Exxon Valdez Oil Spill
Restoration Project Annual Report

Information Systems and Model Development

Restoration Project 95320-J (SEADATA)
Annual Report

This annual report has been prepared for peer review as part of the *Exxon Valdez* Oil Spill Trustee Council restoration program for the purpose of assessing project progress. Peer review comments have not been addressed in this annual report.

Principal Investigator:	Vincent Patrick	Prince William Sound Science Center
Collaborators:	Jennifer R. Allen	Prince William Sound Science Center
	Stephen Bodnar	Prince William Sound Science Center
	Gary Drew	National Biological Service, Anchorage
	David L. Eslinger	Inst. Marine Sci., U. of Alaska Fairbanks
	Charles S. Falkenberg	Adv. Visualization Lab., U. of Maryland
	Edward Jin	Prince William Sound Science Center & U. of Toronto
	Ravi Kulkarni	Prince William Sound Science Center & Adv. Visualiz. Lab., U. of Maryland
	Doran M. Mason	Prince William Sound Science Center & Limnology Lab, U. of Wisconsin
	Christopher N. K. Mooers	Dir., Ocean Pollution Research Center, Rosenstiel Schl. Marine & Atmos. Sci., U. of Miami
	Ricardo H. Nochetto	Prince William Sound Science Center & Mathematics Dept., U. of Maryland
	Sridhar P. Rao	Prince William Sound Science Center
	Jia Wang	OPRC, Rosenstiel School, U. of Miami

Prince William Sound Science Center
PO Box 705
300 Breakwater Avenue
Cordova, AK 99574

April 1995

Exxon Valdez Oil Spill
Restoration Project Annual Report

Information Systems and Model Development

Restoration Project 95320-J (SEADATA)
Annual Report

This annual report has been prepared for peer review as part of the *Exxon Valdez* Oil Spill Trustee Council restoration program for the purpose of assessing project progress. Peer review comments have not been addressed in this annual report.

Principal Investigator:	Vincent Patrick	Prince William Sound Science Center
Collaborators:	Jennifer R. Allen	Prince William Sound Science Center
	Stephen Bodnar	Prince William Sound Science Center
	Gary Drew	National Biological Service, Anchorage
	David L. Eslinger	Inst. Marine Sci., U. of Alaska Fairbanks
	Charles S. Falkenberg	Adv. Visualization Lab., U. of Maryland
	Edward Jin	Prince William Sound Science Center & U. of Toronto
	Ravi Kulkarni	Prince William Sound Science Center & Adv. Visualiz. Lab., U. of Maryland
	Doran M. Mason	Prince William Sound Science Center & Limnology Lab, U. of Wisconsin
	Christopher N. K. Mooers	Dir., Ocean Pollution Research Center, Rosenstiel Schl. Marine & Atmos. Sci., U. of Miami
	Ricardo H. Nochetto	Prince William Sound Science Center & Mathematics Dept., U. of Maryland
	Sridhar P. Rao	Prince William Sound Science Center
	Jia Wang	OPRC, Rosenstiel School, U. of Miami

Prince William Sound Science Center
PO Box 705
300 Breakwater Avenue
Cordova, AK 99574

April 1995

Information Systems and Model Development

Restoration Project 95320-J
Annual Report

Study History: The Sound Ecosystem Assessment (SEA) Program is based upon the *Sound Ecosystem Assessment Initial Science Plan and Monitoring Program,* Rpt No. 1, Nov. 24, 1993. It began in April 1994 (Restoration Project 94320) and was continued in FY95 (Restoration Project 95320). The Information Systems and Model Development Project (SEADATA) is a Restoration Project (94320-J and 95320-J) within the SEA Program. Progress for FY94 is presented in Ch. 6 of the SEA Draft 1994 Final Report. An paper on the SEA database appears in the *Proceedings of Visualization '95*, IEEE, Atlanta, GA, Nov. 1995 (C. Falkenberg and R. Kulkarni, Using spatial access methods to support the visualization of environmental data). A paper on the ocean circulation model appears in *Conf. on Coastal Oceanic and Atmospheric Prediction*, Amer. Meteorological Soc., Atlanta, GA, Jan. 1996 (J. Wang and C. N. K. Mooers, Modelling Prince William Sound ocean circulation).

Abstract: The four main development areas of SEADATA are reviewed: database, Internet and Web based collaboration tools, circulation model, and the fish model. Each of these has reached important milestones, but each is at midstage and facing ahead tasks that will be equally critical. For the database, the technically demanding tasks of the initial architecture have been resolved. The task ahead is to make everything fit within the available resources. The Web tools that were implemented were a significant accelerator for SEA. But more is required and the technology has now expanded unbelievably. The next steps will be all new terrain. The circulation model has been successfully implemented in a remarkably short time. The task ahead of validation and refinement will become increasingly difficult as the limitations of the available data for Prince William Sound become clearer. The fish model now has an efficient scipting code, a new algorithm, and the major parts of its foraging-bioenergetics model. It now must put these together to get the desired mortality estimates. The one message that came consistently from all quarters in this report is the need for more and better communication and coordination.

Key Words: bioenergetics, circulation model, collaborative software, database, diffusion-taxis, dispersion, *Exxon Valdez,* Mellor-Blumberg, Pacific herring, packet-radio, physical-biological model, pink salmon, Prince William Sound, Princeton Model, SEA, Sound Ecosystem Assessment, visualization, World Wide Web

Citation: Patrick, E. V., J. R. Allen, D. L. Eslinger, C. S. Falkenberg, D. M. Mason, C. N. K. Mooers, R. H. Nochetto, S. P. Rao, and J. Wang. 1995. Information Systems and Model Development Project (SEADATA) of the Sound Ecosystem Assessment Program, *Exxon Valdez* Oil Spill Restoration Project Annual Report (Restoration Project 95320-J), Prince William Sound Science Center, Cordova, Alaska.

Table of Contents

List of Figures

EXECUTIVE SUMMARY

The four main development areas of SEADATA are reviewed: database, Internet and Web based collaboration tools, circulation model, and the fish model. Each of these has reached important milestones, but each is at midstage and facing ahead tasks that will be equally critical. For the database, the technically demanding tasks of the initial architecture have been resolved. The task ahead is to make everything fit within the available resources. The Web tools that were implemented were a significant accelerator for SEA. But more is required and the technology has now expanded unbelievably. The next steps will be all new terrain. The circulation model has been successfully implemented in a remarkably short time. The task ahead of validation and refinement will become increasingly difficult as the limitations of the available data for Prince William Sound become clearer. The fish model now has an efficient scripting code, a new algorithm, and the major parts of its foraging-bioenergetics model. It now must put these together to get the desired mortality estimates. The one message that came consistently from all quarters in this report is the need for more and better communication and coordination.

INTRODUCTION

The Information Systems and Model Development Project (SEADATA) is one of the original projects in SEA. SEADATA was organized to deliver to the Program numerical models, a database, computing resources and networks, data visualization, computer- and network-based resources for remote collaborations and interactions, and selected sensor technologies that were judged to be sufficiently mature to have immediate impact on the cost-effectiveness of the final SEA products for long-term monitoring.

As the SEA Program draws nearer to its climax the success of SEADATA and of the newly formed Trophodynamics project for plankton model development become increasingly important to the goals of SEA. Those goals are a) to quantitatively describe, model, and numerically simulate the time evolution of those parts of the Prince William Sound ecosystem that determine the survival and growth of juvenile pink salmon and Pacific herring; and b) to deliver that knowledge to the Trustee Council in formats suitable for use in the restoration objectives of the Council. SEA has argued that an ecosystem perspective is not only necessary but that it is also sufficient for obtaining at the conclusion of the work a restoration tool that describes the status of the injured resource and how that status is likely to change in response to changes in the ecosystem. In particular, SEA is to deliver a restoration tool that will aid in assessing how recovery may be affected by manipulation of the system through resource management policies. For SEA to reach this goal it is necessary that SEADATA and Trophodynamics deliver the model components, the database, and the connectivity resources whereby the SEA collaboration can produce and validate the SEA coupled ecosystem model.

This report reviews the progress of the collaborators in SEADATA during FY95 toward meeting the goals and objectives of SEADATA and SEA. We present an updated and more complete project plan for the development of the SEA numerical models, and then review each of the parts of SEADATA through manuscripts, reports, and project documentation that have been prepared during FY95. These taken together show that SEADATA is very close to conforming to the schedule originally envisioned in the SEA Plan. At the close of FY95 the numerical models planned in the spring of 1994 were operational realities. The internetworking resources now function such that all of SEA is collectively and continuously collaborating online. And major strides have been made in the development of the SEA database during FY95, with the first key components of the database now operational.

FY95 was a year in which there was substantial cost savings to the project through cost sharing. Approximately twenty and forty percent of the costs for the fish model development and for the database development were provided from sources other than EVOS, respectively. This high degree of leveraging was in part possible because the work addressed fundamental development problems with broad applicability. As these development problems are resolved and the work becomes increasingly focused on issues specific to the restoration objectives, there will not be the same opportunities for cost sharing. Increased effort is now required to identify and secure comparable cost-sharing.

OBJECTIVES

The purpose of SEADATA is to deliver to SEA

1 numerical models;
2 database and network resources;
3 selected *measurement technologies* that were judged to be sufficiently mature to be of probable benefit for realizing low-cost long-term monitoring of recovering resources.

FY95 was the second year for the SEADATA project. For that period the Detailed Project Description identified the following task areas.

1. **Information Systems:** Data management for SEA data; local, wide, and metropolitan area networking; data resources required by SEA (realtime and historical AVHRR, weather, stream gauge); scientific database with ecosystem scale query and visualization tools; and SEA coordination utilities.
2. **Numerical Model Development:** Ocean circulation model for Prince William Sound; physical-biological model (time varying phytoplankton and zooplankton spatial distributions); fish model (time varying distributions relative to space and size of interacting species).
3. **Field Data Communications:** Packet-radio repeater network for near realtime data to and from field surveys.
4. **Sampling Technologies:** Implementation of technologies of critical significance for adequate yet affordable long term monitoring.
5. **Interim Products:** Nowcasts and forecasts from models and model parts that are still in development but that are useful for hypothesis testing, adaptive sampling trials, and model testing and parameterization.

This document reviews four task areas: database development, coordination resources, ocean model development, and fish model development. Areas that are now within the Trophodynamics project will be reviewed by D. L. Eslinger, the principal investigator of the Trophodynamics project, in a separate report. The Trophodynamics task areas are the development of the physical-biological models (i.e., the plankton models) and satellite remote sensing.

Task areas that have required primarily technology implementation will not be reviewed in detail other than to note here their completion. Under Sampling Technologies, during FY95 the Aquashuttle and its onboard CTD and Optical Plankton Counter (OPC) were fully integrated into the SEA field survey design and transitioned out of SEADATA. The physical oceanography project now operates and maintains this equipment, and it processes and delivers to SEA the datasets obtained with these devices. The repeater network objectives of the Field Data Communications area have been completed and the network is operational.

METHODS

Project Plan for the SEA numerical models

The overall plan for the SEA model is shown graphically in Figure 1. The charting method used in the figure is a modification of the standard PERT chart. The first version of this chart was created to document a review of SEADATA conducted by Dr. Andy Gunther in June, 1995, in Cordova. Figure 1 is a second version of Dr. Gunther's chart. It is an attempt to provide a more comprehensive chart that can be used for both planning, reporting, and project tracking. The second version provides in a single graphic the approach, present status, the linkages and dependencies, schedules for development and model output, validation stages, and the major endpoints. In this section the chart is used to review the methods for the model development and to provide a brief status report.

The two major endpoints of the model development effort are shown in Figure 1: (1) the coupled models for juvenile fish and (2) the specifications for a continuously running "facility" whereby data is acquired and the models are used and maintained. The method to realize the coupled model endpoint is to conduct in parallel three separate tracks of model development and then at specified stages of the development to couple the models. The three tracks are the ocean circulation model for PWS, the plankton or physical-biological model, and the nekton or fish model. This method exploits the natural partitioning of the modelling effort that occurs due to different modelling issues, different technical methods, and different human resources for each of the tracks. The final coupling of these tracks is shown in the chart as an endpoint in 1998, but the coupling is in fact carried out in stages that are indicated in the figure.

The endpoint for each of the tracks individually is somewhat different. The ocean circulation model provides information that "forces" the other models but is not dependent upon the other models. Consequently, the endpoint for the ocean model track is in fact a "stand-alone facility" with a wide range of applications. This "stand-alone" aspect of the endpoint is noted in the chart by an entry in the 1998 time period titled *98 online nowcast-forecast system*. The various products that the facility will provide for SEA appear twice in Figure 1: there is an entry at the time it is first implemented and a second entry in the catalog of functions in 1998 titled *ecosystem assessment and tracking tools*.

Both the fish and the plankton models are strongly dependent upon the physical environment. Realistic numerical simulations from the biological models depend upon the availability of realistic simulations of the physical system. The large "X" in 1998 represents how the three parts of the model are "mixed" in the coupled model.

The development sequence shown in the chart reflects a bottom-to-top schedule. This is to accommodate the differing degrees of maturity of the various modelling technologies. Ocean models have been under development for decades, combined physical and plankton models for a decade, and combined fish-plankton-physical models for less than a decade. A second rationale for sequencing the effort from bottom-to-top is to promote securing as soon as possible those milestones most readily achieved. Under this sequencing approach the forcing effects due to the circulation processes on plankton or larvae are available first, then the impact upon plankton or larvae abundance and distribution, then the coupled response of plankton and fish to the circulation processes. For example, a catalog of candidate river-lake scenarios are available first. This is followed by the consequences to distribution and growth of the macrozooplankton bloom. This is then coupled to fish through growth and predator switching processes. There are similar bottom-to-top schedules for herring overwintering, herring larval drift, and herring summer growth.

Figure 1 is current regarding the status of the models at the end of FY95. The work during FY94 is not shown in detail. Instead for each of the modelling efforts there is a single entry summarizing the general approach and the status at the end of FY94. The progress through FY95 is shown in greater detail. The following is a brief explanation of the status summary shown in Figure 1.

During the summer of 1994 agreement was reached with Dr. Chris Mooers (RSMAS, University of Miami) to collaborate in SEA. Dr. Mooers and Dr. D.-S. Ko were then just completing a numerical simulation facility for the Florida Straits (Mooers and Ko 1994) that was very similar to what was needed in SEA. Work was to begin in October, 1994, to develop a Mellor-Blumberg model for Prince William Sound. The principal features of the approach are shown under *initial selections* for the ocean model in Figure 1.

A funding delay in FY95 moved the start date from October 1, 1994, to late in February, 1995. This delay was then extended to April due to a gap between the departure of Dr. Ko and the arrival of Dr. Jia Wang at RSMAS. Dr. Wang made remarkably rapid strides and the model objectives were all achieved by the end of FY95. The model was setup and configured and running by early summer. By the end of FY95 the ocean model was operating with the features listed in *model config. 95*. Simulations were run, and work then began on the problem of further model realism and model validation.

The principal features of the approach for the fish model are shown under *initial selections* for the fish model in Figure 1. Roughly speaking the fish model extends to the nekton trophic level the model approach used in so-called "physical-biological" ecosystem models. Examples of these latter are McCreary *et al* (1995), Gallacher and Rochford (1995), and Eslinger (1990). Physical-biological models use systems of evolution equations (i.e., partial differential equations in space and time) to describe plankton population densities and growth. This similarity to the structure to the ocean circulation model equations is convenient in combined models. The fish models are evolution equation models suitable for nekton, namely coupled systems of diffusion-taxis equations (Mason and Patrick, 1993).

The project plan for FY94 and the initial brief project plans for FY95 were constructed with more effort allocated to the ocean model. The reviews of the initial SEA Plan in February 1994 had recommended caution regarding the proposed modelling approach. To minimize risk a weighting of 3:1:1 for ocean, plankton, and fish modelling, respectively, was used in the brief project plan and budget for FY95. The October 1994 reviews, however, emphasized the importance of all of the models. In particular, the reviews emphasized that during FY95 the models, including the fish and plankton models, were expected to provide guidance to the field surveys. Since commitments were in place it was not possible to totally restructure the allocation of effort. However, to the extent possible the work on the fish model was accelerated. On the other hand, because of the allocation of effort it was a near certainty that by mid-year there would be circulation model output that could be used along side the FY95 field sampling. The delay in funding, however, slipped the start time such that the first circulation model results were not available until shortly after the descent of the macrozooplankton.

By the close of FY95 the foregoing development gaps had been greatly reduced. The circulation model was on schedule by late summer. The fish model went through two full implementation revisions for the diffusion-taxis component. Version 4 is an effective and very efficient implementation that enables new scenarios to be set up and analyzed much more quickly. The first version of the bioenergetics component is operational. At the close of FY95 the models have been implemented and the focus will increasingly shift to simulations and validation.

During FY95 no coupling of the three model efforts was scheduled. Model coupling starts in FY96. It is useful to keep in mind that the models are coupled by the SEA hypotheses. The first coupling task is shown in Figure 1: the use of simulations from the ocean model to force the plankton model and herring larval drift. The plankton forcing is for both plankton production (the bottom-up alternative hypothesis) and for macrozooplankton distribution and flushing (river-lake hypothesis). Following this the fish and the plankton models will be coupled in simulations of predator and juvenile fish responses to physically driven changes in the macrozooplankton abundance and distribution. All of the facets of the model coupling will not fit in Figure 1. Some of the ones not shown are juvenile herring overwintering, retention zones, replenishment of zooplankton in embayments, and juvenile herring growth.

RESULTS

The SEA Database

Status

There were four major accomplishments in the development of the SEA database during FY95.

1. The database prototyping phase was completed (at no cost to EVOS); the major questions regarding database architecture and software were resolved; and all of the database software was acquired and installed (with cost sharing across four sites).

2. For the majority of the projects and datasets in SEA the data ingestion steps of *(a)* investigator interview and *(b)* dataset analysis were completed; the subset of the data dictionary consisting of items common to all datasets was completed; and the full specification of the data dictionary for two datasets was completed.
3. For one of the SEA datasets (CTD) a full end-to-end implementation of the database was completed, including a network accessible Web-based (Netscape) query tool.
4. An original software application called the SEA *Cruise Planner* was written and the beta version distributed. The software aids in the design and execution of field surveys that involve multiple vessels, sharing of limited equipment and personnel, and the coordination of dissimilar sampling protocols.

Various aspects of these accomplishments are shown graphically in Figures 2 and 3. Figure 2 shows in a table format a comprehensive list of the SEA datasets (bottom row) and the architecture of the database as reflected in the processing steps (left column) used to ingest a dataset into the database. The table entries are shaded according to the status: blackened indicates completion at the end of FY95, gray indicates completion scheduled for FY96. The complete specification of all processing steps in Figure 2 reflects the accomplishments listed in item 1 above. The accomplishments in item 2 are reflected by the "completed in FY95 status" for interview and analysis processing for most of the datasets. Item 3 is reflected in the completed status for all processing steps for the CTD dataset up through the catalog services query tool. Figure 2 is a working graphic used in project tracking and scheduling and has some imperfections. But it does quickly summarize the scope of both the SEA datasets, the architecture of the database, and the status of the effort to implement the SEA database.

Figure 3 shows the interface to the SEA Netscape query server and reflects the accomplishments of Item 3 regarding network accessible query for CTD data. The Netscape query server is a prototype with respect to ecosystem searches but is fully functional for the CTD dataset. Figure 3 shows a search over a range of positions and time and by station ID. The SEA CTD datasets that are returned by the search are show in Figure 3b, and the position of the CTD cast for each of the returned datasets is displayed on a map (Figure 3c). The query server was implemented using the familiar World Wide Web browser software from Netscape, Inc. This software is available for essentially all computing environments (PCs, macintosh, UNIX, VMS). This means that if a user has the proper login permissions then the query shown can be made from anywhere and from any machine via the Internet. The SEA *Cruise Planner* of item 4 is described in some detail in a later section. As a preview, however, the map used in the Netscape query server is a reuse of the map utility developed for the *Cruise Planner*.

In addition to the four database accomplishments just described, there are two further accomplishments for FY95: *(a)* the substantial cost savings realized by EVOS during FY95, and *(b)* the success of the development group in electing to proceed with Illustra™ as the primary candidate database software for SEA a full four months prior to the announcement that Illustra was selected as the database software for EOSDIS.

EVOS realized substantial cost sharing during FY95 by means of joint efforts and cost sharing. The scope of the joint efforts is shown in Figure 4. During FY95 (as well as during FY94) C. S. Falkenberg, R. Kulkarni, L. Herman, J. R. Allen, and V. Patrick worked on the SEA database. Mark Vallarino of the Herring Project also contributed to the effort. Mr. Kulkarni is a senior developer in the areas of scientific data formats and data models, high performance computer systems, and scientific applications. He participated in the effort throughout FY95 at no cost to EVOS. Mr. Falkenberg is the principle designer of the SEA database. Of his eight months of work on the SEA database half was funded by a research grant from the Advanced Visualization Lab. Throughout all of FY95 we were able to conduct the prototype effort and the initial implementation effort with Illustra under a license grant. The Illustra software was purchased at the end of FY95 with substantial cost savings to EVOS through a multi-user purchase and costs shared by four sites. In total thirty to forty percent of the database development effort was provided by non-EVOS funds.

The SEADATA effort for the development of the SEA database is a collaboration of those with expertise in scientific software and database design with those with expertise in all of the various disciplines of marine biology and oceanography represented in SEA. This collaborative structure provides SEA with a window to the most recent events and trends in the technology as well as to opportunities for substantial cost sharing. In particular, by virtue of this collaboration a quite conservative design approach could be implemented. In particular, it was possible to progress carefully without making premature commitments to an approach that would ultimately prove inadequate and lead to a costly and unsatisfactory conclusion. It was through the prudence of the collaboration that everything was not invested early in the Aurora Dataserver and Xidak, Inc. The design progressed in a manner that maintained the flexibility to easily move to an alternative software. The same approach has been followed with Illustra. Although the database design retains its flexibility and lack of overcommitment to a single solution, it appears increasingly that the selection of Illustra has been prudent. Illustra has been adopted as the database for EOSDIS. In addition, during the spring of 1996 Illustra was purchased by Informix whereby Illustra is now a product of the third largest database vendor.

The remainder of this section provides further details on these accomplishments of FY95, and it references a growing documentation for the database that includes one publication and one non-EVOS funded manuscript that was withdrawn at the eleventh hour.

Xidak and the Aurora Dataserver

Throughout FY94 the primary candidate for the SEA database was an extended relational system from Xidak, Inc. The system consisted of the Orion database with an interface called the Aurora Dataserver. The system was designed specifically to address the problem of managing scientific data. It was at that time unique among commercial-grade databases for the extent of the features it provided for scientific data. Mr. Kulkarni had arranged a co-development agreement with Xidak in which he was to develop a query interface for Xidak using the scientific visualization software AVS™ (from Advanced Visual Systems, Inc.) and in return Xidak provided no cost software licenses to both AVL and to PWSSC for Orion and Aurora, plus a grant to support the interface development. That grant supported the work of Mr. Falkenberg during FY94. The Prince William Sound bathymetry data was used to demonstrate the interface.

In November 1994 an abstract describing the AVS interface for Aurora was submitted and accepted for presentation at the 1995 AVS Users Conference, April 19-21, in Boston. The proceedings are published in both hardcopy and CDROM, and during the winter the manuscript was prepared for the February submission. However, during the winter Xidak decided based on market considerations not to continue the development of Aurora. Development of Aurora stopped and the paper was withdrawn. The manuscript was revised to accurately reflect the suspended development of Aurora and serves as a technical report documenting the work and the findings of the collaboration of Xidak, Inc., the Advanced Visualization Lab, and SEADATA. The manuscript for the Aurora Dataserver work is attached as Appendix 1.

Illustra and the SEA Prototype Design Study

During that same time period the object-relational database Illustra was announced. Illustra Information Technologies, Inc. was a new startup company formed in 1992 by Michael Stonebreaker. Dr. Stonebreaker had led the Postgres database development project at the University of California, he had founded the company that produced Ingres, and he was currently one of the principle investigators for the Sequoia 2000 project (Stonebreaker 1994), a project to develop a database for the earth sciences. The Illustra software drew upon Postgres with the goal of a relational database that could effectively deal with the rapidly expanding arena of non-traditional data types, which included scientific data as well as the larger market arena of multi-media data. The goal of Illustra had much to recommend it. Moreover, Prof. Michael Franklin of the Computer Science Department at the University of Maryland had been awarded a license grant from Illustra. In January a research study grant was made available to Mr. Falkenberg by the Advanced Visualization Lab, University of Maryland. It was agreed that Mr. Falkenberg, under the supervision of Professor Franklin, would develop a prototype design for a scientific database using Illustra, and he would use the SEA data archive as his test case. Mr. Falkenberg was supported by this grant through June 1995. It was through this collaboration that Illustra was available to SEADATA for prototyping and initial trials.

The report for that study grant is included as Appendix 2. Although the Illustra software had advanced object-oriented features that were tailored to scientific applications, the function of the Aurora Dataserver had to be replaced. The solution chosen in January 1995 is outlined in Figure 5. This solution was described in the Integrated SEA Project Description for FY96. Mr. Falkenberg implemented this solution using Illustra in his prototype design study using the CTD datasets from SEA. It is the result of that study that is shown in Figure 3.

The design study implemented an AVS query interface to the Illustra database and to the SEA datasets. A paper (Falkenberg and Kulkarni 1995) describing the interface was presented at the *Vis95* conference and it appears in the *Vis95 Proceedings*. The manuscript is attached in Appendix 3. (It is included conditional upon the granting of appropriate permissions by IEEE.)

SEA data schemas and data dictionary

From June through September 1995 Mr. Falkenberg was supported by SEADATA. During this period he interviewed each of the SEA principle investigators and began the process of implementing the prototype design for the totality of the SEA datasets. An initial analysis for all of the datasets was completed, and the data dictionaries for CTD and acoustics were completed. Some of the documentation for the SEA database is included. Figure 6 shows the working document describing the collection, routing, processing, and disposition of the biological samples collected by SEA. Appendix 4 contains the documentation for the data dictionaries for the CTD and acoustic datasets.

SEA Cruise Planner

Cruise planning for SEA requires the coordinated efforts of investigators that traditionally use different sampling and measurement methods. The field sampling methods divide largely along the same lines as the models: oceanography, plankton, and fish. SEADATA decided to commit some resources to the development of a software application to more quickly design complicated cruise plans. A first rapid prototype was developed by Sri Rao using the public domain software Tcl. The planner development was then passed to C. Falkenberg and Larry Herman who then completed the development of Version 0 of the SEA *Cruise Planner*.

The user interface of the *Cruise Planner* is shown in Figure 7. The software provides a map in which one can place both fixed stations and transects. There is a data entry and display interface for the two cases. Stations and transects can be entered via the map and the data entry panel, or the *Planner* can read a text file of stations. The file format is shown in Figure 8a, a simple comma delimited list of positions and information about the station or transect. The file is easily maintained or modified with a text editor or word processor. An important application is to use the *Planner* in conjunction with a spreadsheet (Figure 8b) as a means of quickly computing cruise times and costs and comparing alternative plans. The current version of the *Planner* is a beta version and is being evaluated by SEA investigators.

January review

For the January 1996 EVOS Workshop and SEA Review C. Falkenberg and R. Kulkarni prepared a short document reporting on the status of the database development effort at the close of FY95. Their report to SEA and the SEA reviewers is quite useful for it presents the status of the effort from the point of view of the database developer. It is attached as Appendix 5.

Coordination resources

The Internet, Intranets, and the SEA Collaboration

The Internet and the World Wide Web continue to be a major news stories. It is now common knowledge that anyone can retrieve information from resources around the globe by means of the Internet and software utilities called Web servers. However, it is no longer this global aspect of the Internet and the Web that is driving the news. Rather, the real force behind the news stories are the methods of staying in touch with those close by using *Intranets*. The same technology and software that had been developed for information at distance has been recognized to have significant value for coordination and communications within organizations spread over any distance. Because the technology now has applications with significant economic consequences, the rate of development has exploded, especially software development.

These new developments are of particular interest in SEA. The success of SEA requires the closest possible collaboration between all of the SEA projects. New knowledge in one project must become the knowledge of all in SEA as rapidly as possible and with the least overhead. Questions that arise in one project must be posed and discussed across SEA with a minimum time delay. SEA has all of the requirements for coordination and tight collaborations plus the additional burden of significant distances between groups. Awareness of this is not new. What is new is that within as little as the past six months the options for enhancing the collaborative process have increased significantly.

SEA Home Page and the SEA Collaboration

During FY95 SEADATA implemented a first set of network resources intended to enhance and facilitate the collaboration processes in SEA. J. R. Allen is the developer of this new collection of Web-based tools. A Web server (httpd, NCSA) was configured and installed at PWSSC in early September and a SEA Home Page at URL http://www.pwssc.gen.ak.us went on-line.

Some of these new tools are shown in Figure 9 and Figure 10. Figure 9a shows the SEA "home page," the first graphic that appears upon connecting to the SEA URL. The home page has a standard assortment of links to public information about the organization and the activity responsible for the page as well as links to closely allied sites such as the EVOS OSPIC site and the Web sites for state agencies and universities. At the bottom of the page is the hypertext link to the password-protected section of the Web-site, the "SEA Program Operations" section.

The first page of the Operations section is shown in Figure 9b. The range of the services available are presented as hypertext links. The start-up pages for two of these services are shown in Figure 9c and 9d. The "What's New?" is a listing of those things that have recently changed or newly appeared. As can be seen from the entries in Figure 9c this like most of the items require the regular attention of the site administrator.

Figure 9d is the case of a common and difficult problem in large collaborations quickly resolved. The "Publications in Preparations" area lists the SEA papers that are planned, in progress, or submitted. Postings for the planned papers include a working title, potential contributing authors, and the targeted publication. Postings for papers underway are updated regularly to indicate the current status of the manuscript. Through timely and current postings in each of these categories the problems of unrecognized conflicts and errors of omission and oversight should be nearly eliminated.

The other items in Figure 9b are

Calendar, Deadlines, Announcements
New Proposals Planned
Cruise Plans
Cruise Reports
Data (i.e., access to the online interface to the SEA database (e.g., Netscape server))
Work Groups

Two of these, the Cruise Plans and the Cruise Reports, are to be moved from the SEA operations section to the open access section. The last item of the list is the link to the section supporting the activities of the three work groups (more recently referred to as focus groups). This area contains the majority of the pages supporting the scientific collaborations. The first and second pages are shown in Figures 10a and 10b. Figure 10b shows the four sections used for each of the focus groups. Figure 10c shows how the four sections are used. The "data" area contains links to new results, typically in the form of tables, diagrams or figures with explanatory captions provided by the investigator, plus the cumulative record of the dialogue about that data. This system was used extensively in preparation for the joint presentations made at the EVOSTC Workshop and SEA Review in January 1996. The activities of December and January provided the first test of the potential of these tools for facilitating collaborative work in SEA, and the potential was confirmed and realized.

The Ocean Circulation Model for Prince William Sound

Status

During FY95 Dr. Christopher N. K. Mooers and Dr. Jia Wang, along with Mr. San Jin, have completed the implementation of a first version of a three-dimensional, primitive equation circulation model for Prince William Sound. A manuscript describing this work was submitted by Drs. Wang and Mooers for presentation at the January, 1996, meeting of the American Meteorological Society. The manuscript "Modeling Prince William Sound Ocean Circulation" was accepted and appears in *Conference on Coastal Oceanic and Atmospheric Prediction*, Atlanta, Jan. 28-Feb. 2, 1996, recently published by the American Meteorological Society (AMS). (Wang and Mooers, 1996) That paper is included here as Appendix 6. (Inclusion of the paper is conditional upon receipt of permission from AMS.)

This section is supplementary to the information in Appendix 6. It documents information about the model development effort that is significant to the record of that effort but outside the scope of the conference paper. It also documents some complementary work undertaken to apply early model results and methods developed to assist in model validation.

Initial selections

The general collaborative structure of the ocean modelling effort is quite similar to that of the database development effort. In particular, for the ocean model development there is the following situation:

> The SEADATA effort for the development of the SEA ocean model is a collaboration of those with expertise in numerical simulation modelling of the ocean with those with expertise in all of the various disciplines of marine biology and oceanography represented in SEA. This collaborative structure provides SEA with a window to the most recent events and trends in the science and technology of 4D simulation models as well as to opportunities for substantial cost savings through efficient and effective model development and by building upon recent previous projects that are very similar to the needs of SEA.

There are somewhat limited opportunities for cost sharing for the development of the Prince William Sound circulation model since the set of potential users is very small in comparison with the potential users of a significantly improved scientific or ecosystem database. Instead, there are cost efficiencies due to prior experience and the ability to quickly and efficiently implement the model.

During 1993 and 1994 Dr. Mooers and Dr. D.-S. Ko implemented what they referred to as the "Straits of Florida Nowcast/Forecast System" (SFNFS) (Mooers and Ko 1994). The requirements for that project closely parallel the requirements of SEA. Briefly, SFNFS ingested realtime data and atmospheric model forecasts, produced 0, 1, and 2 day forecasts of the ocean circulation, and delivered these on a continuous basis as part of an online, realtime resource for use in emergency spill responses and realtime risk management. In addition the system was to provide the capability for "what-if" simulations through the use of hypothetical or historical wind forcing time series. For SEA it suffices that spill response is replaced by plankton and fish models. Indeed, at the American Meteorological Society 1996 Conference on Coastal Oceanic and Atmospheric Prediction, papers on both SFNFS (Mooers and Wang 1996, pp 28-35) and Prince William Sound (Wang and Mooers 1996, pp 36-43) were presented.

Hence, SEA started FY95 with the benefit of Dr. Mooers experience not only with numerical simulation models but also with their application in cross-disciplinary collaborations. Output from SFNFS with the forcing wind fields is still available by anonymous ftp to 129.171.100.26 from the directory /pub/SFNFS. Roughly speaking the SEA circulation model development nearing but not yet half way to the level of refinement represented by SFNFS.

One further note regarding initial selections has to do with the choice of the Princeton Model (alternatively referred to as the Mellor-Blumberg model) for SEA. This model has been used for smaller systems such as Hudson Bay (Wang *et al* 1994) and Tampa Bay (John Wang, RSMAS, personal communication). However, at the end of FY94 some reviewers of SEA expressed varying degrees of concern regarding the suitability and applicability of this selection for a fjord as opposed to the coastal domain in SFNFS. These concerns have apparently been redressed for the FY95 reviews of the ocean model development effort have been uniformly very strong.

Dr. D.-S. Ko left RSMAS in December 1994 prior to the availability of funds in late February 1995. SEA did not have the benefit of his contribution to the Prince William Sound model. In April 1996 Dr. Jia Wang arrived at RSMAS, and he and Dr. Mooers are the developers of the SEA ocean model. Dr. Wang is a valuable addition to SEA. He is highly regarded by all and an effective and congenial collaborator. He brought with him previous experience with the Princeton Model and the modelling of smaller, enclosed systems (Wang *et al* 1994). His curriculum vitae is an accurate indicator of the very substantial contributions that he and Dr. Mooers have made to SEA during FY95 (see the curriculum vitae of Wang and Mooers in the SEADATA Detailed Project Description for NOAA BAA Feb. 1 1996 to Jan. 31 1997).

Setup

The initial "setup" of the model—the selection of the boundaries, grid, depth layers, boundary conditions—was done by Jia Wang during April and May of 1995. A key dataset consists of the exiting bathymetric soundings and the shoreline. The result of this step defines the domain for the circulation model, hence to some extent constrains all of SEA. Figure 12 shows the circulation model domain at zero depth. Figures 16 and 18 show three-dimensional views of the bathymetry data.

The bathymetry data of Figures 16 and 18 are derived from an Alaska Department of Natural Resources (ADNR) GIS data product. The NOAA National Ocean Service (NOS) bathymetry data is too sparse to be used alone. ADNR combined the NOS data with additional depth soundings. From this first generation dataset a second generation dataset was produced consisting of 20m depth contours in the Arc/INFO vector format. This second generation data set is at present the best available bathymetric data for the sound. The bathymetry for the model domain is third generation: the contour data plus shoreline data were interpolated and gridded using the Albers conic equal-area projection (parallels at 55 and 65, origin at -154W, 50) to obtain the needed gridded data product for the model. The first generation data that supplemented the NOS soundings have not been recovered. Only the GIS coverages have been traced.

The limitations of the 20 meter contours can be seen in Figure 18 and in Figures 11 through 15. The 20 m contour is insufficient to retain the shallow water passages into the sound around Hawkins Island. A new first generation bathymetry with coverage of the shallow areas and that extends significantly onto the shelf will be needed for the next version of the model.

Model configuration '95

Three forcing components were implemented during FY95. The dates in parentheses are the approximate time at which the first model output was presented.

inflow at Hinchinbrook Entrance (intrusion of the Alaska Coastal Current) (May, 1995);
the dominant component of the tides, M_2 (June, 1995);
winds (forcing by constant, uniform wind) (August, 1995).
(during the fall 1995 particle advection (tracer) simulations were conducted.)

Additional information on these is contained in Appendix 6.

Simulations using model configuration '95

For all of the following simulations the model output should be understood solely as the product of a simulation exercise; the output should not be expected to closely approximate measured data. Typically here only one or two forcing variables have been applied at one time with only a zeroth order estimate of the proper magnitudes for each.

An example of the type of model simulations obtained first during FY95 is shown in Figure 11. Here a fixed inflow of 0.3 Sv is applied with no other forcing. The model is first run for 17 days to stabilize (spin-up). The days shown in each of the panels of Figure 11 refer to days beyond the first 17 day spin-up. Forcing with inflow of 0.3 Sv is at the upper end of observed the observed range of inflow. It is representative of only the October through December period. Upper layers during the summer can in fact have net outflow at Hinchinbrook Entrance. (Niebauer *et al* 1994). The four panels of Figure 11 are all for velocities at a depth of 3 meters.

The simulation shows the degree to which the circulation is stable after 17 days. At each time step shown the model has generated some new eddy the near surface flow. In Appendix 6 Jia Wang discusses briefly the fact that the model is eddy resolving.

The model output velocity fields for the combined forcing of 0.3 Sv in-flow and M_2 tide (3 m tide range) is shown in Figure 12 (high and ebb at 3 m depth), Figure 13 (low and flood at 3 m depth), Figure 14 (high and ebb at 100 m depth), and Figure 15 (low and flood at 100 m depth). The addition of the tides has added substantial additional structure to the velocity fields of the central sound. The fields at 3m are different from those at 100m.

In Appendix 6 Jia Wang shows the effects that sustained west and sustained east winds have upon the circulation velocity fields.

Model validation

In the last months of FY95 work was begun to incorporate Prince William Sound data into the model. Work was begun to prepare the hydrographic data and acoustic Doppler current profiler data for use in validating and calibration of the model. The plans for the 1996 oceanography cruises were revisited in an effort to identify optimal allocations of measurement resources. These efforts are continuing they remain the top priority. The validation and calibration of the ocean circulation model must be completed before it can be applied to any of the several SEA projects that are waiting for it.

This first step from concept to initial implementation has been a remarkable one. The achievement of a numerical model where none before existed is a clear and unambiguous major step forward. But it cannot be counted yet as a success; success requires further refinement of the model and the validation of the model. That accomplishment is still ahead.

Preliminary applications

In an effort to quickly utilize during FY95 any new insights that might be obtained from the model, during the winter preparations were made to apply the anticipated model simulations to the problem of the initial conditions of stage one neocalanus arriving at the surface waters in the spring.

