

Particles in the Cloud – Running GIS-based Particle Tracking Models for Fish Larval Dissemination in Traditional and Cloud Computing Architectures

Tiffany C. Vance^a and Kyle Wilcox^b

^a San Francisco State University [tcvance@sfsu.edu]

^b Applied Science Associates [kwilcox@asascience]

Abstract: Complex processes involving physical and biological systems can be examined by creating a simplified model of their interactions. A purely physical model might use an ocean circulation model to move particles through space without calculating their effects on the environment. Objects such as marine debris or an oil slick are simply moved from one place to another by the model. More complex models can combine a physical process with a biological process. Modeling the paths of dust particles transported over the oceans involves modeling both the physical transport of the particles and the resulting geochemical and biological effects of the particles when they settle out of the atmosphere into the ocean. Biological particle tracking models, sometimes called individual based models, combine physical models with models of the growth and capabilities of an organism. These are commonly used to study the dispersion of fish larvae. The path that a fish larva follows after hatching, usually driven primarily by oceanic currents, determines the organism's chances of survival and its rate of growth. A GIS can be used to set up the parameters for a model run and also to display the results. As an example, LarvaMap is a tool that uses a map based display for set up and also is tightly linked to ArcGIS for analysis of the output. The GIS can be used to integrate the particle tracking model output with *in situ* measurements of oceanographic and environmental parameters. LarvaMap can use both traditional local computing resources and has been deployed on the Amazon cloud to test the speed and costs of cloud deployment. The transition from a local to a cloud deployment is relatively simple and the performance improvements can be considerable.

Keywords: GIS; particle tracking; environmental modelling; fisheries

1 INTRODUCTION AND BACKGROUND

1.1 Introduction

Particle tracking models utilizing general atmospheric or oceanic circulation models are used to calculate the paths of inanimate particles or organisms that are too small to actively swim against currents. The circulation model provides information on the motion of the fluid or gaseous medium containing the organisms while the particle tracking model describes the transport of the particles in the medium. For more realistic results, the particle tracking model can incorporate the growth and development of the organism to describe its life history. Running these models requires setting a number of parameters and a visual interface for parameter setting can make the process less daunting. The models can return

results on a fine time scale over a large spatial extent and a GIS can be an important tool to display, analyze, and understand the often voluminous results.

Particle tracking models can be used to compute motions over a variety of scales and dimensions. A one-dimensional model describes the motion of particles in a stream or other linear feature. A two-dimensional, or map view, model would describe motions in a plane, for example the circulation of particles on the surface of a lake or small sea. Animating a series of map views through time would add a third temporal dimension. This kind of model could describe surface patterns near the continental shelf where upwelling and downwelling have major biological effects. Adding a depth dimension to a model would provide a fully three-dimensional view of the particle transport. This might be used to study basin-scale motions of particles. Understanding of the evolution of spatial and temporal patterns would be enhanced by animating a three-dimensional model to provide a fully four-dimensional view of the system.

1.2 Examples of Particle Tracking Models in Fisheries Research

In fisheries research, particle-tracking models are used to study the dispersion and survival of a variety of species. Ribergaard et al. (2004) used particle-tracking models based on finite element circulation models to study the transport of shrimp larvae west of Greenland. The Quoddy circulation model is linked with a tracking model that uses Lagrangian advection and diffusion to represent the effects of small-scale turbulence. Grzechnik and Noye [1998] used a similar model on top of the ted circulation model to calculate the spread of western king prawn larvae in Gulf St. Vincent, Australia. Brickman et al. [2007] used the METACOD circulation model and an advective and random displacement particle-tracking algorithm to study the dispersion of cod larvae near Iceland. The particle tracks were mapped but there was no spatial analysis of the results. Models have been used to study the transport of Walleye pollock in Shelikof Strait, Alaska [Hermann et al., 1996] and Lanksbury et al. [2007] looked at the dispersion and transport of rock sole larvae in the Bering Sea.

