterworth filters, and divergence and gradient calculation. The VTK data model allows for fast topology traversal, making these filters very fast, and allows for rapid mesh decimation. VTK also offers powerful 3-D probe "widgets" that allow easy interaction with the data, and has methods to utilize parallel architecture through the Message Passing Interface (MPI).

GeoTools [www.geotools.org] is used for reading in shapefile data. It is an open source library for handling geospatial data. It supports the reading of a variety of data formats including shapefiles and PostGIS databases. GeoFish uses the shapefile reader to allow reading in shapefiles for input to models and as layers in the final display.

The CBOLT application uses an ArcIMS interface to set up model parameters. A three-dimensional circulation model written in FORTRAN and C generates a circulation field for the particle tracking model. The particle tracking model is written in C++. The results of the model are in netCDF and are converted to geospatial features using a Java-based processor. They are stored in a file-based geodatabase (Figure 2).

In implementing the NPZ and IBM models, we plan to use a proprietary package from Environmental Systems Research Institute (ESRI) called ArcGIS Engine and implementations of ArcObjects [www.esri.com] to provide spatial analytical tools. ArcEngine is a simple API-neutral

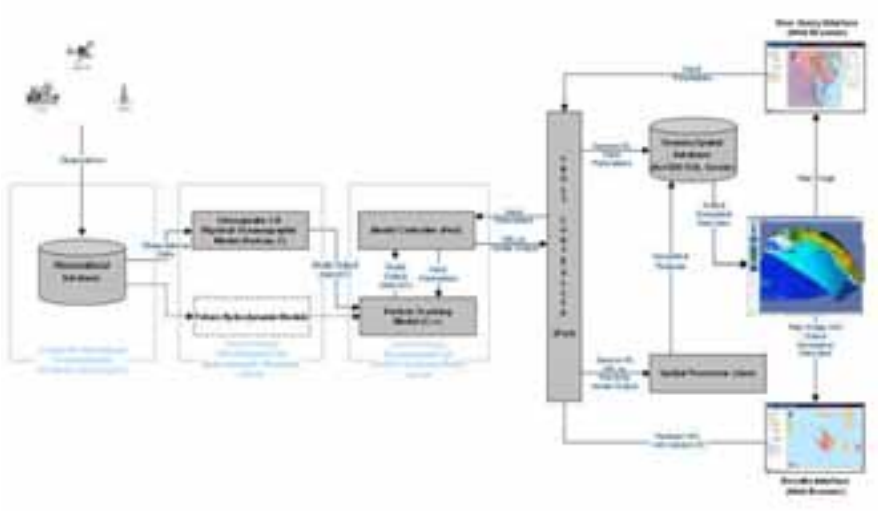

Figure 2 CBOLT/GeoFish architecture

cross-platform development environment for ArcObjects - the C++ component technology framework used to build ArcGIS. ArcObjects are the core of the ArcGIS functionality and include tools such as overlay (union, intersect), proximity (buffer, point distance), surface analysis (aspect, hillshade, slope), and data conversion (shapefile, coverage and DEM to geodatabase). ArcEngine's object library makes full GIS functionality available though fine and coarse-grained components that can be implemented in Java and other environments. Using ArcEngine, solutions can be built and deployed to users without requiring the ArcGIS Desktop applications (ArcMap, ArcCatalog) to be present on the same machine. It supports all the standard development environments, including Java, and C++, and all the major operating systems.

Prototype applications

The GeoFish interface and display use graphical objects to provide context-sensitive functionality related to the type of data being displayed. The tools are designed to be familiar both to GIS users and to users of scientific graphics packages. GeoFish builds upon an earlier tool called OceanGIS (Vance et al., 2005). OceanGIS was initially designed to allow 3-D oceanographic calculations on in-situ data, and to overlay the results. As such, tools were developed to calculate some basic properties of conductivity-temperature-depth measurements, such as mixed-layer depth, geostrophic velocity, and dynamic height. As a data layer is added, the relevant tools for analyses are exposed for use. In the example shown in Figure 1, the addition of a layer of conductivity-temperature-depth (CTD) data causes tools for calculating mixed layer depth and other appropriate oceanographic parameters based on this layer to become available. Data can be read directly from an OPeNDAP (Open-source Project for a Network Data Access Protocol) [www.opendap.org] server by the Java code in GeoFish.