Figure 16 graphically summarizes the problem for the initial conditions for neocalanus. In the lower left of the figure a cross-section of a hypothetical region of the sound with depths exceeding 450 m. The notes accompanying the cross section describe the situation and some of the issues. Briefly, the adult neocalanus overwinter at depths below 350 m, and in early March over a period of approximately 15 days they lay eggs. During a second 15 days these eggs rise to the surface and on the way hatch and become stage one copepidites by the time they reach the surface waters. The notes in the figure identify unknowns regrading the initial distribution of adults at depth.

However, a more substantive question is how the newly hatched early stage neocalanus are distributed when they first reach the surface. If the eggs and nauplii were to rise vertically with no horizontal mixing and if adults at depth were distributed uniformly, then the initial condition would be a population distributed at the surface in proportion to the available volume for overwintering of adults at depth directly below. This is shown in Figure 16 by the surface plot over the sound of the available overwintering volume for adults directly below the position of the plot. Hence, for this hypothetical scenario there would be a very high density of neocalanus directly over the deepest parts of the sound, in particular over the so-called "black hole" in the western sound where bottom depth exceeds 700 m. There would be a very large expanse of midrange density in the east-central sound.

An alternative to the foregoing scenario is substantial advective redistribution of the animals during the 15 day rise to the surface. To illustrate an application of the model, a particle advection simulation prepared by Jia Wang for the velocity fields of Figure 11 is shown in Figure 17. The panels of the two figures correspond. In particular, the velocity fields are due solely to a constant inflow of 0.3 Sv at Hinchinbrook Entrance. The simulation consists of the release of tracer particles throughout the upper 40 m of the model boundary across Hinchinbrook Entrance at a constant rate for the first six days after spin-up. The simultaneous spreading and flushing of the tracer is apparent. It is not shown but the tracer essentially remains within the upper 40m layer.

This simulation scenario, however, does not directly fit the task. According to Niebauer *et al* (1994) the inflow in March approaches zero whereas the simulation assumes 0.3 Sv inflow. However, Jia Wang points out in his paper that the residual tidal flow is approximately that of the assumed in-flow. Hence, it is conceivable that a simulation with tides alone would result in a similar transport but not necessarily along the paths shown. Further model forcing and refinement and model validation is needed before this advective transport feature is meaningful.

A second topic in which there was an effort to pull together field measurements and model results is shown in Figure 18. This shows one of the visualization methods developed to display velocity fields measured by towed ADCP. This picture however is misleading for the time interval over which the measurements were made spans several tide stages. To compare these velocity fields with simulations the time for both the measurement and the model output must be used as well as the spatial position. During FY95 the display method in Figure 18 was further developed so that velocity fields at only a single common tide stages interval are shown. This was then animated so that the display stepped through a tidal cycle.

The Nekton Model for pink salmon and Pacific herring

Status

During FY95 there were many accomplishments, but they all were not achieved at the anticipated time, and for some the need had not been anticipated at all. That is, there were periods in which things were done with temporary solutions until a missing piece was available. There were important pieces that were particularly elusive and the solutions did not all become available until nearly the time of the EVOS Workshop in January 1996. For the fish model development, then, this review of results witll cover the period up through early February 1996. With that warning of some irregularity in the schedule, the key milestones for FY95 are summarized below in approximately chronological order.

During FY95 two versions of the bioenergetic-foraging models were completed. These are referred to as *gut v1* and *gut v2*. The spatial component of the fish model (diffusion-taxis model) has had the name *Alewife* stubbornly remain attached to it (ALaska Experimental Window Interface for Fisheries Ecosystems). This is often shortened to *Aw*. FY95 started with *Aw v1*, a single trophic level, single spatial dimension model. This review will extend to *Aw v4* which allows an unlimited number of interacting trophic levels, one spatial dimension, and implements a new structure that dramatically simplifies all of the coding tasks associated with model development and refinement. The major milestones of the project follow.

1. First foraging-bioenergetics model (*gut v1)* for pink salmon fry. It simulates diel feeding pattern and growth. The model performs well relative to the measurements reported by Godin (1981a, 1981b, 1984) This result was the starting point for *gut v2* below.
   (Feb 95)
2. First results from *Aw v1* configured for three coupled trophic levels and one fixed (forcing) trophic level. This is the code from Tongbio Li from FY94. Although it gave reasonable results the code was too awkward for efficient modelling, it did not scale easily to additional trophic levels, and it did not scale easily to higher spatial dimensions.
   (Apr 95)
3. Limits identified for the taxis to diffusion ratio. These limits are imposed by the numerical implementation of the solver for the partial differential equation; begin process of rewriting the algorithm and the code. Thus begins the need to distinguish the *old* code from the *new* code.
   (Apr 95)
4. Begin work on a graphical user interface intended to facilitate collaborative work with fisheries biologists. This ultimately becomes *Aw v3.*
   (May 95)

The goal here was to have all of the information from Figure 19 readily available. Figure 19 is a list of all possible factors to include in the loss function but typically only a few are used. Even with a few, with Tongbio's code all of the parameters are buried in a large number of small C programs. In addtion, the domain of interest in Figure 20 should be shown. Finally, the interface should facilitate manipulating fish populations with the model as in Figure 21. For example, a linear loss function produces an exponential solution. Hence one can "force" a population to remain at one shore but not another simply by imposing an arbitrary loss function.

5. Arrangements completed with European colleague for collaborative use of recently developed 2-d/3-d mesh generator code.
   (June 95)
6. *Aw v3* with graphical user interface (with *old* code) is ready. See Figures 26 and 27. (Jul 95) Simulation studies performed using fixed *Neocalanus* and coupled pink salmon fry, alternative prey to fry (e.g., juvenile pollock), and adult pollock. See Figures 22 through 25. Present at Fairbanks SEA meeting.
   (Aug, Sep 95)
   One of the September simulations is shown in Figures 22 through 25. This is a four day run starting from atypical distributions in order to see the way the distributions jointly adjust. The loss functions are somewhat forced, and these are described in Appendix 9. In particular, a nearshore preference is applied to fry and a nearshore avoidance to pollock. The feeding rates and rates of encounters with adult pollock are plotted for both fry and age-0 pollock in Figure 23. These are computed during the simulation as part of the calculation of the loss. In Figure 24 the diet of adult pollock on the three prey are shown.
7. Complete second version *gut v2* of bioenergetics-foraging model for both adult pollock and pink salmon fry. Demonstrate the capability to simulate episodic feeding behavior. See Figures 30 and 31. Model performance reviewed within SEA.
   (Dec 95)
8. *Aw v4* (with *new* code) completed. See Appendix 7 and Appendix 8.
   (Dec 95)
9. Prepare updated *Aw* model specification document. See Appendix 9.
10. Conduct simulation studies demonstrating the capability to simulate known or conjectured patterns in fry distribution due solely to current velocity and tides.
    (Jan 95)
11. Conduct first simulations of nearshore interactions of pollock with tidally influenced fry distribution and fixed *Neocalanus* distribution. Present at AGU/ASLO. See Figure 31.
    (Feb 95)

During FY95 the fish model development had the benefit of cost sharing. Dr. Doran Mason and Dr. Ricardo Nochetto, as in FY94, again collaborated with Dr. Vince Patrick on the fish model development. Mr. Sri Rao was added to the group in April 1995. Dr. Mason is contributing to the biology and structure of both the spatial model (Mason and Patrick 1993) and the foraging and bioenergetics model (Mason and Brandt 1996; Brandt *et al* 1992). He is a Research Associate at the Limnology Lab, University of Wisconsin, as well as as Research Scientist at Prince William Sound Science Center. Because his interest is in the successful development of the model in general as well as for Prince William Sound, his contribution to the model development is significantly greater than would be possible with only SEA support.

Dr. Ricardo Nochetto's contributions address the numerical solutions for the fish model. Dr. Nochetto is an Associate Professor of Mathematics at the University of Maryland and a Research Scientist at Prince William Sound Science Center. He is a numerical analyst working primarily on nonlinear partial differential equations. He is interested in the numerical methods for the fish model in general, and in particular he is interested in the problem of multi-dimensional grids for these classes of problems. These overlapping interests and his use of the findings in other areas are of significant cost benefit to SEA and to EVOS.

Both Dr. Mason and Dr. Nochetto bring to SEA and to EVOS insights, resources, and breadth of participation not available any other way. Dr. Mason is, for examle, an invited collaborator on the Lake Superior Technical Committee, a group just now approaching their first success in their effort to restore a viable population of the once native lake trout. His position at the Limnology Lab places him at one of the crossroads of activity in bioenergetic modelling. Indeed, the Hewett and Johnson bioenergetics software was written at the Limnology Lab and supported and published by Wisconsin Sea Grant.

Dr. Nochetto provided the expertise whereby the modelling effort was never limited by the expertise of any of us who are not numerical analysts. He tended to all of the numeric issues and thereby enabled the rest to focus on the model itself and the results. He is an active researcher (over 40 research articles, an editor of the *SIAM Journal on Numerical Analysis,* and the recipient of the 1993 International Giovanni Sacchi Landriani Prize for outstanding contributions to the numerical analysis of partial differential equations). It is through his involvement in the research community that SEADATA has succeeded with the FY95 objective of finding suitable 3-dimensional mesh generation algorithms for trials with the fish model and the SEA coupled model.

Mr. Rao handled all of the programming. All of the new *Aw* software in the foregoing milestones list is due to him.

Versions completed and simulations of distributions
1. Pollock-Fry interactions in the nearshore regions

The most significant result is Item 11 above, which is shown in Figure 31. The purpose of this simulation was to see how the one-dimensional (spatial) cross-channel model would respond for the following scenario:

1. a nearshore pink salmon fry distribution, as shown in Figure 31b;

2. C5 *Neocalanus* at a relatively high density, as shown in Figure 31a; there is a cross-channel distribution that is approximately uniform but which decreases as the shore is approached; the zooplankton density is assumed to not vary in time;
3. a reduced handling time for adult pollock feeding upon C5 *Neocalanus*, as compared to the handling time for fry, to better represent the apparent switch to filter feeding;
4. the adult pollock following an episodic feeding regime, apporximately four hours feeding and eight hours non-feeding.

All of the possible combinations and ranges have not been run, but from a run such as the one in Figure 31 it does seem that the model is quantitatively discriminating in the ways that previously could only be qualitatively described. Note that the simulation is plotted only for five time steps, each step 0.1 day apart, and that the five time steps shown have been "spun-up" for nearly ten days.

Loose all trace of partitioning during non-feeding. The diffusion and taxis coefficients have been set somewhat high. The assumption is that the predator fish swim sufficiently fast and are sufficiently active that during the non-feeding time that the values are realistic. Their distribution becomes uniform when not feeding. The alternative extreme is to have the diffusivity and the taxis coefficient become small when not feeding. This would "freeze" in place the distribution formed during feeding.

Partitioned between fry and zooplankton during feeding. When the hunger increases for the predator, those that are in "contact" with the fry in the nearshore move strongly further inshore. Those predators that are outside the fry distribution remain with the macrozooplankton. In fact, those that do not come in contact with the fry tend to be held within a "loss function well" in the mid-channel.

The prudence of avoiding gradients. Within the confines of the diffusion-taxis model the pollock arrive at the nearshore either by a random walk (diffusion) or by following a gradient. If the fry distribution tailed off into the channel it would draw in a larger population of predators. In the case at hand there was in fact a shortage of prey and an increase in the loss function which insures no taxis by predators toward the fry from midchannel..

The advantage of a loss function maximum. The only reason the loss function does not monotonically decrease to the nearshore is because the fry are further inshore than a certain macrozooplankton density. If the plankton density were uniform, then the loss would decrease with the additional prey and there would be a continual pressure on the nearshore. This can be avoided by the prey if they can position themselves beyond the "roll-off" of the offshore plankton densities. At some point this is in conflict with their feeding, however. It is an optimization problem, and is dealt with automatically by the model if the fry are free to redistribute in response to predators.

Fry mortality is given at each point by multiplying feeding rate by the adult pollock density.

In this study the pollock loss function was set solely by available feeding rate at a point and the degree of hunger. The fry were responding only to diffusion and tidal current avoidance; they were not responding to predators. The interest here was the relative preference between fry and *Neocalanus.* The answer is that at these densities the fry still are better feeding (in grams per unit time) but all of the pollock cannot fit into the small space occupied by fry.

2. Time varying nearshore densities due to tides

The thinking continues that the nearshore serves as a refuge in some way. Some of the mechanisms were suggested from the simulation above. It is often suggested that the smaller juvenile are "avoiding currents." Hence, simulations were run to determine what the end result looks like from a constant velocity current and then a tidal current. The change was quite surprising.

The redistribution of fry from a point source such as a net pen release was simulated, where it was assumed the fry would attach an increased loss to water with high velocity. The time evolution of the density is shown in Figure 28. When a periodic tide is switched on, the fry distributions tend to "leak" from the shore during slack tides. A new quasi-equilibrium is obtained wherein the fry densities increase during flood and ebb as they move back inshore, then the densities decrease as the population disperses off shore. See Figure 29.

3. Time varying nearshore density due to feeding

In the foraging-bioenergetics component the issue of satiation and its modulation of feeding behavior is of top interest. In *gut v2* better simulations were constructed as alternative mechanisms for modelling satiation and hunger were tried. We can reproduce the intense morning feeding reported in the literature. This is shown in Figure 30. This process is a candidate for modulating the nearshore density of fry.

DISCUSSION

The collaborative process in SEA

Throughout the preparation of this review there was the repeated discovery: there is an immediate and urgent need for a dramatic increase in the communications and collaborations in SEA. Not enough information is changing hands. This is not to say something has not been done, but rather that the tasks have matured in scope and complexity and now require more comprehensive links.

Individual new information must become collective knowledge almost instantly. An unexplained finding must become a global question of which all are aware.

LITERATURE CITED

Blumberg, A. F. and G. L. Mellor, 1987. A description of a 3-D coastal ocean circulation model. In *Coastal and Estuarine Sciences 4*, N.S. Heaps, ed., Amer. Geophys. Union, Washington D.C. pp 1-16.

Brandt, S. B., D. M. Mason, and E. V. Patrick. 1992. Spatially-explicit models of fish growth rate. *Fisheries,* 17(2) pp 23-35.

Eslinger, D. L., 1990. The effects of convective and wind-driven mixing on springtime phytoplankton dynamics as simulated by a mixed-layer model, 127pp., Ph. D. Dissertation, Florida State University.

Falkenberg C. S. and R. Kulkarni 1995. Using Spatial Access Methods to Support the Visualization of Environmental Data. In *Proceedings of Visualization '95*, Atlanta GA, November 1995, IEEE Computer Society Press, pp 400-403.

Gallacher, P. C. and P. A. Rochford, 1995. Numerical simulations of the Arabian Sea using tracers as proxies for phytoplankton biomass. *J. Geophys. Res.,* 100, pp 18,565-18,579.

Gerritsen, J., and J. R. Strickler. 1977. Encounter probabilities and community structure in zooplankton: a mathematical model. *J. Fish. Res. Board Can.* 34 pp 73-82.

Godin, J.-G. J. 1981a. Effect of hunger on the daily pattern of feeding rates in juvenile pink salmon, *Oncorhynchus gorbuscha Walbaum. J. Fish Biol.* 19 pp 63-71.

Godin, J.-G. J. 1981b. Daily patterns of feeding behavior, daily rations, and diets of juvenile pink salmon (*Oncorhynchus gorbuscha*) in two marine bays of British Columbia. *CJFAS* 38 pp 10-15.

Godin, J.-G. J. 1984. Temporal variation in daily patterns of swimming activity and vertical distribution in juvenile pink salmon (*Oncorhynchus gorbuscha*). *Can. J. Zool.* 62 pp 72-79.

Howick, G. L., and W. J. O'Brien. 1983. Piscivorous feeding behavior of largemouth bass: an experimental analysis. *Trans. Am. Fish. Soc.* 112 pp 508-516.

Mason, D. M., and S. B. Brandt. 1996. Effects of spatial scale, capture efficiency, and the spatial distribution of predators on the predictions made by spatially-explicit models of fish growth rate. *Environmental Biology of Fishes,* 45(3) pp 283-298.

Mason, D. M. and E. V. Patrick. 1993. A model for the space-time dependence of feeding for pelagic fish populations. *Trans. Am. Fisheries Soc.,* 122(5) pp 884-901.

McCreary, J. P., K. E. Kohler, R. R. Hood, and D. B. Olson, July 1995. A four-component ecosystem model of biological activity in the Arabian Sea. Technical Report.

Mooers, C. N. K. and D.-S. Ko. 1994. Nowcast system development for the straits of Florida. *Estuarine and Coastal Modelling III, Proc. of the 3rd Intern. Conf.,* pp 158-171.

Mooers, C. N. K. and Wang, J., 1996. The second generation of the Straits of Florida nowcast/forecast system. In *Conference on Coastal Oceanic and Atmospheric Prediction,* Atlanta, Jan. 28-Feb. 2, American Meteorological Society, Boston, pp 28-35.

Niebauer, H. J., T. C. Royer and T. J. Weingartner, 1994. Circulation of Prince William Sound, Alaska. *J. of Geophysical Research,* 99, pp 14,113-14,126.

Stonebreaker, M. 1994. Sequoia 2000: A Reflection on the First Three Years. *IEEE Computational Science & Engineering* Winter 1994, pp 63-72.

Wang, J. and C. N. K. Mooers, 1996. Modelling Prince William Sound ocean circulation. In *Conference on Coastal Oceanic and Atmospheric Prediction,* Atlanta, Jan. 28-Feb. 2, American Meteorological Society, Boston, pp 36-43.

Wang, J., L.A. Mysak and R.G. Ingram, 1994. A 3-D numerical simulation of Hudson Bay summer circulation: Topographic gyres, separations and coastal jets. *J. Phys. Oceanogr.*, 24, pp 2496-2514.

FIGURES

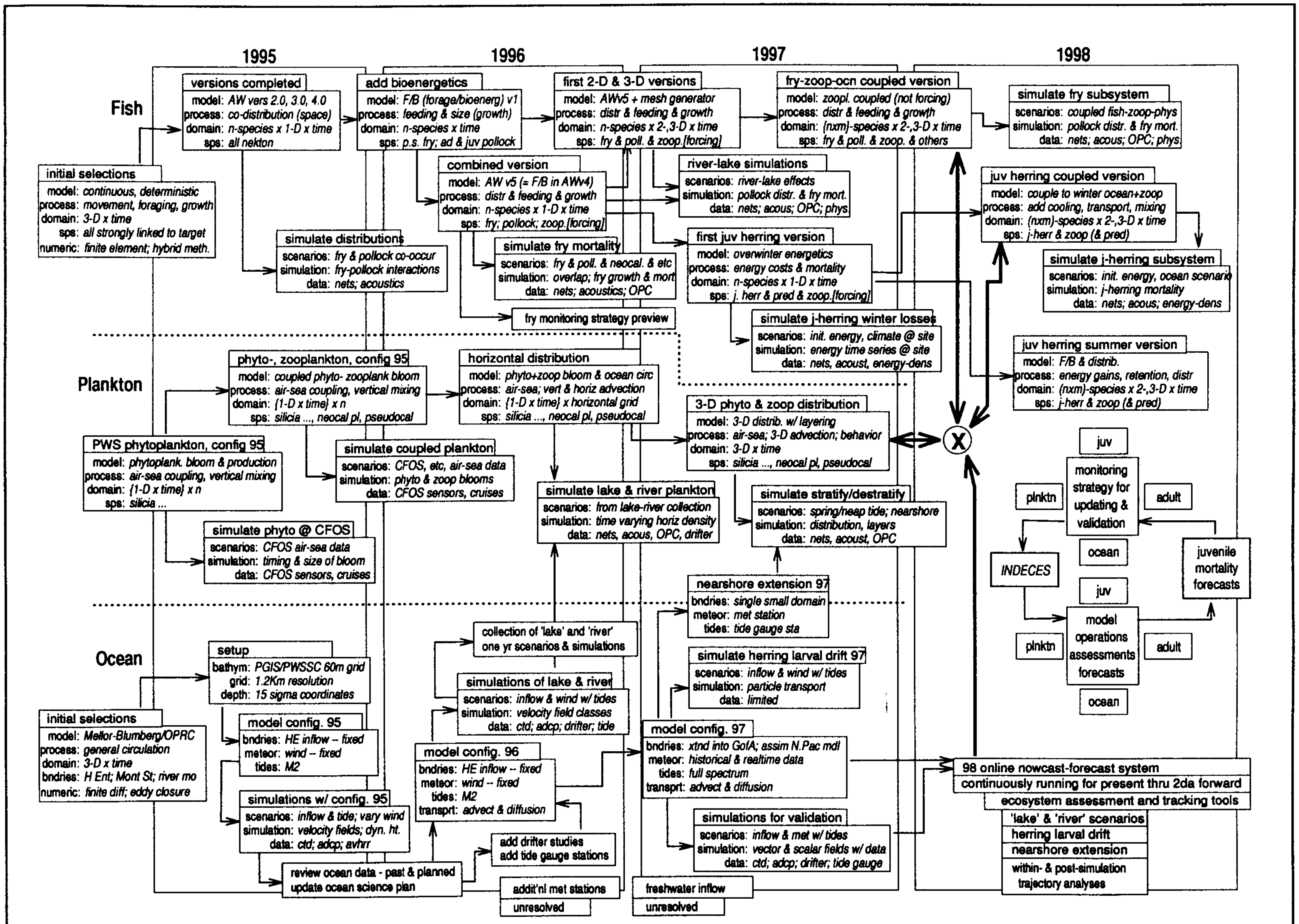


Figure 1. The SEA models: status, development plan, dependencies, and endpoints.

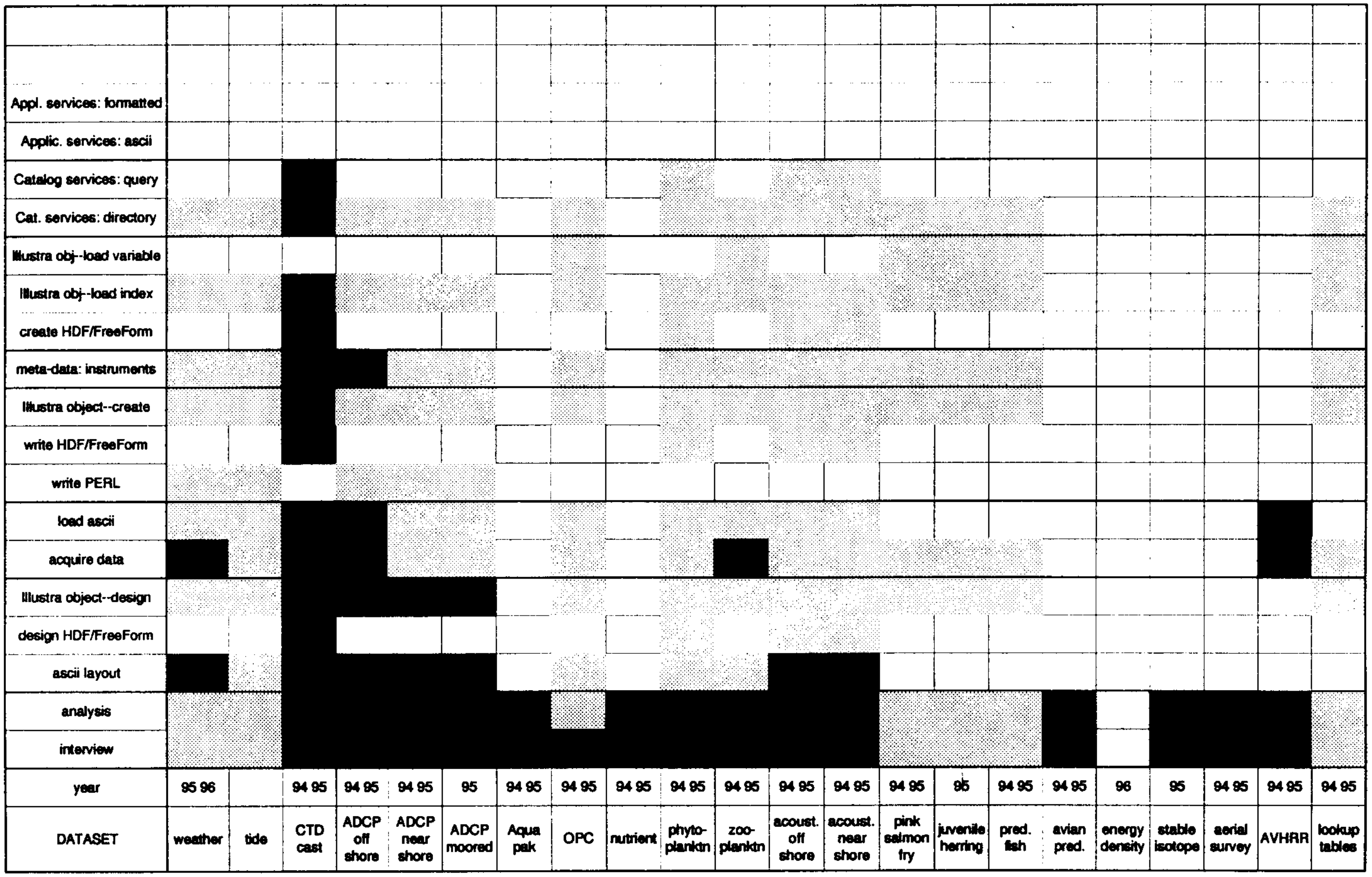


Figure 2. The SEA Datasets and the SEA Database; status for Oct. 95 (black), expected status for Oct. 96 (gray).

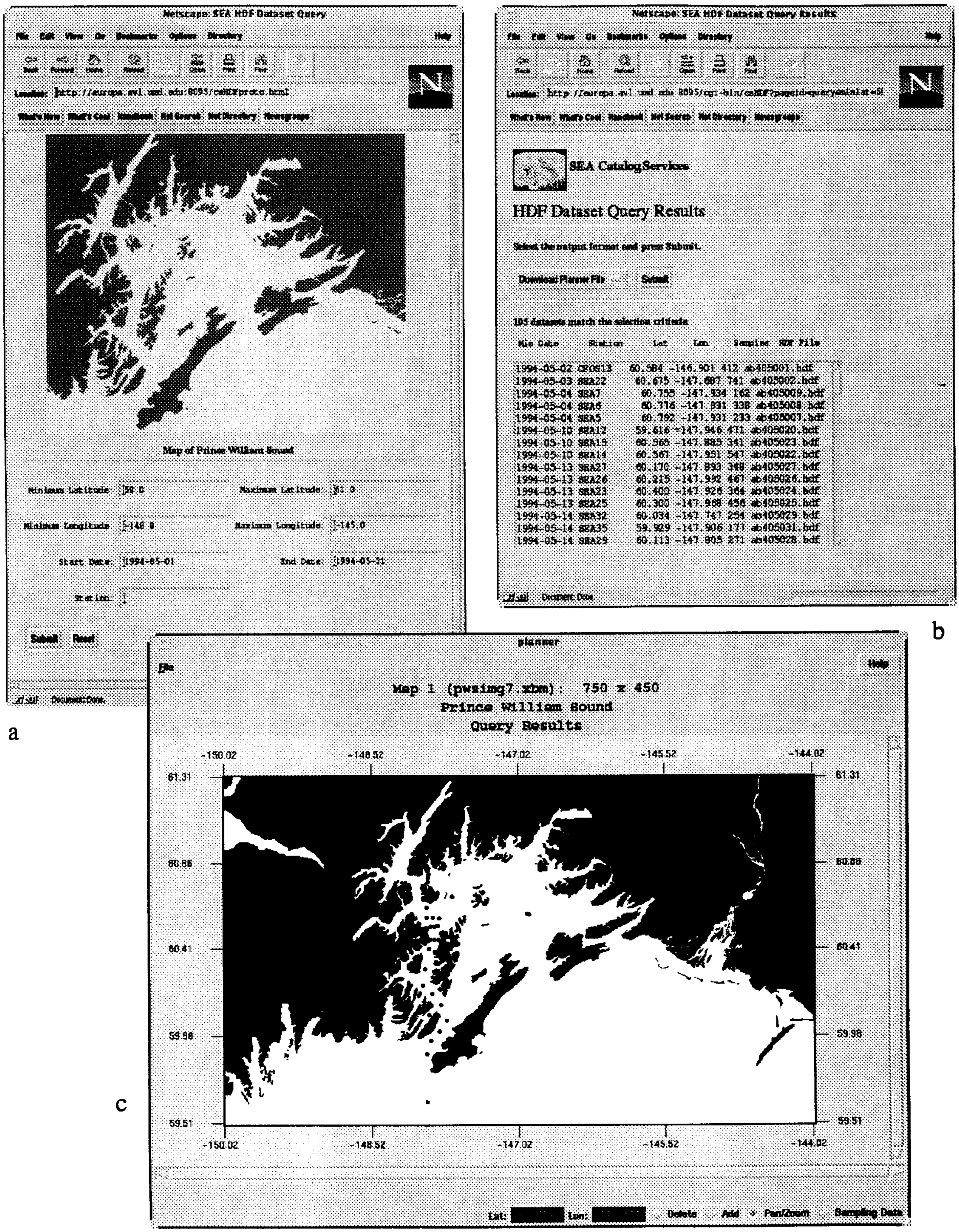


Figure 3: SEA prototype Netscape server: query form (a), query results (b), locations of results (c).

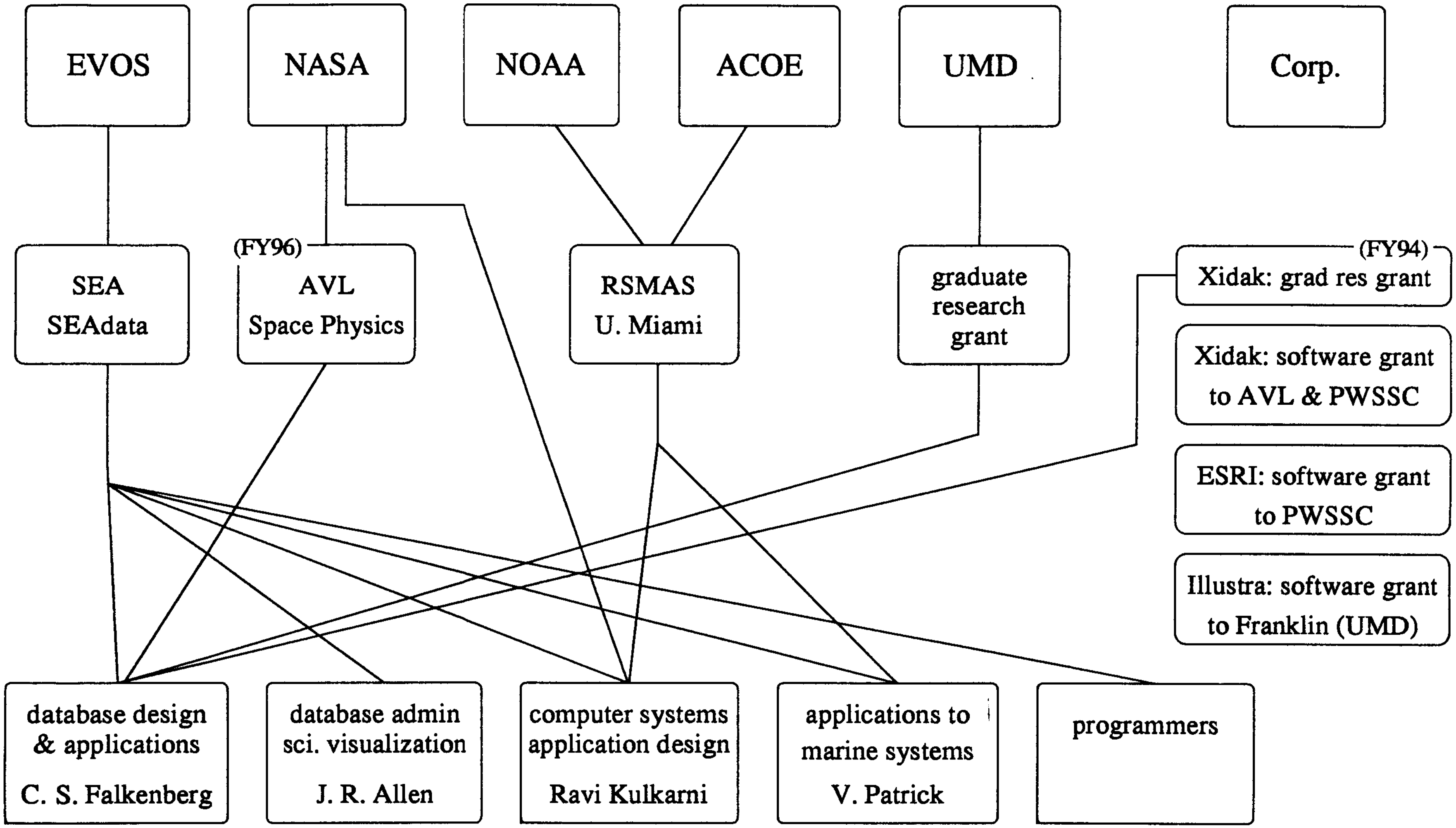


Figure 4. Cost-sharing for the SEA database development during FY95.

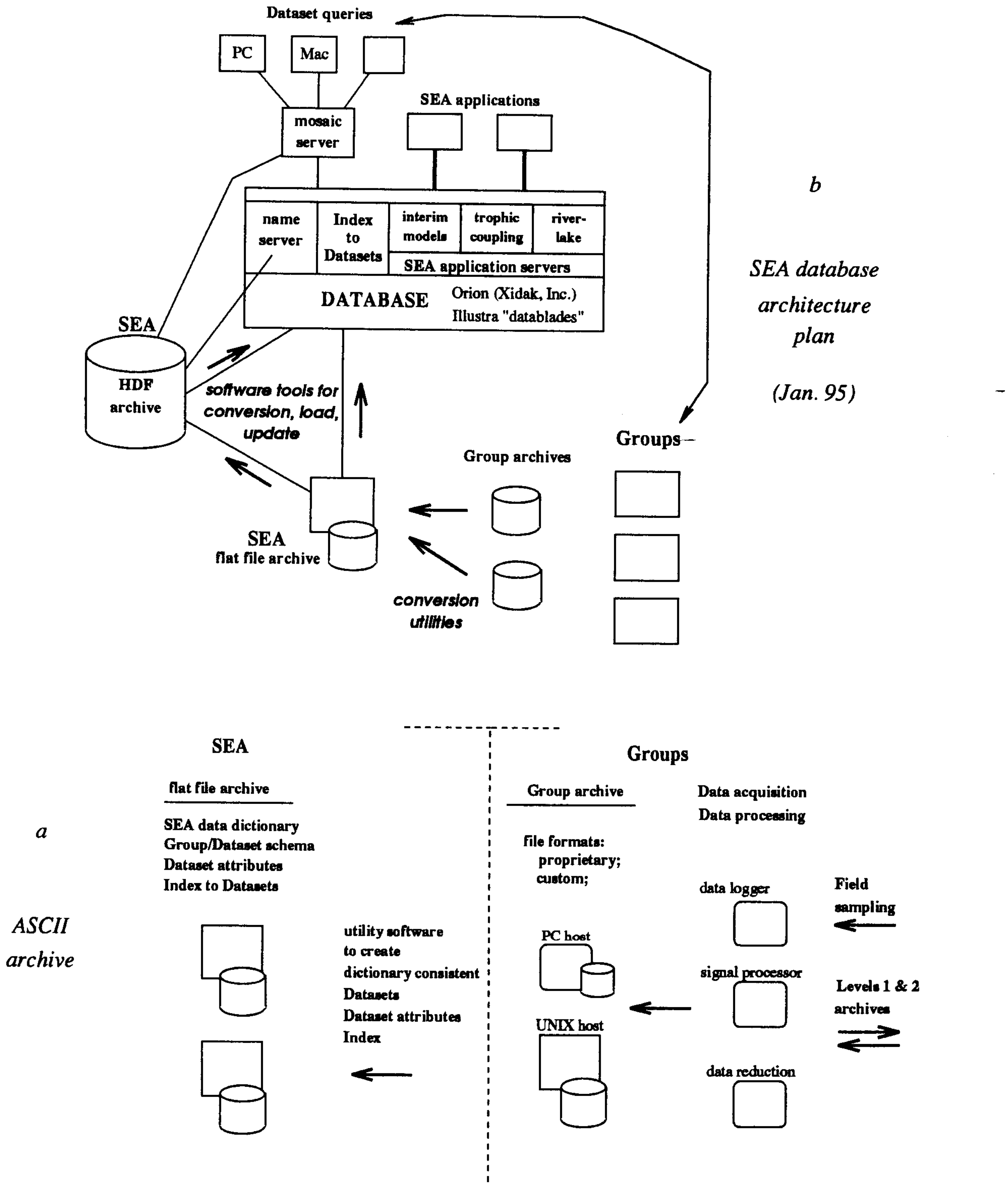


Figure 5. Plan for the SEA database, Jan. 1995.
ASCII archive (a) serves also as first ingestion step.
Database plan (b) show full complement of anticipated services.

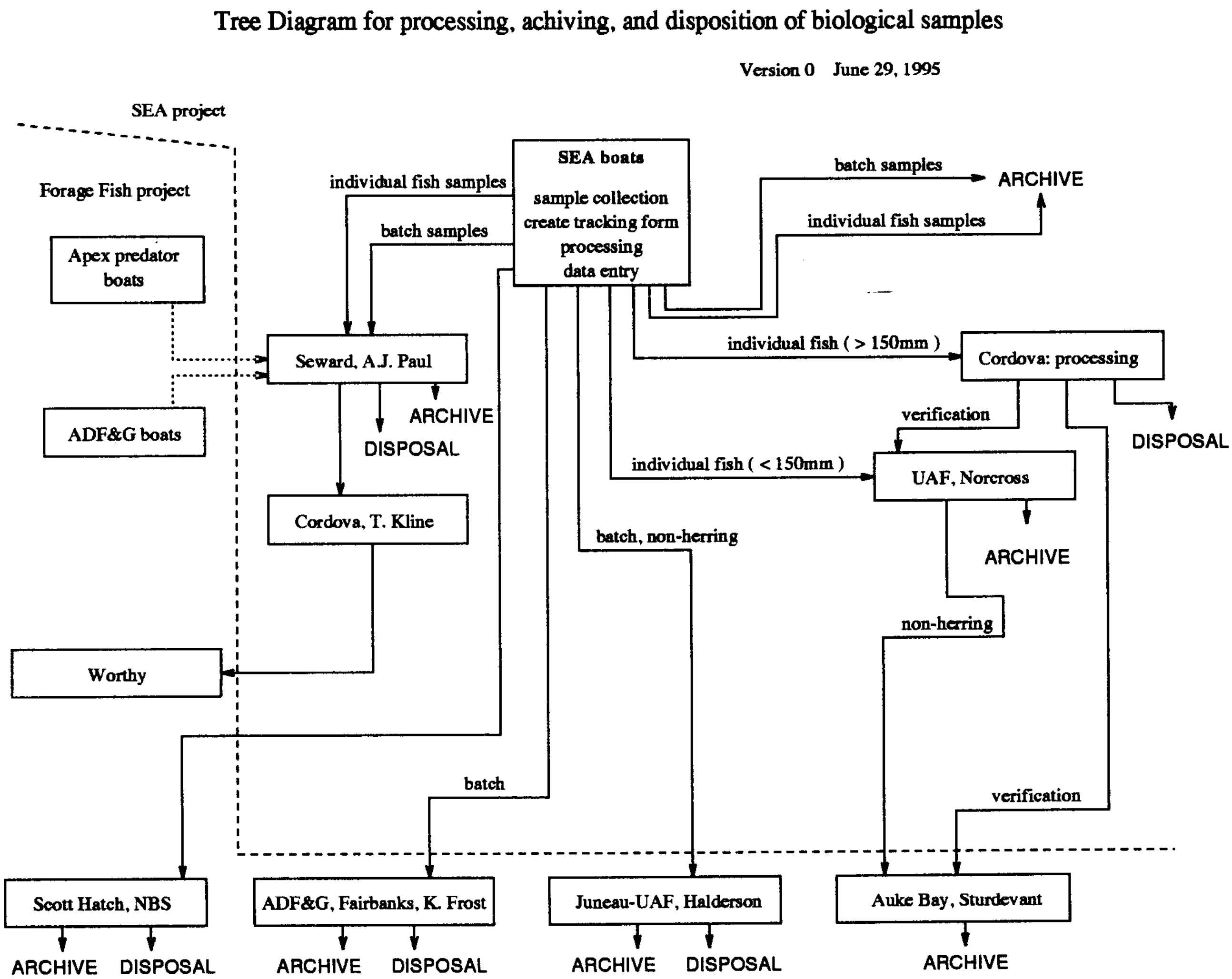


Figure 6. Diagram prepared to guide the development of the data dictionary entries for biological samples.

