HYPERSPECTRAL AQUATIC RADIATIVE TRANSFER MODELING USING CLUSTER COMPUTING

Anthony M. Filippi[*], Assistant Professor
Department of Geography, 3147 TAMU, College of Geosciences, Texas A&M University,
College Station, TX 77843-3147 USA
filippi@tamu.edu

Budhendra L. Bhaduri, Computational Specialist, GIST Group Leader
Thomas J. Naughton III[**], Research Associate
Amy L. King, Project Manager
Stephen L. Scott[**], Senior Research Scientist
Oak Ridge National Laboratory, One Bethel Valley Road, P.O. Box 2008, Bldg. 5600,
Oak Ridge, TN 37831 USA
bhaduribl@ornl.gov
naughtont@ornl.gov
kingal@ornl.gov
scottsl@ornl.gov

ABSTRACT

Estimating aquatic biophysical quantities from remote sensor imagery is an inverse problem, which is still unsolved. In this research, forward-modeling was considered to ultimately address the inverse problem of hydrologic optics. Above-surface hyperspectral remote-sensing reflectance can be inverted to yield bottom depth and various inherent and bottom optical properties. Retrievals can be obtained by using an artificial neural network (ANN) or similar algorithm. However, due to the paucity of measured oceanic field/organic training and testing sets, synthetic data generation is usually necessary to train such an algorithm in this application domain. Such modeled data sets, which can be quite large, are typically computationally expensive and time-consuming to generate. To accomplish this efficiently, a cluster computing method was developed to produce a forward-modeled, hyperspectral bio-optical database, based on radiative transfer (RT) theory. This method enables multiple instances of the *Hydrolight* RT model to be run in parallel. The *Hydrolight* runs are independent of one another, making them good candidates for parallelization, thus reducing the time required to generate the synthetic training/testing data. The resultant input-output pairs in the forward-modeled database can be used to develop remote-sensing inverse algorithms. Bathymetry and water-column and bottom optical models can thus be developed while minimizing the need for *in situ* data, which is useful in an operational environment. Simulated remote-sensing reflectance pseudodata are of utility in hydrologic inversion since adequate field data for model development are often difficult or impractical to acquire.

INTRODUCTION

Predicting optical properties of the water column from remote sensor data is important in many application domains, including providing biophysical input parameters to aquatic ecological models, as well as providing critical information in a military/security-related operational environment. Estimating aquatic biophysical quantities from remote sensor imagery is an inverse problem. In this research, the inverse problem of hydrologic optics was addressed by considering the forward ocean optics problem. The *forward problem* of hydrologic optics deals with finding the radiance distribution of a water body given the inherent optical properties (IOPs) and the physical properties of the boundaries. This problem is essentially solved, as the computed radiances are restricted in accuracy primarily according to the accuracy of the input IOPs and boundary conditions, as well as allowable computation time. However, the *inverse problem*, where the task is to determine the IOPs, given radiometric measurements of an

[*] This research was supported in part by an appointment to the U.S. Department of Energy (DOE) Higher Education Research Experiences (HERE) for Faculty at the Oak Ridge National Laboratory (ORNL) administered by the Oak Ridge Institute for Science and Education. This project was conducted using the joint resources of the Geographic Information Science & Technology Group (GIST), Computational Sciences & Engineering Division and the Computer Science & Mathematics Division, ORNL.

[**] Research supported by the Mathematics, Information and Computational Sciences Office, Office of Advanced Scientific Computing Research, Office of Science, U. S. Department of Energy, under contract No. DE-AC05-00OR22725 with UTBattelle, LLC.

aquatic light field, is still unsolved (Mobley, 1994). Given the above-surface hyperspectral remote-sensing reflectance, inversion can yield estimates of bathymetry, IOPs and bottom optical properties (BOPs), and constituent concentrations (e.g., chlorophyll concentration, etc.). One method for obtaining such retrievals lies with artificial neural networks (ANNs) or similar algorithms, where spectral remote-sensing reflectance $R_{\mathrm{rs}}(\lambda)$ data serve as input vectors, and the desired water properties and possibly bathymetry constitute typical output variables. However, due to the paucity of measured oceanic field/organic training and testing sets, synthetic data generation is usually necessary to train such an algorithm in this application domain. This paucity of field data is related to the logistical difficulties of acquiring *in situ* data in dynamic oceanic environments. Given the common impracticality and expense associated with *in situ* ocean optics data collection, it is often necessary not to relay on such data for remote-sensing inversion model development. Simulated $R_{\mathrm{rs}}(\lambda)$ pseudodata are thus useful for this application. However, such modeled data sets are typically computationally expensive and time-consuming to generate. To accomplish this in a temporally efficient manner, a cluster computing method was developed to produce a forward-modeled, hyperspectral bio-optical database, based on radiative transfer (RT) theory.