An example of a particle-tracking model linked to a GIS is the Chesapeake Bay Oyster Larvae Tracker (CBOLT) [<http://ches.communitymodeling.org/models.php>] CBOLT used an Internet Map Server interface to set up the parameters (start location for larvae, duration of run, run of a hindcast or a forecast etc.) for submission to a model controller running the Chesapeake Bay 3-D Physical Oceanographic (C3PO) Model. This finite element hydrodynamic model was developed by the NOAA Coast Survey Development Lab to simulate circulation in the Bay. The output of the hydrodynamic model drove a particle-tracking model and the results were stored in an ArcSDE database. The results were displayed back to the user using the same ArcIMS interface used to set up the parameters to run the model. The results were two-dimensional and static. CBOLT did not take into account the biological development of the larvae. This behavior is being developed in a more recent project described in North et al. [2008].

GeoFish, described in Vance et al. [2008] was a prototype of linking a particle tracking model with a GIS for analysis and display of the results. The particle tracking model from CBOLT was linked to an ArcIMS and VTK-based interface. The interface was used to set the release location for the larvae, the number of larvae released, the release date and time, the duration and temporal direction, i.e. forecast or hindcast, of the model run, and the type of spatial output. The results were returned as a netCDF file of a three-dimensional grid of the trajectories of the larvae over time. The features can be shown as a two-dimensional map in the map server and as a three-dimensional visualization in GeoFish. The ArcIMS renderer provided analyses such as spatial queries and buffering. The GeoFish renderer allowed for color-coding of the tracks, overlaying data stored in shapefiles and netCDF files, and interaction with the visualization for user exploration [Figure 1].

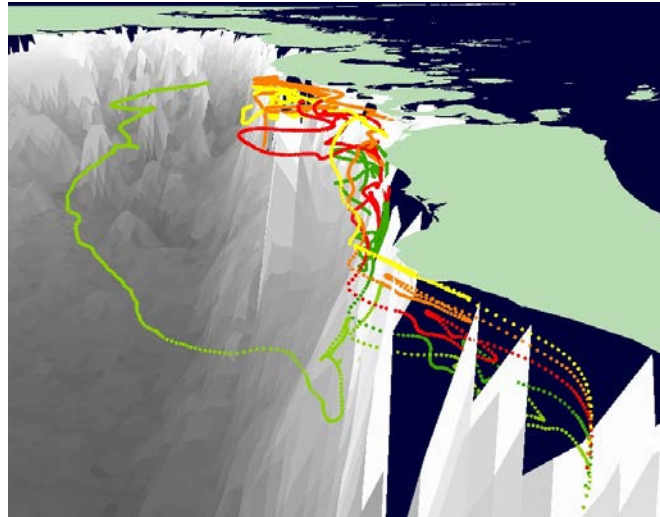


Figure 1. GeoFish visualization of rock sole larval trajectories. The larvae were seeded in a vertical column near the shore. Most drift along shore close to the shoreline but one is carried offshore into deeper water.

2 PARTICLES IN THE CLOUD: LARVAMAP

LarvaMap is a tool for scientists and resource managers to use to create scenarios for larval dispersion (species, start location, start depth, duration of dispersion), to run the particle tracking model, and to display the results for visualization and analysis. LarvaMap (Figure 2) uses a map based display for set up and also is tightly linked to ArcGIS for analysis of the output. The GIS can be used to integrate the particle tracking model output with *in situ* measurements of oceanographic and environmental parameters. LarvaMap can use both traditional local computing resources and can be deployed in the cloud to test the speed and costs of cloud deployment. The transition from a local to a cloud deployment is relatively simple and the performance improvements can be considerable.

The motivation for developing LarvaMap was a desire to make it easy for researchers to create and run their own scenarios. The advent of high performance computing resources and the availability of circulation model data via OPeNDAP and other servers makes it possible to run this kind of model on the user's desktop via a web interface. This rapid turn around and easy visualization makes it possible to use these models for exploratory analyses.

2.1 Model Architecture

The OpenScales framework and Adobe Flex were used to write the web-based map interface for LarvaMap. The interface shows a map of the region being modeled and enables the user to set parameters for the model run including the point or region where the particles will be released, the number of particles released, and the depth at which they are released. The interface also allows the user to choose the underlying circulation model to use and the date and duration of the model run. A variety of circulation models can be used as LarvaMap implements the Unidata Java netCDF libraries for connecting to local or OPeNDAP servers hosting hydrodynamic models.