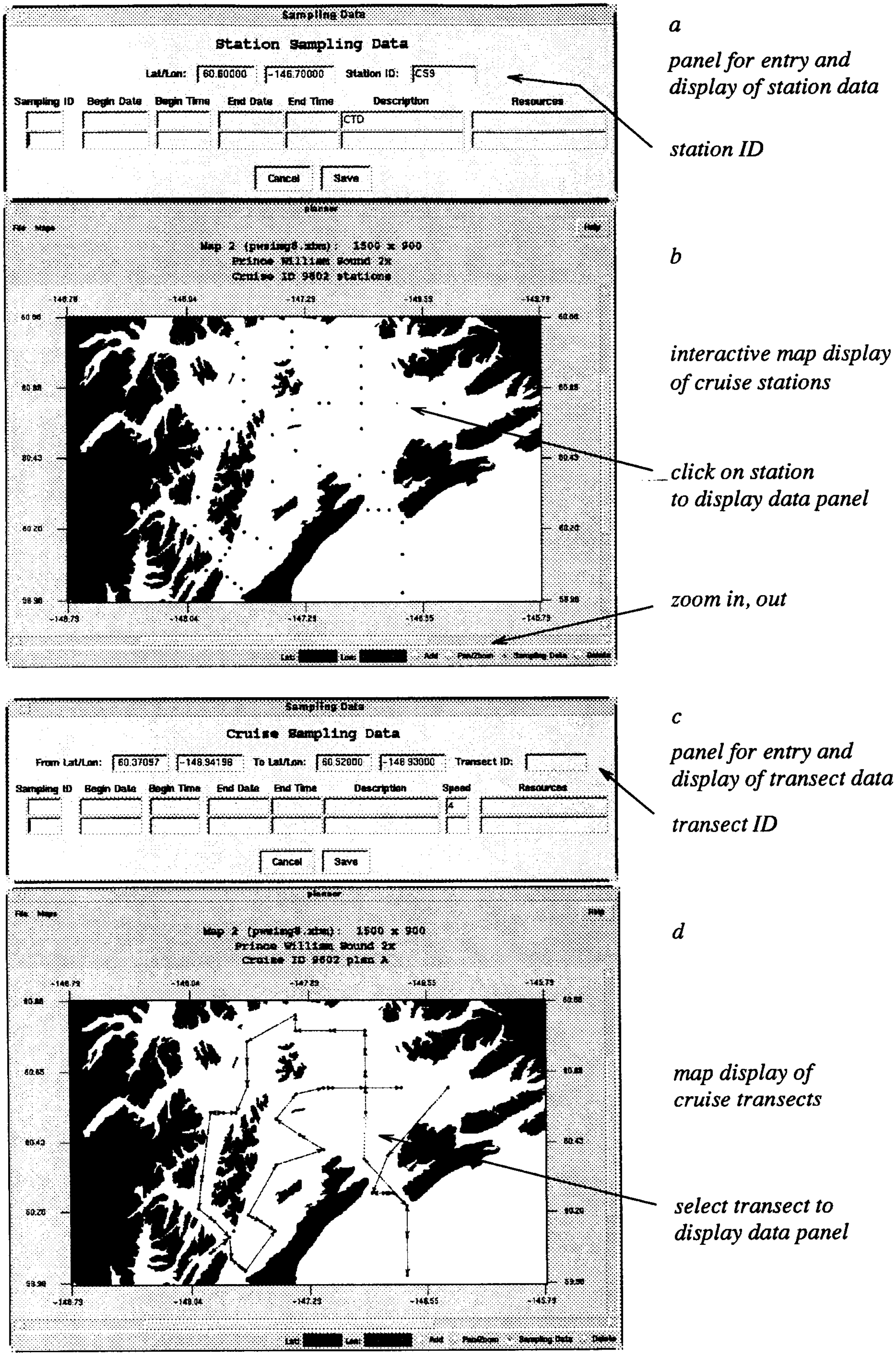


Figure 7. The user interface of the SEA cruise planning software.

```
Cruise ID 9602 plan A
HE13,60.2600000000,-146.8900000000,60.2600000000,-146.8900000000,,,,,,CTD,,
HE11,60.2600000000,-146.7550000000,60.2600000000,-146.7550000000,,,,,,CTD,,
SEA33,60.0150000000,-147.7000000000,60.0150000000,-147.7000000000,,,,,,CTD,,
SEA31,60.0500000000,-147.7850000000,60.0500000000,-147.7850000000,,,,,,CTD,,
SEA18B,60.5200000000,-147.7500000000,60.5200000000,-147.7500000000,,,,,,CTD,,
SEA16,60.5200000000,-147.9100000000,60.5200000000,-147.9100000000,,,,,,CTD,,
SEA26,60.2100000000,-147.9900000000,60.2100000000,-147.9900000000,,,,,,CTD,,
SEA25,60.3000000000,-147.9700000000,60.3000000000,-147.9700000000,,,,,,CTD,,
  ...

SEA32,60.0300000000,-147.7400000000,60.0300000000,-147.7400000000,,,,,,CTD,,
SEA36,59.9100000000,-148.0700000000,59.9100000000,-148.0700000000,,,,,,CTD,,
SEA37,59.8800000000,-148.0100000000,59.8800000000,-148.0100000000,,,,,,CTD,,
MS8,60.4000000000,-147.2000000000,60.4000000000,-147.2000000000,,,,,,CTD,,
MS8,60.4500000000,-147.3500000000,60.4500000000,-147.3500000000,,,,,,CTD,,
MS5,60.1400000000,-147.5100000000,60.1400000000,-147.5100000000,,,,,,CTD,,
MS5A,60.1700000000,-147.6000000000,60.1700000000,-147.6000000000,,,,,,CTD,,
MS5B,60.1900000000,-147.6800000000,60.1900000000,-147.6800000000,,,,,,CTD,,
CFOSBY,60.6000000000,-147.2000000000,60.6000000000,-147.2000000000,,,,,,CTD,,
CS5,60.3800000000,-146.7900000000,60.3800000000,-146.7900000000,,,,,,CTD,,
,60.5992213571,-146.4016010670,60.3800000000,-146.7900000000,,,,,,,4,
,60.2608453838,-146.7538358910,60.2200000000,-146.6700000000,,,,,,,4,
,60.2208008899,-146.6697798530,60.1200000000,-146.6700000000,,,,,,,4,
,60.1206896552,-146.6697798530,60.0000000000,-146.6700000000,,,,,,,4,
,60.0005561735,-146.6697798530,60.2200000000,-146.6700000000,,,,,,,4,
,60.2208008899,-146.6697798530,60.3700000000,-146.9400000000,,,,,,,4,
,60.3709677419,-146.9419613080,60.5200000000,-146.9300000000,,,,,,,4,
,60.5191323693,-146.9299533020,60.6000000000,-146.9300000000,,,,,,,4,
  ...

,60.2608453838,-146.7538358910,60.2600000000,-146.8200000000,,,,,,,4,
,60.2608453838,-146.8218812540,60.2600000000,-146.8900000000,,,,,,,4,
,60.2608453838,-146.8899266180,60.2600000000,-146.7550000000,,,,,,,4,
,60.5191323693,-147.7505003340,60.5200000000,-147.9100000000,,,,,,,4,
,60.5191323693,-147.9106070710,60.5200000000,-147.8300000000,,,,,,,4,
,60.5191323693,-147.8305537020,60.5200000000,-147.7500000000,,,,,,,4,
,60.5191323693,-147.7505003340,60.5200000000,-147.9100000000,,,,,,,4,
```

a

cruise plan data is stored in a comma delimited test file

planner can read and display cruise lists from text editors and spreadsheets

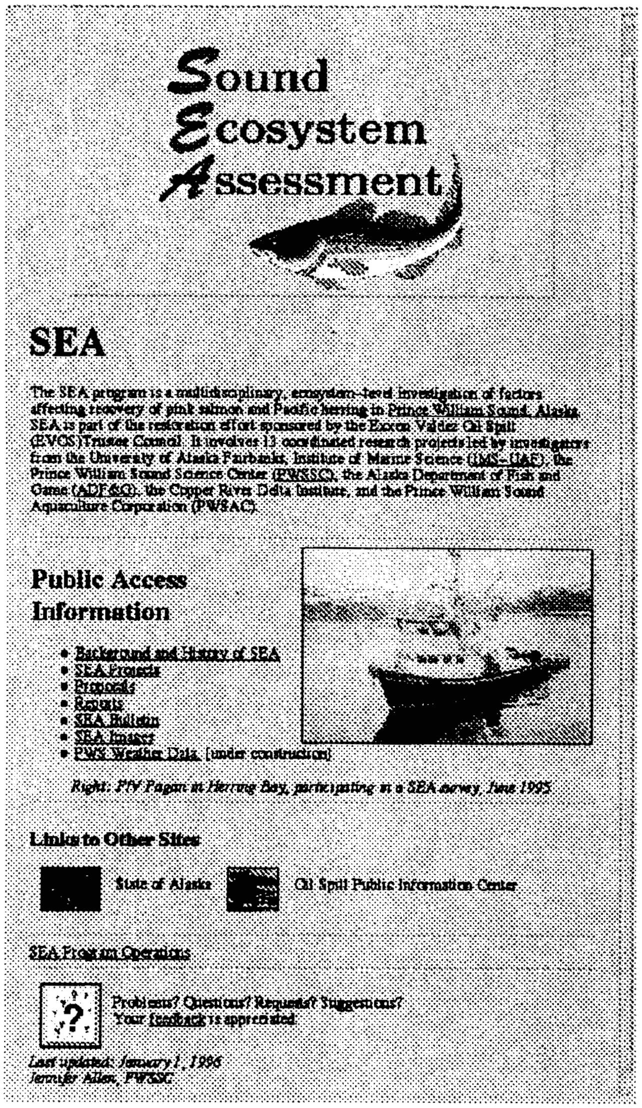
cruise planner file used with spreadsheet to compute transect times and resolve resource conflicts

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1												nts/hr			km	hours
2	HE13	60.2600	-146.8900	60.2600	-146.8900						CTD				0.0	
3	HE11	60.2600	-146.7550	60.2600	-146.7550						CTD				0.0	
4	SEA33	60.0150	-147.7000	60.0150	-147.7000						CTD				0.0	
5	SEA31	60.0500	-147.7850	60.0500	-147.7850						CTD				0.0	
6	SEA18B	60.5200	-147.7500	60.5200	-147.7500						CTD				0.0	
7	SEA16	60.5200	-147.9100	60.5200	-147.9100						CTD				0.0	
8	SEA26	60.2100	-147.9900	60.2100	-147.9900						CTD				0.0	
60		60.5992	-146.4016	60.7800	-146.9300						OB2->NS1	4.0			41.3	5.6
61		60.7794	-146.9300	60.7200	-146.9300						NS1->NS22	4.0			6.6	0.9
62		60.7194	-146.9300	60.6500	-146.9300						NS2->NS3	4.0			7.7	1.0
63		60.6493	-146.9300	60.5200	-146.9300						NS3->NS4	4.0			14.4	1.9
64		60.5191	-146.9300	60.3700	-146.9400						NS4->CS12	4.0			16.6	2.2
65		60.3710	-146.9420	60.2600	-146.8900						CS12->HE13	4.0			13.1	1.8

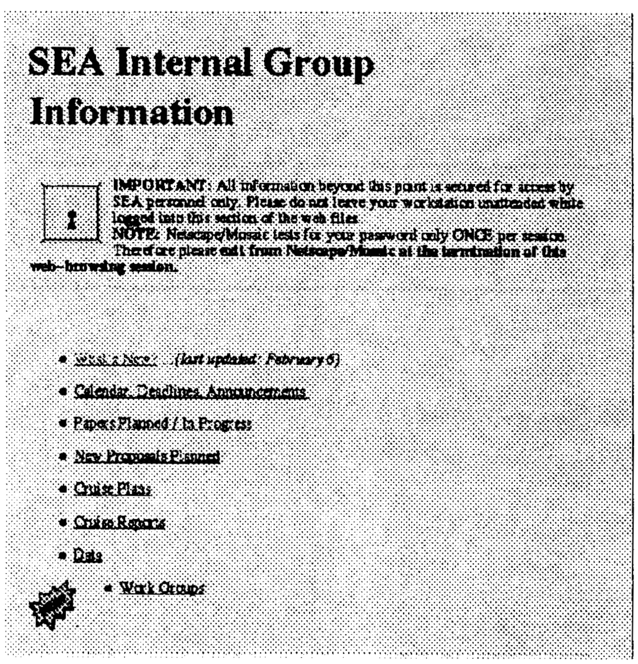
Sheet1

b

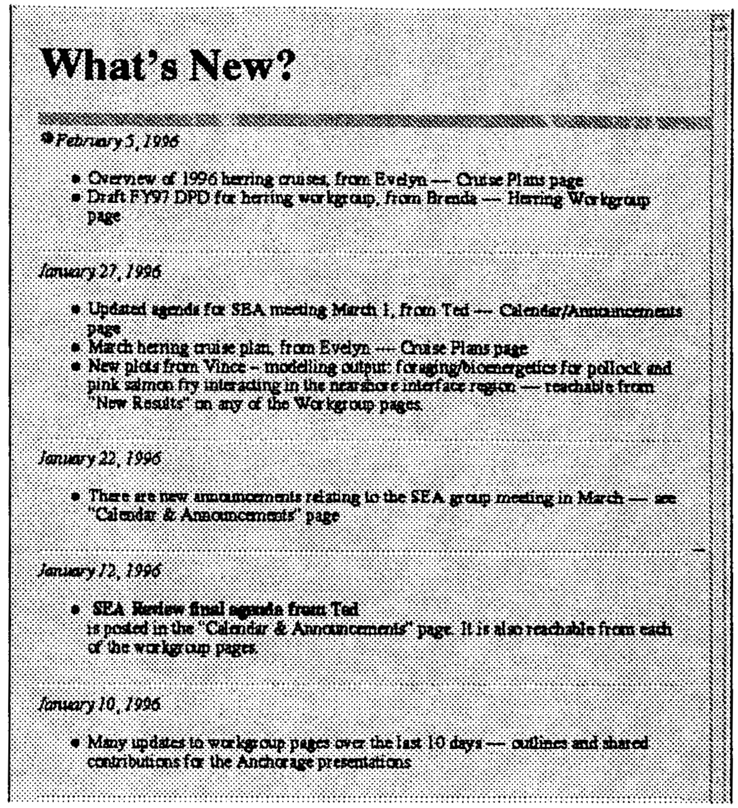
Figure 8. Cruise planner file structure (a).
Planner file used in a spreadsheet for confllict resolution and cruise optimization (b).



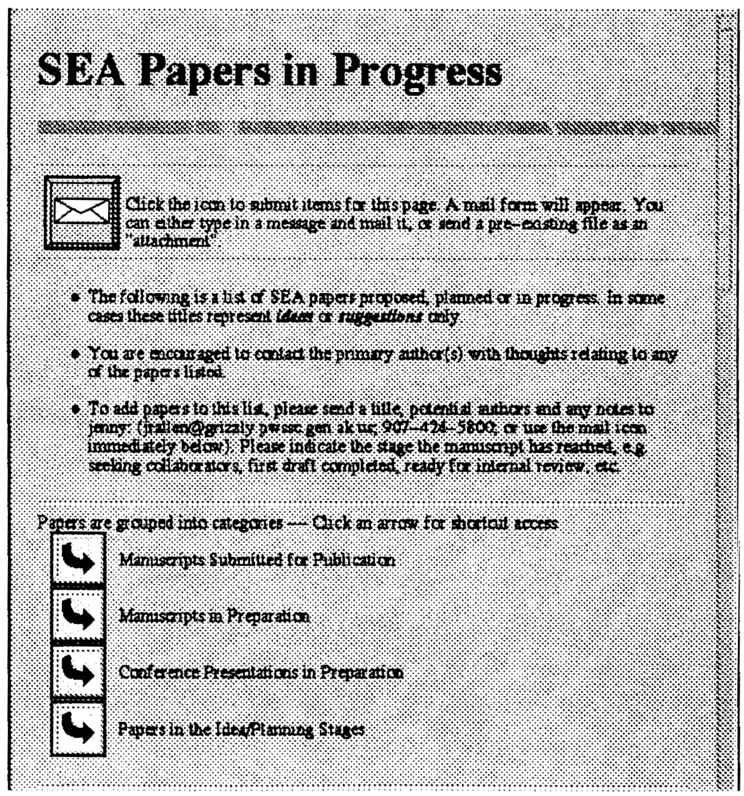
a



b



c



d

Figure 9. The SEA home page (a); samples of the project pages implemented to facilitate the SEA collaboration (b-d).

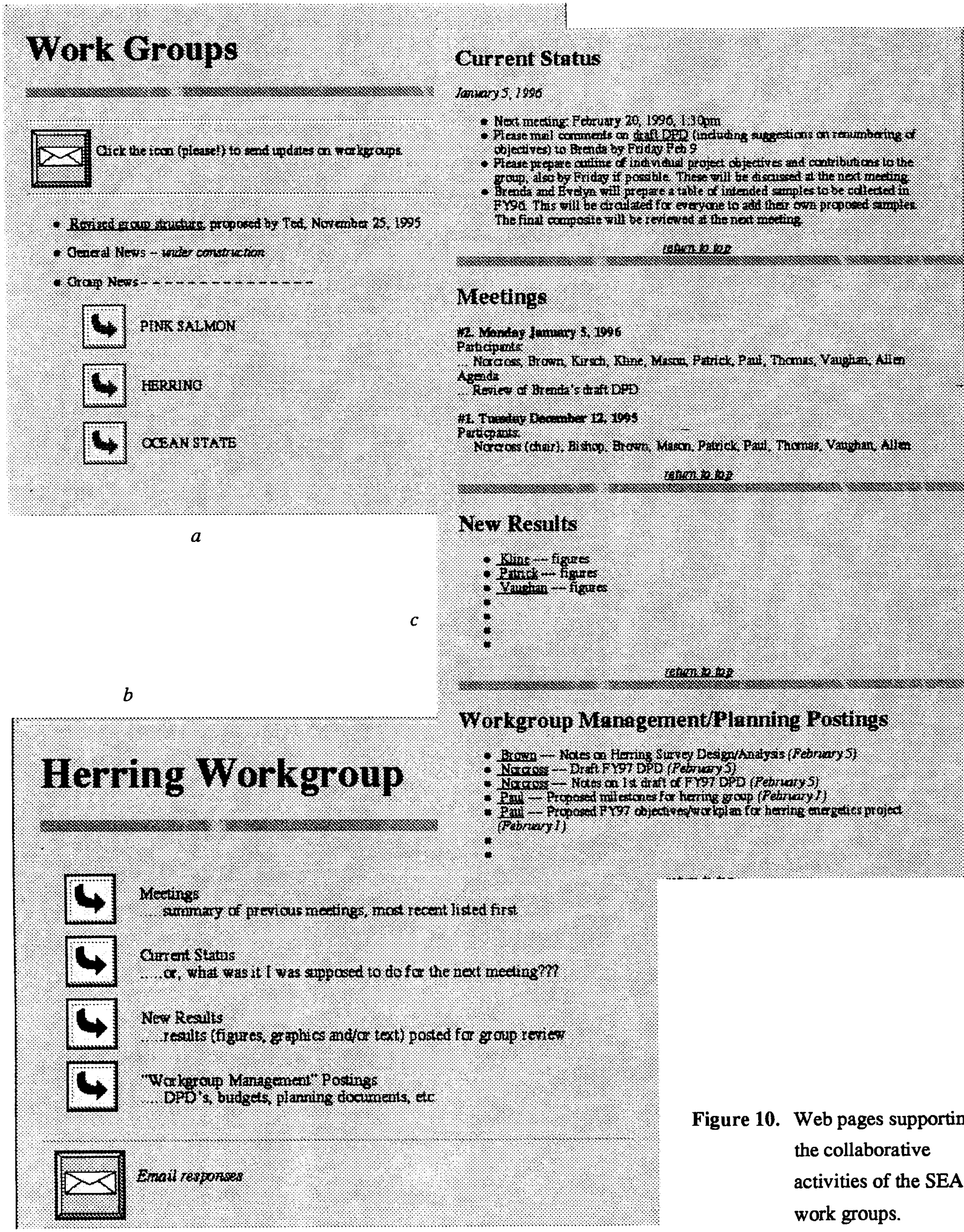


Figure 10. Web pages supporting the collaborative activities of the SEA work groups.

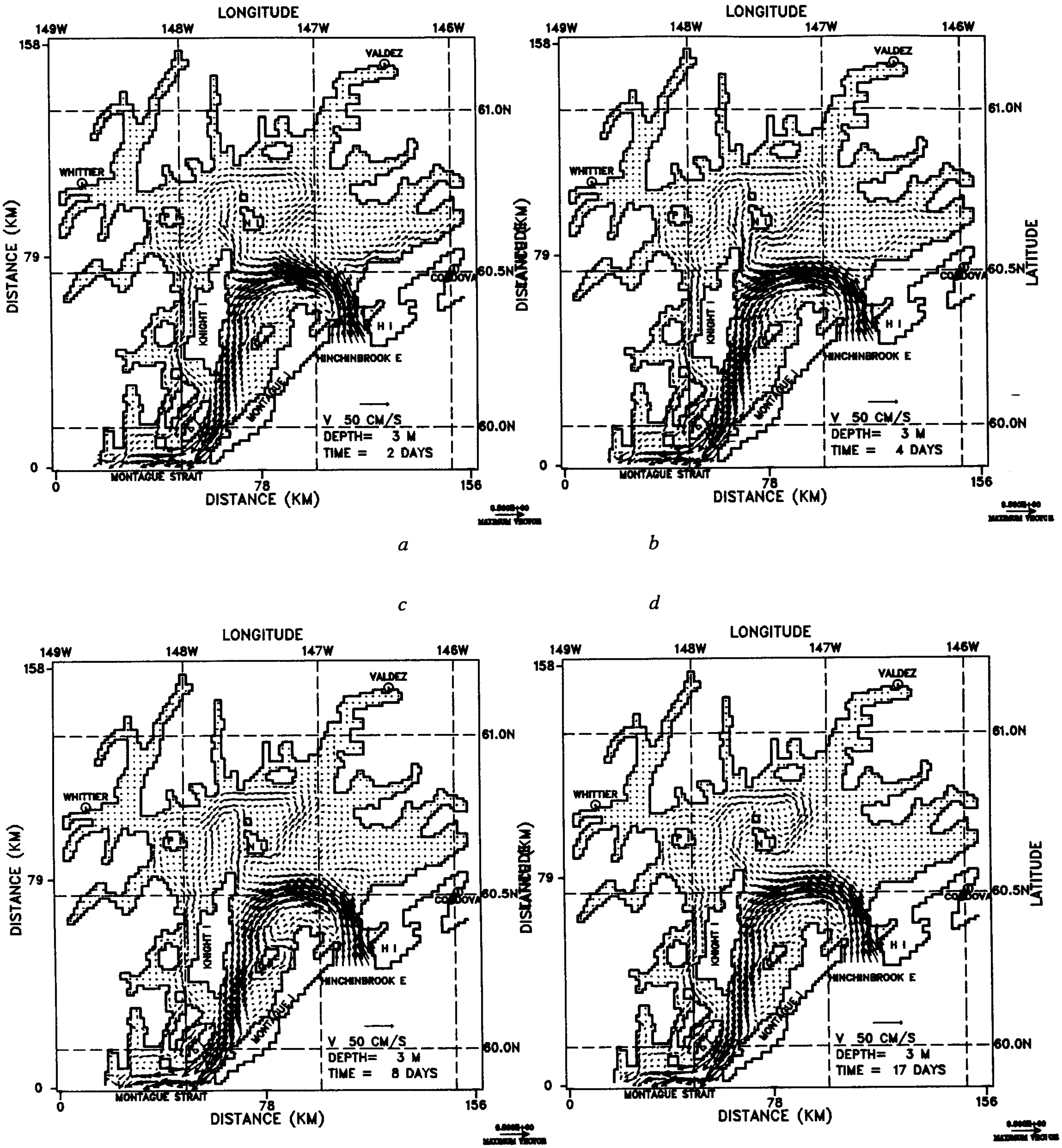


Figure 11. Velocity fields at 3m depth at 2 days (a), 4 days (b), 8 days (c), and 17 days (d) following 17 day spin-up. The sole forcing is in-flow of 0.3 Sv at Hinchinbrook Entrance.

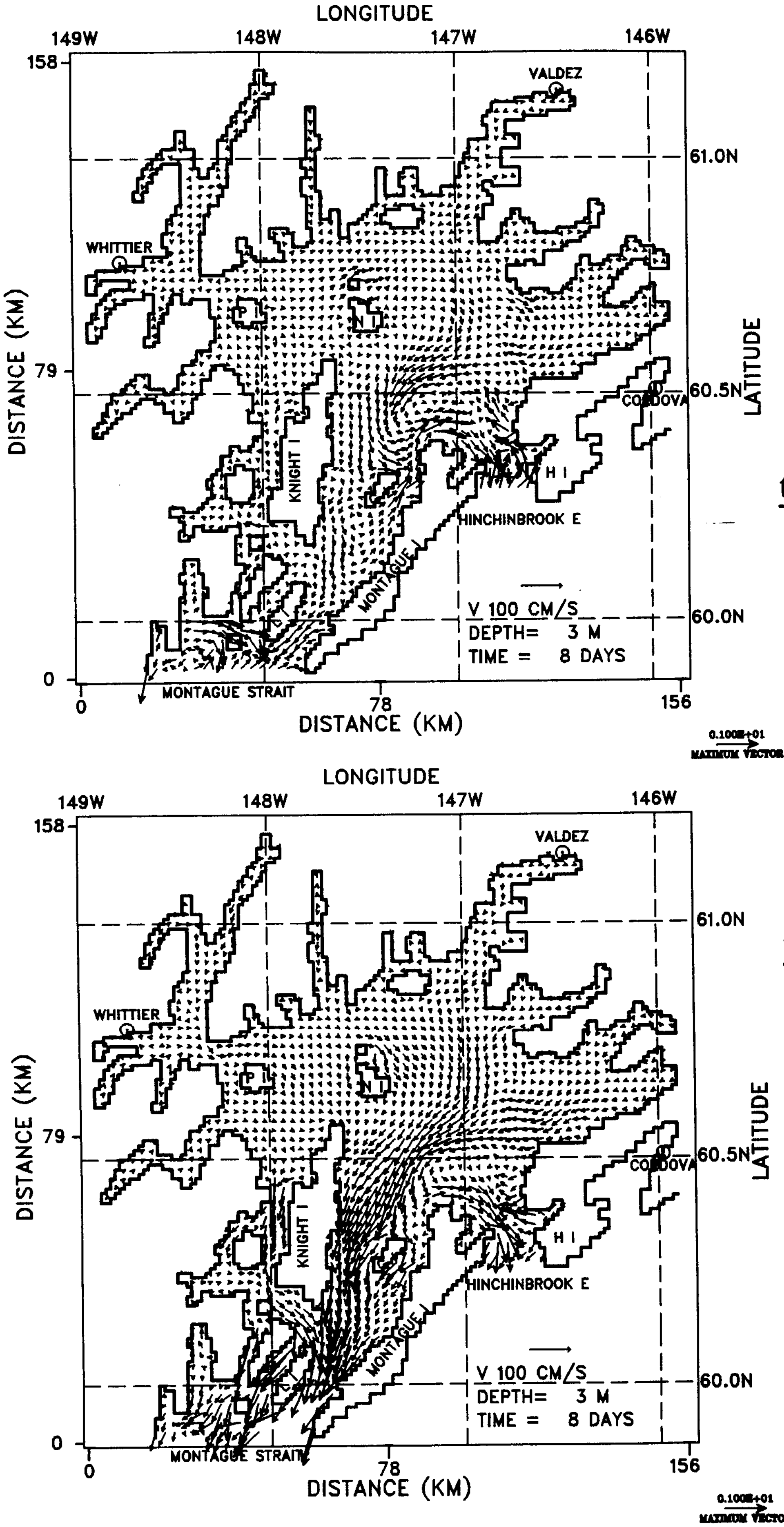
forcing:

in-flow at Hinchinbrook Entrance (HE)

M_2 tide

tide stage:

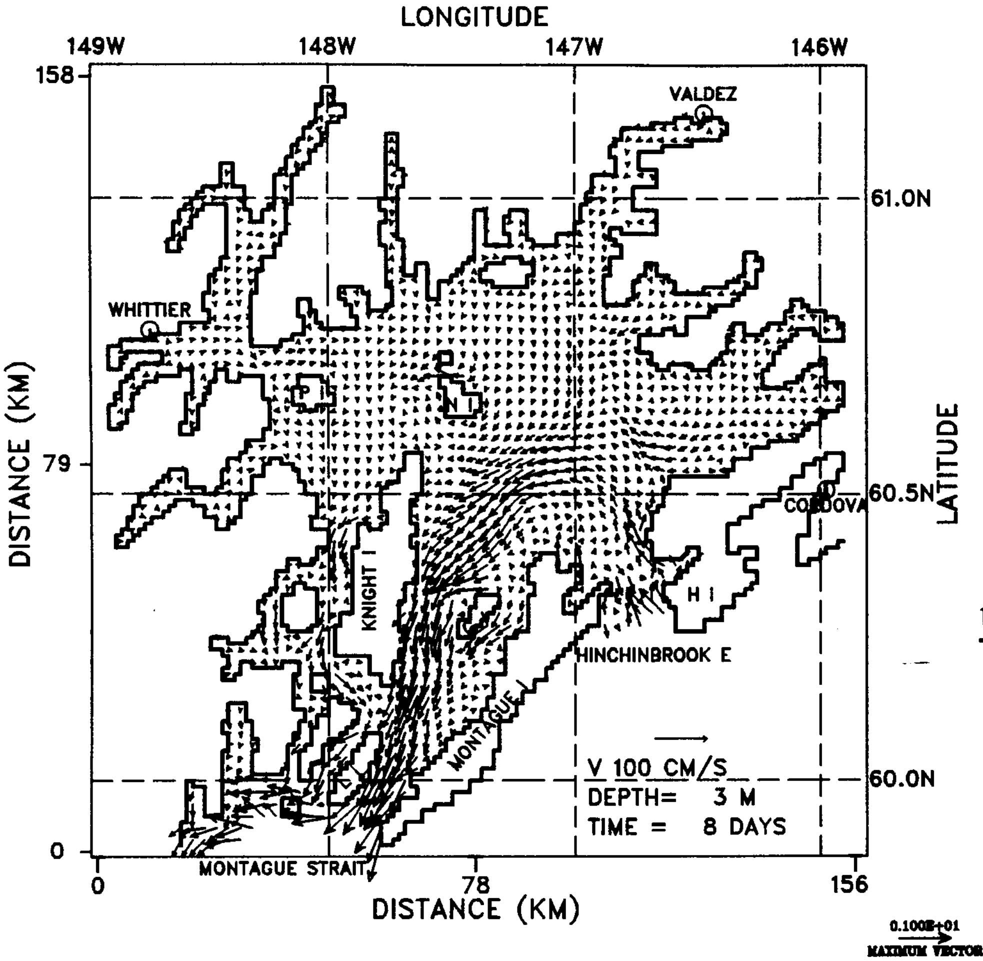
high at HE



tide stage:

ebb at HE

Figure 12. Velocity fields at 3m depth at high tide and at ebb tide.



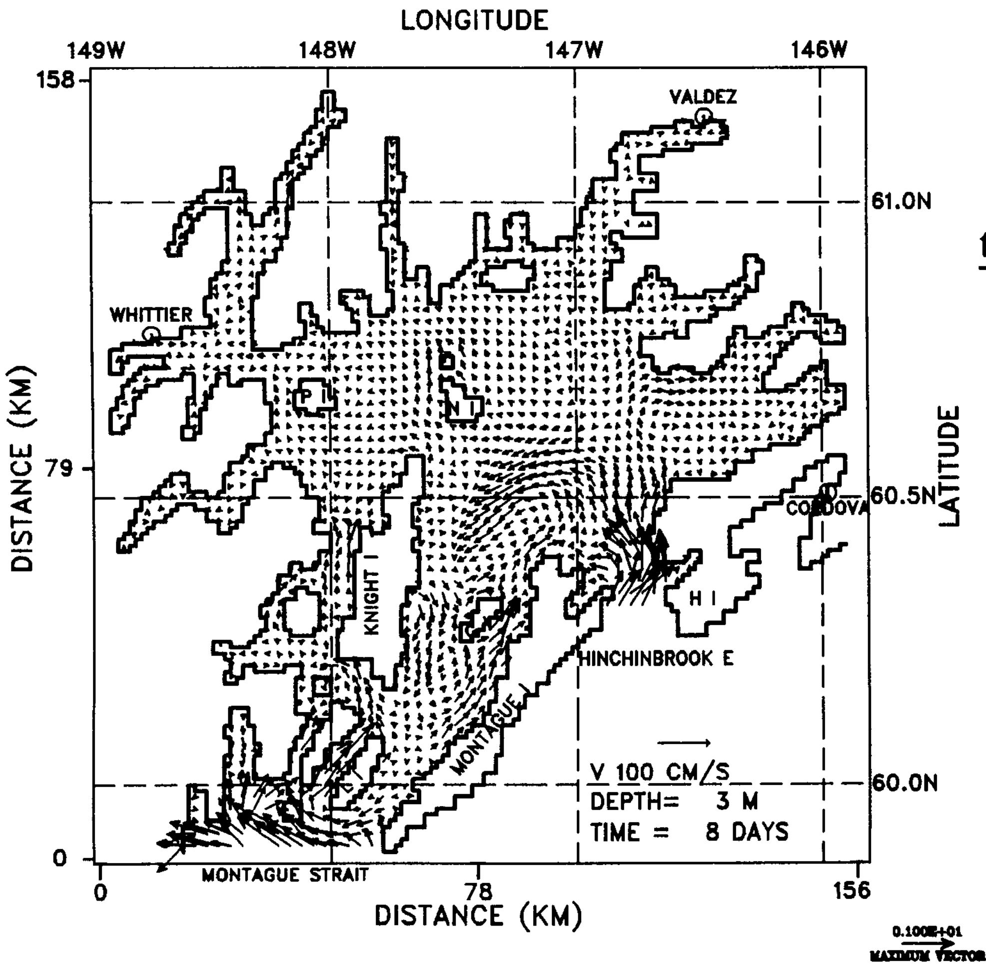
forcing:

in-flow at Hinchinbrook Entrance (HE)

M_2 tide

tide stage:

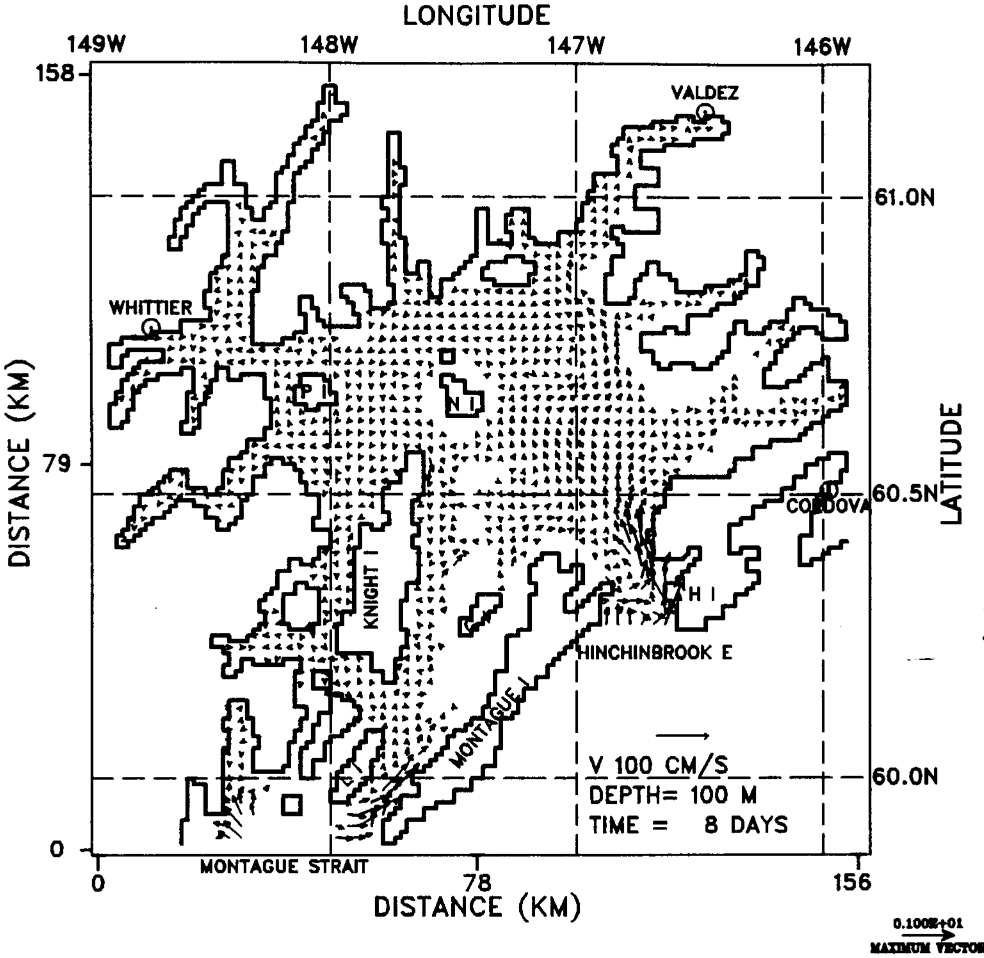
low at HE



tide stage:

flood at HE

Figure 13. Velocity fields at 3m depth at low tide and at flood tide.