One method of developing ANN-based models, for instance, would be to use observed, measured data sets that cover a wide range of environmental conditions for ANN training and testing; however, if such a database is limited in scope, as is often the case, it may not yield models that are sufficient for broad applicability or spatial extension. Models more universal or regional in scope are often desired, and many ANN training/testing pairs are also often needed for model development. The *Hydrolight* radiative transfer model (Mobley et al., 1993; Mobley, 1994) offers the capability of performing experiments in a constrained environment, where pseudodata can be generated for any selected optical regime.

The *Hydrolight* software package (Sequoia Scientific, Inc.) was utilized in this research to address the forward problem in hydrologic optics. *Hydrolight* is a numerical model used to calculate radiance distributions and derived quantities for natural water bodies (Mobley and Sundman, 2001a; 2001b). Radiance $L(\mathrm{z},\theta,\phi,\lambda)$, which is a function of geometric depth z, nadir angle θ, azimuthal angle ϕ, and wavelength λ, is the key quantity that describes the time-independent, 1-D oceanic light field. To predict the spectral radiance, *Hydrolight* numerically solves the integro-differential radiative transfer equation (RTE), as well as its boundary conditions, using a computationally efficient invariant imbedding method (Mobley et al., 1993; Mobley, 1994). The standard form of the monochromatic RTE is (Mobley, 1994):

$$\mu \frac{dL\left(z;\hat{\xi};\lambda\right)}{dz} = -c\left(z;\lambda\right)L\left(z;\hat{\xi};\lambda\right) + \int_{\Xi} L\left(z;\hat{\xi}';\lambda\right)\beta\left(z;\hat{\xi}';\rightarrow \hat{\xi};\lambda\right) d\Omega\left(\hat{\xi}'\right) + S\left(z;\hat{\xi};\lambda\right), \qquad (1)$$

where units = (W m^{-3} sr^{-1} nm^{-1}); μ is the cosine of the zenith angle (dimensionless); c is the beam attenuation coefficient (m^{-1}); $\hat{\xi}'$ is the incident photon direction; the unit sphere Ξ is the set of all directions $\hat{\xi}$; β is the volume scattering function (VSF) (m^{-1} sr^{-1}); Ω is a solid angle (sr); and S is a source function. More computationally efficient extensions to *Hydrolight* have been developed in research mode (Liu et al., 1999; Liu et al., 2002), though the original *Hydrolight* program was used in the present research, providing very accurate forward-modeled data (i.e., at the full level of accuracy that *Hydrolight* is capable of) (Mobley and Sundman, 2001a). The purpose of this research was to establish the degree to which the wall-clock time for hyperspectral Hydrolight simulations can be reduced via a cluster computing approach. This may be critical for large or time-sensitive high-dimensional modeling jobs.

METHODS

This paper discusses work based on *Hydrolight* version 4.1, which is a sequential Fortran model available for UNIX based systems[*] (Mobley and Sundman, 2001a). *Hydrolight* also runs on Windows OS-based systems and provides a Graphical User Interface (GUI) for this platform only to assist in the execution of the model. However, this GUI is only a frontend and is not directly tied to the actual numerical simulator. A simple forward simulation example for case 1 water was considered in this research, where computed $R_{\mathrm{rs}}(\lambda)$ pseudodata were the output.

[*] The UNIX code base in v4.1 does not differ from that in v4.2, as primary feature addition in v4.2 was the Windows frontend for *Hydrolight*.