We have two prototype applications to test both the parameter setting and the display and visualization aspects of the GeoFish framework. The first involves setting the parameters for a regional ocean modeling system (ROMS) [http://marine.rutgers.edu/po/index.php?model=roms&page=] model and displaying the results draped over a three-dimensional globe. The second implementation is for setting parameters for setting parameters for a Lagrangian particle tracking model for larvae and displaying the results in a map and as a visualization.

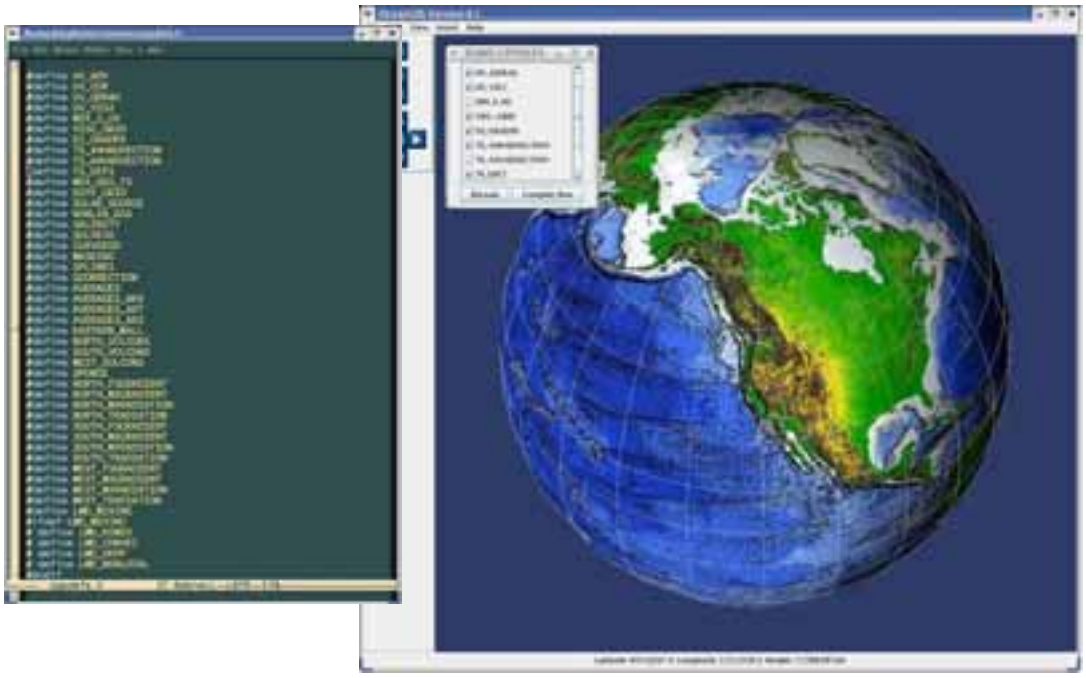

Figure 3 Interface for setting ROMS model parameters

Regional ocean models

The initial implementation of GeoFish was done using a VTK-based interface to ROMS. Complex ocean-atmosphere models such as ROMS are intended to provide a multi-purpose, multi-disciplinary oceanic modeling tool. These models can be daunting in the complexity of setting up their parameters. Visualizing the three-dimensional output is not simple, the learning curve for using the models is steep, and existing tools tend to only portray two-dimensional slices of the output. GeoFish's aim is to remove some of these obstacles and allow non-modelers to explore the utility of these models.

A basin-scale ROMS has been developed for regions of the Pacific Ocean with a spatial resolution of 12.5 km [http://ourocean.jpl.nasa.gov/]. The test application uses an interface that allows the user to modify any of the C-preprocessor directives that ROMS uses in its build-script to enable various physical and numerical options (Figure 3). The initialization file can also be modified to reflect changes in timestep, tiling, and initial conditions. The model is configured, compiled and launched through the GeoFish interface, and model results are output to an OPeNDAP server directory, allowing either viewing in GeoFish or sharing results with remote colleagues. The GeoFish ROMS Data Reader is a GUI class that reads netCDF output of ROMS model data, and allows 3-D renderings to be created and animated in a geo-referenced framework. Since GeoFish utilizes the Geotools library, shapefiles of data using standard projections can be rendered simultaneously (as opposed to simply overlaying). Then ROMS model output is loaded through the ROMS Data Reader, and the user selects the variable of interest, contour levels, color maps, etc., and animates the resulting rendering (Figure 4).