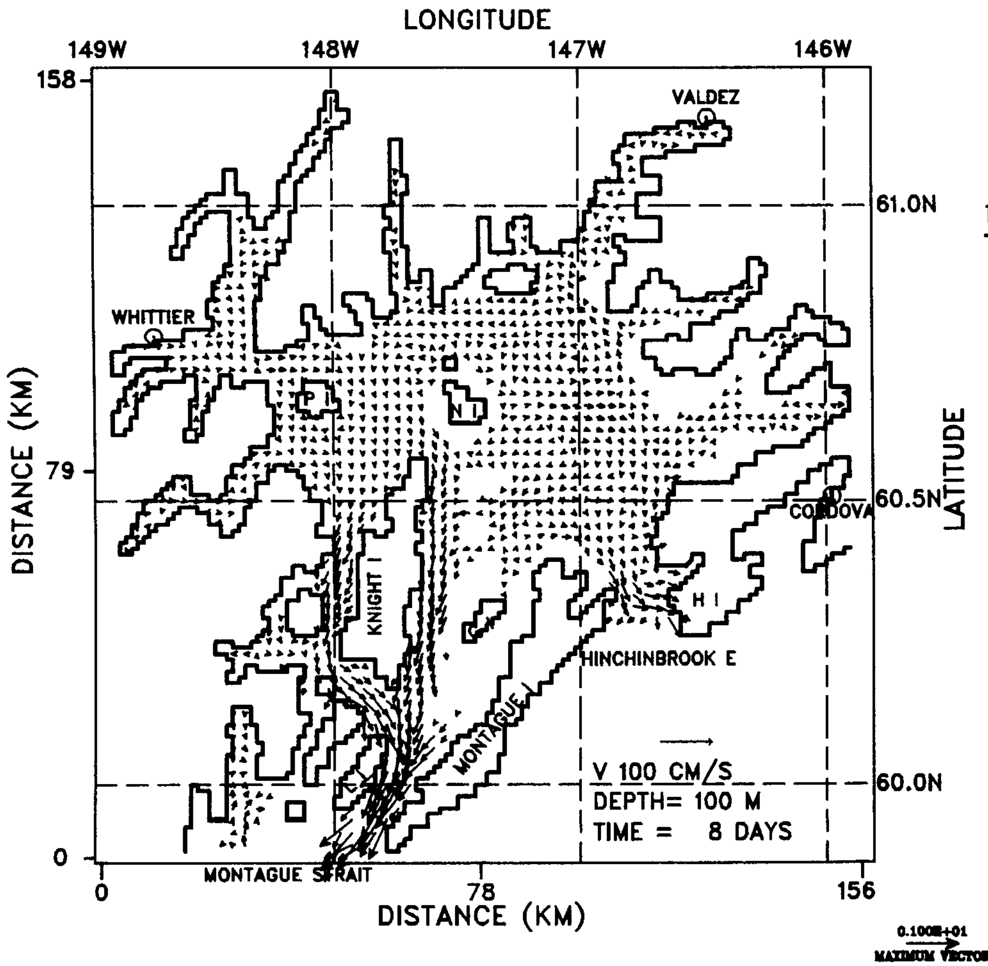
forcing:

in-flow at Hinchinbrook Entrance (HE)

M_2 tide

tide stage:

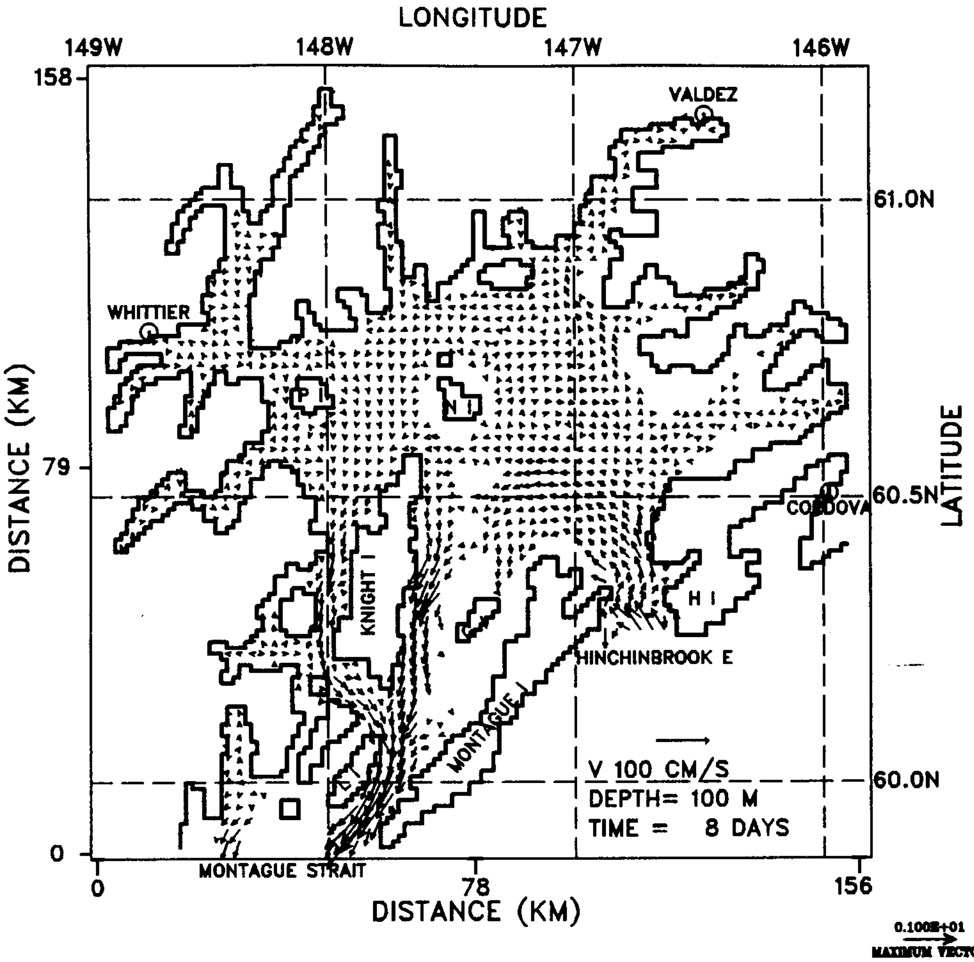
high at HE



tide stage:

ebb at HE

Figure 14. Velocity fields at 100m depth at high tide and at ebb tide.



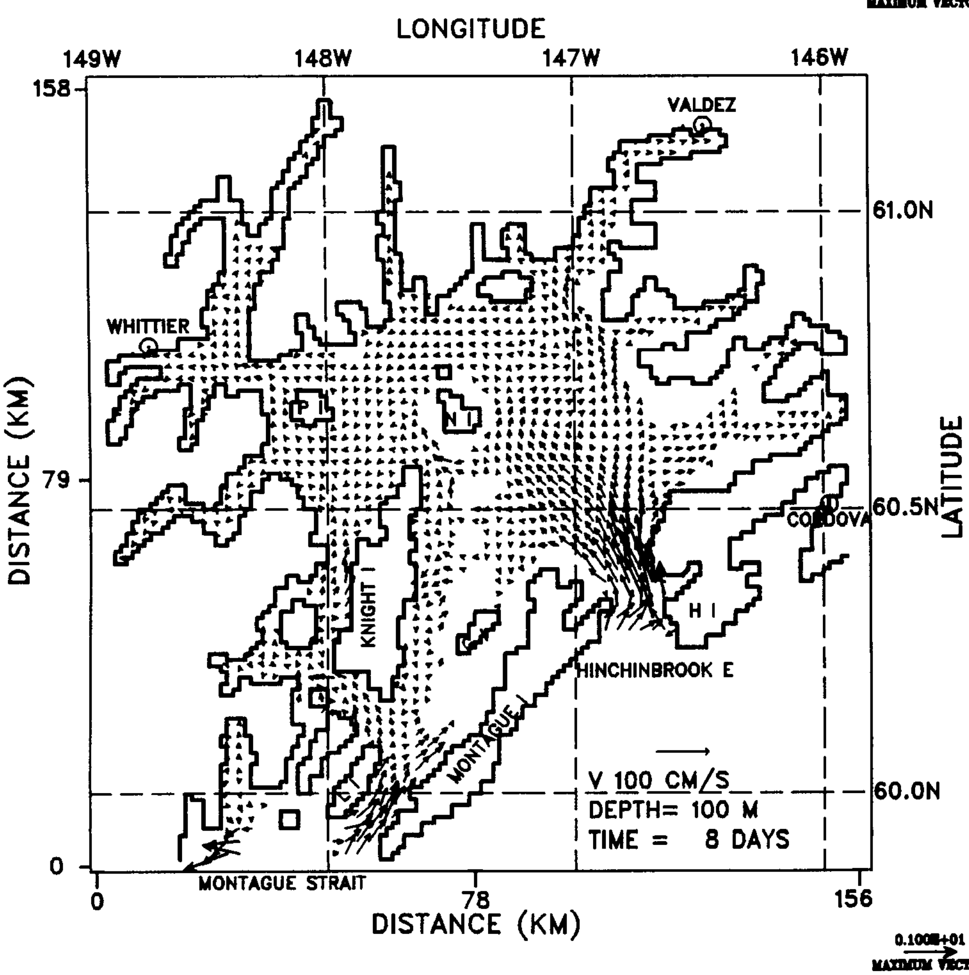
forcing:

in-flow at Hinchinbrook Entrance (HE)

M_2 tide

tide stage:

low at HE



tide stage:

flood at HE

Figure 15. Velocity fields at 100m depth at low tide and at flood tide.

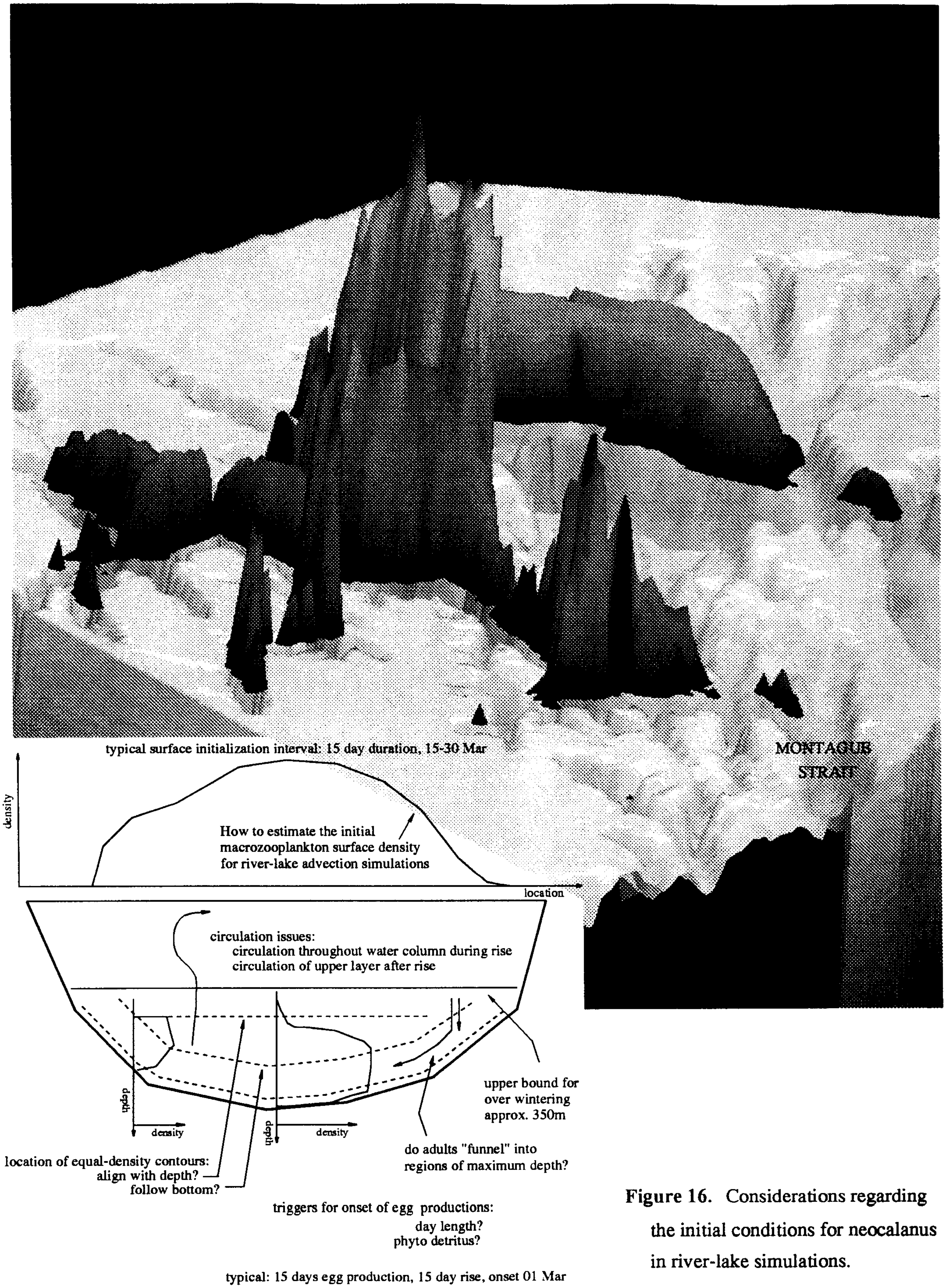


Figure 16. Considerations regarding the initial conditions for neocalanus in river-lake simulations.

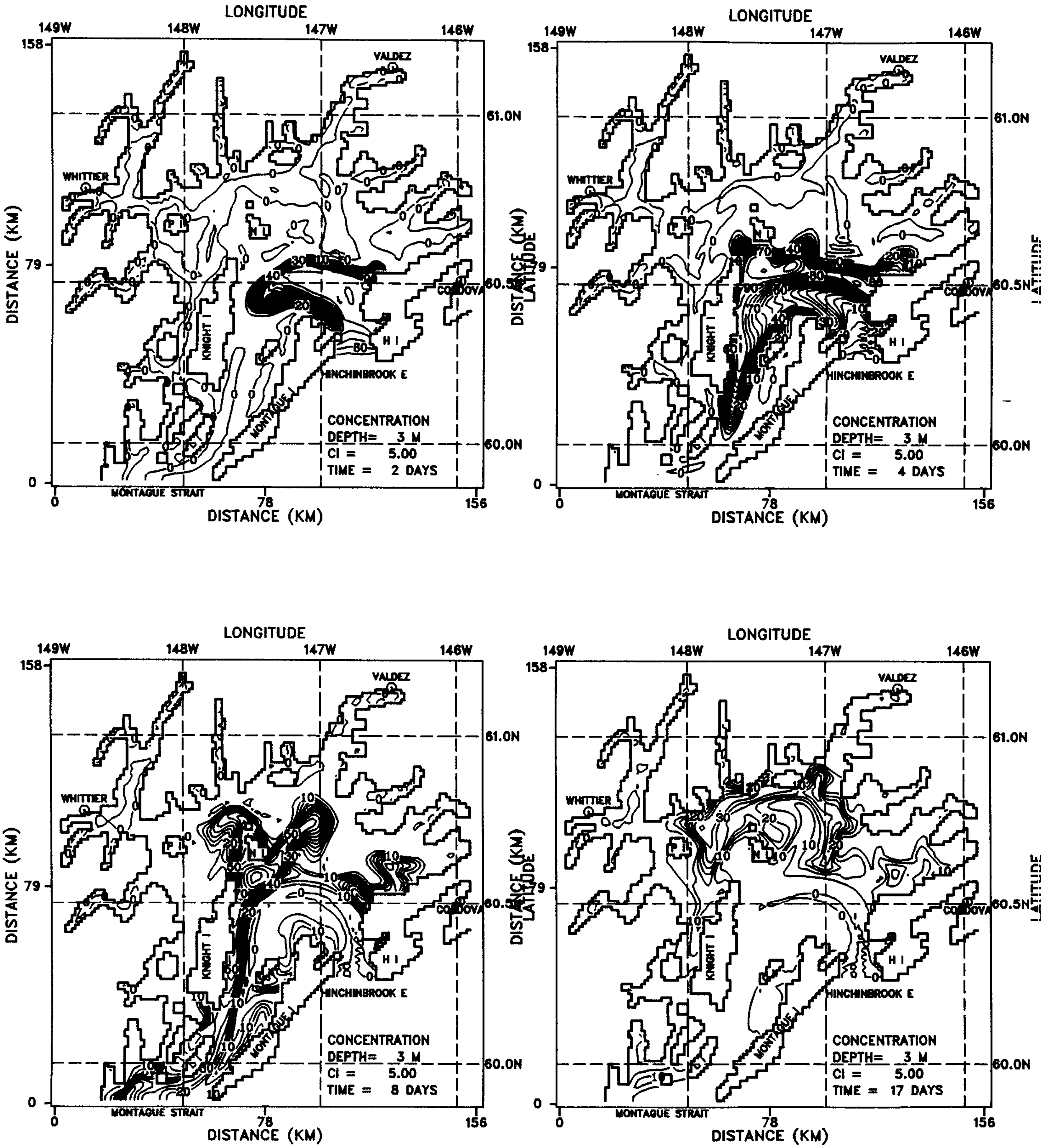


Figure 17. Time varying tracer concentration at 3m depth due to the circulation fields shown in Figure 11. After a 17 day spin-up the tracer is released along the upper 40m of the model boundary at Hinchinbrook Entrance at a uniform rate and for a period of 6 days.

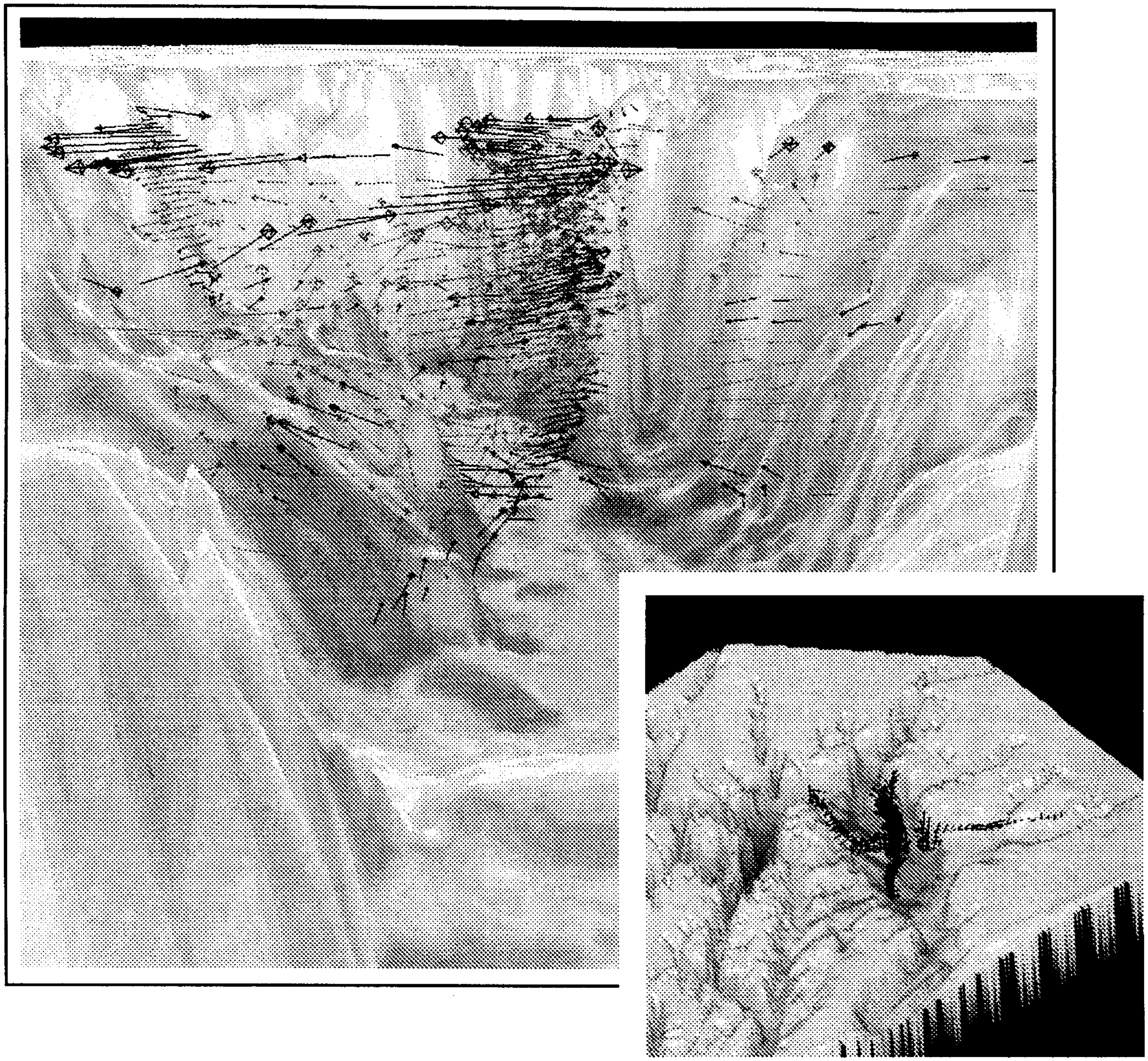


Figure 18. Velocity fields from ADCP transects. The time of the transects extends over multiple tide stages. Both time and position are needed to compare tide model output with towed ADCP data.

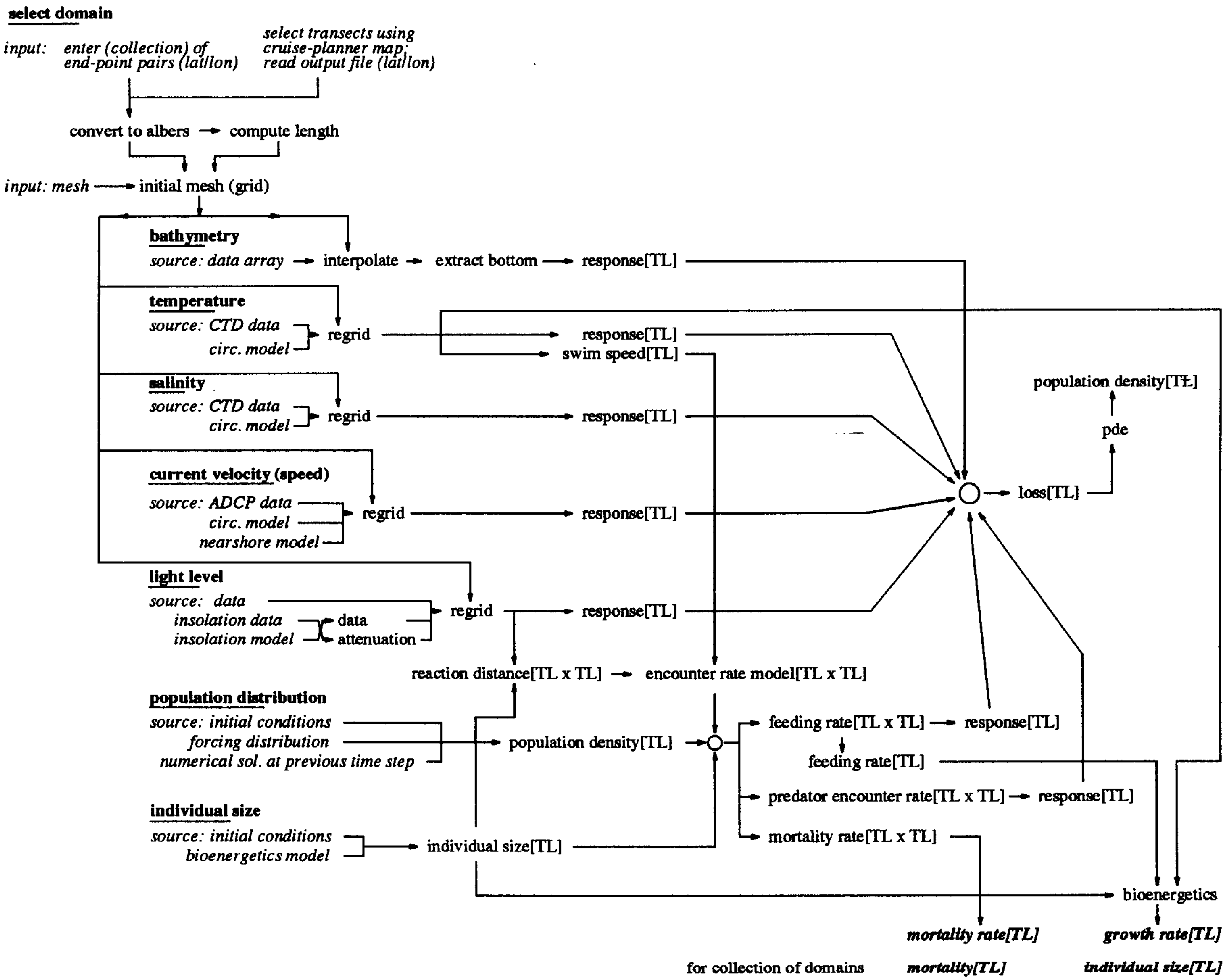


Figure 19. Informal chart representation of the fish model. The arrows indicate the flow of information from "input" to "output."

Figure 20. Set of one-dimensional domains suitable for the one-dimensional fish model.

population density $u(x,t)$

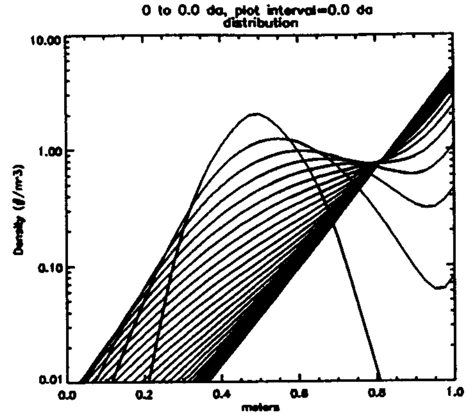
population *flux* (redistribution)

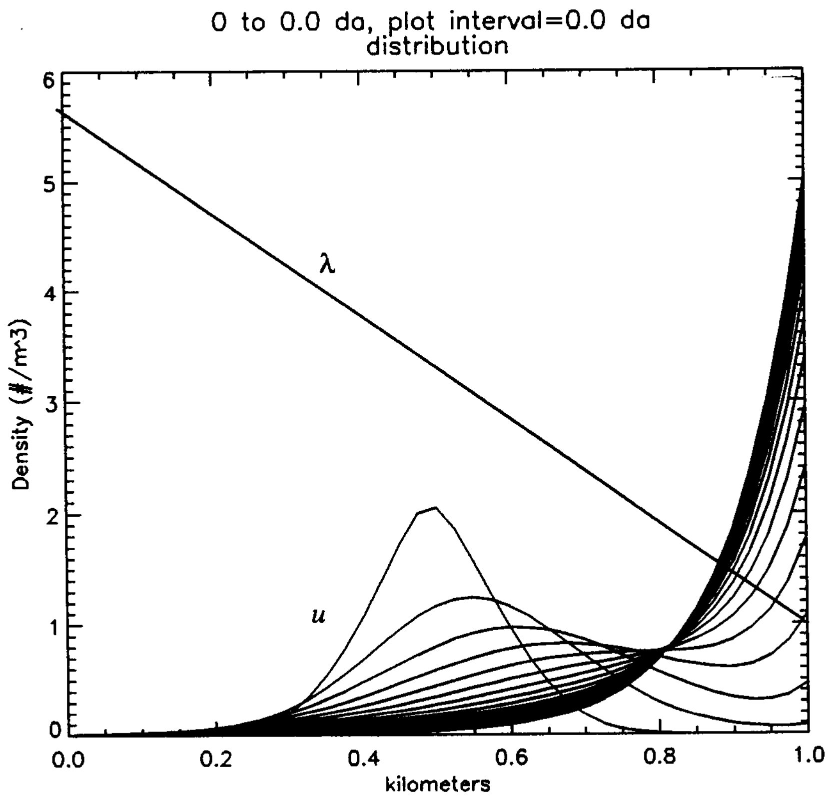
by diffusion $-D\frac{\partial u}{\partial x}$ (from high density u to low)

by taxis $-\chi u\frac{\partial \lambda}{\partial x}$ (from high loss λ to low)

$$D\frac{\partial u}{\partial x} + \chi u\frac{\partial \lambda}{\partial x} = 0 \quad \text{for } x \text{ on boundary}$$

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial x^2} + \chi\frac{\partial}{\partial x}\left(u\frac{\partial \lambda}{\partial x}\right) \quad \text{for } x \text{ in domain}$$





to increases loss :

decrease feeding rate ↘ ↙ increase predation pressure

λ goes up

↗

change environment to increase stress
(e.g., change current velocity, salinity, temperature)

Figure 21. **Debugging example: linearly decreasing loss function with distribution initially concentrated near the middle of the domain.**

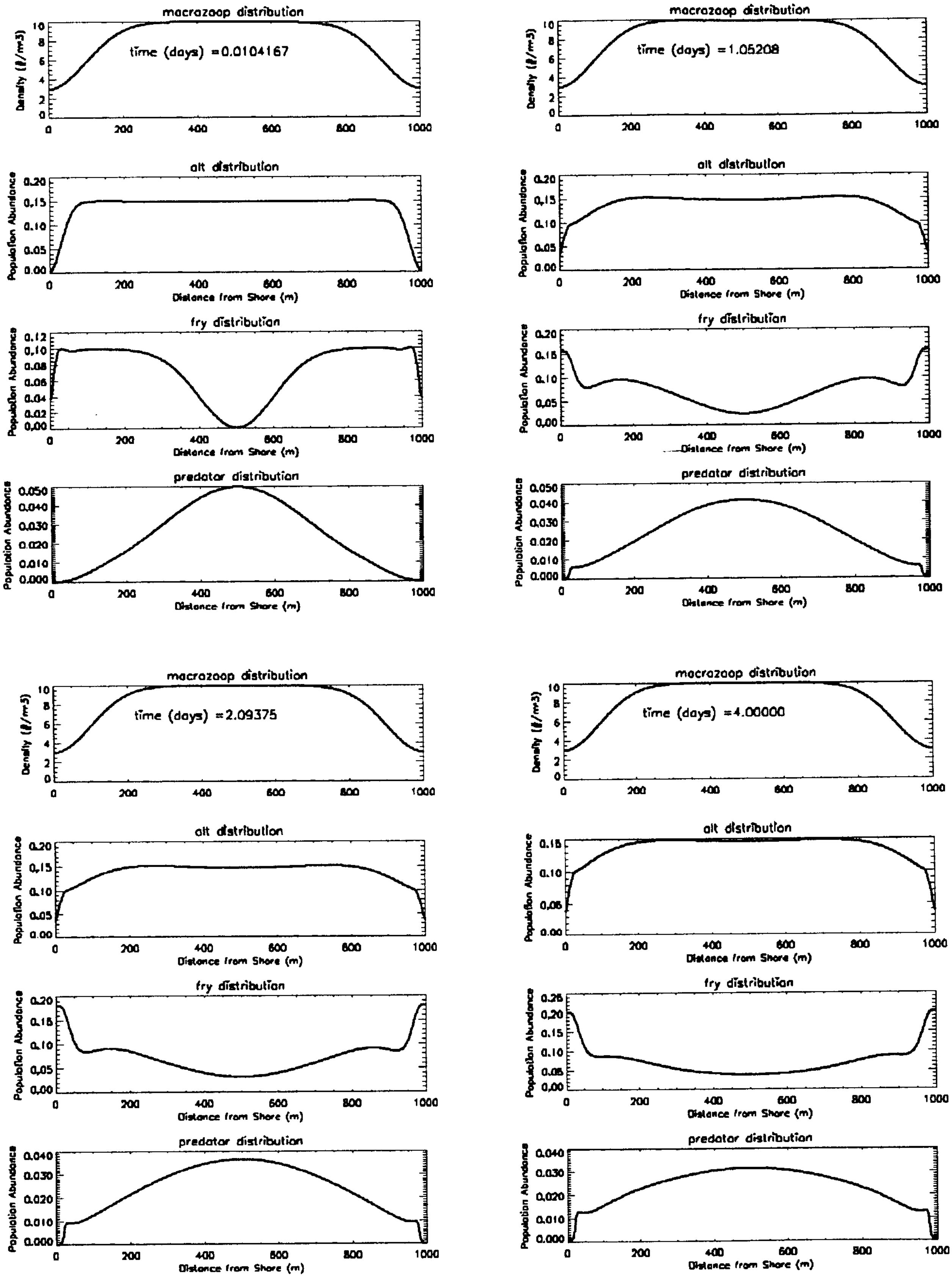
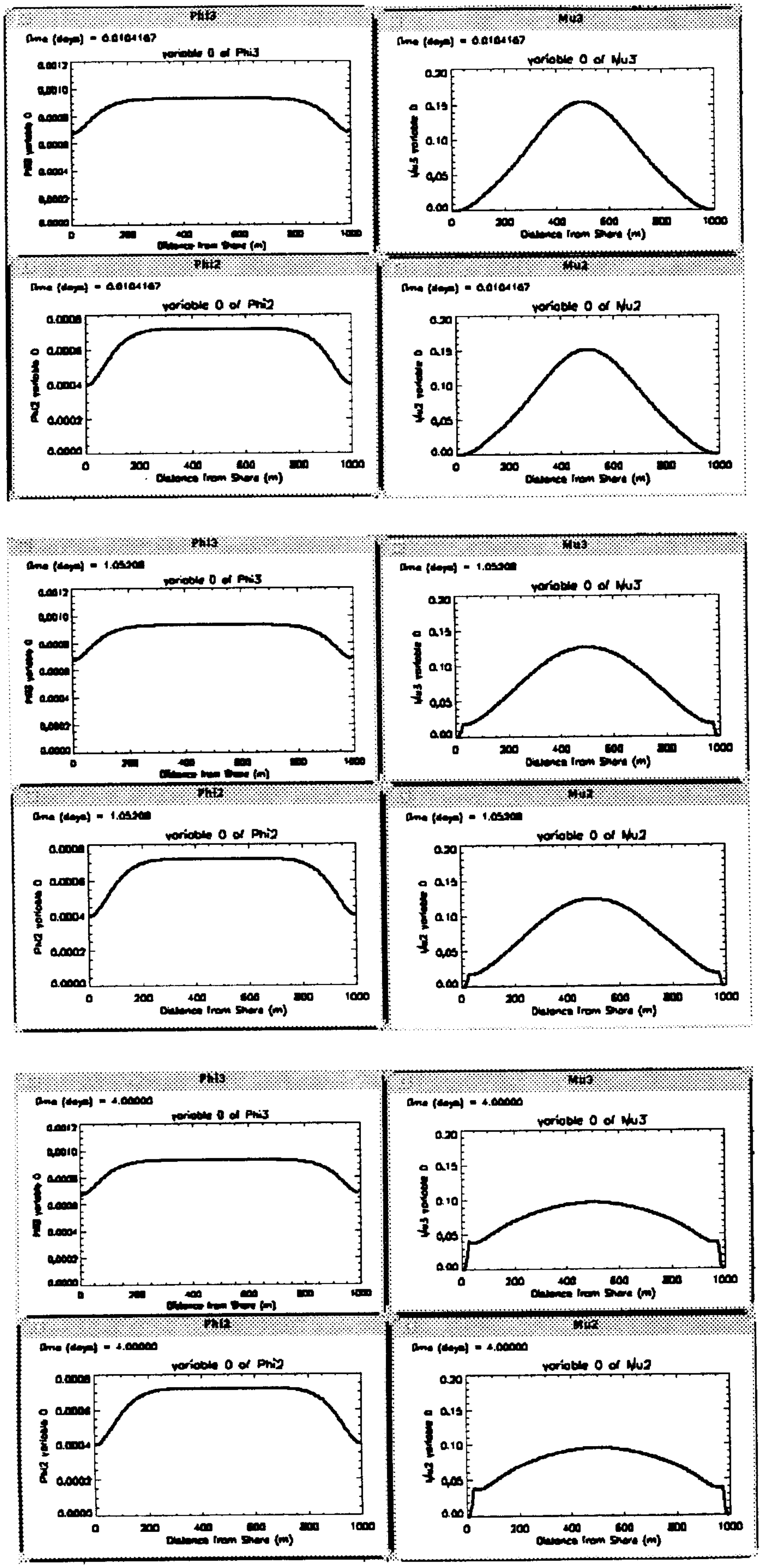


Figure 22. Four day spin-up of coupled system with four trophic levels.
The trophic levels are neocalanus, age-0 pollock, pink salmon fry, and adult pollock.



upper panels (xx3): pink salmon fry

left panel: feeding rate

right panel: predator encounter rate

lower panels (xx2): age-0 pollock

Figure 23. The manner in which the rates for feeding and for the encounter with predators change during the four day spin-up

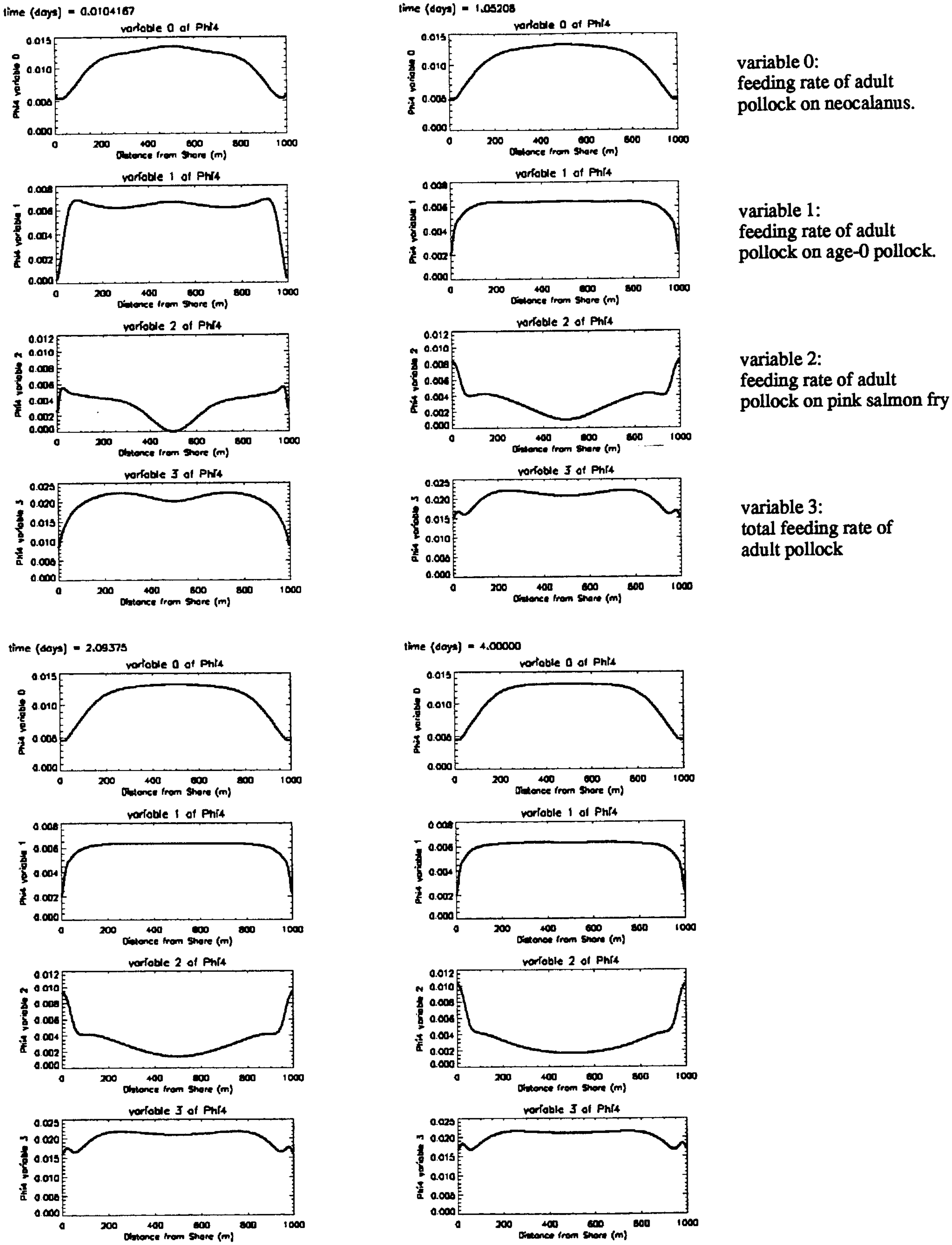


Figure 24. The change in the species specific and total feeding rates of adult pollock during the four day spin-up.