The basic approach to running a simulation with *Hydrolight* involves the creation of an input file (*Iroot.txt*) that contains various control parameters to the model. Additionally, a Fortran source file (*root.for*) is generated that includes the necessary routines based on the parameters selected for the simulation. Once these files have been setup and placed in a predefined directory structure the actual Fortran source code is compiled and placed in the execution directory. A brief summary and outline of the steps for use on UNIX/Linux follows:

Note: The top-level directory is assumed to be at $H41_HOME. The *Iroot.txt* and *root.for* are the generic names which would normally be named more meaningfully for individual runs, such as "Itest3-nwave-5.txt" and "test3-nwave-5.for". Note that all paths and directory names are predefined by *Hydrolight*.

Step 1 Place the *Iroot.txt* file (input parameters) into $H41_HOME/run/batch/.

Step 2 Place the *root.for* file (Fortran incl file) into $H41_HOME/maincode/batch/.

Step 3 Create a symbolic link (or replace existing file): incfiles.f → root.for, e.g.,
cd $H41_HOME/maincode/ ; ln -s batch/root.for incfiles.f

Step 4 Build the maincode, e.g.,
cd $H41_HOME/maincode/ ; make -f makeunix

Step 5 Run the test, e.g.,
cd $H41_HOME/run ; ../maincode/maincode.exe < batch/Iroot.txt

Step 6 After the program completes and returns the "Success" message, collect results from $H41_HOME/output/*.

Once this process was established, the input parameters (*Iroot.txt*) and Fortran include file (*root.for*) that were generated using the Windows GUI could be transferred to the UNIX/Linux based system. The eXtreame Tennessee Oak Ridge Cluster (XTORC) was used for these experiments. The XTORC system is maintained in the Computer Science and Mathematics Division at Oak Ridge National Laboratory (ORNL). The system is comprised of sixty-four 2Ghz Pentium IV compute nodes and two 1.7Ghz Pentium IV head nodes, all running Linux. The cluster has both FastEthernet (100Mb) and Gigabit Ethernet (1Gb) interconnects and each node has a 40GB local disk drive and 768MB of random access memory. The head nodes are dual-homed and have 120GB of local disk space, which is partially shared via Network File System (NFS) to the compute nodes, and 1GB of random access memory. The system is used for computer science research and development for cluster management software (Mugler et al., 2003; Mugler et al., 2004; Bode et al., 2005).

The specific objectives for this initial investigation were to (a) decrease the wall-clock time for running *Hydrolight* simulations, while (b) limiting the changes to the software base for maintenance purposes. Therefore, our first approach involved the parallelization of the batched simulations, which are typically run in a sequential fashion on Windows workstations that are equipped with the *Hydrolight* software. This involved the modification of the build system to facilitate compilation on the Linux cluster for the Fortran code based upon the routines used for the individual run. Additionally, the actual input parameter file was modified to break the selected wavelengths used in the computation into *N* equal parts. These subdivided wavelengths were run in parallel without modification to the Fortran code base[*] .

RESULTS AND DISCUSSION

The initial approach to lower the time for *Hydrolight* simulations limited modifications for parallelization to the build/execution process and the input parameters (*Iroot.txt*). As mentioned in Methods Section, this was to assist with long-term maintenance and initial investigation costs. A basic experiment compared the wall-clock time for a single simulation using both sequential and parallel approaches. The sequential approach used a single compute node from the XTORC cluster running a basic simulation over 35 wavelengths that were defined in the input parameter file (*Iroot.txt*). The parallel approach used seven compute nodes from the XTORC cluster running the same basic simulation but over a subset of the overall wavelengths, which was achieved by slightly augmenting the input

[*] Some experimentation was undertaken to augment the Fortran code base to isolate particular parameters for customized logs, but this was not required or directly applicable to the experiment for wavelength parallelization.

parameter file. The results for these runs are shown in Table 1; this simple experiment suggests a 77% reduction in wall-clock time with the parallel-processing method, relative to the sequential approach.

Table 1. Wall-clock time for sequential versus parallel simulation over 35 wavelengths.