The user entered parameters are passed via an XML file to the Java-based REST web service. The web service packages the parameters with a unique ID for the model run and adds the job to the end of the model runner job queue. The Java-based backend model runner then takes the XML file, parses out the model

parameters and uses them when it launches the Java-based particle tracking model. The model returns results to the web interface via an XML file.

2.2 Traditional Deployment and Use

The interface for LarvaMap is a web page where the user can set the starting points(s) for the particles as a point location or as a polygonal area. The user can also specify the number of particles to release, the depth(s) at which they are released and the duration of the model run. A variety of underlying circulation model outputs can be used and the horizontal and vertical diffusivity can also be specified. The input parameters are verified before the model run is submitted. Once the user approves all of the parameters the model run is launched. While the model is running, the web interface is updated to show the progress of the model run. When the run is complete, a summary map of the locations of the centroid of the particle “cloud” is drawn, the total model run time is given and the full results are available for download as a netCDF file to allow for further mapping and analysis in a tool such as ArcGIS.

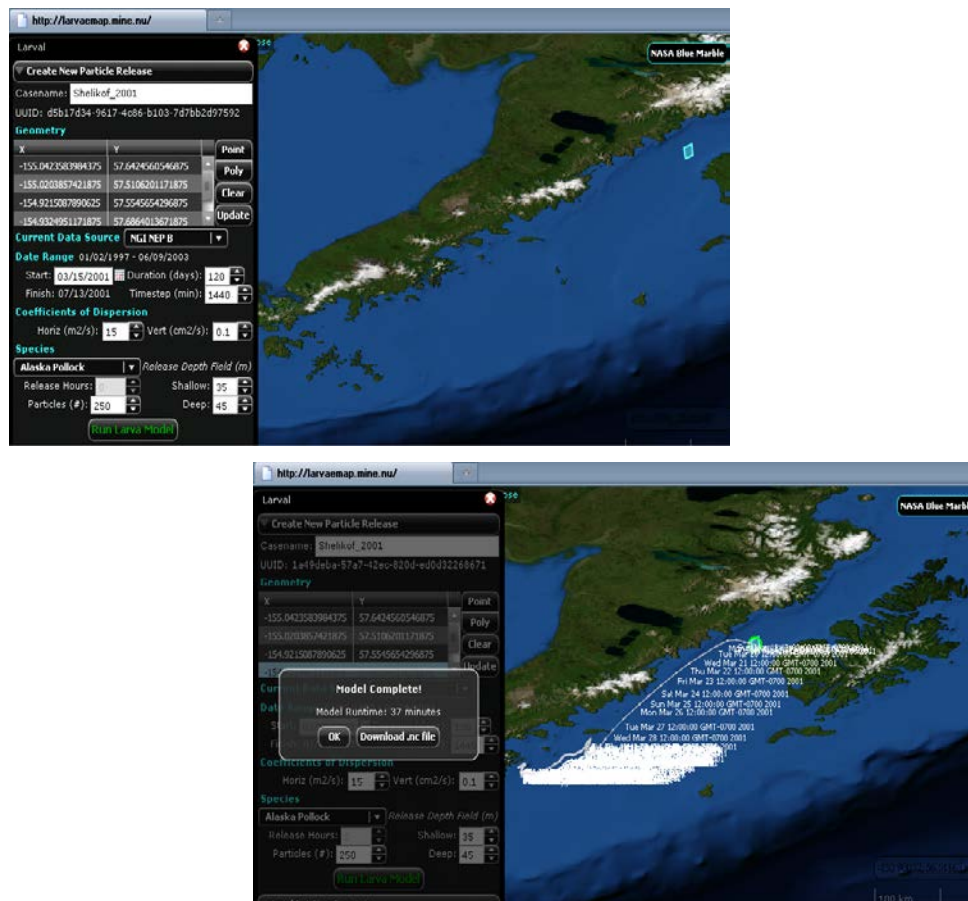


Figure 2. The LarvaMap interface. The left-hand image shows the setup of model parameters. The right-hand image shows the results with the daily centroids of the particle cloud plotted on a map and the full results available for download as a netCDF file.