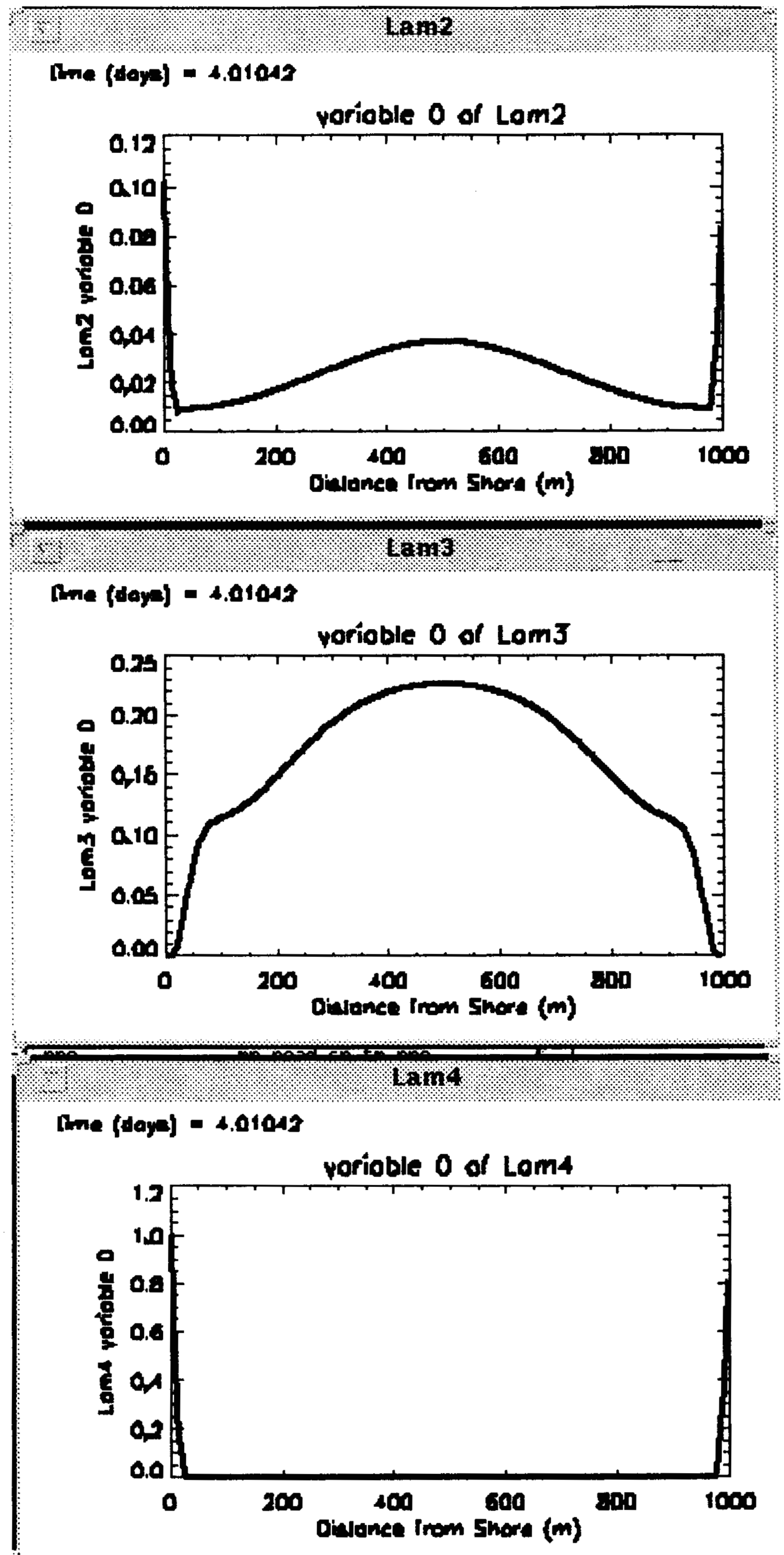
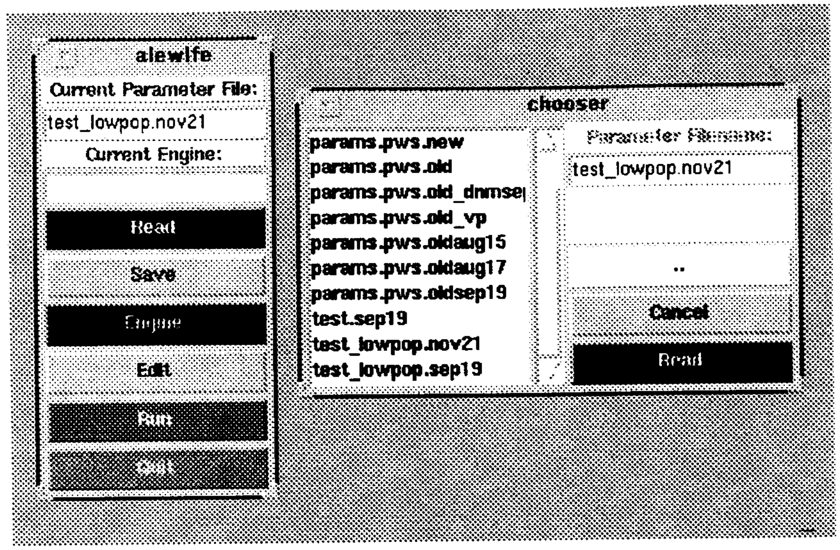


Figure 25. For population densities in a steady state the gradient of the loss function is balanced by the gradient of the density function, up to a constant scaling factor.

Read-in the selected parameter file.

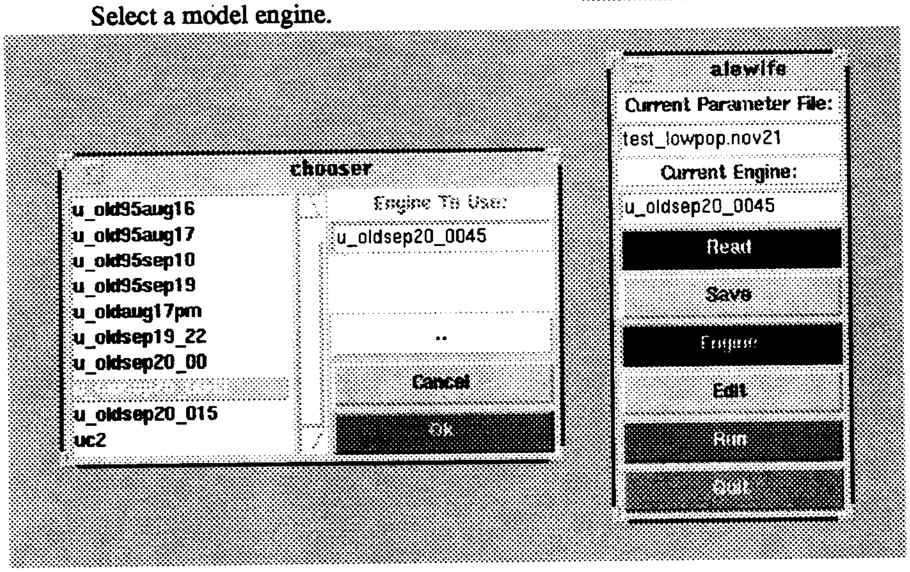
Open the "engine" chooser window.

Open "Model Parameter Categories" window.



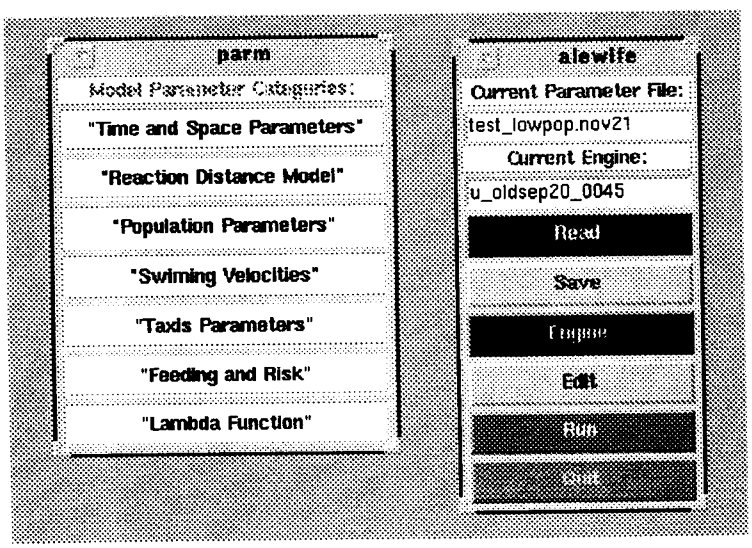
Select a model engine.

Select a parameter file to be used when the selected engine runs.



Model parameters are grouped into tables by category.

Open one or more of the parameter tables.



This parameter file will be used.

This engine will execute.

Run the engine with the parameter file

Figure 26. The startup windows of the graphical user interface in *Alewife* Version 3.

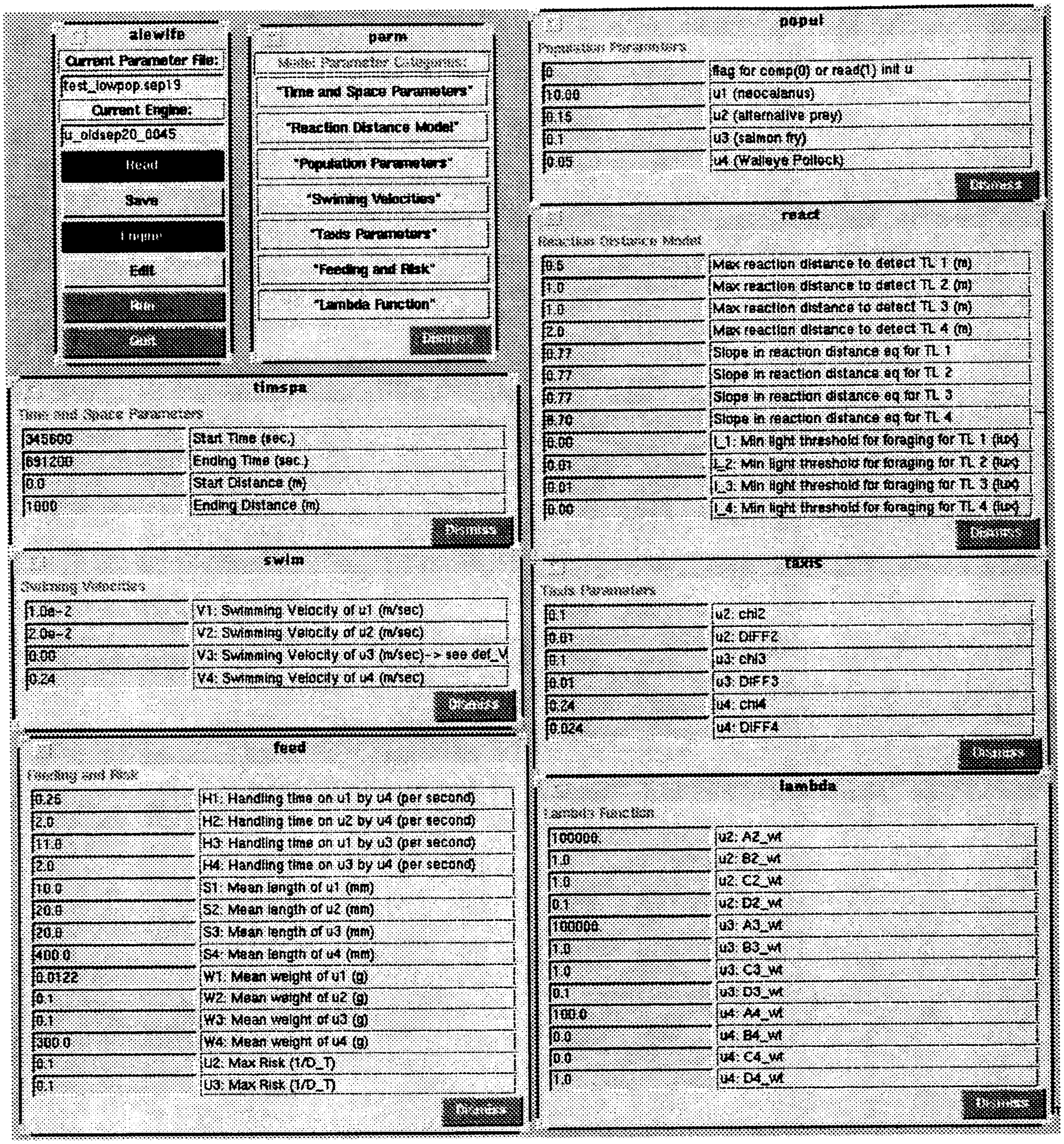


Figure 27. The editable parameter tables in *Alewife* v3.

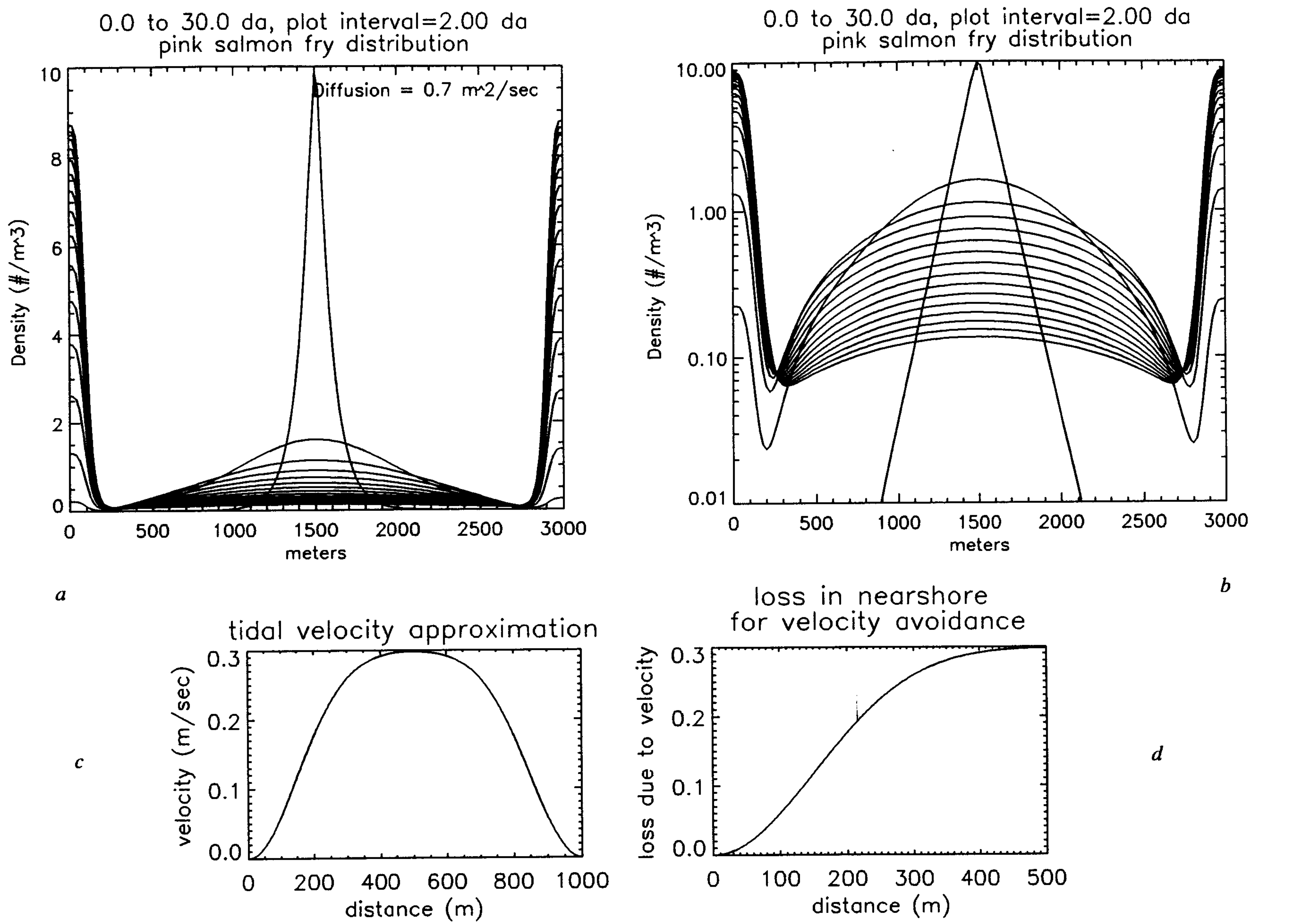


Figure 28. Evaluation of the consequences of avoidance of off-shore currents by pink salmon fry using *Alewife* Version 4. The system consists of a single trophic level (fry) and a loss functiion that depends only on current velocity. A constant (non-tidal) velocity profile as in (c) was used with a loss function as in (d).

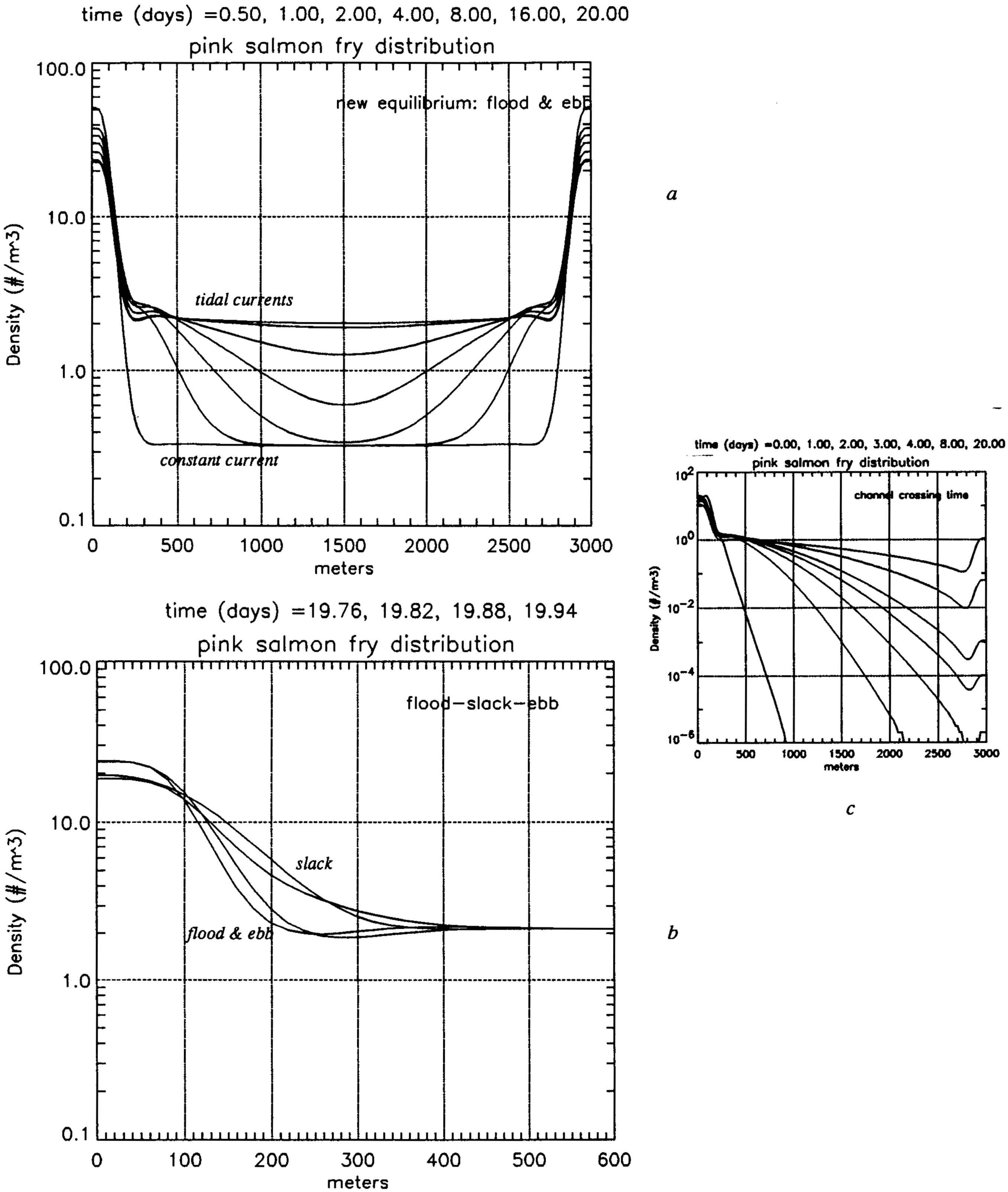


Figure 29. The consequeces of off-shore current velocity avoidance for the case of tidal currents. After spin-up to steady state the constant current of Figure 28 is made tidal resulting in substantial "leakage" from nearshore to off-shore (a) and a tidally synchronized inshore-offshore shuffle (b). Examination of channel crossing time (c) for the same system assumptions.

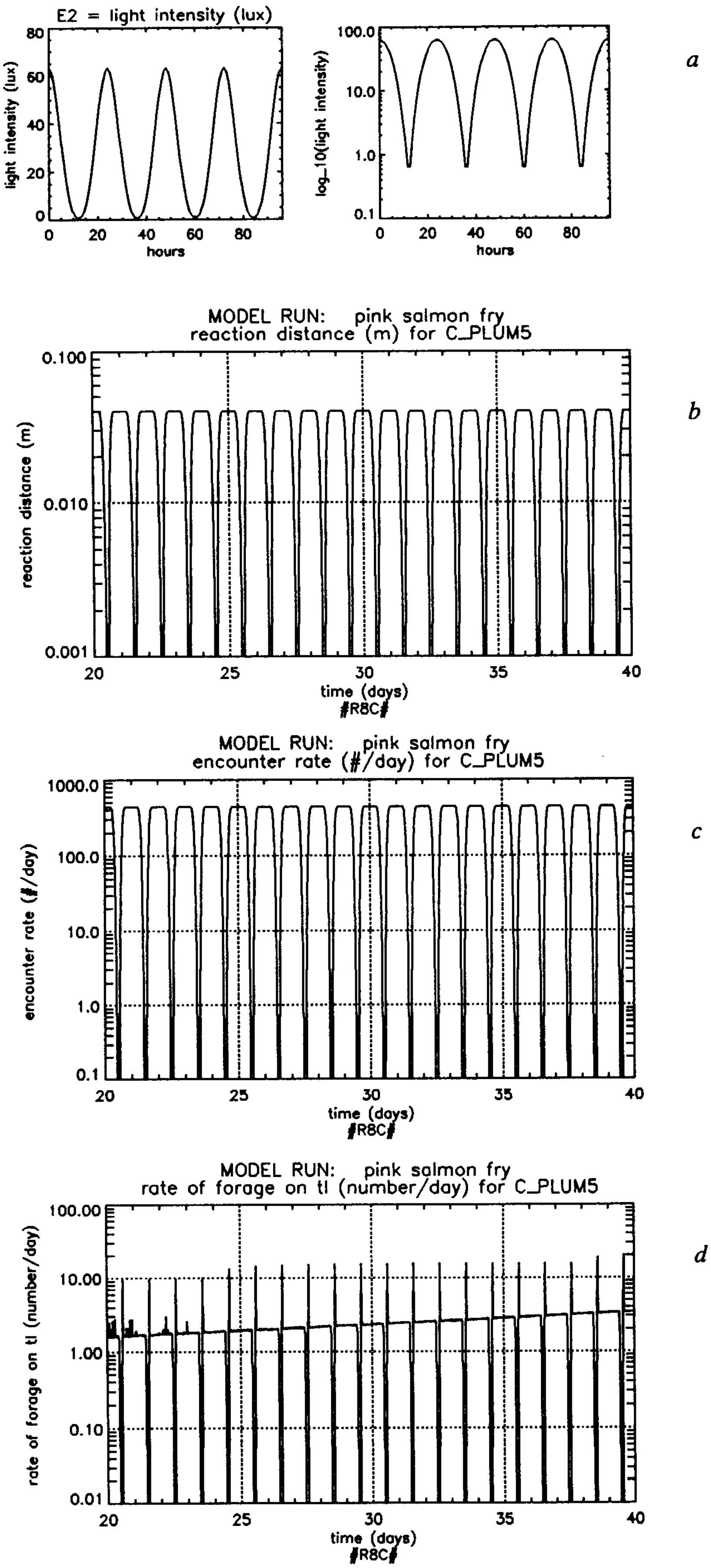


Figure 30. Foraging model for pink salmon fry is of sufficient scope to simulate well night fasts, high morning feeding rates, and day feeding that maintains gut fulness.

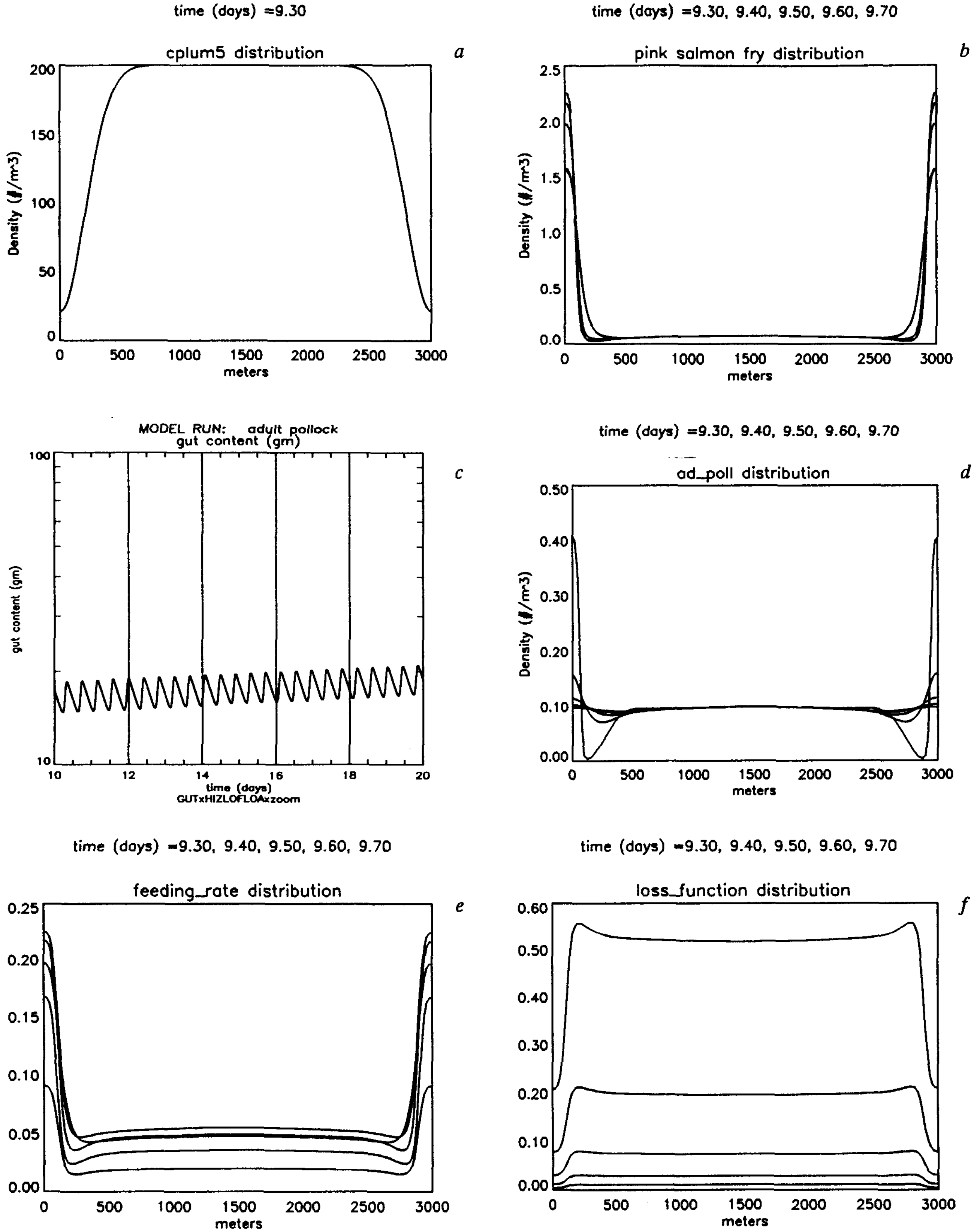


Figure 31. Simulation of pollock - fry interactions using glut-like feeding at 12 hour intervals (c), fixed neocalanus distribution (a), and the tidally shuffling fry distribution (b). Pollock disttribution diffuses to uniform during non-foraging interval. At onset of feeding the distribution quickly becomes disjoint as pollock move to one or the other feeding area.

Appendix 1

An AVS Interface to the Aurora Dataserver

An AVS Interface to the Aurora Dataserver^{TM}

Charles Falkenberg
Computer Science Department and
Advanced Visualization Laboratory
University of Maryland
College Park, MD 20742

Ravi Kulkarni
Advanced Visualization Laboratory
University of Maryland
College Park, MD 20742

ABSTRACT: To investigate database support for scientific visualization, we built several AVS modules which provide access to scientific datasets in stored in the *Aurora Dataserver.* Aurora is a scientific database which provides the essential functions to define and access multi-dimensional scientific data. The modules we built allow datasets to be selected from Aurora and converted to the internal representation needed by the AVS scientific visualization system. In addition to providing database support for scientific visualization, the modules were intended to be used by a large research project in Prince William Sound, Alaska. This project, called SEAData, is collecting a wide range of oceanographic and biologic data to assess the long term impact of the Exxon *Valdez* oil spill. In this paper we briefly describe the functionality of the Aurora system, the use of the modules we built, and the tests we ran using scientific datasets from Prince William Sound. Although the Aurora system will not be extended in the future, our work exposes some of the essential database features needed to support visualization.

This research was supported by Xidak Inc. Aurora Dataserver is a registered trademark of Xidak Inc.

Scientific Datasets and the Aurora Dataserver

Most scientific data are characterized by large multidimensional arrays of samples collected during experiments. Modeling these data posses several problems for a database management system. The sample points can fall into a fixed or regular grid, they can be scattered throughout space, or they can represent a mapping from one space to another. For instance, a two dimensional surfaces may be positioned in a three dimensional space. Capturing these geometries is not the only problem associated with scientific data management however. Meta-data which describes the sampling environment, instrumentation and calibration are all critical if the data are to be analyzed properly. Part of this meta-data is the historical record which is needed to trace the evolution of data from a raw state to an interpreted state. These problems all challenge current database management systems.

The *Aurora Dataserver* was designed by Xidak Inc. to provide scientists and engineers with the database functionality needed to deal with scientific data. The Dataserver implements a data model which incorporates many common geometries and it includes the facility for defining meta-data and historical data. The Aurora data model allows scaler sample types, vector and matrix sample types as well as tuple sample types, which are structures of variables for each sample point. The sampling geometry can be a fixed grid, a variable grid, or a warped field which maps points between spaces of different dimension. Warped fields allow lines and surfaces to be positioned in higher dimensional spaces. Together this range of sample types and coordinate geometries describe a large number of scientific datasets.

In addition, the data model allows meta-data (data describing data) and the historical data to be defined. The meta-data pertain to the dataset as a whole, describing all of the samples in the set. These data can include dates, times, and calibration or instrumentation information. The evolution of the data can also be tracked using the history records. Some history records are created automatically by Aurora when a dataset is modified, but history records can also be added by the user.

The functions of the Aurora Dataserver are accessed through an application programming interface (API) which is used with a program to declare and manipulate datasets in Aurora. Each dataset is a member of a 'class' which defines the sample type, sampling geometry and meta-data. The class for a dataset must be created before the dataset is created. Using the API, datasets can be found by navigating through a hierarchical name space or by using an SQL query string. The SQL query also uses the class definition to make a selection based on the values of the meta-data or the dataset history. Whether an SQL query is used or the name space is searched the result is a list of one or more datasets which match the users request.

Once a dataset has been identified, a subset of the samples can be selected. The subset can be based on the value of the samples or the coordinates of the samples. Value based selection can produce a new geometry which includes only those points whose value satisfies the request. Coordinate based selection can specify sub-ranges along each axis or sub-sampling of points at regular intervals. These two types of coordinate based selection can be combined, creating a hyper-slab at a lower resolution. Together these selection operations give users of visualization, or other scientific analysis software, a wide range of functionality.

AVS Access to Aurora Datasets

The AVS scientific visualization system provides visual manipulation and analysis of scientific data. It has its own data model to describe multi-dimensional data which is similar in some respects to the Aurora data model. Our project was to create AVS modules which bring the functionality of Aurora to a user of AVS through a single graphic interface panel. We created two modules, one to select and read an Aurora dataset into an AVS field and another to create an Aurora class and write out an Aurora dataset from an AVS field. This section describes the use of these two modules.

Read Module The read module allows the selection of a dataset and the selection of samples from within a dataset. The samples are converted into and AVS field which is the output of the module. The dataset can be selected by browsing the Aurora name space or by entering an SQL command which queries the meta-data. The name space is an internal hierarchy of Aurora datasets and classes similar to the Unix directory hierarchy. A pattern can be provided by the user and datasets and classes whose names match the pattern are displayed in the AVS user panel. The pattern can contain normal Unix wild card characters. Figure 1 shows the read module as part of a simple AVS network. A space is provided to type in a pattern for a name search or an SQL string for a database query.

The SQL query selects datasets based on the values of meta-data for each dataset. A particular class must be selected first and then the SQL statement can include any of the meta-data defined for that class. The datasets satisfying the query are displayed on the panel for subsequent selection. Whether a name pattern or an SQL statement is given, the user must manually select one of the datasets by clicking on the user interface panel.

Once a dataset is selected several attributes of that dataset are displayed on the panel to allow a user the ability to read all or part of the samples in the dataset. Sliders are provided to select sub-ranges or a sub-sample interval. A mapper string can also by typed in which selects a single variable from a tuple sample type. The dataset can be converted into an AVS field by 'pressing' the button labeled 'load dataset' on the user panel. The Aurora dataset is loaded into memory using the sub-sampling criteria given by the user and converted to the AVS field format. Figure 1 shows the load dataset button and the sliders used to select sub-ranges or the interval for sub-samples.

Write Module The write module allows for the creation of new Aurora classes and datasets. Each new dataset must become a member of an existing class and so the module allows the navigation of the name space to check for existing datasets and classes. A class can be selected from this name space and used to create a new Aurora dataset or a new class can be created from a class specification which is generated from the AVS field. If an existing class is used when a new dataset is created, the existing class and the dataset must be compatible. If a new class is created, the specification is displayed in an editor to allow new meta-data fields to be added to the class.

Aurora Support for SEAData

One of the goals for developing AVS modules which access the Aurora Dataserver was to support a large project analyzing the oil spill in Alaska's Prince William Sound. Over the next 5 years part of this project, called SEAData, will be collecting, analyzing and distributing a large number of scientific datasets. These data include oceanographic and biologic samples from throughout the Sound collected from buoys, ships, and satellites. We implemented some of the Alaska datasets in Aurora and tested remote access to these datasets through the Aurora server. The rest of this section describes how we used our modules with the Alaska datasets.

The earliest dataset collected by the SEAData project was the bathymetry of Prince William Sound. This is a large dataset which was well suited for testing our modules. The dataset could be selected from the read module by browsing the name space or through an SQL command. Once selected, sliders are displayed which allow a sub-ranges and sub-sampling to be specified. The sub-ranges of latitude and longitude can be bounded by the minimum and maximum of the region and the sub-sampling allows up to every 10th point to be selected.

Figure 1 shows the read module after the the dataset has been loaded. The Aurora name space includes the bathymetry dataset and the class associated with that dataset. The dataset has been selected and the class body for the dataset and a set of sliders for sub-setting the samples are displayed. No sub ranges have been specified but an interval of 8 samples has been given to lower the resolution and eliminate the need for a 'down size' module.

The second major test we performed to support the Alaska project was to evaluate the performance of remote access to the Dataserver. We configured a server at the Prince William Sound Science Center in Alaska and a client at the University of Maryland in College Park. Using a query facility which was specially designed by Xidak to minimize network overhead, we read the bathymetry data which was stored in Alaska from an AVS network running in Maryland. One of the significant advantages of sub-setting the data at the server is the reduced amount of network overhead needed to transfer the data. Our performance results were reasonable but more importantly the scaled well as sub-ranges or sub-sampling was specified.

Conclusions

The scientific data model implemented by Aurora greatly simplified the conversion into the AVS field structure. The Aurora data model incorporated most of the AVS field model with the exception of unstructured cell data which is a key feature of AVS. Aurora does, however, provide a much richer set of sample types. Our hope was to build a new set of AVS modules which were tailored the needs of the Alaska project but the development on the Aurora system has been put on hold. It does not appear that Aurora will be expanded to include the additional sampling geometries and class inheritance that we need to support the SEAData project and so we are currently investigating other alternatives. Our current choice is an extensible database management system which provides the necessary class hierarchy and external file references. It will, however, require some additional effort to implement some of the basic functionality provided by Aurora.

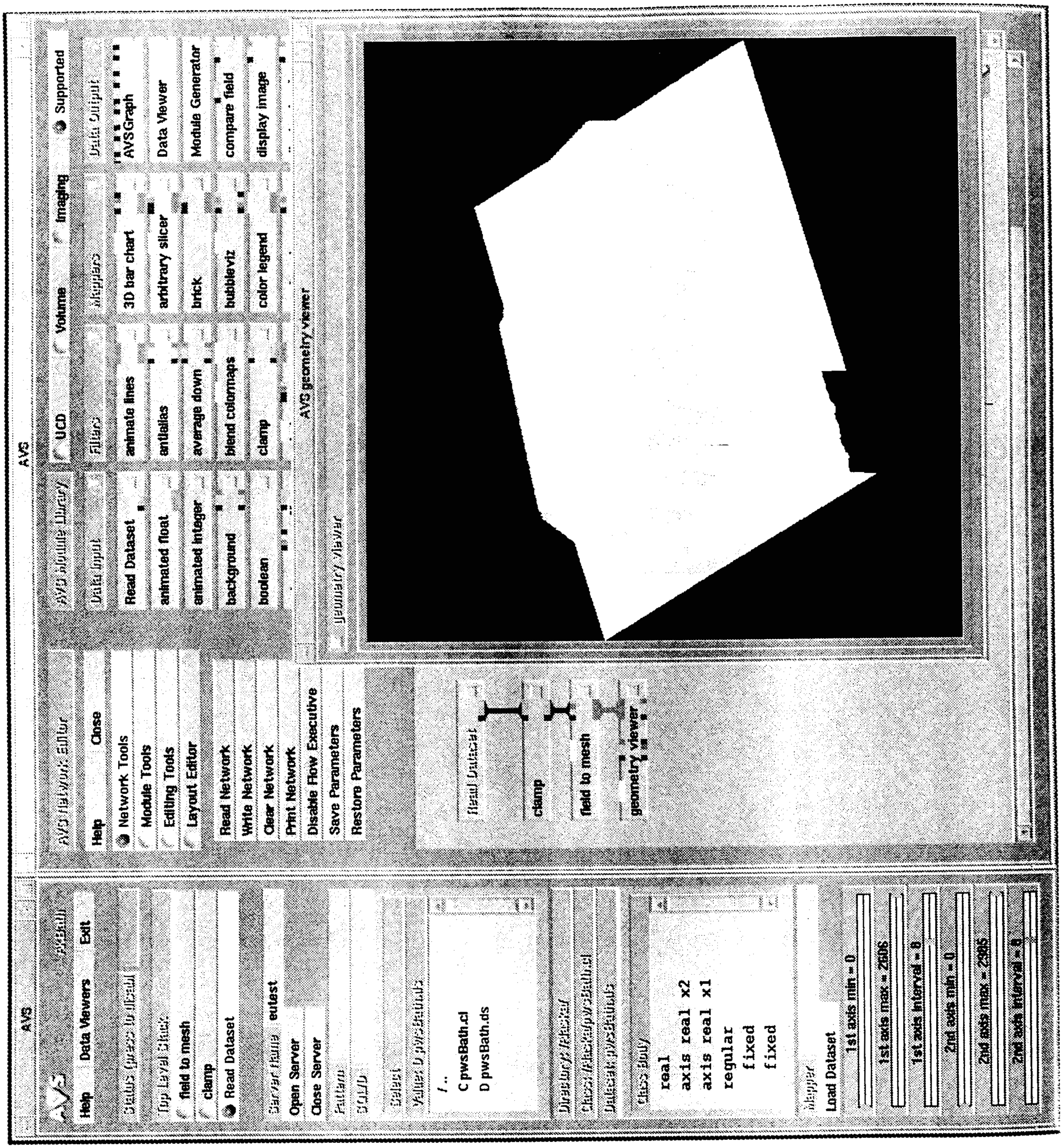


Figure 1: AVS network using a read module to select an Aurora dataset containing bathymetry data from Prince William Sound, Alaska

Appendix 2

Prototyping an Architecture for the Management of Scientific Data

Prototyping An Architecture for the Management of Scientific Data

Charles Falkenberg

Computer Science Department and
The Advanced Visualization Laboratory
University of Maryland
College Park, MD 20742

Introduction

The current state of the art in scientific data management is based on self-describing file formats. Formatting standards like HDF, CDF and FITS, allow scientists a great deal of flexibility in defining and sharing their data. The software needed to read and write data in one of these formats is freely available and the API for defining and accessing data is fairly easy to use. These formats do not, however, provide the functionality available with modern database management systems (DBMS). As the volume of scientific data grows, many scientific applications need the indexing and search features of a DBMS, but modern database management systems do not offer the portability and modularity of data which many researchers have come to depend upon. Since these features are not currently provided with modern DBMS, many sophisticated scientific applications are denied data management functionality which has been available to transaction oriented applications for many years.