CBOLT/GeoFish integration

The second implementation is an ArcIMS and VTK-based interface to a Lagrangian particle tracking model. CBOLT uses an ArcIMS-based front end to set the source for the larvae, the duration of the model run and whether the model should be run forwards or backwards in time

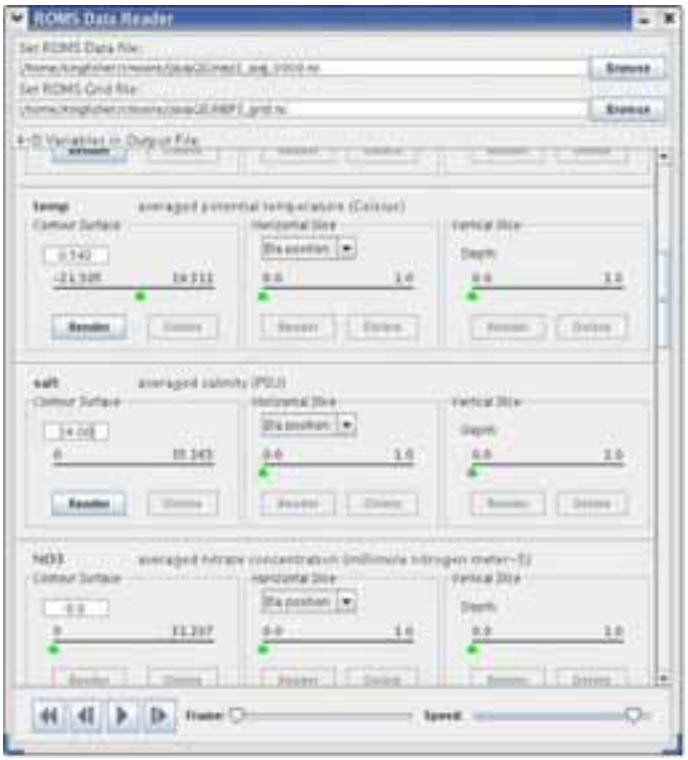

Figure 4 Interface for setting ROMS display parameters

(http://gis2.esri.com/library/userconf/proc06/papers/abstracts/a2095.html). CBOLT allows the user to "seed" a region or area with larvae and watch how they disperse based upon a hydrodynamic circulation model. The user sets parameters for the model run including the release location for the larvae, the release data and time, the run duration and direction, the number of larvae to release and the type of spatial output. The particle tracking model is run and the results are returned as a netCDF file which is translated and stored at geospatial features. The output is a three-dimensional grid of the trajectories of the larvae over time. The features can be shown as a two-dimensional map in an ArcIMS and as a three-dimensional visualization in GeoFish. The ArcIMS renderer (Figure 5) provides analyses such as spatial queries and buffering. The GeoFish renderer allows for color-coding of the tracks, overlays of other parameters stored in shapefiles and netCDF files, and zoom, pan and rotation of the visualization for user exploration. CBOLT is composed of modules and is intended to allow the insertion of new modules such as different types of circulation models or other databases or storage schema. The CBOLT/GeoFish architecture will also be applied to a model of the distribution and transport patterns of northern rock sole, *Lepidopsetta polyxystra*, larvae in the southeastern Bering Sea (Lanksbury et al., 2007).

Conclusions

GeoFish prototypes the direct integration of GIS and modeling capabilities in support of fisheries recruitment and forecasting. Through the use of Java-based application programming interfaces (APIs) and connectors, a GIS front-end is directly linked with models. Scientists and managers are provided with a GIS-based graphical interface to display datasets, select the data to be used in a scenario, set the weights for factors in the model and run the model. The results are returned to the GIS-based application for display and spatial analysis.

Figure 5 CBOLT/GeoFish interface

The results have been applied to two types of models - ROMS ocean models and the CBOLT particle tracking model. The ease with which these models have been implemented suggest that the implementation of further models should be straightforward. Making it simpler for modelers to integrate new GIS-based functionality has made them more open to implementing GeoFish. The enhanced visualization capabilities and the ability to easily include GIS-based data, such as socioeconomic data, will enhance the results of the existing scientific models and new implementations of multi-trophic level fisheries IBMs embedded in three-dimensional ocean circulation models such as ROMS.

Acknowledgments and disclaimers

Funding for this project was provided by the High Performance Computing and Communications (HPCC) project of the NOAA Office of the Chief Information Officer. For more details please see http://nwweb.nwn.noaa.gov/hpcc/nwg/. Additional support for this research was provided by the NOAA Fisheries-Oceanography Coordinated Investigations (FOCI). This publication was supported by the Joint Institute for the Study of the Atmosphere and Ocean (JISAO) under NOAA Cooperative Agreement #NA17RJ1232. PMEL contribution nnnn. This research is contribution EcoFOCI-0632 to NOAA's Ecosystems & Fisheries Oceanography Coordinated Investigations. The views expressed herein are those of the author(s) and do not necessarily reflect the views of NOAA or any of its sub-agencies. Mention of software products does not imply endorsement of these products.