Since the results are in a netCDF file, they can easily be ingested and analysed in a GIS such as ArcGIS. Figure 3 shows the output of two model runs as both a 3-D perspective plot and as a 2-D map which can also be displayed as an animation. Analyses of the tracks, such as the location of the centroid of a daily or hourly “cloud” of particles or along path plots of temperatures (from the underlying circulation models) or path lengths can be made using the tools intrinsic to the

GIS. The results could also be integrated with *in situ* environmental or biological data gathered in the same geographic locations or the results of other types of models such as wind and wave forecasts.

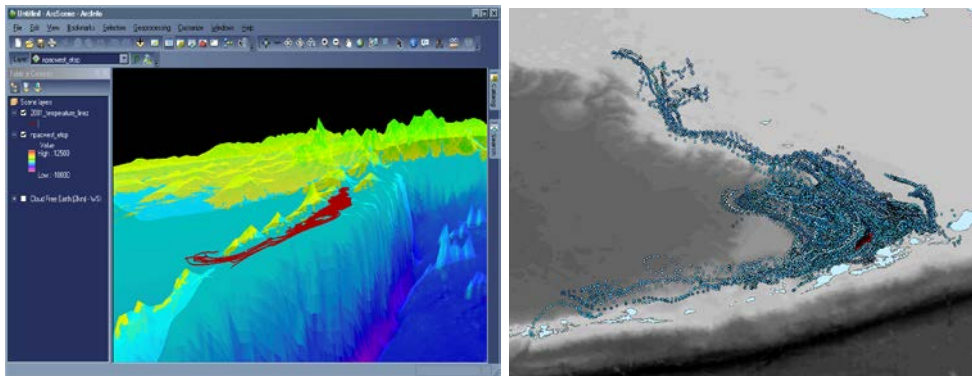


Figure 3. Visualizations of the LarvaMap output. The left-hand view is a 3-D display of the particle tracks. The right-hand display shows the positions of all the particles at all time steps of the model run. The red symbols show the particle locations for a single day and can be animated to show the travel of the particles for the entire model run.

2.3 Deploying LarvaMap in the Cloud

LarvaMap was originally designed to run on a single computer system. The speed of running models depended upon the number of CPUs available on the system and their speed. Since each CPU core calculated the results for a single particle, having multiple cores allowed the system to work on more than one particle calculation at a time. While the models are compute intensive, they do not require large amounts of memory - 512MB is sufficient. To simplify managing the output of the model, LarvaMap was designed to run models sequentially. A model run would have to be complete before the next model could be run. While this is efficient for a single user running a single model, it is not efficient for a user wanting to run multiple models simultaneously for scenario testing or multiple users wishing to run multiple models simultaneously.

Re-architecting and deploying LarvaMap in the cloud allowed the models to be scales horizontally so multiple instances could run simultaneously. The architecture for this is described in Wilcox and Vance [2011]. Deploying LarvaMap did provide some challenges for managing multiple users, multiple models, and multiple outputs. The Amazon Load Balancer was used to create and manage multiple instances of the web front end and also of the middleware web services used to pass information to the model runners. The Amazon Simple Queue Service was used to manage the scheduling of model runs and the activities of the multiple model runners. Using this, any model runner can honor a request for a new model run if it is free. It is also easy to scale out to more model instances if the load demands.

Amazon provides a variety of system configurations in its cloud services [aws.amazon.com/ec2/instance-types]. Since we needed CPU power over RAM, we choose an instance with 8 cores but only 1.7GB of RAM. Amazon charges by the instance so it was actually less expensive to have a multiple core instance rather than multiple single core instances. Having the multiple cores also allowed each model run to run faster as it could make computations on multiple particles simultaneously. Fine tuning the type of Amazon configuration we choose turned

out to be critical for keeping good response times and model performance while controlling costs.