To investigate a solution to this problem I built a prototype of a scientific data management architecture which incorporates the advantages of a full function, DBMS, without sacrificing the portability and flexibility of a scientific formatting standard. The prototype uses an extensible database management system to store some of the meta-data associated with an archive of HDF datasets. The DBMS will provide spatial indexing and search facilities for datasets in the archive, and objects which are designed to support specific application. The datasets can also be used independently, without referencing the DBMS.

The prototype is a subset of the data management architecture which is proposed to support a large research project which studying Prince William Sound in Alaska. This project, called SEA-data,is responsible for organizing and disseminating the data collected by 11 groups researching the effects of the *Exxon Valdez* oil spill. In addition, SEAdata is responsible for several scientific visualization applications which will use data from all of the groups to produce visual models of the oceanography and biology of the Sound. The data management architecture proposed for SEAdata includes a large archive of HDF files and a database management system to support the visualization applications. The prototyping activity was tailored to answering the questions which will reduce the risk of developing this data system by providing information to the design and implementation phases.

The Illustra, extensible, DBMS was used in the prototype to maintain the meta-data for the datasets. Extensions will be created, in Illustra, which support the indexing and access to the HDF datasets. These extensions were designed to support one of the visualization applications which the responsibility of SEAdata with the intent that a generalized model can be drawn from the specifics. Part of the goal of the prototyping exercise is to experiment with the Illustra DBMS and address concerns about its viability for the management of SEAdata's scientific information.

The project lasted throughout the spring of 1995 and meet the requirements of a independent study in scientific data management. This document is intended for the faculty members overseeing this project, the researchers at the Advanced Visualization Laboratory who will be responsible for the SEAdata visualization applications, and the SEAdata information system developers who will be implementing the target data management system. The remainder of this report includes a section describing the HDF standard and the role of scientific data formats and a section describing the SEAdata project in more detail. The proposed SEAdata architecture is presented and the final section includes a description of the prototype and future work.

HDF and Scientific Data Formatting Standards

Like other data formatting standards, the Hierarchical Data Format (HDF) standard is a simple data storage system. It is implemented by a set of accessors which are used to read and write multi-dimensional arrays [RDE93, Nat94]. In addition to multi-dimensional arrays, HDF includes 5 other types of data as well. Multiple datasets of any of the 6 types can be written to a single HDF file in a hierarchical fashion. The possible types are summarized below.

- Two type of raster image data, 8 bit and 24bit
- A text data type for storing documents, annotations and text meta-data.
- Color pallets which are used to assign colors to values when visualizing data.
- Multi-dimensional arrays for scientific data.
- Tables for relational data or complex scientific data.

The software which supports this format is developed and distributed by the National Center for Super Computer Application at the University of Illinois. It includes a library of accessors for each standard data type and a set of low level accessors for extending the system to handle a new type. The library calls are accessible from a C or Fortran program and they are available, along with the documentation, free of charge. Recently, the HDF standard was chosen by NASA as the primary format in which the EOS data will be made available.

The flexibility and portability of a standard format like HDF has a great deal of appeal to scientists collecting or using data. These formats are called *self describing* because they provide accessors which reveal the definition of the sample data or the attributes in any file. Attributes can be defined in a simple way, and given a value, which further describe the data. This internal description, along with a facility to translate data across multiple hardware platforms, make these formats *portable* as well.

The research autonomy enjoyed by scientific community demands this level simplicity and portability of data. A single dataset may be useful to a wide variety of scientists for many different purposes. A research project may demand that data be stored with a minimum of meta-data or, alternatively, with complex catalogs of meta-data. Datasets may also be passed between different types of research projects or revisited years after the initial use. For all of these reasons the scientific community has created these self contained, flexible, formats, which allows the user to define the level of data complexity.

These reasons have also steered the scientific community away from commercial DBMS product. Although there is a great deal of database functionality which would be valuable in managing scientific data, the restrictions imposed by DBMS are to great. The transaction overhead and database administration are unnecessary in most research applications. Primarily, however, committing to a single data and meta-data model which may or may not be shared by another research group is not practical for most scientists. In addition, commercial database systems are seen to need a support staff which will not always be affordable or even available in a research environment. Finally, many of the data used by the scientific community has a more complex structure than can be captured by current relational database systems.

As the volume of scientific data grows, however, the need for database functionality will grow as well. Large numbers of datasets, each of which could be several megabytes in size will require sophisticated indexing and query facilities. In addition, the nature of meta-data is growing more complex as more computer interpolation is applied to raw instrument data. Sophisticated scientific applications are also increasing the demand placed on locating and sub-setting of large arrays. Finally, the performance of these high volume, data intensive applications will suffer without intelligent prefetching and low resolution sub-sampling.

These conflicting demands challenge the designers of database management systems. Our goal, here, is to prototype a system architecture which supports both the functionality of database management systems and the autonomy of a scientific data standard. Although some of this design has been implemented in other research projects we hope to use our prototype as a foundation for further research into scientific database management systems.

SEAdata and Scientific Visualization

The SEAdata project is part of large effort to collect information about the oceanography and biology of Prince William Sound, Alaska. This project is part of the remediation effort following the *Exxon Valdez* oil spill, which, in March of 1989, added 11 million gallons of oil to the sound.

The overall project, called the Sound Ecosystem Assessment (SEA) project is funded by the *Exxon Valdez* trustee council. SEA includes 12 separate research groups, all of which are collecting data on the sound. In addition to the oceanography of the Sound, the groups are researching the phytoplankton, zooplankton, salmon, herring and birds, in and around the Sound.

The goals of the SEAdata portion of the project are to collect a portion of the oceanographic data, produce several scientific visualization applications, and create a data repository for the data collected by all of the SEA groups. The oceanographic data is being collected in a variety of ways. A fleet of ships is collecting a set of measurements across the surface of the Sound and vertically through the water column. Buoys, located around the sound are collecting similar parameters from the water column. SEAdata is also responsible for collecting and analyzing AVHRR data transmitted directly from the AVHRR satellite.

SEAdata is also responsible for creating several scientific visualization applications for analyzing the Sound. One of these will present all of the data collected by SEA, within a given time interval, above a bathymetric surface of the Sound. This interim model will be used to get a picture of the all of the data which has been collected. The final model will allow a time period and sub region to be specified, and the datasets which satisfy the query will be displayed in an image, which can be manipulated by the visualization software. Complex regions, based on shorelines, depths or the spatial extents of existing datasets may also be put into the query.

A second visualization application will focus on the fish populations around an acoustic transect of part of the Sound. By using the transect data, along with any other data collected in the region around the transect, this application will couple all of the trophic levels of the sound in a coherent model of the activity of necton and plankton. The queries needed to support this activity will select several datasets based on specific parameters for this model. The datasets which satisfy the query will be converted into a form which can be manipulated by the visualization software. The modules which prompt for the query, select and subset the data, and convert it into a form which is usable by the visualization software will conform to requirements of that visualization system.

The third goal, of the SEAdata project is to provide a data archive of all of the data collected by the SEA project. As a central repository SEAdata will ingest the data from each of the research groups and providing that data to all research groups in a standard format. This will require a common data dictionary and standards for the HDF files. In addition, SEAdata will maintain a database to support those applications which need mechanisms to locate and process the data in the archive. The architecture of the SEAdata management system as well as a prototype implementation of that system is the focus of the rest of this document.

The Data Management Architecture of SEAdata

An overview of the proposed data management architecture for the SEAdata project is shown in figure 1. The four main components of this architecture are the a DBMS which supports a set of visualization and statistical applications; an archive of HDF files which is indexed by the DBMS;

a mosaic server; and processing required to ingest data into the archive and DBMS.

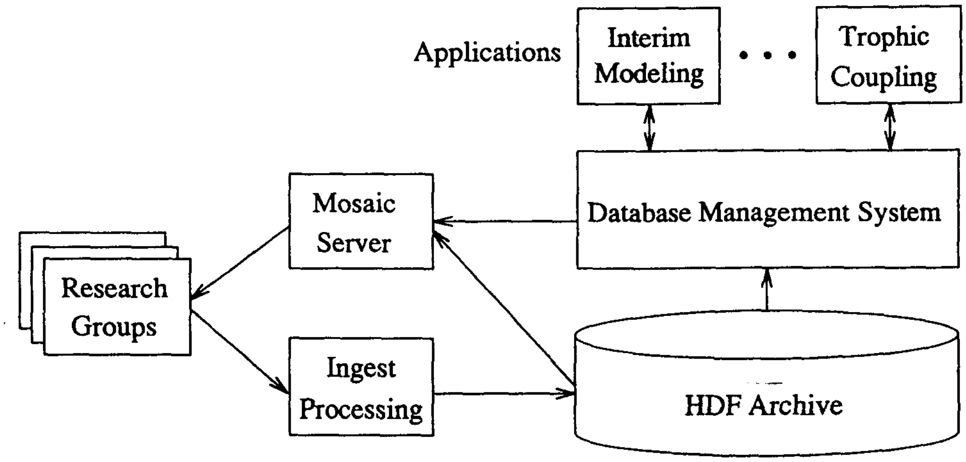


Figure 1: The proposed data management architecture for the SEAdata project

The architecture is designed to maintain an archive of datasets which is essentially independent of the database management system. The DBMS will be used by some of the more sophisticated applications but the HDF archive will be available to any application which does not need the additional functionality provided by the DBMS. The HDF archive could span the disks of several different systems. The DBMS will contain the meta-data associated with each dataset and the pointer needed to locate the dataset in the archive. Some of the meta-data in the DBMS may reference datasets which are not generated by the SEAdata project. The meta-data for NOAA or NASA datasets may be included in the DBMS in order to provide the scientists with a consistent view of relevant data.

HDF Archive The datasets will be stored and retrieved from HDF files stored in the archive. Each dataset will have a unique identifier and a single HDF file may contain multiple datasets. The files can be stored on several different disks, on several different machines, but the DBMS will include a pointer to the disk and directory in which the file is located. If the HDF file is moved, the database can be updated by referencing the unique dataset identifiers in file and in the database. Shadow files might also be used to notify the database that an HDF file has moved.

The system may need to index datasets which are outside of the HDF archive. These dataset could be located at NOAA or NASA or at another research location. The meta-data would be stored in the database and files could be down-loaded when they were requested or copied into the HDF archive and treated like an internally generated file.

Database Management System The database system will provide general functionality, which is available for all applications, as well as functionality which is application specific. Figure 2 shows the two levels of the database management system. The general functionality includes a name service and a variety of indexes over the meta-data. In addition, some sample data may be replicated inside the DBMS. Specific functionality in the form of query objects will also become part of the high level DBMS interface.

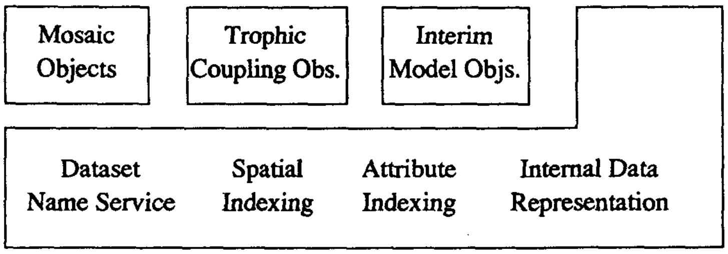


Figure 2: The database components for general and application specific functionality.

The name service will provide applications with simple query facilities to locate files in the HDF archive. This service will also tie a unique file identifier with a disk and directory and file name in the archive. The spatial index will use the meta-data from each dataset to construct a bounding box for that data set. These 2d boxes will be indexed by the database to support spatial queries. Other indexes of the meta-data will be available, also, to speed up queries on non-spatial attributes.

Some applications will need objects which have be created to support that application alone. As an example, the trophic coupling application will tie together several different data sets which are located around the acoustic transect data. The relationship of the data sets to a given transect may be stored in a query object which will accept some parameters and return the list of data sets and the clipping information necessary to execute the application. These query objects would be specific to an application but the would be part of the general interface to the DBMS.

The Illustra database management system is one of the DBMS products proposed for this component of the SEAdata architecture. Illustra is an extensible DBMS which will allow new data types to be defined and optimized by the Illustra kernel [Ill94]. To create a new data type in Illustra, the complete set of functions which are needed to manipulate that type, are created and stored in the database. Illustra can then treat these user defined types as an integral part of the database, and therefore, queries which reference these types can be optimized effectively.

Illustra also provides a several *datablades* which are sets of types for a given application domain. The 2 and 3 dimension datablades provide spatial applications with types for points, lines, polygons, an other shapes in 2 or 3 dimensions. Other datablades provide the functionality for time series analysis or image analysis. These datablades extend the functionality of Illustra, for a given domain of applications, by providing the data types which are standard in that domain.

SEAdata is considering Illustra as the extensible DBMS from which to build the database component. The intent is to create new types in Illustra which represent the datasets in the HDF archive. These would be simple types which are made up of the meta-data of the datasets. The attributes could indexed by standard b-trees and with the 3 dimensional datablade. The query objects which are needed to support some of the applications also be new types in Illustra.

Data Ingestion The ingestion process in figure 3 will insert a dataset into an HDF file in the archive and update the database with the necessary meta-data. The input to this process will be the ASCII or HDF files which have been generated by the individual research groups. The format of the data from each group will be unique and therefore require processing which is specific to that groups files. If a group chooses to use HDF as the format for the group archive, the ingestion process may need to augmented these HDF files with additional attributes needed for the DBMS.

The update of the database may occur when the datasets are added to the archive or by a separate process, as shown here, which only references the HDF files. In either case, when the process is complete, the database management system will be able to index that dataset and locate it in the HDF archive or across the network.

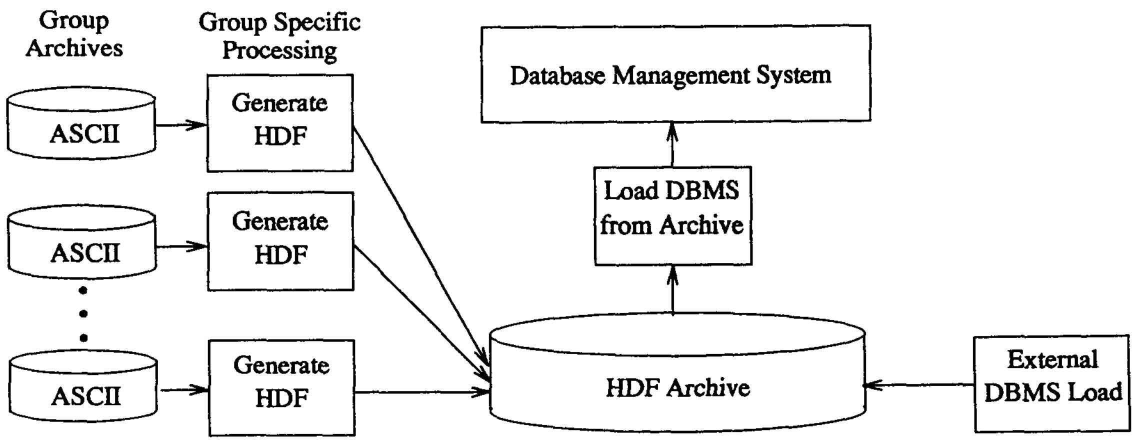


Figure 3: The ingest processing component of the data management architecture

Mosaic Server The last part of this architecture is the mosaic server. This is the interface used by the individual research groups, as well as external researchers, to access the HDF archive. The mosaic interface would display a 3d map of the sound and allow regions and specific datasets to be selected. The database would return a list of HDF files which satisfy the query. The HDF files in which the datasets are located could be down-loaded through the standard mosaic interface and operated on locally.

A group could store its own data at the SEAdata archive and down-load those files which were needed for a given research effort. In addition, groups will have access to any of the datasets which have been generated by another group. The ability to process these standard datasets in HDF format will make a wide variety of data accessible to each group.

This preliminary architecture meets the initial set of requirements outlined by the SEAdata project. Several aspects of this architecture, however, are untried or require new technology and therefore pose a certain risk to the project as a whole. The goal of prototyping is to identify those risks and build a system which is tailored to reducing risk by providing information to design or implementation phase of the target system.

Prototyping the Data Management Architecture for SEAdata

The goal of prototyping the data management architecture is to provide constructive proof of the design concept, expose the potential problems with the system, and experiment with solutions to those problems. The initial prototype meets the first two of these. It provides a demonstration of the architecture, an experimental interface to that architecture, and it exposes some of the problems which will be encountered with the design of the target system. The prototype can now be used in a second phase, to experiment with potential solutions to problems which have been revealed by phase 1. This section includes a description of the implementation of the prototype, a presentation of the benefits of this prototyping effort, and a summary of open issues and future prototyping goals.

Implementation of the Prototype The prototype is a vertical implementation of the proposed data management architecture for SEAdata. This includes all three phases of data management implemented for a single kind of dataset. The first phase is the ingestion of raw, ASCII data, into HDF format; the second phase is the loading of the database objects from the HDF files; and the final phase is an application which uses the database and the HDF files. The datasets selected contain LTD data which include temperature, salinity and water pressure measurements, sampled at every few meters in a column of water. About 300 such columns were sampled in the 1994 summer season. The prototyping effort included ingesting these 300 datasets and displaying them with a visualization application. The rest of this section describes these phases of the implementation and the data structures and objects which were used.

After a discussion with the PI's of SEAdata, a prototype data structure was designed to hold the meta-data and sample data for all of the SEAdata datasets. This data structure is a hierarchy of inherited attributes which is shown in figure 4. This is the hierarchy of meta-data within the datasets maintained by SEAdata. As an example, datasets collected from ships have particular meta-data attributes but they share geographic attributes with all other SEAdata datasets. A data structure which could describe this hierarchy was designed as used as a model for the meta-data in the HDF files and in the Illustra objects. This data structure and the functions which operate on it make up a *dataset object* within SEAdata. All of the details of this dataset object were not worked out but an initial design was used in the prototype.

The ingestion of the CTD data required the adaptation of a program which read the ASCII files containing header data and sample data. Dataset objects were created using the ASCII input and then passed to a routine which created the HDF files. Some of the HDF functionality is encapsulated in these routines which can be reused when subsequent ingestion programs are written. The ingestion program itself, therefore, contains only the functionality which is specific to the format of the ASCII input files.

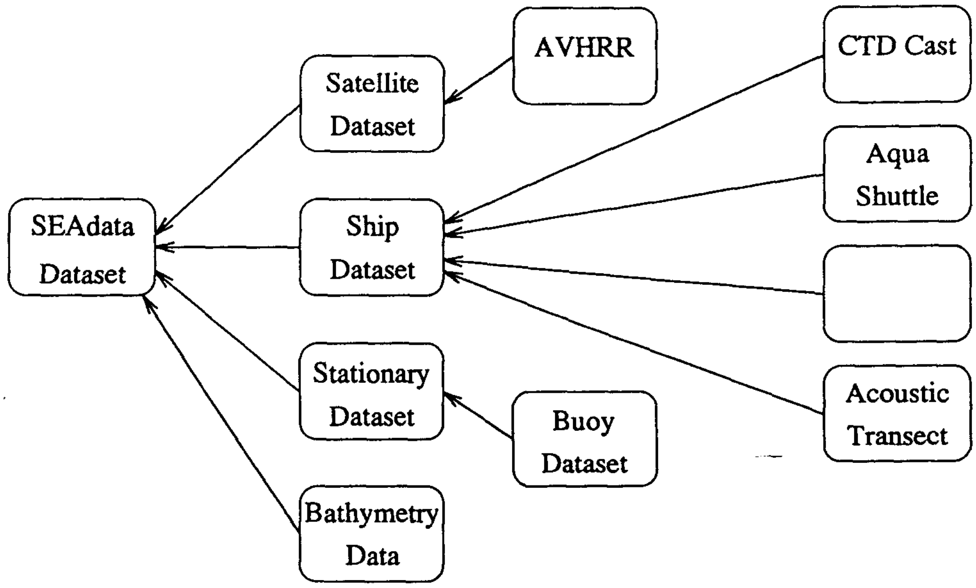


Figure 4: The inheritance hierarchy of the dataset objects in SEAdata

The meta-data for the CTD datasets was read from header files and stored in attributes within the HDF files. These attributed mirror the meta-data members of the SEAdata dataset objects. Four sample arrays were stored in each HDF file. These include the depth, salinity, temperature, and pressure, of each sample. Although depth is a logical coordinate array it would need to be replicated for each of the other datasets and so it was stored separately.

Loading the Illustra database objects was done entirely from the data in the HDF files. This again allows reuse of these routines when other HDF files are used to load the database or when data is loaded from outside of SEAdata. The objects in Illustra mirrored the objects and inheritance of the SEAdata dataset objects. SEAdata objects were created with the meta-data from the HDF files and then passed to a routine which created the Illustra objects. This creation consists of the construction and execution of an Illustra insert statement in SQL. As with the HDF phase, this encapsulates the Illustra functionality within a few routines.

The Illustra objects include a two dimensional bounding box which is constructed from the minimum and maximum of the latitude and longitude of all of the sample points. This additional data member is not in the SEAdata dataset object. The box might be large for some datasets which are collected as a ship moves across the Sound, but for the CTD samples, which all have the same latitude and longitude, it is only a point on the surface. The bounding box data member is a data type from the Illustra 2d datablade. This type is described by the x and y coordinates of two corner points.

An application was created which exercised the access to the Illustra database objects and the HDF files containing the sample points. This application, shown in figure 5, includes a graphic interface for specifying the a database query and a button which will retrieve and display the selected sample points. This scientific visualization application is written within AVS and provides a working example of how the data management architecture will be used.

The selection of datasets is done with a SQL query of the database in Illustra. The selection criteria include non-spatial attributes like ship name and sample date, as well as the spatial attributes of latitude and longitude. A list of the datasets which meet the selection criteria are shown in a results window which displays some of the meta-data attributes. If the list of dataset meets the users needs a separate 'Load Datasets' button is provided to load the samples into the application and display them.

The application uses some of the previously written routines for loading SEAdata dataset objects from Illustra and HDF. The results of the Illustra query are used to create an internal list of dataset objects which contain no samples. When the button is used, this list of objects is passed to a routine which loads the sample arrays in these objects from the HDF files. The complete list is then passed to a routine which creates AVS fields needed to display the samples. Several other AVS modules manipulate the data including one which projects the sample coordinates from latitude and longitude onto a flat surface.

The utilization of the SEAdata dataset objects is shown in figure 6. Routines were written to convert raw data, HDF data and Illustra objects into a SEAdata dataset object. Routines were also written which create HDF data, Illustra objects and application structures from the SEAdata dataset objects. This provides a high degree of modularity and reuse as well as allowing for the change from HDF or Illustra to another storage manager or DBMS, respectively.

Results of Prototyping The implementation of a prototype provided insight into several aspects of the target system. These include the capabilities the designers can expect from Illustra and HDF, preliminary documentation of the database schema, and a user interface which demonstrates the functionality to potential users. The other major result of the initial prototyping effort is the identification of potential problems with the architecture and the clarification of the experiments needed to reduce the risk which these problems present to the design of the target system.

The capabilities of Illustra which were used in the initial prototype of the SEAdata architecture include type inheritance and spatial access methods. The inheritance hierarchy of the datasets in SEAdata is readily described by the Illustra type system. Although the description of the objects must be replicated in the programming language, the Illustra specification is comprehensive and easier. Several tedious routines had to be written to extract convert the results of the Illustra query into the internal objects but once done these routines can be used any time Illustra is used, within SEAdata or not.

The 2 and 3 dimensional datablades were investigated for use with this prototype. The selection of dataset will nearly always include sample depths and date/time in addition to latitude and longitude. The 2d datablade was used for the prototype, however, because no 4d datablade is available and the 3d datablade does not include the needed functionality for selecting objects in space. The 2d datablade provides several valuable functions for manipulating and selecting spatial objects. This datablade includes an R-tree index which can be built on boxes and polygons and this feature was used in the prototype. The other parameters can be selected using standard relational selection operations.

The capabilities of HDF are quite limited but the interface is simple and I/O performs well. No inheritance hierarchy can be described in HDF but the meta-data attributes can be represented easily and can mirror the structure of an inherited object. The current HDF API does not provide any type of search facility. Subsections of the arrays can be read but only index values can be used to specify those subsections. No facility is provided to read samples within a range of depth, for instance, or for a range of given values. The API does provide a functions from which a description the attributes and arrays within the file can be displayed. This is vast improvement over the raw, ASCII, representation.

Although the clear understanding of the capabilities of Illustra and HDF is an important result of the prototyping exercise, the refinement of the database schema definition is perhaps the most valuable. The preliminary definition of the meta-data needed for all of the SEAdata datasets includes several different kinds of data. Some of the meta-data summarizes the sample points by giving a minimum and maximum values of the sample arrays. Other meta-data describes the origin of the sample data, providing an audit trail back to the raw data. Finally a third type of meta-data is created to support the administration of the data management system. These data include dataset identification numbers and the bounding box for the R-tree index.

Using meta-data to summarize the sample points will greatly enhance the potential selectivity of the database. As queries get more sophisticated and select dataset based on the values of the sample points, summary data can be used to trim the search space significantly and reduce the number of samples which must be examined. Although summary data is a replication of some of the sample values, these sample data do not change and therefore data consistency is not a concern. One of the benefits of building this prototype was the increased awareness of the value of this technique. Currently we are only using minimum and maximum values for sample points but other methods are being considered too.

Another important benefit of the prototyping activity is the development of a user interface which can demonstrate the functionality of the data management system to the potential users. This demonstration not only provides a framework from which the details of the architecture can be discussed, it allows scientists to consider how they will be utilizing the system for their own applications and challenges them to find ways in which the both the user interface and the basic functionality can be improved. This is a common benefit of prototyping and for good reason.

Finally, the creation of the prototype and a demonstration application exposes several open issues. An initial prototyping step is often followed by several experiments which can address the major risk factors in the target system. This second phase will address the following open issues.

Open Issues and Future Goals Several issues surfaced once the prototype was in place which need to be addressed before the prototyping activity is complete. Perhaps the most important of these is the selection and retrieval of sample points. Other issue include performance evaluation and more sophisticated spatial queries, and the integration of other kinds of datasets.

The basic data management architecture provides extensive selection of dataset based on the meta-data stored in the database. Once the datasets have been identified, however, the samples must be retrieved from the HDF files. This means that the architecture does not support any selection on the values of the sample points. One of the goals was to consider the possibility that sample points could be stored in the database, at least temporarily, giving the user the ability to select on sample values. However, because Illustra does not cache objects at the client, the sample points will need to be down-loaded for each query making solution unworkable as well.

Therefore, additional experiments must be done to see how to provide sample selection on local data. Illustra does provide the ability to run functions at the client, but it is not clear that the basic indexing facility can be provided on local objects. This and other alternative need to be tried and results made available to the designers of the target system.

An evaluation of the performance of Illustra and the 2d datablade is also needed in order to provide confidence. This study needs to quantify the performance of the R-tree index for a large number of datasets as well as quantify the performance of the remote access to the server. The current prototype operates well but some idea of the degradation due to this remote access should be understood before Illustra can be fully recommended.

Part of the performance evaluation should include more sophisticated spatial selection criteria. SEAdata needs to select datasets within an polygon surrounding an irregular region of interest. Additional datasets are being added to the prototype and this evaluation could give insight into the performance of Illustra's spatial access methods.

Finally, as a demonstration tool the prototype does not illustrate the capabilities of the DBMS for integrating different kinds of datasets. The hierarchical structure of the object in Illustra allow the multiplicity of datasets to be searched together and return in a single list of results. This capability is not well demonstrated in the current application which visualizes only one kind of dataset. The addition of another kind of dataset contain biologic parameters will push the application and the interface toward are more realistic view of the final result

Results of the Project As an activity, the constructive prototyping effort replaced an initial set of concerns with a deeper, more informed, set of concerns. No prototype can hope to flush out all potential problems but as a research tool the prototyping activity can provide the design phase with a much clearer idea of the viability of the basic concepts as well as the potential pitfalls.

References

[FJP90a] J.C. French, A.K. Jones, and J.L. Pfaltz. "Scientific database management: Final report". Technical Report 90-21, Department of Computer Science, Univeristy of Virginia, Charlottesville, VA, August 1990.

[FJP90b] J.C. French, A.K. Jones, and J.L. Pfaltz. "Summary of the NSF workshop on scientific database management". *SIGMOD Record*, 19(4), December 1990.

[Fre91] James C. French. "Support for scientific database management". In Zibigniew Michalewicz, editor, *Statistical and Scientific Databases*, chapter 4. Ellis Horwood Limited, 1991.

[Ill94] Illustra Technology. *The Illustra^{tm} Users Manual*, 1994.

[Kim91] Won Kim. "Object-Oriented databases for statistical and scientific applications". In Zibigniew Michalewicz, editor, *Statistical and Scientific Databases*, chapter 15. Ellis Horwood Limited, 1991.

[Nat93] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. *Getting Started with HDF*, May 1993.

[Nat94] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. *HDF Reference Manual*, February 1994.

[RDE93] Russ Rew, Glenn Davis, and Seve Emmerson. *NetCDF User's Guide*. Unidata Program Center, University Corporation for Atmospheric Research, April 1993.

[Sam90] Hanan Samet. *The Design and Analysis of Spatial Data Structures*. Addison Wesley, 1990.

[Sho91] Arie Shoshani. "Properties of statistical and scientific databases". In Zibigniew Michalewicz, editor, *Statistical and Scientific Databases*, chapter 2. Ellis Horwood Limited, 1991.

[SW85] A. Shoshani and H.K.T. Wong. "Statistical and scientific database issues". *IEEE Transactions on Software Engineering*, SD-11(10):1040–1047, October 1985.

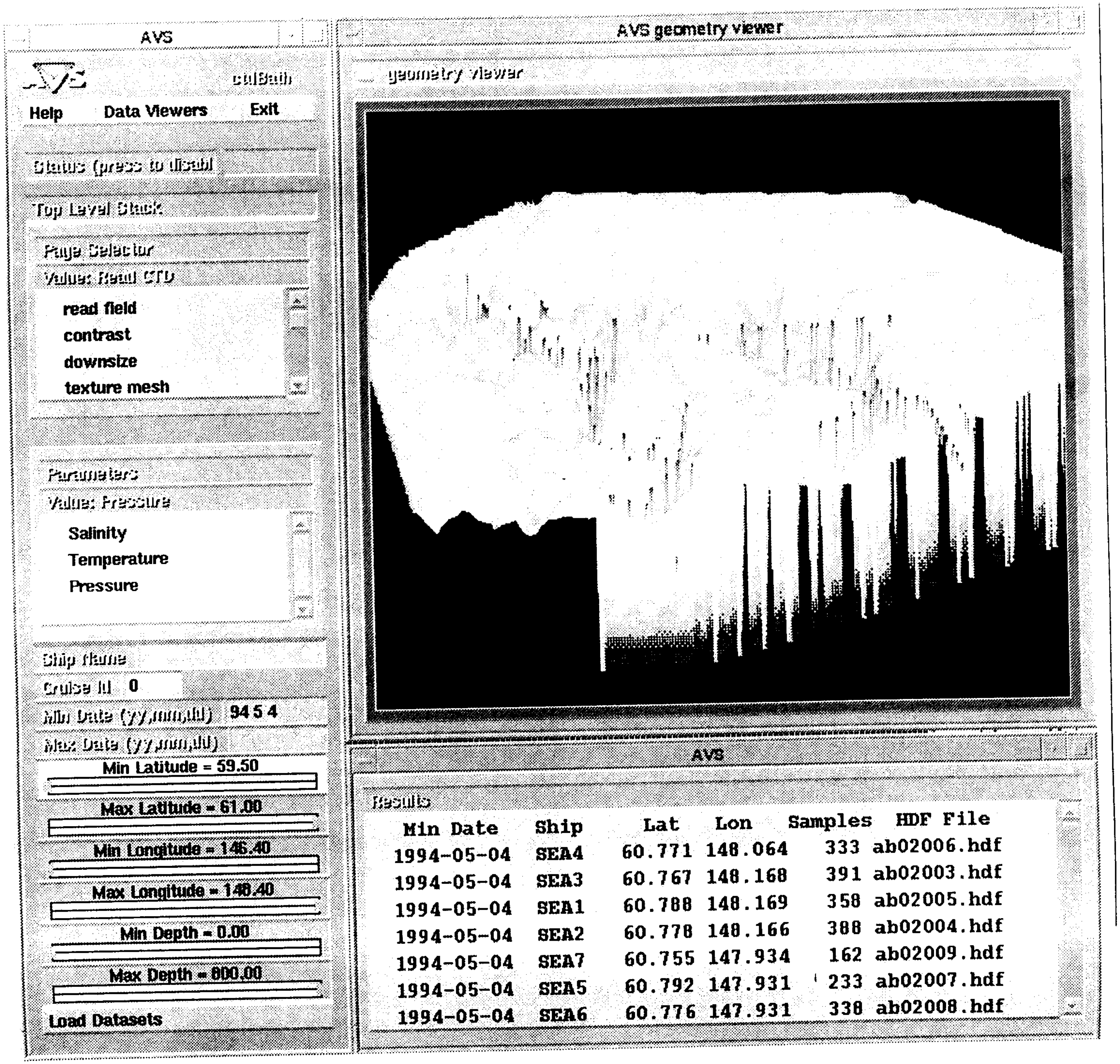


Figure 5: The application which allows visual display of the datasets within the SEAdata data management system.

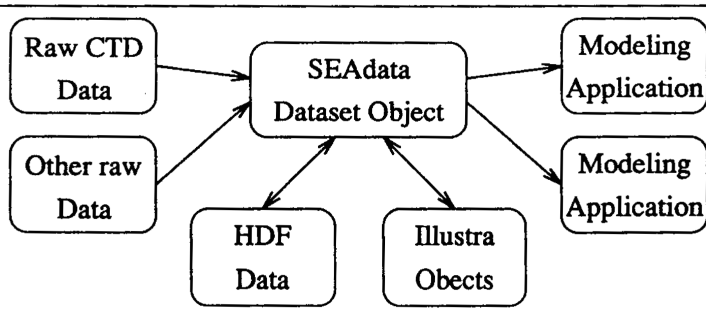


Figure 6: The data flow across the interface to the SEAdata dataset object and the other objects and structures in the system.

Appendix 3

Case Study: Using Spatial Access Methods to Support the Visualization of Environmental Data

Case Study: Using Spatial Access Methods to Support the Visualization of Environmental Data

Charles Falkenberg

Department of Computer Science
Advanced Visualization Laboratory
University of Maryland
College Park, MD 20742
csfalk@cs.umd.edu

Ravi Kulkarni

Advanced Visualization Laboratory
University of Maryland
College Park, MD 20742
ravi@avl.umd.edu

Abstract

As part of a large effort evaluating the effect of the Exxon Valdez oil spill, we are using the spatial selection features of an object-relational database management system to support the visualization of the ecological data. The effort, called the Sound Ecosystem Assessment project (SEA), is collecting and analyzing oceanographic and biological data from Prince William Sound in Alaska. To support visualization of the SEA data we are building a data management system which includes a spatial index over a bounding polygon for all of the datasets which are collected. In addition to other selection criteria the prototype provides several methods for selecting data within an arbitrary region. This case study presents the requirements and the implementation for the application prototype which combines visualization and database technology. The spatial indexing features of the Illustra^{tm} object-relational database management system are linked with the visualization capabilities of AVS to create an interactive environment for analysis of SEA data.

1 Introduction

As the quantity and disparity of scientific data increases so do the problems of managing these data. Part of the management goal is to provide consistent methods of access to scientific data which are collected with different techniques for different purposes. Scientific visualization and scientific database technology have both been used to address the problem of managing large numbers of datasets [6, 3]. Meta-databases have been used to index scientific datasets and the problems of combining database technology and visualization techniques are being explored as well [1, 5]. This case study describes the use of the spatial selection capabilities of an object-relational database management system to support queries from a visualization tool. The prototype displays and manipulates scientific data which are being used to analyze biological activity after the Exxon *Valdez* Oil spill.

In 1994, a 5-year effort called the Sound Ecosystem Assessment was begun to analyze the biological impact of the Exxon *Valdez* oil spill in Prince William Sound, Alaska (PWS). As a response to the low production of pink salmon and Pacific herring in the last few years, the SEA project is collecting a wide range of oceanographic and biological data. These data will be used to analyze the ecosystem of the Sound and provide the foundation for simulating sound-wide processes. The project is funded by the Exxon Valdez Oil Spill Trustee Council and includes 13 separate research groups, each of which is specializing in some aspect of the ecosystem [2]. These aspects include the oceanography, plankton, fish, and birds in and around the Sound. Each group is collecting data relevant to its own focus and one group is charged with archiving these data and providing an integrated view of that archive to all groups in SEA. As part of this integration effort we are implementing the data management architecture as well as several visualization tools which take advantage of the spatial capabilities of a database management system.

Our approach to integrating these data includes the use of both visualization and database management techniques. We have built a prototype of our visualization tools which utilizes the Illustra^{tm} [4] database management system. The spatial access methods provided by Illustra become a valuable element in the iterative set of operations for refining the selection criteria within a query and manipulating the results. This case study describes the use of irregular polygons to interactively define regions of interest for selecting datasets collected by SEA. We describe some of the datasets, and then present the motivation for the visualization tools. This is followed by a discussion of how Illustra is being used to implement spatial searches within our visualization tools.

2 The Challenge of SEA Data

SEA is collecting 12 main datasets, each of which has from 3 to 10 measured variables. These datasets are of different sampling geometries and ranks, different resolution in both time and space, and several different levels of interpretation. These data are collected by scientists in different disciplines with occasionally divergent requirements. Each discipline, however, will need access to most of data, in a format which is suitable to the specific needs of the analysis. Although the total volume of data is not unmanageable, the number and disparity of the datasets presents a data management challenge.

The oceanographic data include remotely-sensed sea surface temperature, Doppler profiles of ocean currents, and vertically and horizontally sampled temperature and salinity. The biological data record phytoplankton and ocean nutrients, zooplankton, herring spawn, salmon fry, adult salmon, herring and walleye pollock, and sea birds. The number, biomass, or feeding behavior are documented for many of the species. These oceanographic and biological data will be augmented with weather data, hydrologic data and tide information which is available from sources outside of SEA. Finally, these data will be analyzed in the context of the bathymetry (water depth) of Prince William Sound.

The level of interpretation of the data will also vary depending on the source instrument or modeling application. Simple values like temperature can be recorded with little or no interpretation. However, acoustic soundings of fish population will require several steps of manual interpretation to produce species counts and biomass. At the highest level of interpretation, simulation models using all relevant data will generate large volumes of 3- and 4-dimensional (space and time) output.

