INTEGRATING OCEANOGRAPHIC MODELS WITH GIS - ROMS AND BEYOND

Tiffany C. Vance[1], Sharon Mesick and Christopher W. Moore
NOAA/NMFS, Seattle, Washington; NOAA/PMEL, Seattle, Washington;
NOAA/NCDDC, Stennis Space Center, Mississippi

1. 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 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 starting to provide programming constructs, by exposing code and objects, to allow closer coupling of core GIS functionality and analytical/modeling tools.

In creating GeoModeler, we have provided 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. 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 scenario testing. 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 a tsunami-generating earthquakes in the Pacific and using the specified sources to calculate, from pre-computed model outputs, the height and travel time for potential tsunamis.

The final goal is an application to visualize and analyze the results of two fisheries models; a model for predator-prey interactions and a model to look at the effects of climate change on the recruitment of an economically important fish species. The first model is an individual-based model (IBM) that models the fate of individuals using characteristics such as age, size and prey consumption. The second model is a nutrient-phytoplankton-zooplankton (NPZ) model looking at the effects of climate change on the early life history of fish.

2. BACKGROUND

Optimally, 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. In the past, these two have not been well integrated for scientific uses. GIS-based models are unable to handle high computational loads and communication between a PC-based GIS and a UNIX/Linux-based model was problematic. Lack of true integration hinders the ability of managers and scientists to create interactive,

---

[1] *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

GIS-based models for management and research. However, GIS packages are starting to provide programming constructs, i.e. by exposing code and objects, to allow closer coupling of core GIS functionality and analytical/modeling tools. 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.

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; or 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 and 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 (Monaco et al., 2005; Greene 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 then 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).

In creating GeoModeler (Figure 1), we have
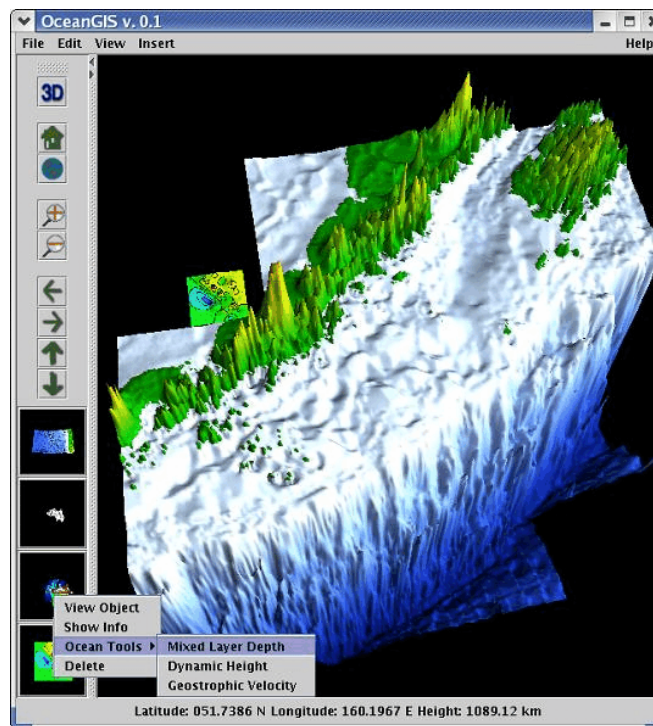
Figure 1 GeoModeler interface for CTD data

provided a prototype of how one might integrate a GIS with a number of oceanographic and fisheries models. Through the use of Java-based application programming interfaces and connectors, a GIS is directly linked with the Regional Ocean Modeling System (ROMS) model and with the Method of Splitting Tsunami (MOST) model. With this tool, scientists and managers are able to use a graphical interface to display input datasets, select the data to be used in a scenario, 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 back-end fisheries, oceanographic, and ecosystem models written in a variety of programming languages.

The Java3D API extension is used to create visualizations [https://java3d.dev.java.net/]. It is designed as a high-level, platform independent 3D 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
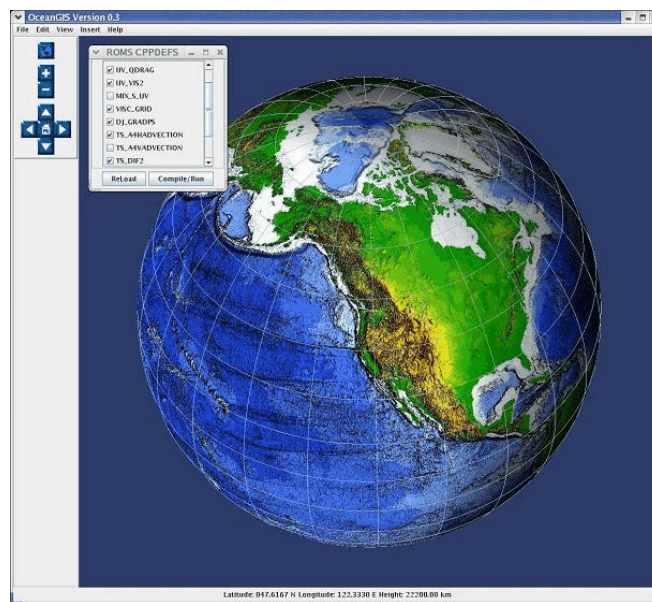

Figure 2 Setting up ROMS parameters

APIs.