Approach	Wall-Clock Time	Node Details
Sequential	3 min, 51 sec	1 Node, P4 2Ghz
Parallel	53 sec (1 outlier)	7 Nodes, P4 2Ghz

However, note that in the parallel version of the experiment, one machine was delayed for undetermined reasons and therefore slowed the overall wall-clock time. The wall-clock times for the individual nodes in the parallel run are shown in Table 2 and indicate that even though the overall wall-clock time was approximately 53 seconds, the majority of the computations (six nodes, 30 wavelengths) were completed in approximately 30-38 seconds. The execution of the simulations was driven by the cexec (parallel execution) command which runs the command across a set of nodes in the cluster (Luethke et al., 2002; ORNL, 2006). In this example, nodes 50-52, 54-56, and 58 (seven nodes) were used with a timing command issued for the overall execution context and individual timing commands for each sub-portion of the execution, i.e., 5-wavelengths per node calculation. (See the Appendix: Detailed Listings for further information.)

Table 2. Individual wall-clock information, sorted by elapsed time (m = minutes; s = seconds).

Node Number	Wall-Clock Time	CPU Usage
node50.xtorc	0m33.397s	99%CPU
node51.xtorc	0m30.418s	98%CPU
node52.xtorc	0m28.475s	97%CPU
node58.xtorc	0m27.071s	98%CPU
node54.xtorc	0m28.196s	98%CPU
node55.xtorc	0m38.792s	99%CPU
node56.xtorc	0m51.186s	98%CPU

It is important to note that the GNU g77 compiler, which is not an optimizing compiler, was utilized in this research. All tests, however, were executed using this *same* non-optimizing compiler. The GNU g77 tool may not produce the most optimal executable code compared with other popular parallelizing Fortran compilers; the implication is that the temporal results in this study could possibly be improved if an optimizing compiler were employed.

Also, while this experiment only included one *Hydrolight* simulation run, a radiative transfer (RT)-modeled database useful for inverse model-development may include tens of thousands of RT runs in order to include a sufficiently large range of enviromental conditions in the bio-optical database necessary for spatial extension, including variability in water properties and constituent concentrations, bottom depth, solar and viewing geometries, and meteorological conditions. Thus, scaling this grid-computing method up to tens of thousands of runs using many more nodes in the grid would result in a marked decrease in database generation time relative to the sequential method.

CONCLUSION

A temporally efficient cluster computing method for generating forward-modeled, hyperspectral bio-optical databases was presented. Such databases contain input-output relationships among inputs (e.g., inherent optical properties (IOPs) of the water column, constituent concentrations, water/bottom depth, bottom albedo, and other enviromental parameters) and computed spectral water-leaving radiances or spectral remote-sensing reflectances. These pseudodata can be utilized for remote-sensing inversion algorithm development. Planned investigations will incorporate many thousands of RT runs and more processing nodes to better establish the improvements possible in temporal efficiency with the parallel-processing approach. In addition, future experimentation with optimizing compilers is expected to decrease computation time even further.

REFERENCES

Bode, B., Bradshaw, R., DeBenedictus, E., Desai, N., Duell, J., Geist, G. A., Hargrove, P., Jackson, D., Jackson, S., Laros, J., Lowe, C., Lusk, E., McLendon, W., Mugler, J., Naughton, T., Navarro, J. P., Oldfield, R., Pundit, N., Scott, S. L., Showerman, M., Steffen, C., and Walker, K. (2005). Scalable System Software: A component-based approach. *Journal of Physics: Conference Series*, 16:546-550.

ORNL (Oak Ridge National Laboratory). (2006). Cluster Command & Control (C3) Power Tools, http://www.csm.ornl.gov/torc/C3. Last accessed 10 February 2006.

Luethke, B., Naughton, T. and Scott, S. L. (2002). C3 Power Tools: The Next Generations... In *DAPSYS 2002*, September 29-October 2, 2002, Kluwer Academic Publishers, Johannes Kepler University, pp. 82-89.

Liu, C.-C., Carder, K. L., Miller, R. L., and Ivey, J. E., (2002). Fast and accurate model of underwater scalar irradiance. *Appl. Opt.*, 41(24):4962-4974.

Liu, C.-C., Woods, J. D., and Mobley, C. D. (1999). Optical model for use in ocean ecosystem models. *Appl.Opt.*, 38(21):4475-4485.

Mobley, C.D., (1994). *Light and Water: Radiative Transfer in Natural Waters*. Academic Press, San Diego, 592 p.