The sampling strategies result in multiple resolutions and geometries. The resolution, or size, of the sample cell varies across each sensing instrument, bottle or net used for collection. The smallest cells are less than a cubic meter and the largest can span hundreds of meters. The sampling geometries may be linear, or planer in 3-dimensional space, or include 3- and 4-dimensional output generated by the models. The geometries may be regular, irregular or scattered.

These data will be used to test several hypotheses about the processes which regulate the population of pink salmon and Pacific herring in PWS. The processes include loss of herring eggs, salmon and herring feeding behavior and predators, the magnitude and direction of the current in the Sound and the overwinter behavior of the herring. Testing these hypothesis requires correlating many of the variables across both time and space. To facilitate this, the collection of the variables is coordinated between all groups to provide simultaneous measurements. Standard locations in the Sound are revisited and the variables are re-measured to allow time series analysis as well.

As an example, understanding the movement and growth of the juvenile fish requires an understanding physical variables such as current, water temperature, the availability of food, and presence of predators. The interaction between these may depend upon the depth, proximity to the shore, and the season. Each scientist will need to correlate a different group of these variables, within given regions, across time.

3 Visualization Tools

In order to provide a visual integration of the disparate datasets within SEA we are creating several tools for selecting and displaying SEA data. The tools use AVS as a visualization engine and allow the selection of data from Illustra and from files which use the HDF standard. Our prototype of these tools includes a set of iterative operations for displaying data in a spatial context as well as in a graph form. The requirements of the interactive visualization tools are driven by the following goals:

- Provide an integrated view of the datasets in SEA in order to support multidisciplinary data analysis and planning for future data collection.
- Allow dataset browsing with various search criteria including space, time, and meta-data attributes.
- Support the selection of individual datasets and alternative displays such as line plots, images, or tables

The data flow architecture of AVS allows iterative cycles of user interaction to be implemented with an upstream flow of data to selected modules. Selecting a geometric object can result in an alternate display of the data or the use of that object to refine the definition of a query. The network can operate as a browse tool or it can assist in more detailed analysis.

For an integrated view we created special subnetworks for each type of dataset with default visuals and a common projection. Alternate visuals were added to allow for detailed views of each dataset. The integrated view provides a perspective on the sampling coverage of each type of dataset in the sound. It also provides the basis for our ability to browse and investigate data in finer detail.

The tools provide several ways to define the regions of interest in space and time by specifying queries visually. A user defined region of interest can be drawn on the geometry viewer. The resulting polygon is used in the selection criteria when the datasets are retrieved. A second method uses a bounding polygon around a selected dataset, or group of datasets, to define a proximity query. These polygons can be entered by the user or generated automatically using the bounds of the selected data. A third approach generates a polygon based on the values of other datasets. Isolines of temperature, salinity, or depth could be used to create a set of polygons used in the selection process. These three types of spatial queries are described more fully later and shown in Figures 2 through 4. Finally, polygons may also be predefined around study areas in which research is being concentrated.

In order to support these types of queries, as well as other types of non-spatial queries, we are implementing a data management architecture which includes an archive of HDF files and a database of meta-data. The database uses Illustra to store meta-data along with other spatial objects including the predefined polygons around study areas. To allow transportability the variables are stored in HDF files. These variables will also be cached in the database in order to answer content based queries.

We chose the Illustra database management system because it includes features of both object-oriented and relational database management systems (DBMSs). Like a relational DBMS the data is organized into tables, views and indexes. Illustra is a commercial database management system which includes a database kernel with a standard set of data types and functions. Illustra is also extensible, allowing the user to create new data types and the functions which operate on those types. Several additional packages, called 'datablades', are available for Illustra, which include a set of types and functions which are tailored to a particular type of application.

Illustra's two dimensional spatial datablade provides a suite of spatial functions and types, including a polygon type. In addition, this datablade provides an rtree index over a subset of the spatial data types. Our meta-database includes a table of datasets with a record for each dataset in the archive. One of the attributes in this table is a 2D polygon which describes the bounds of each dataset in latitude and longitude. An rtree is used to index the set of polygons in the table. A spatial query supplies another polygon which circumscribes the region of interest and the result of the query is a list of datasets from the table which fall within that polygon.

4 Demonstration Prototype

Figures 1 through 4 are images from our prototype application. Figure 1 displays a subset of the SEA datasets over the bathymetry of Prince William Sound. The depth of the ocean floor is exaggerated and all of the topography above sea level has been flattened. The view is looking North East from the gulf of Alaska. The port of Valdez is in the top right hand corner.

Several of the datasets have been lettered in Figure 1. The wavy lines behind the letter A display oceanographic data which is collected with an aquashuttle which moves up and down in the water column as it is towed behind a ship. Acoustic transects of the fish populations appear as blue an green curtain near the letter B. These data represent the echos of objects in the water as the sonar instrument is moved along the surface. The vertical columns display the salinity from the surface to depth. These are called CTD casts and include temperature and oxygen measurements. Finally, the line near the letter D, is the path of an aerial survey (video) of birds and other visible effects like rip tides. The other three figures show different methods of using a polygon to select SEA datasets.

Figure 2 shows a polygon which has been drawn by a user around a local area of interest. This is done to reduce the retrieval time and get a general view of the data collected for a region. Figure 3 displays the selection criteria for a more detailed analysis. Polygons, defined in close proximity to 3 acoustic transects, are used to select oceanographic data which may be correlated with the acoustic data. These polygons could be automatically generated by selecting the transect from the geometry viewer.

Figure 4 shows CTD casts within a polygon which was created from a depth contour. Such an isoline could connect points of equal temperature or salinity rather than points of equal depth. This figure also shows a graph of temperature vs. salinity for a single CTD cast which has been inverted in the display. This type of alternate display provides additional analytic capability from within the visualization application.

5 Conclusion

By combining advanced visualization techniques and database management, our prototype provides a framework for integrating a disparate set of data. The user can visualize the data and specify spatial queries on the display, which then refine the query selection. Our work with the prototype is being followed by a full scale implementation of the data management architecture and several other visualization tools. This will test the scalability of the design with larger volumes of data.

Acknowledgments

This work is supported by the SEA project and the Prince William Sound Science Center. Dr. G. Thomas provided the acoustic data, Dr. V. Patrick and E. Jin provided the aquashuttle data, Dr. D. Salmon provided the CTD data, and Dr. D.Sheel provided the aerial survey data. We thank them for all their help. Dr. M. Franklin from the University of Maryland provided insight into the scientific database issues and Illustra^{tm} was made available from through the "Engines for Innovation" grant program.

References

[1] Manish Arya et al. "Database issues for data visualization: System integration issues". In *Lecture Notes on Computer Science*. Springer Verlag, October 1994.

[2] R. Ted Cooney et al. Sound Ecosystem Assesment (SEA) - An Integrated Science Plan for the Restoration of Injured Species in Prince William Sound. Submitted to the Exxon Valdes Oil Spill Trustee Council, April 1994.

[3] J.C. French, A.K. Jones, and J.L. Pfaltz. "Summary of the NSF workshop on scientific database management". *SIGMOD Record*, 19(4), December 1990.

[4] Illustra Technology. *The Illustra^{tm} Users Manual*, 1994.

[5] Peter Kochevar, Zahid Ahmed, Colin Shade, and Jonathan Sharp. "A simple visualization management system: Bridging the gap between visualization and data management". In Gregory Nielson and Dan Bergeron, editors, *Proceedings of Visualization 93*, San Jose, California, October 1993. IEEE.

[6] L.A. Trenish. "Unifying principles of data management for scientific visualization". In R. Earnshaw and D. Watson, editors, *Proceedings of the British Computer Society conference on Scientific Visualization and Animation*, pages 141–169. Academic Press, December 1993.

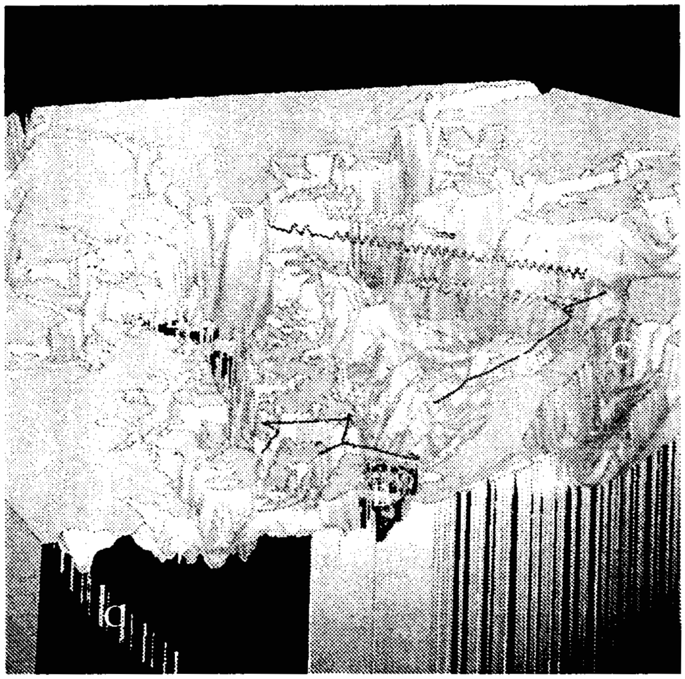


Figure 1: Bathymetry overview of Prince William Sound displaying 4 types of datasets: A) aquashuttle tows; B) acoustic transects; C) CTD casts; D) aerial survey.

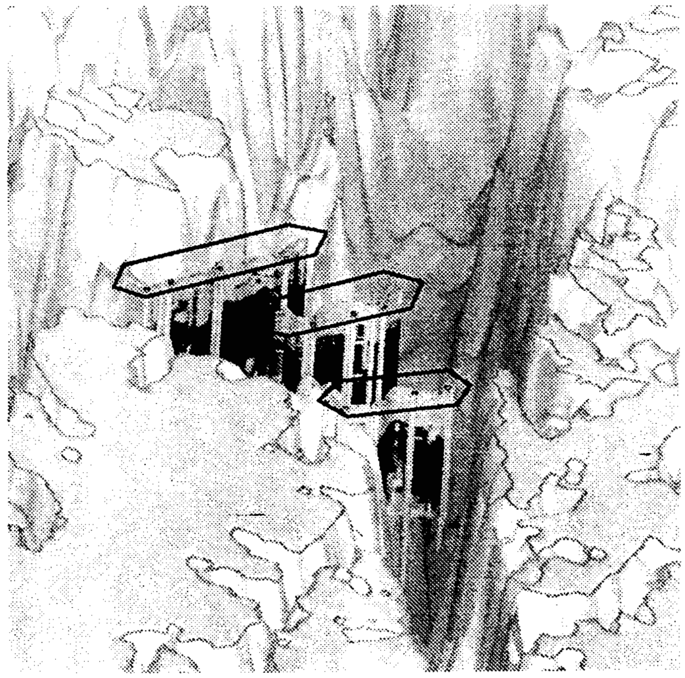


Figure 3: A set of three polygons are used to query within a close proximity of selected transects. The polygons can be drawn by the user or generated automatically from the bounds of the transects.

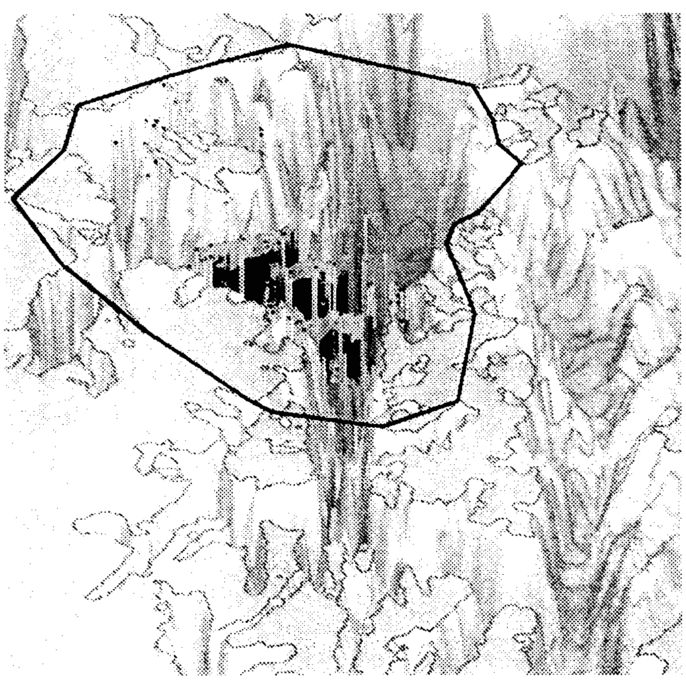


Figure 2: Region of interest specified by the drawing a polygon on the AVS geometry viewer. The polygon is used by Illustra^{TM} to select datasets within the region.

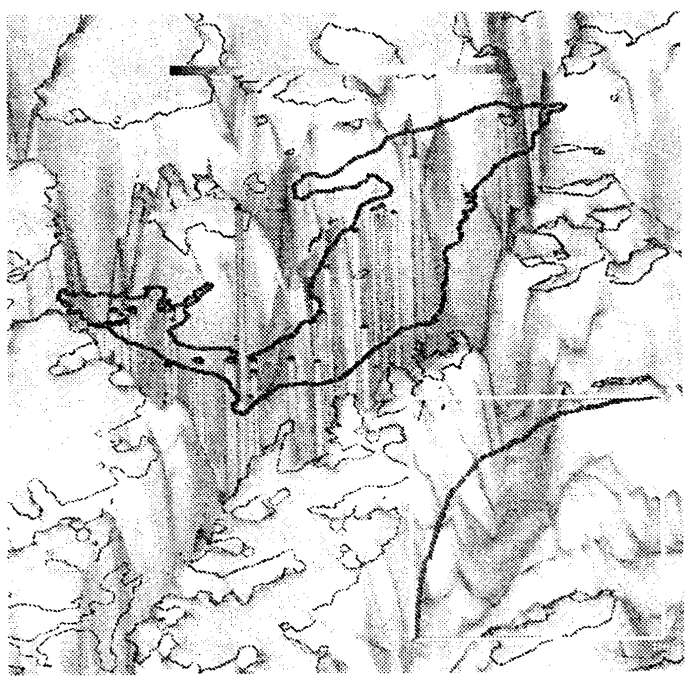


Figure 4: A region is specified using a depth contour of the bathymetry. A single CTD cast has been selected and a graph of the data is displayed.

Appendix 4

SEA Data Dictionaries
CTD casts
Echo Acoustics for Pollock

1.1 CTD casts

General Description

The CTD instruments measure temperature and salinity at a given interval of pressure. Using the salinity value, pressures are converted to depths. CTD cast are made at regular stations during the broad area survey cruise and at index sites. The raw data is averaged and then bined at 1 decibar intervals.

Variables

Temperature	Water temperature in °C.
Pressure	The water pressure in decibars of the measurements.
Oxygen	Some CTD casts include oxygen content of the water in ml/Liter.
Salinity	A unitless salinity derived from conductivity.

Size and Coordinates of Sample Cell

The data are collected for each decibar of pressure from the surface to the bottom. The length of a decibar is very close to a meter for the entire water column The depth array is derived using the pressure a conversion equations. There is only one latitude, longitude and date/time for the entire column of data. A weighted average is applied to the raw data and it is stored for each cell.

The coordinates of the sample include the latitude, longitude, depth and date/time of the cast. In addition, pressure can be used as a coordinate instead of depth. Both pressure and depth are represented as arrays. These coordinates describe a point at the top of the sample cell and the lat/lon is recorded at the beginning of the sampling (some vessel drift may be experienced during the sampling).

Searchable Meta-Data

These data can be used to select CTD casts from the database.

Date/time	Range of dates and times when the data were collected
Location	A bounding polygon in lat/lon
Depth	A range of depth
Station name	The standard station name on which is was collected
Cruise Id	The id given to the cruise on which the data were collected

Ship name	The name of the ship on which the data were collected
Operator	The name of the individual who operated the CTD
Instrument Id	The id of the CTD instrument which collected the data
Temperature	A range of temperature
Pressure	A range of pressure
Salinity	A range of salinity

Estimate of Data Volume

In 1994 there were approximately 1000 ctd cast with an average of about 300 samples each. None of these casts included oxygen. The storage requirement for the HDF archive and the Illustra objects are based on these averages.

In HDF the overhead for the common meta-data (4000) plus the cruise meta-data (4000) plus the ctd meta-data (1000) come to total storage over-head of 9000 bytes. The overhead for 4 arrays (ctd without oxygen) with 300 samples per array is 9600 bytes. Together the average HDF file will occupy 18600 or about 20k per ctd dataset. With a seasonal average of 1000 datasets, 20Mb will be needed to store for the HDF archive of one seasons ctd data.

The storage requirements of Illustra will be much less. Only meta-data will be stored in Illustra and much of the data will not need to be replicated for each dataset. An estimate of 500 bytes per dataset will result in about .5Mb per season.

We also have between 2000 and 4000 historical ctd casts, each with about 300 samples. At 20k per dataset the HDF requirements for these datasets will be about 40Mb to 80Mb of space.

Open questions

- Does the number of data sets look right?
- What is the time zone are the dates and times for?
- Am I right that the coordinates are at the top of the cell?
- What percent of the casts will include oxygen?

1.1.1 ASCII Layout

The ASCII files include a header file, a CTD data file, and a free form text file for instrument calibration or notes. The header file has one line per dataset and the file names of the other two files. All of the dataset meta-data is located in the header file and the text file and the data arrays come from the data file.

ASCII Header data for CTD (id ctd-01) This is a comma delimited header file which contains one line per ctd dataset. Each line includes the dataset type and the format identifier (01). It also includes the information about the location and collection of the dataset. Finally it includes the name of the file which could include a varying amount of text describing the calibration of the CTD or other notes about the dataset.

	Variable	Units/Type	Description
1	Dataset Type	≤ 10 bytes	This dataset type 'ctd'
2	Format version	integer	This file format id: '01'
3	Dataset Id	≤ 16 bytes	A unique dataset reference id set by the submitting group and used to uniquely identify the dataset to the database.
4	Dataset PI	≤ 20 bytes	The name of the PI person who is responsible as the author of the data: 'Dave Salmon'
5	PI Affiliation	≤ 35 bytes	The affiliation of the PI for the data: 'PWSSC'
6	Origin of Dataset	≤ 10 bytes	This is the name of the group which originated the dataset. In SEA this is 'SEA-x' where x is the letter abbrievation for the group: 'SEA-M'
7	Dataset Comment	≤ 25 bytes	A comment used to describe this dataset. (optional)
8	Samples file	≤ 16 bytes	Name of the file containing the sample measurements
9	Samples format id	integer	The file format id number of the samples file

	Variable	Units/Type	Description
10	Cruise id	integer	The id which identifies the cruise on which the samples were collected. The format is xxnn where xx is the year and nn is sequence number within that year.
11	Sampling id	integer	The id which identifies a logical point during the cruise at which several samples were collected.
12	Location name	≤ 8 bytes	The name of the station or transect where the samples were collected.
13	Ship name	≤ 20 bytes	The name of the ship from which the samples where collected.

	Variable	Units/Type	Description
14	Instrument Model	$\leq$ 10 bytes	Instrument model number.
15	Instrument Number	$\leq$ 10 bytes	Instrument serial number.
16	Operator Id	$\leq$ 8 bytes	The id of the person who operated the instrument during the sample collection
17	Log book Id	$\leq$ 10 bytes	The id of the log book used during the sample collection
18	Calibration/Text file name	$\leq$ 16 bytes	Name of the file containing text describing the calibration or other text about the instrument, operation or dataset.
19	Latitude	deg.dec	Latitude of dataset.
20	Longitude	deg.dec	Longitude of dataset.
21	Date	yyyy-mm-dd	Date of dataset collection
22	Time	hh:mm:ss	Time of dataset collection
23	Time zone	4 bytes	Time zone of dataset collection (eg. "YDT")
24	Oxygen Flag	char	Oxygen concentration collected (y/n)
25	Number of samples	integer	The number of samples (lines) in the samples file
26	Salinity Model	$\leq$ 12 bytes	The name of the model used to derive salinity.

ASCII data for CTD casts (id ctd-11) This blank delimited ASCII format includes the variables collected during the CTD drop. The first line of the file will not contain data and is available for column header information.

	Variable	Units/Type	Description
1	Pressure	decibars	
2	Temperature	°C	
3	Salinity	unitless	
4	Flag	none	

1.1.2 HDF and Illustra Schema

A single HDF file will store one CTD dataset. This includes all of the data in the ASCII files along with generic data common to multiple SEA datasets. The data in each HDF file is broken down as follows. The Illustra objects will contain all of the data in the first group, 'dataset attributes'.

- Dataset Attributes
  - Common SEA dataset attributes
  - Names of the HDF file and the ASCII files used for the dataset

  - Cruise level attributes
  - Instrument/Operation attributes
  - CTD specific meta-data
  - Minimum and Maximum values of the CTD arrays
- Dataset Arrays
  - CTD arrays (1-d; size is number of depths)
  - Coordinate array (1-d; depth)

Dataset Attributes

Common meta-data for all SEA datasets

Every dataset will include meta-data which will identify the dataset, describe the data, describe the source of the data, and provide the geographic coordinates of the data. The various meta-data elements shared by all datasets are described below.

Dataset classification and identification

The dataset type, along with the format verion number, can be used to identify the variables and meta-data which are contained in the dataset. Over time, if variables are added or removed from a dataset or the meta-data describing those variables change, a new format version number will be assigned which identifies the new format of the dataset. The type-version is a concatenation of the the dataset type and the version number.

The dataset type and the dataset id unquely identify each dataset in the database. This is independent of the format and can be used to select a single dataset from the database. The dataset id is assigned by the group which submits the dataset in order to prevent duplicate copies of the same data.

Name	Type	Units	Description
dsetSetType	char[11]	none	This is the type of data set. Possible values: {echo, adcpoff, adcpnear, aquapack, opc, avhrr, surbird, surherr, avfry, avspawn, ctd, eggmort, fishcatch, isotope, phynuto, tide, weather, zoops}
dsetSetId	char[17]	none	A unique dataset id established by the submitting group.
dsetFmtVersion	integer	none	This identifies the variables and meta-data in the dataset.
dsetTypeVer	char[15]	none	Dataset type and version combined together to form a unique dataset type id.
dsetGeom	char[13]	none	This is the geometry of the samples within the dataset. Possible values: {relational, uniform, rectilinear, irregular}
dsetLevel	char[4]	none	This is the level of interpretation of this dataset. Possible values: {2, 3, 4, 5}

Dataset geo-location

The following meta-data items describe the space and time in which the data values are located. This includes a bounding polygon projected on the surface of the sound. This polygon is made up of a series of lat/lon points which are stored in the lat/lon arrays. Min and max depth and date/time are also stored for each dataset.

Name	Type	Units	Description
dsetLatArr	real	deg.dec	Array of latitude values for bounding polygon.
dsetLonArr	real	deg.dec	Array of Longitude values for bounding polygon.
dsetMinDepth	real	meters	This is the minimum sample depth of all the points in the Dataset.
dsetMaxDepth	real	meters	This is the maximum sample depth of all the points in the Dataset.
dsetBegDTime	timestamp	yyyy-mm-dd hh:mm:ss tz	Earliest sample time of the variables in the dataset.
dsetEndDTime	timestamp	yyyy-mm-dd hh:mm:ss tz	Latest sample time of the variables in the dataset.

Dataset origin

The following general meta-data items describing the dataset origin of the dataset and include a text comment which can be used for any identifying features. The PI is individual who is responsible for the data in the dataset and the submitting user is the individual who provided the data for the archive.

Name	Type	Units	Description
dsetOrigin	char[11]	none	This is the name of the group which originated the dataset. In SEA this is 'SEA-x' where x is the letter abbrievation for the group. (eg. SEA-M)
dsetPI	char[21]	none	The name of the PI who is the author of the dataset
dsetAffiliation	char[36]	none	The affiliation of the dataset PI.
dsetSubmitId	char[9]	none	The user id of the person who submitted the dataset to the archive.
dsetArrSize	char[18]	none	The size of the data arrays. (eg. 100x230)
dsetComment	char[26]	none	A comment used to describe this dataset.

File names and locations for SEA datasets

The data will be stored in ASCII file archive and in and HDF archive. These attributes provide the pointers to these files. Both types of files could be moved to CD or tape storage. If URL is specified then the directory string contains the URL type (eg. file://) along with the URL directory and the name includes the file name.

Name	Type	Units	Description
dsetASCIIFiles	text	none	A blank delmited list of ASCII files from which the dataset was taken.
dsetASCIIDir	char[129]	none	The name of the directory containing the ASCII files.
dsetASCIILocation	char[13]	none	Storage location of the ASCII files. Possible values: {disk, tape, CD, URL}
dsetHDFName	char[17]	none	The name of the HDF file in which the data is stored.
dsetHDFDir	char[129]	none	The name of the directory containing the HDF files.
dsetHDFLocation	char[13]	none	Storage location of the HDF files. Possible values: {disk, tape, CD, URL}
dsetHDFDTime	timestamp	yyyy-mm-dd hh:mm:ss tz	Date and time the HDF file was created.

Cruise meta-data for SEA datasets

Datasets collected on a survey cruise planned by SEA contain the id of the cruise and the ship name from which the samples were collected. The location is a unique transect or station and the sampling id is used to group together several different datasets which were collected at the same logical point during the cruise

Name	Type	Units	Description
crsCruiseId	long	none	This identifies the specific cruise on which the dataset was collected. The format is xxnn where xx is the year and nn is sequence number within that year.
crsSamplingId	integer	none	This number is attached to all datasets which are collected at the same logical point during a cruise.
crsShipName	char[21]	none	The name of the ship on which the dataset was collected.
crsLocationName	char[11]	none	The name of the station or transect where the dataset was collected.

Instrument/operation meta-data for SEA datasets

Some datasets will include data which describe the operation of the collection instrument. These data document the collection of the data and the calibration of the instrument. The calibration text is a free form description of how the collection instrument was set when the samples were taken.

Name	Type	Units	Description
igoInstrId	char[11]	none	Unique instrument identification.
igoOperatorId	char[9]	none	The id of the person who operated the instrument during the sample collection
igoLogId	char[11]	none	The id of the log book used during the sample collection
igoCalText	char[var]	none	Variable length text for calibration data or notes.

CTD File level meta-data attributes (id ctd-101) These are CTD specific, file level meta-data generated during ingestion of the CTD ASCII files.

Name	Type	Units	Description
ctdSalModel	char[13]	none	The name of the model used to derive salinity.
ctdNbrSamples	integer	none	The number of depths for each array
ctdOxyIncluded	char	none	Oxygen included in dataset (Y/N).
ctdAvgCellHght	integer	meters	Average height of the sample cell over which the data averaged.
ctdCoordLoc	char[8]	none	Location of the coordinate relative to the cell. 'top'

CTD min-max values (id ctd-111) These are minimum and maximum values for the CTD arrays. They are used to answer value based queries.

Name	Type	Units	Description
ctdTempArrMin	real	°C	Minimum value in temprature array.
ctdTempArrMax	real	°C	Maximum value in temprature array.
ctdSalArrMin	real	unitless	Minimum value in salinity array.
ctdSalArrMax	real	unitless	Maximum value in salinity array.
ctdOxyArrMin	real	ml/Liter	Minimum value in Oxygen array.
ctdOxyArrMax	real	ml/Liter	Maximum value in Oxygen array.
ctdPresArrMin	real	decibars	Minimum value in Pressure array.
ctdPresArrMax	real	decibars	Maximum value in Pressure array.

Dataset Arrays

CTD coordinate array (id ctd-111) The only coordinate array for CTD is depth. The depths are derived from pressure and a model of the relationship between depth and pressure.

Name	Type	Units	Description
ctdDepthArr	real	meters	Coordinate array of depths at which the measurements were taken.

CTD data arrays (id ctd-111) These are CTD data arrays. Not all datasets will include Oxygen.

Name	Type	Units	Description
ctdTempArr	real	°C	Array of temprature measurements.
ctdSalArr	real	unitless	Array of salinity measurements.
ctdPresArr	real	decibars	Array of pressure measurements from which depth is calculated.
ctdOxyArr	real	ml/Liter	Array of Oxygen measurements.

1.1 Echo Acoustics for Pollock

General Description

These datasets contain the results of echo soundings collected in offshore or nearshore surveys. The backscatter from echo sounder can be manually interpreted into numeric and biomass density of pollock, and biomass density of herring and zooplankton. The current datasets contain the biomass density and numeric density of pollock. These data were collected from broad area, offshore and nearshore surveys of the sound. Many were collected in the pink salmon migration corridor. The ground truth of some of the acoustic datasets is established from tow net samples.

Variables

Pollock biomass density	The biomass density of pollock in Kg/m^3 for each sample cell.
Pollock numeric density	The numerical density of pollock in $number/m^3$ for each sample cell.

In future, other variables might include the biomass density of zooplankton and herring.

Size of Sample Cell

The variables are retrieved from a given number of pings of the sonar equipment made along a cruise transect. The length of a sample cell depends on the resolution needed but it can range between 4m and 240m, 90m is common. This range is calculated from the ranges shown in the table below.

Pings/cell	8 – 60
Pings/second	1 – 3
Meters/second	1.5 – 4

These data are collected within cells which vary in size with depth. The vertical bin size is standardized during data interpretation to a 1 meter of depth. The horizontal dimension varies from 4m to 240m The average cell size of a dataset could be 90m by 1m.

Coordinates

The coordinates of the sample include the latitude, longitude and depth of each sample. The beginning and ending date and time are available for the the dataset as a whole. The coordinates of the sample values represent the middle of the top of the cell.

Estimate of Data Volume

Each acoustic dataset is approximately 125 by 50, or in the neighborhood of 6250 values. Each value requires 8 bytes or 50k for each array in the dataset.

With one variable in each dataset and an overhead of 8k for the HDF files the average file holding an acoustic dataset would need 58k bytes. In 1995 there were approximately 200 survey datasets and 100 near shore acoustic datasets. If this average holds true for other years, then the HDF files holding the acoustic data will require approximately 17.5Mb of space per season.

The storage requirements of Illustra will be much less. Only meta-data will be stored in Illustra and much of the data will not need to be replicated for each dataset. An estimate of 500 bytes per dataset results in about 150kb for the Illustra objects.

Searchable Meta-data

The following data items can be used to search for acoustic datasets.

Date/time	Range of dates and times when the data were collected
Location	A bounding polygon in lat/lon
Depth	A range of depth
Biomass Density	A range of pollock biomass density
Numerical Density	A range of pollock numerical density
Transect name	The standard transect id used by ADFG
ADFG Set number	The set number assigned by ADFG to net tow.
ADFG Site id	The ADFG numeric site id.
Survey Type	The code used to indicate nearshore, offshore, or broad scale survey.
Cruise Id	The id given to the cruise on which the data were collected
Ship name	The name of the ship on which the data were collected
Operator	The name of the individual who operated the echo sounder
Instrument Id	The id of the echo sounder instrument which collected the data

Tools used for Analysis

These data are acquires and reduced using the BioSonics Echo Signal Processer (ESP); they are processed, regridded and exported with IDL. These data are visualized using AVS.

1.1.1 ASCII Layout

The ascii files include a sampling header file, an instrument operation file, and a variables or samples file. The header file has one line per dataset and there is one samples file per dataset.

The instrument operation file is described below and could pertain to several different datasets. The file names can be 16 characters long and the date/time values are yyyy-mm-dd hh:mm:ss tz in Alaska time.

ASCII Header data for echo acoustic data (id echopol-01) This is a comma delimited header file which contains one line per acoustic dataset. Each line includes the dataset type and the format identifier (01). It also includes the information about the location and collection of the dataset. Finally it includes the name of the file which could include a varying amount of text describing the calibration of the sonar equipment or other notes about the dataset including model parameters used when the data was derived.

The location field is the ADFG transect number which is a number between 1 and 88. The group cruise number is assigned by SEA-N to different cruise legs.

	Variable	Units/Type	Description
1	Dataset Type	$\leq$ 10 bytes	This dataset type 'echopol'
2	Format version	integer	This file format id: '01'
3	Dataset Id	$\leq$ 16 bytes	A unique dataset reference id set by the submitting group and used to uniquely identify the dataset to the database.
4	Dataset PI	$\leq$ 20 bytes	The name of the PI person who is responsible as the author of the data: 'Gary Thomas'
5	PI Affiliation	$\leq$ 35 bytes	The affiliation of the PI for the data: 'PWSSC'
6	Origin of Dataset	$\leq$ 10 bytes	This is the name of the group which originated the dataset. In SEA this is 'SEA-x' where x is the letter abbrievation for the group: 'SEA-N'
7	Dataset Comment	$\leq$ 25 bytes	A comment used to describe this dataset. (optional)
8	Samples file	$\leq$ 16 bytes	Name of the file containing the sample measurements
9	Samples format id	integer	The file format id number of the samples file

	Variable	Units/Type	Description
10	Cruise id	integer	The id which identifies the cruise on which the samples were collected. The format is xxnn where xx is the year and nn is sequence number within that year.
11	Sampling id	integer	The id which identifies a logical point during the cruise at which several samples were collected.
12	Location name	$\leq$ 8 bytes	The name of the station or transect where the samples were collected.
13	Ship name	$\leq$ 20 bytes	The name of the ship from which the samples where collected.

	Variable	Units/Type	Description
14	Instrument Id	$\leq$ 10 bytes	Unique instrument identification.
15	Operator Id	$\leq$ 8 bytes	The id of the person who operated the instrument during the sample collection
16	Log book Id	$\leq$ 10 bytes	The id of the log book used during the sample collection
17	Calibration/Text file name	$\leq$ 16 bytes	Name of the file containing text describing the equipment calibration or other text about the instrument, operation or dataset.
18	Begin Date	yyyy-mm-dd	Begin Date of dataset collection
19	Begin Time	hh:mm:ss	Begin Time of dataset collection
20	End Date	yyyy-mm-dd	End Date of dataset collection
21	End Time	hh:mm:ss	End Time of dataset collection
22	Time zone	4 bytes	Time zone of dataset collection (eg. "AKDT")
23	Group Cruise Id	$\leq$ 8 bytes	The cruise number used by a specific group.
24	ADFG Set Number	$\leq$ 9 bytes	The ADFG set number. This is the set number of the trawl which was underway during echo sounding.
25	ADFG Site Number	integer	The ADFG numeric site id.
26	Survey Type	1 byte	The type of survey (O=Offshore, N=Nearshore, B=Broadscale).
27	Model Name	$\leq$ 12 bytes	A name which describes the models used to derive the data values.

ASCII data for Acoustics datas (id echo-11) This blank delimited ASCII format includes the variables derived from the acoustic soundings. Biomass density of pollock is the variable and the other data items are coordinates.

	Variable	Units/Type	Description
1	Latitude	deg.dec	Latitude of sample
2	Longitude	deg.dec	Longitude of sample
3	Depth	meters	Depth of sample
4	Biomass density	Kg/m^3	Biomass density of pollock
5	Numeric density	$number/m^3$	Numerical density of pollock

1.1.2 HDF and Illustra Schema

A single HDF file will store one Acoustic dataset. This includes all of the data in the ASCII files along with generic data common to multiple SEA datasets. The data in each HDF file is broken down as follows. The Illustra objects will contain all of the data in the first group, 'dataset attributes'.

- Dataset Attributes
  - Common SEA dataset attributes
  - Names of the HDF file and the ASCII files used for the dataset
  - Cruise level attributes
  - Instrument/Operation attributes
  - Acoustic specific meta-data
  - Minimum and Maximum values of the Acoustic arrays
- Dataset Arrays
  - Acoustic arrays (2-d; cols x depth)
  - Coordinate arrays (1-d; lat, lon, & depth)

Dataset Attributes

Common meta-data for all SEA datasets

Every dataset will include meta-data which will identify the dataset, describe the data, describe the source of the data, and provide the geographic coordinates of the data. The various meta-data elements shared by all datasets are described below.

Dataset classification and identification

The dataset type, along with the format verion number, can be used to identify the variables and meta-data which are contained in the dataset. Over time, if variables are added or removed from a dataset or the meta-data describing those variables change, a new format version number will be assigned which identifies the new format of the dataset. The type-version is a concatenation of the the dataset type and the version number.

The dataset type and the dataset id unquely identify each dataset in the database. This is independent of the format and can be used to select a single dataset from the database. The dataset id is assigned by the group which submits the dataset in order to prevent duplicate copies of the same data.

Name	Type	Units	Description
dsetSetType	char[11]	none	This is the type of data set. Possible values: {echopol, adcpoff, adcpnear, aquapack, opc, avhrr, surbird, surherr, avfry, avspawn, ctd, eggmort, fishcatch, isotope, phynuto, tide, weather, zoops}
dsetSetId	char[17]	none	A unique dataset id established by the submitting group.
dsetFmtVersion	integer	none	This identifies the variables and meta-data in the dataset.
dsetTypeVer	char[15]	none	Dataset type and version combined together to form a unique dataset type id.
dsetGeom	char[13]	none	This is the geometry of the samples within the dataset. Possible values: {relational, uniform, rectilinear, irregular}
dsetLevel	char[4]	none	This is the level of interpretation of this dataset. Possible values: {2, 3, 4, 5}

Dataset geo-location

The following meta-data items describe the space and time in which the data values are located. This includes a bounding polygon projected on the surface of the sound. This polygon is made up of a series of lat/lon points which are stored in the lat/lon arrays. Min and max depth and date/time are also stored for each dataset.

Name	Type	Units	Description
dsetLatArr	real	deg.dec	Array of latitude values for bounding polygon.
dsetLonArr	real	deg.dec	Array of Longitude values for bounding polygon.
dsetMinDepth	real	meters	This is the minimum sample depth of all the points in the Dataset.
dsetMaxDepth	real	meters	This is the maximum sample depth of all the points in the Dataset.
dsetBegDTime	timestamp	yyyy-mm-dd hh:mm:ss tz	Earliest sample time of the variables in the dataset.
dsetEndDTime	timestamp	yyyy-mm-dd hh:mm:ss tz	Latest sample time of the variables in the dataset.

Dataset origin

The following general meta-data items describing the dataset origin of the dataset and include a text comment which can be used for any identifying features. The PI is individual who is responsible for the data in the dataset and the submitting user is the individual who provided the data for the archive.

Name	Type	Units	Description
dsetOrigin	char[11]	none	This is the name of the group which originated the dataset. In SEA this is 'SEA-x' where x is the letter abbrievation for the group. (eg. SEA-M)
dsetPI	char[21]	none	The name of the PI who is the author of the dataset
dsetAffiliation	char[36]	none	The affiliation of the dataset PI.
dsetSubmitId	char[9]	none	The user id of the person who submitted the dataset to the archive.
dsetArrSize	char[18]	none	The size of the data arrays. (eg. 100x230)
dsetComment	char[26]	none	A comment used to describe this dataset.

File names and locations for SEA datasets

The data will be stored in ASCII file archive and in and HDF archive. These attributes provide the pointers to these files. Both types of files could be moved to CD or tape storage. If URL is specified then the directory string contains the URL type (eg. file://) along with the URL directory and the name includes the file name.

Name	Type	Units	Description
filSetType	char[11]	none	This is the type of data set. Possible values: {echopol, adcpoff, adcpnear, aquapack, opc, avhrr, surbird, surherr, avfry, avspawn, ctd, eggmort, fishcatch, isotope, phynuto, tide, weather, zoops}
filSetId	char[17]	none	A unique dataset id established by the submitting group.
filFileName	char[17]	none	The file name.
filFileDesc	char[31]	none	The description of the file.
filFileLoc	char[129]	none	The path to the file.
filFileFmt	char[7]	none	The format of the file (ASCII, HDF, DBMS).
filFileCont	char[4]	none	The type of data stored in the file: (hdr, dta, cal, txt, ...).
filFileOnline	boolean	none	(t/f) t: The file is online f: The file is nearline or on tape.
filFileURL	boolean	none	(t/f) t: The file name is a URL f: The file name is not a URL.
filFileDTime	timestamp	none	The date and time the file was last modified.
filFileOnline	integer	bytes	The size of the file in bytes.