3. SOFTWARE PACKAGES

GeoModeler 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.

We also make use of a second 3D API called the Visualization Toolkit (VTK) [www.kitware.com] for creating the output visualizations. VTK is a cross-platform 3D 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 3D 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 3D probe "widgets" that allow easy interaction
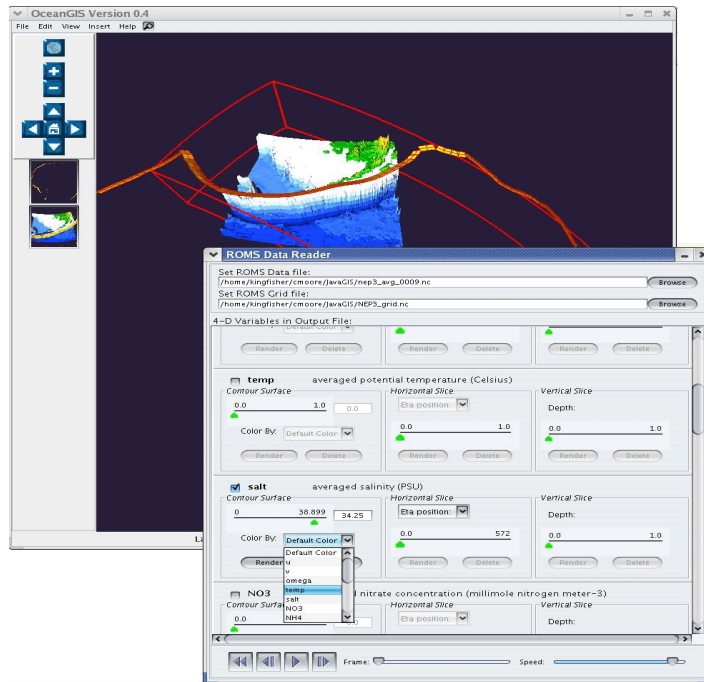
Figure 3 Setting ROMS output display parameters

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. GeoModeler uses the shapefile reader to allow reading in shapefiles for input to models and as layers in the final display.

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 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.

4. PROTOTYPES

Current applications include two examples - one an oceanographic model and the other an example of scenario testing. 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 the Method of Splitting Tsunami (MOST) model and allows a user to specify the source region(s) for a tsunami and to calculate the resulting wave heights and travel times. ROMS and MOST implementations in GeoModeler are used to introduce oceanographic modelers to GIS techniques.

The GeoModeler 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. GeoModeler builds upon an earlier tool called OceanGIS (Vance et al., 2005). OceanGIS was initially designed to allow 3D 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 GeoModeler.

Regional ocean models

The initial implementation of GeoModeler was done using a VTK-based interface to the Regional Ocean Modeling System (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

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 2). 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 GeoModeler interface, and model results are output to an OPeNDAP server directory, allowing either viewing in GeoModeler or sharing results with remote colleagues.

The GeoModeler 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 GeoModeler 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 3).
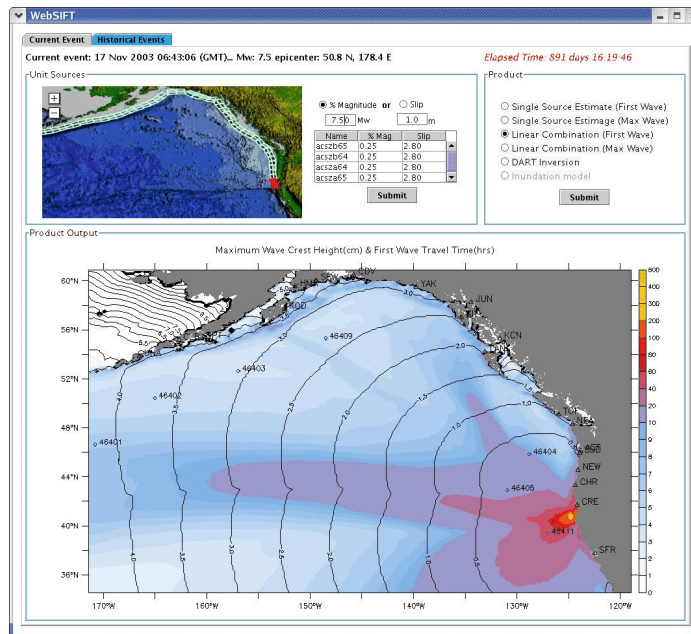
Figure 4 Tsunami model output

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. GeoModeler'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

Tsunami modeling

A second test model for GeoModeler is the implementation of an interface to launch tsunami models and allow the integration of results into a GIS framework. The initial results for this implementation are shown in Figure 4. The model output renderer/animator reads results of the MOST model (http://nctr.pmel.noaa.gov/research.html and Titov and

Gonzalez, 1997), and surface height is rendered at each of 1440 timesteps. This model run includes "runup", or the adjusting of the model boundary condition to simulate inundation.