9. REFERENCES

Akçakaya, H. Renullit, Burgman, M.. A., Kindvall O., Wood, C. C., Per Sjögren-Gulve, Hatfield, J.S. and McCarthy, M.A., 2004. Species conservation and management : case studies. New York : Oxford University Press.

Arctur, D. and Zeiler, M., 2004. Designing Geodatabases: Case Studies in GIS Data Modeling, Redlands, CA, ESRI Press.

Brodie R.S. 1998. Integrating GIS and RDBMS technologies during construction of a regional groundwater mode, Environmental Modeling and Software, Volume 14, Number 2, December 1998, pp. 119-128(10)

Carerra, Jamie, http://grass.gdf-hannover.de/twiki/bin/view/GRASS/JaimeCarrera, viewed 3/25/2006.

Carrera-Hernandez, J. J. and Gaskin, S. J. 2006. The groundwater modeling tool for GRASS (GMTG): Open source groundwater flow modeling, Computer and Geosciences, 32(3): 339-351. Methuen;

Cromley, Ellen K., 2003. GIS and Disease, Annual Review of Public Health, January 2003, Vol. 24, Pages 7-24

ESRI. 2005. http://www.esri.com/news/arcnews/winter0506articles/for-puget-sound.html, viewed 4/10/2006.

Greene, H.G., Bizzarro, J.J., Tilden, J.E., Lopez, H.L., and Erdey, M.D., 2005. The benefits and pitfalls of geographic information systems in marine benthic habitat mapping, in Wright, D.J. and Scholz, A.J., Place Matters: Geospatial Tools for Marine Science, Conservation, and Management in the Pacific Northwest, Corvallis, OR, Oregon State University Press.

Hermann, A.J., Hinckley, S., Megrey, B. A., and Napp, J.A.. 2001. Applied and theoretical considerations for constructing spatially explicit individual-based models of marine larval fish that include multiple tropic levels. ICES J. Mar. Sci., 58, 1030–1041

Lanksbury, J.A., Duffy-Anderson, J.T., Mier, K.L., Busby M.S., and Stabeno P.J. 2007. Distribution and transport patterns of northern rock sole, Lepidopsetta polyxstra, larvae in the southeastern Bering Sea. Prog. Oceanogr., 72, 39–62.

Monaco, M., Kendall, M., Higgins, J., Alexander, C., and Tartt, M., 2005. Biogeographic assessments of NOAA national marine sanctuaries: The integration of ecology and GIS to aid in marine management boundary delineation and assessment, in Wright, D.J. and Scholz, A.J., Place Matters: Geospatial Tools for Marine Science, Conservation, and Management in the Pacific Northwest, Corvallis, OR, Oregon State University Press.

Murphy, J., Marshall, J., Ulmer, J.,Boulware, J., and Eslinger D., 2006. CBOLT: Visualizing NetCDF Model Output as Spatial Features in ArcSDE, Proceedings of the ESRI User Conference. San Diego. August 7–11, 2006.

Tarboton, D. G., 2005, Terrain Analysis Using Digital Elevation Models (TauDEM) http://hydrology.neng.usu.edu/taudem/ , viewed 12/20/2005.

UNIDATA, 2006, 'Example netCDF files', http://www.unidata.ucar.edu/software/netcdf/examples/files.html, viewed 4/6/2006.

Vance, T.C., Merati, N., Moore, C., 2005. Integration of Java and GIS for visualization and analysis of marine data. International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences of the ISPRS. Working Group II/IV

CBOLT, http://ekman.csc.noaa.gov/CBOLT_Input/index.htm , viewed 5/2/07.

ESRI Arc Engine, http://www.esri.com/software/arcgis/arcgisengine/index.html , viewed 5/10/07

GeoTools home page, www.geotools.org, viewed 4/28/2007.

OPeNDAP pages, www.opendap.org, viewed 4/18/2007.

ROMS model pages, http://ourocean.jpl.nasa.gov, viewed 4/20/2006.

VTK, [www.kitware.com], viewed 5/8/07.

Contact author: Tiffany C. Vance,
NOAA/NMFS, 7600 Sand Point Way NE,
Seattle, Wa. 98115 USA
tel: +01.206.526.6767
email: tiffany.c.vance@noaa.gov