NEW DEVELOPMENTS IN THE GENERAL NOAA OPERATIONAL MODELING ENVIRONMENT (GNOME)

CHRISTOPHER H. BARKER*

March 25, 2015

ABSTRACT

The General NOAA Operational Modeling Environment (GNOME) is an oil spill transport model developed and used by the United States' National Oceanic and Atmospheric Administration (NOAA) Emergency Response Division (ERD). It is ERD's primary tool for trajectory analysis in support of oil spill response in the United States. In addition, it is widely used by the general public for use in planning, intuition building, research, and education. In the last few years, GNOME has seen a great deal of development. This paper presents the new features, a road map for future development, and information about how third parties can use and/or extend the model.

GNOME is a particle tracking model designed for oil spill fate and transport modeling, but is highly flexible and adaptable. It can be applied anywhere in the world, receive input from virtually any oceanographic model, and handle problems of any scale and complexity. The major new features are an extensible, scriptable framework for the core model, fully integrated oil weathering, and a full featured interactive web interface for the model. In addition to the new code structures, the individual algorithms have been updated to reflect the latest understanding of spill science. This paper will outline the core features and algorithms, provide information about how others can contribute or use the model, and present a few examples of GNOME's very broad applications.

* *National Oceanic and Atmospheric Administration. Office of Response and Restoration. Emergency Response Division. Seattle WA, USA.* `Chris.Barker@noaa.gov`

1 INTRODUCTION

The National Oceanic and Atmospheric Administration (NOAA) is designated by the United States National Contingency Plan as the agency to provide scientific support to the US Coast Guard for the response to oil and chemical spills in US coastal waters. NOAA provides this support through the deployment of Scientific Support Coordinators (SSCs) stationed in the field throughout the country. In the event of a spill, NOAA's SSCs play a key role in the command post, advising the Federal On Scene Commander (FOSC). In order to provide this support, the SSCs rely on NOAA's Emergency Response Division (ERD) Technical and Scientific Support Branch, a group of modelers, oceanographers, chemists, biologists and software developers, based in NOAA's Seattle office.

One of the key roles of ERD during spills is to provide Fate and Trajectory Analysis: the modeling, analysis, and resulting forecast of where the oil (or chemical) is likely to go, and how its properties change with exposure to the environment. In order to meet this mission, NOAA ERD has developed most of its modeling software in-house since its inception in the 1970s. This was originally due to the lack of commercial options, but the tradition has continued as it is key to our mission that the operational modelers are intimately familiar with the working of the models. In addition, having the development in-house allows ERD to to be nimble in a response – sometimes fixing bugs or adding features as needed to support ongoing response.

As a result, ERDs modeling tools have been at the forefront of oil and chemical fate and trajectory modeling for decades. NOAA's GNOME model is a key tool to support response, and it has also been adapted for us for other emergency and non-emergency uses, such as hind-casting the trajectory of floating mammals, Harmful Algal Bloom (HAB) modeling, and fish and coral larval transport.

In the last few years, the core modeling tools have undergone substantial development. The major components of this development have been to re-structure the code-base to allow easier selection and updating of new algorithms, the merging of the weathering and transport models, the development of a fully cross-platform scripting framework, and new web browser-based user interface. The code is fully available as open source software to encourage contributions and use outside of NOAA.

2 HISTORY

NOAA's suite of oil spill models dates back to the 1970s with the On Scene Spill Model (OSSM). In the late 1990's a full re-implementation, using object oriented programming techniques and a modern user interface resulted in the General NOAA Operational Modeling Environment (GNOME). GNOME provided an easy to use interface for non-experts, and "location files", downloadable from ORR's web site (`http://response.restoration.noaa.gov/gnome`). Location Files provide pre-set up models for select regions, allowing minimally trained users to use the model for education, intuition building, and preparedness drills.