Mobley, C. D., Gentili, B., Gordon, H. R., Jin, Z., Kattawar, G. W., Morel, A., Reinersman, P., Stamnes, K., and Stavn, R. H. (1993). Comparison of numerical models for computing underwater light fields. *Appl. Opt.*, 32: 7484-7504.

Mobley, C. D., and Sundman, L. K., (2001a), *Hydrolight 4.2: Technical Documentation*. Sequoia Scientific, Inc., Redmond, WA, second printing edition, October 2001, 79 p.

_____. (2001b). *Hydrolight 4.2: Users' Guide.* Sequoia Scientific, Inc., Redmond, WA, second printing edition, October 2001, 88 p.

Mugler, J., Naughton, T., and Scott, S. L. (2004). The Integration of Scalable Systems Software with the OSCAR Clustering Toolkit. In *Proceedings of 2nd Annual OSCAR Symposium (OSCAR 2004)*, Winnipeg, Manitoba Canada, May 16-19 2004.

Mugler, J. Naughton, T., Scott, S. L., Barrett, B., Lumsdaine, A., Squyres, J. M., des Ligneris, B., Giraldeau, F., and Leangsuksun, C. (2003). OSCAR Clusters. In *Proceedings of the 5th Annual Ottawa Linux Symposium OLS'03)*, Ottawa, Canada, July 23-26, 2003.

APPENDIX: DETAILED LISTINGS

Full cexec listing

The full listing for the execution across seven nodes of the XTORC cluster to perform the simulation over five wavelengths per node in parallel is given below.

```
node0: $ time cexec /home/tjn/para_hydrolight/myruns.pl
************************ hydro ************************
--------- node50.xtorc---------
real    0m33.397s
user    0m32.920s
sys     0m0.180s
32.92user 0.18system 0:33.40elapsed 99%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (1631major+2209minor)pagefaults 0swaps
--------- node51.xtorc---------
real    0m30.418s
user    0m29.800s
sys     0m0.170s
29.80user 0.17system 0:30.42elapsed 98%CPU (0avgtext+0avgdata 0maxresident)k
```

```
0inputs+0outputs (1626major+2208minor)pagefaults 0swaps
--------- node52.xtorc---------
real    0m28.475s
user    0m27.700s
sys     0m0.170s
27.70user 0.17system 0:28.47elapsed 97%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (1626major+2208minor)pagefaults 0swaps
--------- node58.xtorc---------
real    0m27.071s
user    0m26.520s
sys     0m0.180s
26.52user 0.18system 0:27.07elapsed 98%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (1626major+2212minor)pagefaults 0swaps
--------- node54.xtorc---------
real    0m28.196s
user    0m27.520s
sys     0m0.190s
27.52user 0.19system 0:28.20elapsed 98%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (1626major+2208minor)pagefaults 0swaps
--------- node55.xtorc---------
real    0m38.792s
user    0m38.600s
sys     0m0.140s
38.60user 0.14system 0:38.79elapsed 99%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (1626major+2205minor)pagefaults 0swaps
--------- node56.xtorc---------
real    0m51.186s
user    0m50.500s
sys     0m0.170s
50.50user 0.17system 0:51.19elapsed 98%CPU (0avgtext+0avgdata 0maxresident)k
0inputs+0outputs (1626major+2215minor)pagefaults 0swaps
real    0m53.496s
```

Simple ’myruns.pl’ script

Below is a simple Perl script used to run the actual *Hydrolight* simulation on a given node using the hostname to determine which section of the wavelength to run. The time command is used to report the elapsed time and cpu utilization.

```
#!/usr/bin/perl
my $host=`hostname`;
chomp($host);
my $cmd = "/home/tjn/para_hydrolight/nodes/$host.myrun.sh";
system("time $cmd");
```

Individual node script / Hydrolight startup

A very basic BASH shell script to run the *Hydrolight* simulation for a given section of the wavelength that is conveyed via the node specific input parameter file, e.g., *Iugex2_nwave-5_350-400.txt*.

```
#!/bin/sh
time $HYDRO_HOME/run/run.pl \
  $HYDRO_HOME/run/batch/Iugex2_nwave-5_350-400.txt \
  >& rslts
```