3 CONCLUSIONS AND RECOMMENDATIONS

Running particle tracking models and integrating the results with environmental or *in situ* data can be a powerful tool to understand the early life history and success of marine fish species. But, running these models is frequently a complex and time-consuming process. An intuitive interface for parameter setting combined with harnessing the power of cloud computing resources can make running these models intuitive and rapid. This kind of model deployment can make it possible for scientists and resource managers to use these tools for “what if” scenario testing and the exploration of thought models.

In the future, we plan to expand LarvaMap by including biological behaviors and responses for the particles, currently a notable gap in our tool. We are working on a joint project to create a framework where users can specify a set of biological behaviors, such as time spent as passive particles or swimming strength or timing of the development of eyes capable of seeing well enough for predator avoidance, for a species of interest. We will also implement links to THREDDS or other data servers to allow a variety of circulation models to be implemented. These improvements will build upon a robust computational structure to create a fully functional biological model.

ACKNOWLEDGMENTS

The US National Oceanic and Atmospheric Administration (NOAA) funded the development of LarvaMap and associated work through generous support from its High Performance Computing and Communications Office and the IT Innovation Program [<http://www.cio.noaa.gov/HPCC/innovation/index.html>]. The NOAA/NMFS AFSC Recruitment Processes Program also supported the project. Deployment of LarvaMap in the cloud was supported by the US Federal Geospatial Data Committee (FGDC) via their GeoCloud Sandbox initiative [cisc.gmu.edu/scc/presentation/GeoCloud-doug-panel.pdf]. The authors thank the three anonymous reviewers for their helpful comments.

REFERENCES

- Brickman, D., G. Marteinsdottir, K. Logemann, and I. H. Harms, Drift probabilities for Icelandic cod larvae. *ICES Journal of Marine Science: Journal du Conseil Advance Access* Jan 1, 2007, 49 – 59, 2007.
- Chesapeake Community Modeling Program, <http://ches.communitymodeling.org/models.php>, viewed 2/24/2102.
- Grzechnik, M. P., and B. J. Noye, A Lagrangian-Stochastic particle tracking procedure for coastal seas. In Noye, B. J., M. D. Teubner, and A. W. Gill, editors, *Computational Techniques and Applications: CTAC-97*, 257- 264. The University of Adelaide, South Australia, World Scientific, Singapore, 1998.
- Hermann, A.J., W.C. Rugen, P.J. Stabeno, and N. A. Bond, Physical transport of young pollock larvae (*Theragra chalcogramma*) near Shelikof Strait as inferred from a hydrodynamic model, *Fish. Oceanogr.*, 5(Suppl. 1), 58-70, 1996.
- Lanksbury, J.A., J.T. Duffy-Anderson, K.L. Mier, M.S. Busby, and P.J. Stabeno, Distribution and transport patterns of northern rock sole, *Lepidopsetta polyxstra*, larvae in the southeastern Bering Sea. *Prog. Oceanogr.*, 72, 39–62, 2007.

North, E. W., Z. Schlag, R. R. Hood, M. Li, L. Zhong, T. Gross, and V. S. Kennedy, Vertical swimming behavior influences the dispersal of simulated oyster larvae in a coupled particle-tracking and hydrodynamic model of Chesapeake Bay, *Marine Ecology Progress Series* 359: 99-115, 2008.

Ribergaard, M.H., S.A. Pedersen, B. Aadlandsvik, and N. Kliem, Modeling the ocean currents on the West Greenland shelf with special emphasis on northern shrimp recruitment, *Continental Shelf Research*, **24** (13-14), 2004.

Vance, T.C., S.L. Cross, B.A. Megrey, S.M. Mesick, and C.W. Moore, GeoFish – visualization and analysis of particle tracking model output for fish and shellfish larvae, ICES 2008 Annual Science Conference - theme session R (Data Management) Halifax, Nova Scotia. 22-26 September 2008.

Wilcox, K. and T.C. Vance, Ride the wave: ocean scientists use cloud computing to model dispersion, *GeoWorld*, September 2011, 18-21, 2011.