GNOME is focused on the transport side of the oil spill forecasting problem – where will the oil go? What shorelines are likely to be affected? To answer the fate part of the problem – How will the oil change as it weathers?, How much will remain on the water surface?, How much might evaporate, disperse, emulsify, etc? NOAA developed the Automated Data Inquiry for Oil Spills (ADIOS) model. The first version was developed in the 1980s, with ADIOS2 coming out in 2000. ADIOS2 provided a modern user interface and embedded oil properties database, and became an industry standard tool for evaluating the mass balance of spills during response and planning activities.

In the current development, updated versions of the weathering algorithms from ADIOS2 have been integrated with the transport capabilities of GNOME. The ADIOS oil database has also been included, while also providing a separate API and user interface with much improved search capabilities. The new combined model continues to support ERD's response mission, while also providing enhanced access, usability, and greater extensibility to the modeling community.

3 COMPUTATIONAL STRUCTURE

One of the key features of GNOME its ability to be applied at virtually any location and time and spatial scale. Oil spill transport is primarily a function of the winds and ocean currents. The availability of meteorological and oceanic models varies greatly with time and location, and there may be multiple models for some locations. It is key to the design of GNOME that it not be tied to any particular hydrodynamic model (or grid type), and is able to be easily configured to ingest information from multiple sources, including hand-entered data.

In order to achieve this flexibility, GNOME is built in a very modular structure, with each key component as independent as possible. In order to achieve this flexibility, the model is designed around two core structures: particle tracking, or Lagrangian Element modeling, and Linear Superposition of physical processes. Used by most oil spill modeling systems, the particle tracking approach allows GNOME to adapt to a wide variety of time and spatial scales and allows highly variable concentrations in an efficient manner without issues of numerical diffusion. Linear superposition of physical processes allows GNOME to add and remove various algorithms without implications as to how they interact. With appropriate numbers of elements and well-selected time steps, these simplifications can capture the core processes required to understand the fate and transport of an oil spill.

3.1 Core Components

GNOME is designed around a set of core components that can be added, removed, or combined as best fit the modeling problem at hand. The operational GNOME code provides multiple versions of many of these components, and third parties can build new components that might introduce a new algorithm for one particular process.

3.1.1 *Spills*

A Spill in GNOME is responsible for creating Lagrangian elements (particles)that are used to to track the location and properties of the spill being modeled. It consists of a two components: A Release object that specifies where and when elements are introduced into the model, and a Substance object that specifies the properties associated with the elements. Elements can be simple with only a location as properties, or very complex with all the properties required to support the full weathering model.

Currently available release objects include simple point and line sources (instantaneous or continuous), a plume model, pipeline leak, and pre-specified locations (for initializing from satellite or aerial observations).

3.1.2 *Maps*

A Map defines the location where the model is running – it specifies where the boundaries of the model are, including shoreline and the bottom. The Map also defines how the elements interact with the boundaries – sticking and/or refloating on the shoreline, leaving the domain, etc.

Maps can be build from hydrodynamic model grids so as to exactly match the boundaries of the model, or independently from GIS definitions of the shoreline.

3.1.3 *Movers*

Movers represent any physical process that might move the oil. These might include the wind, currents, diffusion processes, stokes drift, etc. Included with the existing code base are movers for simple isotropic diffusion (horizontal and vertical), wind drift, current patterns scaled by a time series (wind, tidal currents, river flow), and full time dependent currents from a variety of hydrodynamic models and grid types.

GNOME uses latitude / longitude for the spatial positions, so that different components can all communicate in one reference system. Vertical position is in meters down from the water surface.

3.1.4 *Weatherers*

Weatherers represent physical and chemical processes that change the properties of the oil (properties associated with the elements), or remove oil from the water surface. The GNOME code currently includes evaporation, dispersion, emulsification, sedimentation (Oil Sediment Aggregate formation), bio-degradation, and simple Dissolution.

3.1.5 *Clean Up*

Clean up objects represent response clean up options that remove oil from the water. Currently included are simple skimming and burning options.

3.2 Environment

Environment objects represent physical aspects of the environment that may not directly act on the oil, but change how the movers and weatherers act on the elements. Currently included are wind, tides, waves, and water properties such as temperature, salinity, and sediment load. THis allows the various movers and weatherers to share the same environmental parameters, but also also allows them to use independent ones if the need arises. One example is a time-averaged wind field to drive a coastal current pattern that may be different than the localized wind used to move the surface oil.