The MOST model parameters include a series of uniform-sized seismic faults, called "unit sources", shown in Figure 5. These sources are initially set using an interface that allows the user to set the magnitude for each source, as well as the vertical distance the fault moves (the slip). An initial condition for running the inundation model is built up of a linear combination of model runs for each unit source. This initial condition is passed to the inundation model, with default parameters pre-set to give a rapid estimate of inundation in time for emergency managers to view results well before the approaching wave strikes populated areas. Future work includes getting all the parameters for MOST model runs into the interface so the GeoModeler tool can be used for research.

interactions and studying the consequences of changing environmental conditions. The second model is a nutrient-phytoplankton-zooplankton (NPZ) model looking at the effects of climate change on the early life history of walleye pollock (*Theragra chalcogramma*) (Parada et al., submitted). The adults of these species are the target of major commercial fisheries. Both of these projects will expand upon the techniques developed for the ROMS model effort, as ROMS is a part of these models. We also hope to expand the tool to work with the S-coordinate Primitive Equation Model (SPEM) [http://marine.rutgers.edu/po/models/spem.html]. GIS formatted files may be used for input to the models and this part of the project would take advantage of the new direct reading of netCDF data available via ArcGIS and ArcEngine. The output of the models is three-dimensional and will take advantage of the multi-dimensional analysis techniques described above.
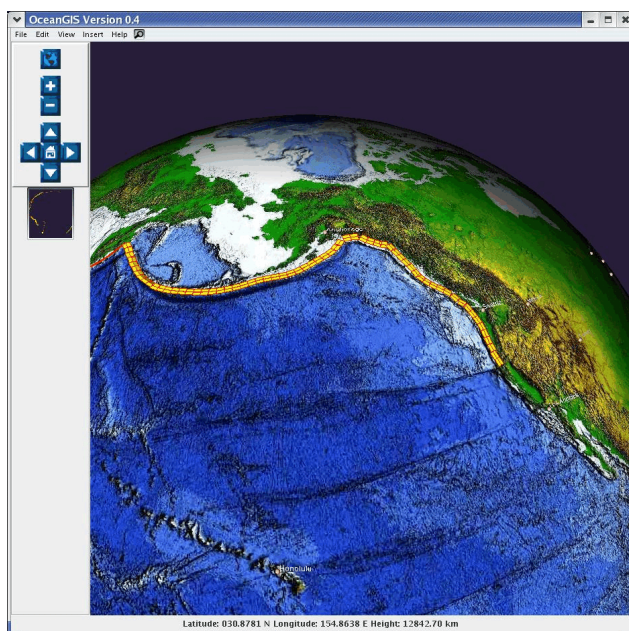
Figure 5 MOST model source regions

Plans for NPZ/IBM implementation

In progress is an application to visualize and analyze the results of two fisheries models; a model for predator-prey interactions and a model to look at the effects of climate change on the recruitment of an economically important fish species. The first model is an individual-based model (IBM) that models the fate of individuals using characteristics such as age, size and prey consumption (Hermann et al., 2001). As such, it is a typical example of the models used for ecosystem

6. CONCLUSIONS

GeoModeler prototypes the direct integration of GIS and modeling capabilities in support of management and decision making. 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.

The results have been applied to two types of models - ROMS ocean models and the MOST tsunami model. The ease with which these models have been implemented suggest that the implementation of further models should be straightforward. Making is simpler for modelers to integrate new GIS-based functionality has made them more open to implementing GeoModeler. The enhanced visualization capabilities and the ability to easily include GIS-based 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.

7. ACKNOWLEDGMENTS AND DISCLAIMERS

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

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.

Parada, C. , Hinckley, S., Dorn, M. and Megrey, B.. Submitted. Estimating walleye pollock recruitment in the Gulf of Alaska using a biophysical model: analysis of physical processes and comparison with stock assessment models and data. Marine Ecology Progress Series.

Tarboton, D. G., 2005, Terrain Analysis Using Digital Elevation Models (TauDEM)

http://hydrology.neng.usu.edu/taudem/ , viewed
12/20/2005.

Titov, T.V., and Gonzalez, F.I, 1997. Implementation
and testing of the Method of Splitting Tsunami (MOST)
model. NOAA Technical Memorandum ERL
PMEL-112, 11 pp

UNIDATA, 2006, 'Example netCDF files',
http://www.unidata.ucar.edu/software/netcdf/examples/f
iles.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

ESRI Arc Engine, www.esri.com , viewed 7/29/06

GeoTools home page, www.geotools.org, viewed
4/28/2006.

MOST model pages,
http://nctr.pmel.noaa.gov/research.html ,viewed
8/15/06.

OPeNDAP pages, www.opendap.org, viewed
4/18/2006.

ROMS model pages, http://ourocean.jpl.nasa.gov,
viewed 4/20/2006.
         http://marine.rutgers.edu/po/index.php?model=
roms&page=, viewed 8/17/06.

SPEM model,
[http://marine.rutgers.edu/po/models/spem.html],
viewed 8/12/06.

VTK, [www.kitware.com], viewed 7/28/06.