4 THE OIL DATABASE

GNOME works very closely with the ADIOS oil database. This database consist of thousands of individual oils, each with a variety of physical and chemical properties. As the data available for any given oil may be incomplete, the database also includes approximation algorithms for approximating physical properties that have not been measured. The database and approximation code are independent of the rest of the GNOME model and could be used in other applications.

4.0.1 *Pseudo Components*

The oil weathering algorithms use a pseudo component (PC) approach – the complex chemical makeup of an individual oil is approximated by breaking down the oil into a manageable set of components. Each PC is defined by a boiling point (usually from distillation data), and chemical category: Saturate, Aromatic, Resin, or Asphaltene (SARA). This allows the weathering algorithms to treat each component differently, and allows the final results to report these individual components that may effect spill impact analysis.

As SARA analysis is not always available, approximation algorithms are included, so that the final database always has the full data required for the weathering analysis.

5 SCRIPTING

The core model is written in the Python programming language, with many components written in C++ for performance. As such, fairly simple scripts can be written that set up the model and run it as required for an individual use-case. Each component can be optionally added as needed, with parameters specified. And example script with annotation, is below:

```
start_time = datetime(2013, 2, 13, 9, 0)

# Set up primary model parameters
model = Model(start_time=start_time,
              duration=timedelta(days=2),
              time_step=15*60 # fifteen minutes in seconds
              )

# adding a map from a vector land definition
# and setting the refloating half-life
model.map = MapFromBNA('./GuamMap.bna',
                       refloat_halflife=6 # hours
                       )

# Adding netcdf output for the results
model.outputters += NetCDFOutput('script_guam.nc')

# Adding a spill
model.spills += point_line_release_spill(num_elements=1000,
                    start_position=(144.664166, #longitude
                    13.441944, # latitude
                    0.0), # depth
```

```
                    release_time=start_time)

# Adding Random walk diffusion
model.movers += RandomMover(diffusion_coef=50000)

    print 'adding␣a␣wind␣mover:'

# Adding winds fromgridded wind model output
model.movers += GridWindMover(wind_file)

# Adding Currents from a gridded hydrodynamic model
model.movers += GridCurrentMover('./hycom_ouput.nc')

# Run the model:
model.full_run(log=True)
```

6 OUTPUT

GNOME has a variety of output options, including visual images and animations of the spill transport, tabular data about the oil weathering processes, and a netcdf format that can store all the element positions and data associated with the particles, allowing for any number of post-processing options.

7 EXTENDING

GNOME can be easily extended by writing new movers or weatherers that implement new algorithms. The API has been carefully designed to allow users to write a new component with minimal knowledge of how the rest of the system works. Components can be written in Python, or an language that can be called fro Python, such as Fortran, C or C++.

8 WEB INTERFACE

In addition to scripting and batch mode, an extensive easy-to use web browser based user interface has been written to facilitate easy and casual use. The model can be accessed and run on NOAA servers at `http://gnome.orr.noaa.gov`. The public NOAA version includes pre-initialized locations for use at select locations primarily in the United States. In addition, users can set up and run a non-specific location oil weathering model and access the oil database through the NOAA web interface.

The web interface can also be run locally on a users personal machine, or set up on a server at any location.

9 PARTICIPATING IN DEVELOPMENT

The GNOME source code is all open source. You can find the code on gitHub at: `https://github.com/NOAA-ORR-ERD/PyGnome`. Technical documentation is available at: http://noaa-orr-erd.github.io/PyGnome/. With the source code, anyone is free to run teh code, fix issues, add features, and re-purpose the code to work with an new system. If anyone makes a fix or addition that may be useful to others, we welcome submissions. gitHub's "pull request" feature is an excellent way to submit suggestions for review.

10 CONCLUSION

The new NOAA oils spill modeling tools have been developed to support NOAA ERD's mission, but have also been designed to be as flexible and extensible as possible.