Cruise meta-data for SEA datasets

Datasets collected on a survey cruise planned by SEA contain the id of the cruise and the ship name from which the samples were collected. The location is a unique transect or station and the sampling id is used to group together several different datasets which were collected at the same logical point during the cruise

Name	Type	Units	Description
crsCruiseId	long	none	This identifies the specific cruise on which the dataset was collected. The format is xxnn where xx is the year and nn is sequence number within that year.
crsSamplingId	integer	none	This number is attached to all datasets which are collected at the same logical point during a cruise.
crsShipName	char[21]	none	The name of the ship on which the dataset was collected.
crsLocationName	char[11]	none	The name of the station or transect where the dataset was collected.

Instrument/operation meta-data for SEA datasets

Some datasets will include data which describe the operation of the collection instrument. These data document the collection of the data and the calibration of the instrument. The calibration text is a free form description of how the collection instrument was set when the samples were taken.

Name	Type	Units	Description
igoGearName	char[11]	none	Gear/Instrument name.
igoGearNumber	char[11]	none	Gear/Instrument serial number.
igoOperatorId	char[9]	none	The id of the person who operated the instrument during the sample collection
igoLogId	char[11]	none	The id of the log book used during the sample collection
igoCalText	char[var]	none	Variable length text for calibration data or notes.

Echo acoustics file level meta-data attributes (id echo-101) These are Acoustic specific, file level meta-data generated during ingestion of the acoustic ASCII files.

Name	Type	Units	Description
echSurveyType	char	none	The flag indicating the type of survey. (O=Offshore, N=Nearshore, B=Broadscale)
echNetSet	char[10]	none	The ADFG set number.
echSiteNbr	integer	none	The site number used by ADFG.
echGroupCruise	char[9]	none	The cruise number used by a specific group.
echNetSet	char[10]	none	The ADFG set number.
echModel	char[13]	none	The name of the model(s) used to derive data.
echAvgCellLen	integer	meters	Average length of the sample cell over which the data averaged.
echAvgCellHght	integer	meters	Average height of the sample cell over which the data averaged.
echCoordLoc	char[8]	none	Location of the coordinate relative to the cell. vertical:horizontal {{top, mid, bot} : {beg, mid, end}} 'top:mid' for Acoustic.

Echo acoustic min-max values (id aco-111) These are minimum and maximum values for the acoustic data arrays. They are used to answer value based queries.

Name	Type	Units	Description
echPolBioMin	real	Kg/m^3	Minimum value in the Pollock biomass array.
echPolBioMax	real	Kg/m^3	Maximum value in Pollock biomass array.
echPolNbrMin	real	$number/m^3$	Minimum value in the Pollock numeric density array.
echPolNbrMax	real	$number/m^3$	Maximum value in Pollock numeric density array.

Dataset Arrays

Echo acoustic coordinate array (id echo-111) The coordinate arrays for acoustic datasets includes the latitude, longitude, and depth.

Name	Type	Units	Description
echLatArr	real	deg.dec	Array of latitude values of samples points.
echLonArr	real	deg.dec	Array of Lonitude values of samples points.
echDepArr	real	meters	Array of depth values of samples points.

Acoustic data arrays (id aco-111) The acoustic data array maintains the biomass density of pollock for each sample cell.

Name	Type	Units	Description
echPolBioArr	real	Kg/m^3	Array of biomass of Pollock.
echPolNbrArr	real	$number/m^3$	Array of numeric density of Pollock.

Appendix 5

Design and Implementation of the SEA Data Archive

The Design and Implementation of the SEA Data Archive

Charles Falkenberg

Department of Computer Science
Advanced Visualization Laboratory
University of Maryland
College Park, MD 20742
csfalk@cs.umd.edu

Ravi Kulkarni

Advanced Visualization Laboratory
University of Maryland
College Park, MD 20742
ravi@avl.umd.edu

Abstract

The data archive for the EVOS Sound Ecosystem Assessment (SEA) project contains a wide range of datasets from several different scientific disciplines. The methodology for developing this archive is broken into three tracks: the documentation of the data, the implementation of the data ingestion system, and the development of the tools used to retrieve the data. This paper outlines the functionality of the data archive and describes the tasks undertaken in each of these three tracks. A brief status as of January 1996 is presented as a conclusion.

Overview

The data archive, or database, for the Sound Ecosystem Assessment project (SEA) is designed to store datasets from the 14 projects which are under the SEA umbrella. In addition, the archive will contain historical datasets and other datasets from outside of SEA. The data include physical variables describing the oceanography, hydrology and weather, as well as biologic variables describing the plankton, salmon, pollock, and herring in the Sound. Currently 18 different types of datasets have been identified and it is estimated that between 4000 and 5000 datasets will be added to the archive each year. The goal is to build a long lasting data archive along with the query tools to retrieve datasets from the archive.

The development of the SEA database is progressing along three tracks:

1. The creation of a dictionary of SEA datasets and data elements
2. The design and implementation of a SEA data management architecture
3. The development of software tools for managing the database and retrieving datasets

Although these tracks are proceeding in parallel the begin dates must be staggered. Some of the data dictionary must be complete before the ingestion portion of the architecture can begin, and some of the datasets must be ingested before the query tools can be fully designed and developed. These three development tracks are described below and together offer an overview of components and functionality of the SEA data archive.

Data Dictionary

The data dictionary for SEA includes the definitions of all of the datasets and data elements in the database. The dictionary entry for each data item includes the item's name, type, unit of measure, and description. The dictionary entry for each dataset includes the following categories:

- A brief description of the dataset
- The list of the variables in the dataset
- The sampling geometry and sample cell size for each variable
- An estimate of the volume of data per year
- A list of the meta-data items which can be used in searches to select the dataset from the DBMS
- The list and layout of the ASCII files which are submitted to the archive as the dataset
- The layout of the HDF files and the DBMS tables which contain the dataset in the archive

One of the prime goals of creating a SEA data dictionary is to identify and standardize shared data elements. These shared data elements will allow query access to multiple types of datasets using a common data definition. Shared elements include the data variables themselves (eg. temperature), geographic coordinates, and dates and times. Shared, collection related, meta-data items such as the survey cruise id, or the name of the PI should also prove to be valuable selection criteria.

The SEA data dictionary will be the comprehensive documentation of the all SEA data. A complete dictionary of the datasets and data elements in the

SEA database will allow effective use of the database by the wide range of research disciplines within SEA. In addition, the long term research potential of these datasets will depend upon comprehensive and consistent documentation.

Data Management Architecture

The structure of the SEA database system is shown in figure 1. It includes an archive of datasets stored in HDF files and an index of those datasets in the Illustra^{tm} DBMS. The ingestion of datasets from ASCII input files includes the creation of the HDF file for each dataset and an entry for the dataset in the DBMS indexes. Queries are submitted to the DBMS through one of several software tools and the result is a list of datasets which meet the search criteria. These components and the relationships between them make up the SEA data management architecture.

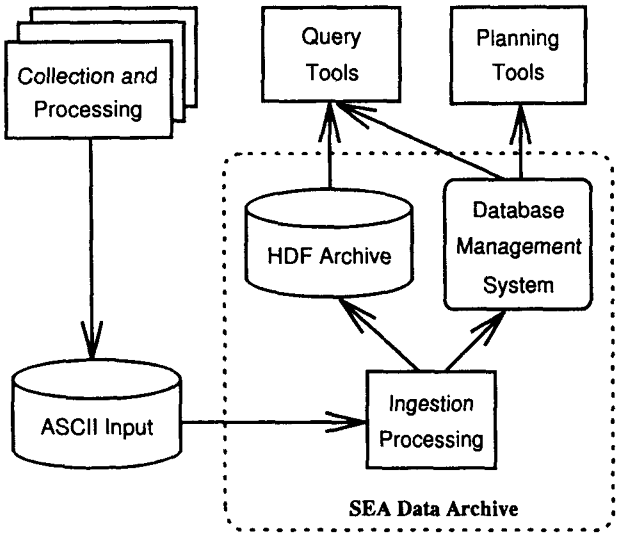


Figure 1: Architecture of the SEA data management system showing the main processing and data components. Arrows indicate the flow of data between components.

The primary storage mechanism in the data archive is the Hierarchical Data Format [4] which was designed and is maintained by the National Center for Supercomputing Applications (NCSA) at the University of Illinois. This scientific data format is called self-describing because each data element is accompanied by a name and definition. This feature, in addition to some other facilities, allow HDF files to be transported to and accessed on different hardware platforms including workstations and PCs.

Encapsulation and portability were part of the HDF specification in order to support long lived data archives. The HDF standard has been chosen by NASA's EOS project as the primary dataset storage technology. This ensures that HDF will continue to receive excellent support and will be available well into next century.

The SEA Software tools

The SEA software tools will utilize the Illustra^{tm} DBMS and will be used to plan the data collection process and query and manage the data archive. The three main tools are:

- Planning tools for organizing the survey cruises.
- Netscape based query tools for selecting and displaying datasets
- Query tools which run at a local workstation or PC

A planning tool is currently running and provides several maps of the Sound on which a survey cruise can be drawn. Once the path of the cruise is laid out, dates and resources can be assigned to each leg or station along the cruise path. The output from the cruise planner can be imported to a spreadsheet program to allow costs to be assigned to each leg of the cruise.

The query tools use the indexing facilities of the Illustra^{tm} DBMS to identify and locate the datasets which meet the given selection criteria. Illustra is an Object-Relational DBMS which allows new data types to be defined and stored using a traditional table structure. A package of spatial data types which includes two dimensional points, lines, and polygons is being used in the SEA database to provide a spatial index of datasets. These spatial queries are the first of the three basic types of queries which the SEA database supports:

- Spatial and temporal queries
- Other meta-data based queries
- Variable value base queries

Spatial and temporal queries result in a list of datasets which are in a given region of space or range of time. Meta-data queries allow selection based on meta-data which describe the collection or interpretation of the data values. An example would be all datasets collected with a given instrument or on a particular survey cruise. The third type of query results in a list of datasets which have variables within a range of values. The result list can be then used to download all or part of the variables in the HDF file or to display the location of the datasets on a map.

The Netscape based query tool allows access to all SEA data from any machine which is able to run Netscape. Once the proper authorization is provided, HDF or ASCII files can be downloaded or displayed using Netscape facilities.

The local query tools have the additional advantage that they can check the local archive of datasets and only download new or changed datasets. In addition, a local tool can be used in conjunction with an application which can accept a list of datasets and process and display the data. This will be an important feature for the large modeling efforts which are part of SEA.

Current Status of Development

Together these three development tracks provide a framework for understanding the design and development of the SEA database. As of January 1996, interviews with all but a few of the groups within SEA have been conducted and 5 of the 18 datasets have been documented. The SEA data dictionary currently includes these 5 datasets and approximately 330 data elements. Most of data elements common to all datasets have been identified and standardized.

A prototype of the data management architecture was implemented last spring and is available for review. The ingestion of the CTD datasets into the production version of the SEA database underway and should be completed in February. A major portion of ingestion processing must be developed as part of the ingestion of this first dataset. A generic data structure is being used which is able to describe most of the datasets within SEA reducing the development effort required for each new dataset.

A planning tool was designed and implemented last summer and fall. It was used to plan parts of the October cruise and is now available for the 96 planning season. This planning tool is also being used as a display tool for the CTD datasets which are in the prototype.

The completion dates for rest of the ingestion process and the query tools will depend upon the resources available. The complexity of the datasets varies widely but a general estimate of bringing a new dataset online is 4 to 8 weeks of individual effort. This includes 2-3 weeks of discussion with each group to define and document each part of the dataset. This time includes the layout of all files and database objects and might include reformatting existing ASCII files to meet the design specifications. Another 2-3 weeks will be needed to create the new programs needed to read in and ingest the dataset. Finally 1-2 weeks are needed to incorporate the new dataset into the query tools. In addition, several weeks will be needed for the completion of the initial ingestion processing and the initial development of the query tools.

The new year begins with the data definition work done for several of the datasets and most of the initial design and development of the ingestion processing. In addition, the planning tool and the database prototype provide a solid set of design criteria for the planned query tools.

References

[1] R. Ted Cooney et al. Sound Ecosystem Assessment (SEA) - An Integrated Science Plan for the Restoration of Injured Species in Prince William Sound. Submitted to the Exxon Valdes Oil Spill Trustee Council, April 1994.

[2] J.C. French, A.K. Jones, and J.L. Pfaltz. "Summary of the NSF workshop on scientific database management". *SIGMOD Record*, 19(4), December 1990.

[3] Illustra Technology. *The Illustra^{tm} Users Manual*, 1994.

[4] National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign. *HDF Reference Manual*, February 1994.

Appendix 6

Modeling Prince William Sound Ocean Circulation

MODELING PRINCE WILLIAM SOUND OCEAN CIRCULATION

Jia Wang* and Christopher N.K. Mooers

University of Miami, Miami, Florida 33149

1 . INTRODUCTION

Prince William Sound (PWS) is a combination of multiple fjords, basins, and estuaries along the coast of Alaska. Its spatial scale is approximately 80x80 km with an average depth of about 150m. Its deep basins (with a maximum depth of about 750m) and channels were formed by a combination of preglacial erosion, glacial excavation, and tectonism.

The water exchange between PWS and the coastal Gulf of Alaska strongly influences the circulation pattern and biomass distribution (Schmidt, 1977). The spatial scale of the basin is barely large enough for a recirculation to develop (Niebauer et al. 1994). PWS is very rich in the production of salmon, halibut, herring, and other fish species. Its economic potential strongly depends on how well the fishcatch can be managed and how well the pollution can be minimized in the presence of a major oil tanker route. Therefore, understanding the circulation patterns is basic to understanding PWS ecology and environmental risks. Our goal is to establish a 3-D nowcast/forecast system (and later, possibly a coupled physical-biological (ecosystem) numerical model, including nutrients, phytoplankton, zooplankton, and fish concentrations and their interactions). In this paper, thus, a 3-D model of PWS is presented and the dynamical origin and nature of the circulation are examined.

Tidal currents are very strong in PWS due to the tidal range of 3 to 4 meters. Such a strong tide (with the M_2 constituent being the largest) helps to flush PWS, by mixing coastal waters with those of PWS. The magnitude and pattern of the tidal residual current is another important aspect. These circulation features are not understood. Observations have been conducted since 1973 in PWS (Royer et al. 1979) and along the coast of Alaska (Royer, 1975); it was found that the coastal current usually intrudes into PWS from Hinchinbrook Entrance and drives the basin-scale cyclonic circulation.

A modified Princeton Ocean Model (POM, Blumberg and Mellor 1987), which has been successfully applied to the circulation of Hudson Bay (Wang et al. 1994), is used. It uses the primitive equations (with hydrostatic and Boussinesq approximations) and has the following features: (1) horizontal curvilinear coordinates; (2) an "Arakawa C" scheme; (3) sigma (terrain-following) coordinates in the vertical with realistic bottom topography; (4) a free surface; (5) a level 2.5 turbulence closure model for the vertical viscosity and diffusivity (Mellor and Yamada, 1982); (6) a mean flow shear parameterization for horizontal viscosity and diffusivity (Smagorinsky, 1963); (7) a semi-implicit scheme to solve for surface elevation in the shallow water equations; and (8) a predictor-corrector scheme for the time integration to avoid inertial instability (Wang and Ikeda, 1995).

**Corresponding author address:* Jia Wang, Ocean Prediction Experimental Laboratory, Univ. of Miami, 4600 Rickenbacker Cswy., Miami, FL 33149

The model domain includes the entire PWS with two open boundaries allowing water exchange with the Alaskan coastal water, Hinchinbrook Entrance and Montague Strait (Fig 1). The model grid spacing is 1.2 km, which is eddy-resolving because the internal Rossby radius of deformation is about 5 km (Niebauer et al. 1994). There are 11 vertical sigma levels. The integration time step is 62.1 seconds.

According to the observations in Hinchinbrook Entrance (Niebauer et el, 1994), the coastal inflow varies seasonally: from 0.1 to 0.3 Sverdrup (1 Sv=10^6 m^3s^{-1}). Similarly, the outflow through Montague Strait is of the same order of magnitude according to the conservation of volume transport, though the water volume in PWS may increase or decrease in response to transient forcing. Hence, an inflow of 0.3 Sv was specified through the Hinchinbrook Entrance, while a radiation boundary condition (with self-adjusted outflow of 0.3 Sv) was applied to the Montague Strait. Outflow may also occur through Hinchinbrook Entrance. A coastal M_2 tide with an amplitude of 1.5 m (i.e., the tidal range is 3 m) is specified at both open boundaries.

Section 2 describes the initial simulations. Section 3 presents the future nowcast/forecast system. Section 4 summarizes the present results and the next steps in the future study of the PWS circulation.

2. INITIAL SIMULATION RESULTS

The initial temperature and salinity fields use typical summer profiles with no horizontal variations, because comprehensive spatial data are not yet available. Thus, the results shown below exclude the density-driven component which is smaller than, but perhaps comparable in magnitude to the barotropic current. The model is spun up from these initial conditions for 8 days. At this time a dynamical steady-state is reached, and the restart file is saved for use as the initial condition in the next runs. There are no heat and salt fluxes specified yet at the ocean surface. Vertical viscosity is determined from the Mellor-Yamada 2.5 turbulence closure model with a background viscosity of 10^{-5} m^2s^{-1}, and the horizontal viscosity is determined from the Smagorinsky horizontal mixing closure with C=0.2. The typical computed horizontal viscosity is about 5 to 10 m^2s^{-1}.

2.1 INFLOW-DRIVEN CIRCULATION. The inflow from the Alaskan coastal waters is an important factor in determining the circulation pattern in PWS. The vertical distribution of the inflow decreases linearly from the surface to 150m depth, the sill depth. Tidal forcing is specified at the two open boundaries; there is no wind forcing. The inflow temperature and salinity profiles were taken to be the same as the boundary point values, i.e., the zero horizontal gradient (no heat and salt fluxes) condition was used.

Based on the circulation pattern at depths of 10m and 100m on day 25 (Fig. 2), a coastal inflow of 0.3 Sv enters from Hinchinbrook Entrance and exits through the Montague Strait at 10m, and there is an anticyclonic eddy in the eastern PWS. Inside the cyclonic main stream (inflow), there is a cyclonic eddy (Fig. 2a). At 10m, the outflow is largely channelled through the Knight Inlet (i.e., east of Knight Island).

At 100m, the inflow flows to the north following the deep channel and forms a large cyclonic circulation. The outflow is split between Knight Channel and Montague Strait. Interestingly, the inflow separates into two branches near 60.5N, 147W, perhaps due to two small seamounts (Fig. 1). In the northeastern PWS, there is a cyclonic eddy. These mesoscale eddy features reveal an important fact that PWS, in contrast to typical estuaries, has a large enough horizontal scale to not constrain mesoscale eddy development.

Figure 3 shows an 8-day time series of surface velocity vectors taken from three grid points (see locations in Fig. 1). Over the deepest basin (grid 1), the current is weak, due to the topography and geometry of the interior PWS. In Montague Strait (grid 2), because it is shallow, the tidal current superimposed on the mean southwestward current is very energetic, while the currents near Hinchinbrook Entrance are relatively weak because it is deep (grid 3).

2.2 WIND-DRIVEN CIRCULATION. Wind regimes over PWS vary seasonally with changes in the position and strength of the Aleutian Low. Eastward wind (i.e., Alaskan coastal upwelling favourable wind) tends to occur during summer because of the influence of the North Pacific High, while in winter, there are strong, steady westward winds (i.e., Alaskan coastal downwelling favourable winds) when the Aleutian Low deepens. The surface current fields under eastward and westward wind forcing of 7 ms^{-1} (Fig. 4) (with the same inflow, but no M_2 tide) have the same circulation patterns at 10m, except for small differences. The eastward wind displaces the main stream of inflow to the south through the effects of southward surface Ekman transport, while the westward wind displaces the main stream northward, through the northward surface Ekman transport. There are also numerous, weak mesoscale eddies in the northern PWS. The westward wind produces stronger circulation in the northwestern PWS. Additionally, the eastward (westward) wind piles up the water on the eastern (western) shore by typically 0.3 m (not shown).

2.3 TIDAL-DRIVEN RESIDUAL FLOW. Comparing Figs. 2a and 4, the tide-induced residual current is prominent, modifying the circulation pattern. Because of the presence of many sills, shoals, and seamounts, the interaction of the strong M_2 tidal current and topography leads to a significant tidal-residual mean flow, typically 0.05-0.1 ms^{-1}, i.e., one order of magnitude smaller than the tidal current. This residual flow is comparable to, even though smaller than, the mean current due to the coastal inflow. Therefore, the contribution of the tide to the mean circulation is very important in PWS. It remains to add the other major tidal constituents to the model.

2.4 GENERAL CIRCULATION PATTERN. With tidal, throughflow, and eastward wind (summer case) forcing (Fig. 5), as expected, the main stream is displaced to the south by the surface Ekman transport compared with Fig. 2a. In northwestern PWS, the alongshore current is strengthened.

3. NOWCAST/FORECAST SYSTEM PLAN

To establish the Prince William Sound nowcast/forecast system (PWSNFS), similar to the Straits of Florida Nowcast/Forecast System (SFNFS, Mooers and Ko, 1993; Mooers et al. 1995), the following two initial tasks will be accomplished:

1) Observational System. PWSNFS not only involves the sophisticated 3-D model presented here, but also on a reliable observational network. This network should include the historical temperature and salinity profiles, meteorological stations and buoys, moored current meters, moored and shipboard ADCPs (acoustic doppler current profilers), coastal tide gauges, and surface drifters, as well as other elements. As with atmospheric weather analyses and forecasts, the final stage of PWSNFS will include a robust data assimilation scheme.

The initial observational network (in Fig. 6) commenced from 1994 and will hopefully continue and be extended for several years. These data will be used in PWSNFS as much as possible. For example, the ADCP mooring at the Hinchinbrook Entrance will be analyzed using empirical orthogonal function analysis to determine modal (baroclinic and barotropic) distribution of variance for the transient flow in order to more accurately specify the inflow profile. The mean and variable inflow transport will be estimated based on the time series. The weather stations and the buoys will provide synoptic winds and other atmospheric variables for the model. The tide gauge data will be harmonically analyzed to determine the tidal amplitudes and phases, and the low frequency variability which is essential to evaluating the model transient response to atmospheric storms, etc. Current meter mooring and surface drifter deployments are anticipated in the future; they will facilitate Eulerian and Lagrangian comparisons between the model and observations.

2) Atmospheric Forcing. A synoptic wind field is essential for PWSNFS, similar to SFNFS (i.e., the atmospheric Eta model winds from the National Meteorological Center will be used). The winds from the new version of the Eta model (which provides a finer horizontal resolution of 28 km) will be evaluated for use in PWSNFS. The six-hourly wind will be used to force the model. Heat and moisture fluxes from the Eta model output will also be evaluated for use as the surface boundary conditions in driving the circulation model, especially the ocean mixed layer and ecosystem dynamics. Due to coastal orographic effects, the Eta model output will probably be combined with local data using an air-sea boundary layer model. The influence of freshwater inflows from PWS coastal orography and the Alaskan Coastal Current, as well as precipitation, are yet to be examined.

4. SUMMARY AND FUTURE WORK

POM has begun to be applied to PWS and some important dynamical factors influencing the circulation pattern have been determined. The initial simulation results indicate that POM has produced basically correct circulation patterns. Inflow/outflow between PWS and the Alaskan coastal waters, wind forcing, and tidal forcing are all essential factors. Vigorous mesoscale eddies are a prominent phenomenon and may be important for biomass distribution, because they influence the biomass abundance, concentration, and residence time. Furthermore, different wind conditions may change the residence time by changing the circulation pattern and stratification.

A seasonal cycle simulation (with and without synoptic forcing and freshwater inflow) will be conducted for both physical and biological interests. The wintertime convection that produces the deep water mass of the major basin will be examined as well. When these steps are accomplished, the physical model will be coupled to the biology, providing as much information for ecological and fishery science as possible in the near future.

ACKNOWLEDGEMENT

This research is sponsored by the EVOS Trusteeship Council, via the Prince William Sound Science Center, Cordova, Alaska.

REFERENCES

Blumberg, A.F. and G.L. Mellor, 1987. A description of a 3-D coastal ocean circulation model. In Coastal and Estuarine Sciences 4, N.S. Heaps, ed., AGU, Washington D.C.: 1-16.

Mellor, G.L. and T. Yamada, 1982. Development of a turbulence closure model for geophysical fluid problem. **Rev. Geophys. Space Phys.**, 20: 851-875.

Mooers, C.N.K. and D.-S. Ko. Nowcast system development for the Straits of Florida. Estuarine and Coastal Modeling III, Proceedings of the 3rd Intern. Conf., pp 158-171.

Mooers, C.N.K., D.-S. Ko, and J. Wang. Straits of Florida nowcast/forecast system. OPRC tech. Rep. 95-1, RSMAS, Univ. of Miami, 40pp.

Niebauer, H.J., T.C. Royer and T.J. Weingartner, 1994. Circulation of Prince William Sound, Alaska. **J. Geophys. Res.**, 99: 14,113-14,126.

Royer, T.C., 1976. Seasonal variations of waters in the northern Gulf of Alaska. **Deep Sea Res.,** 22: 403-416.

Royer, T.C., D.V. Hansen and D.J. Pashinski, 1979. Coastal flow in the northern Gulf of Alaska as observed by dynamic topography and satellite-tracked drogued drift buoys. **J. Phys. Oceanogr.,** 9: 785-801.

Smagorinsky, J., 1963. General circulation experiments with the primitive equations, I. The basic experiment. **Mon. Weather Rev.**, 91: 99-164.

Schmidt, G.M., 1977. The exchange of water between Prince William Sound and the Gulf of Alaska, M.S. thesis, 116 pp, Univ. of Alaska, Fairbanks.

Wang, J., L.A. Mysak and R.G. Ingram, 1994. A 3-D numerical simulation of Hudson Bay summer circulation: Topographic gyres, separations and coastal jets. **J. Phys. Oceanogr.**, 24: 2496-2514.

Wang, J. and M. Ikeda, 1995. Stability analysis of finite difference schemes for inertial oscillations in ocean general models. in **Computer Modeling of Seas and Coastal Regions**, eds. C.A. Brebbia et al., Computational Mechanics Publications, Southampton.

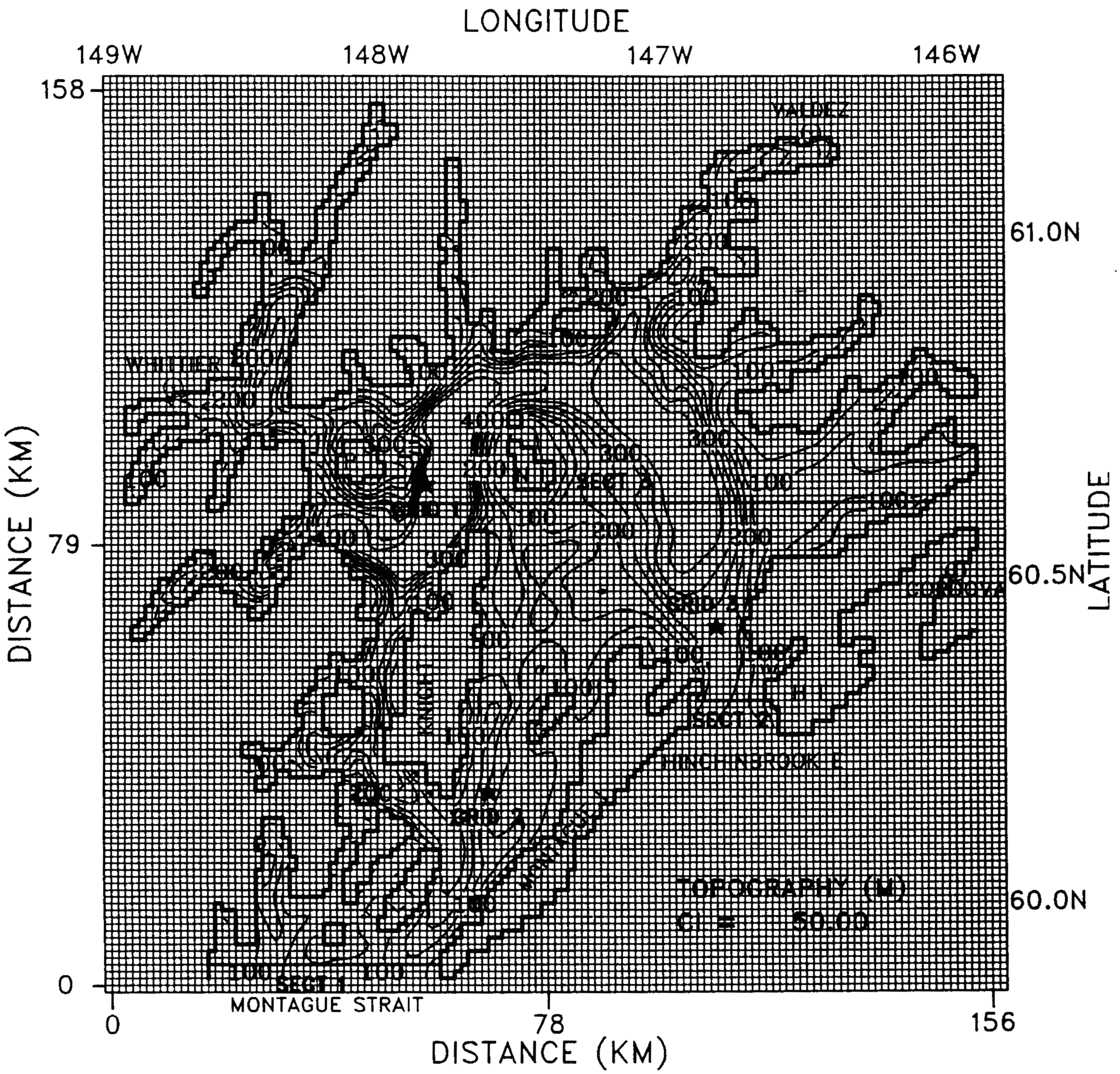
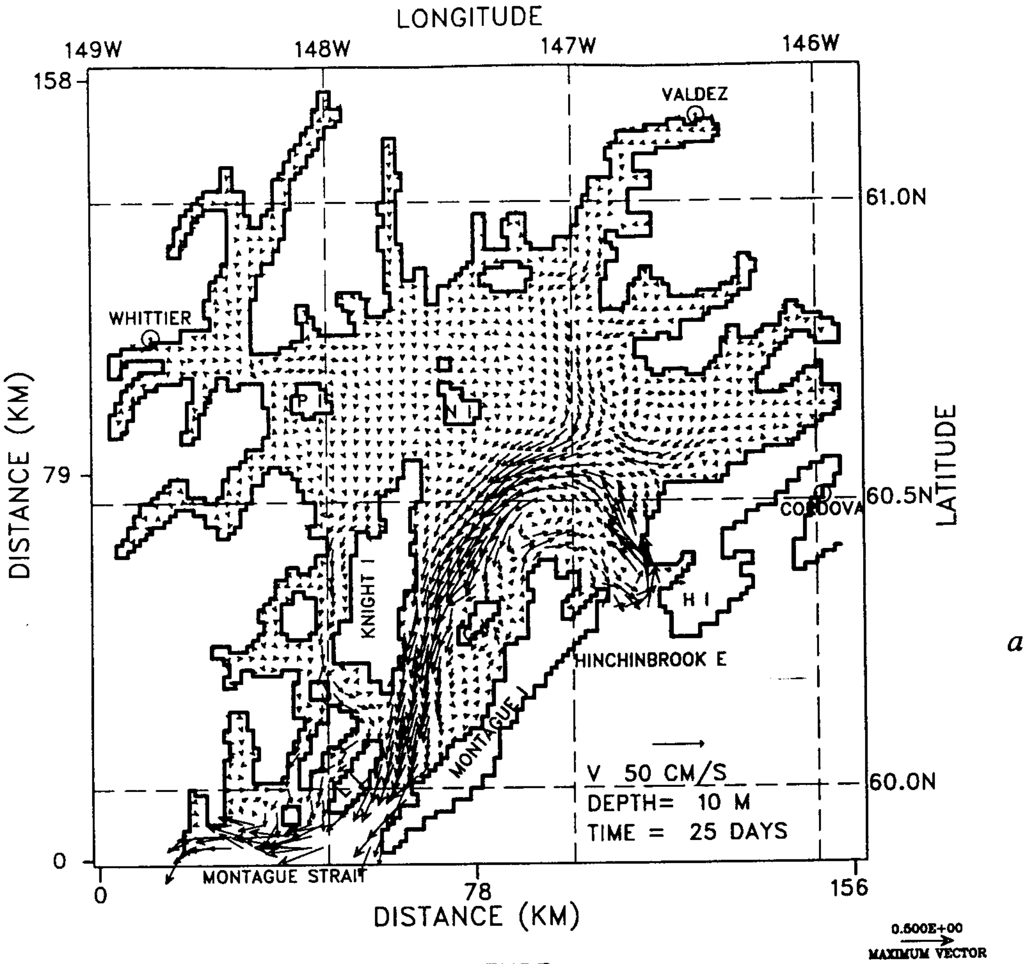


Figure 1 The PWS model domain with topography (depths in meters). The stars denote three grid points from which the time series of current velocity are taken.



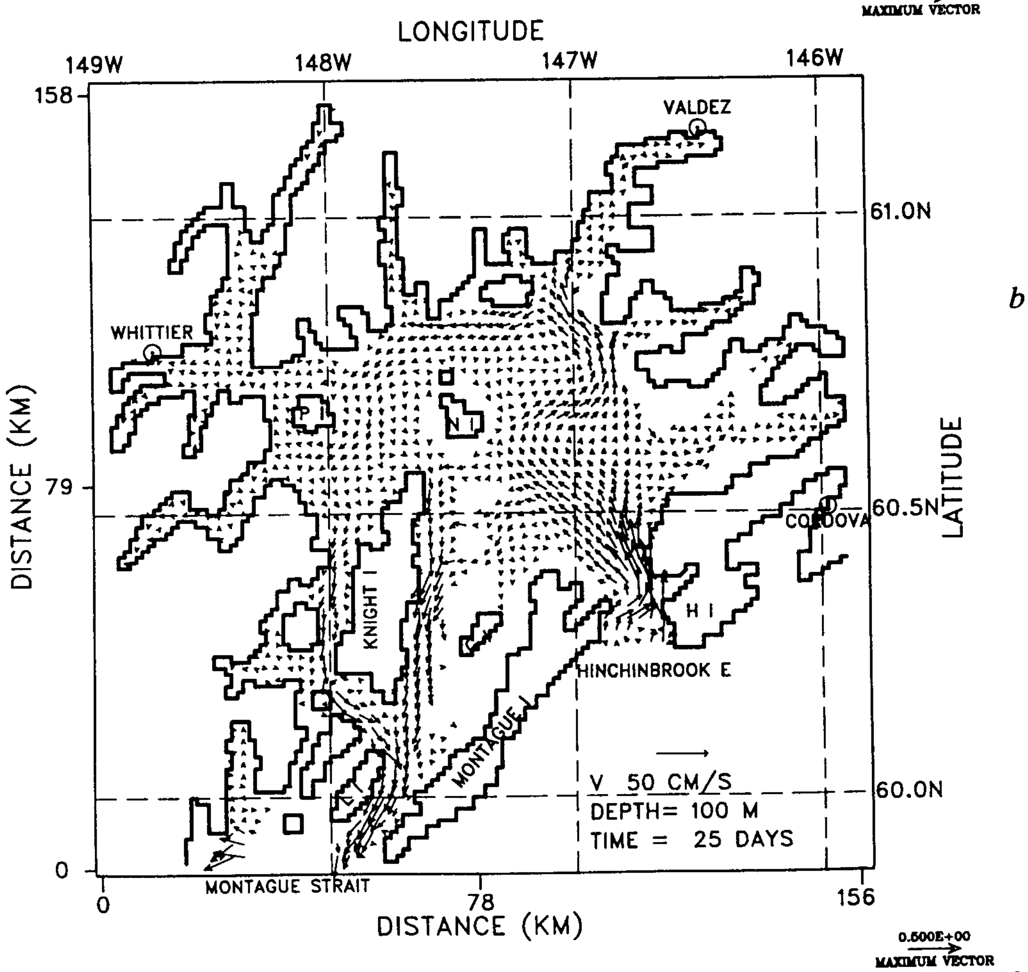


Figure 2 Mean velocity field under forcing of the M_2 tide and inflow/outflow of 0.3 Sv at 10m (a) and 100m (b)

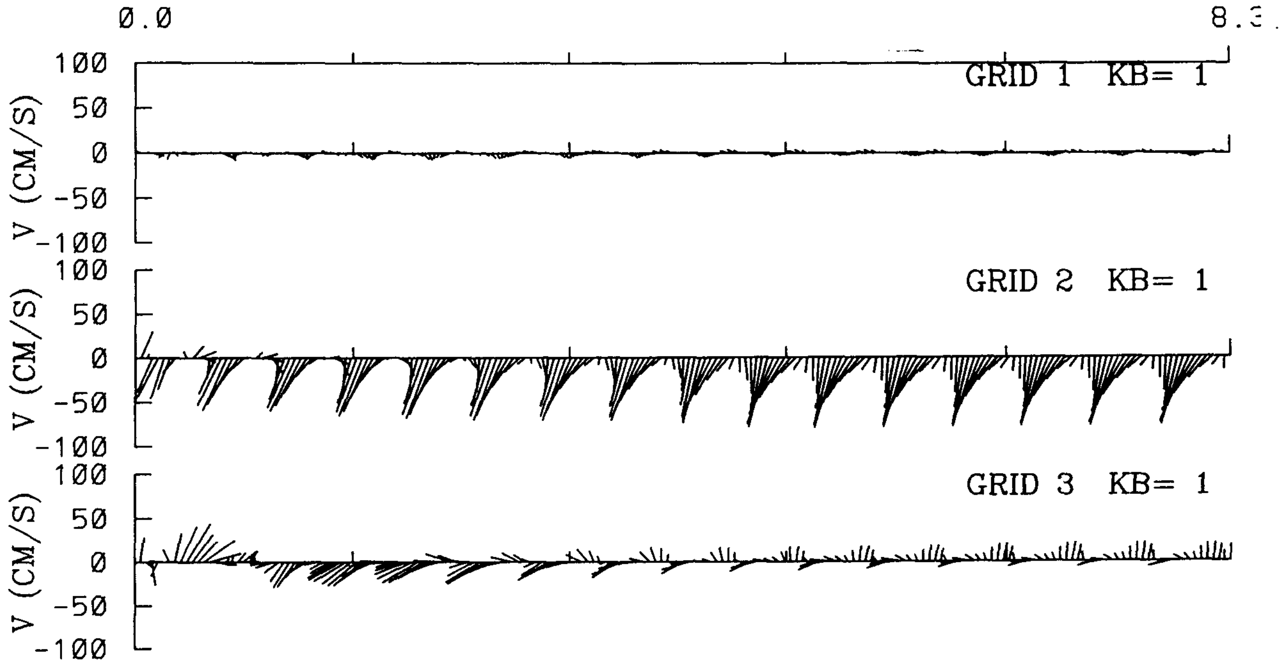


Figure 3 An eight-day time series of surface velocity vectors at grids 1, 2, and 3, as shown in Fig. 1.

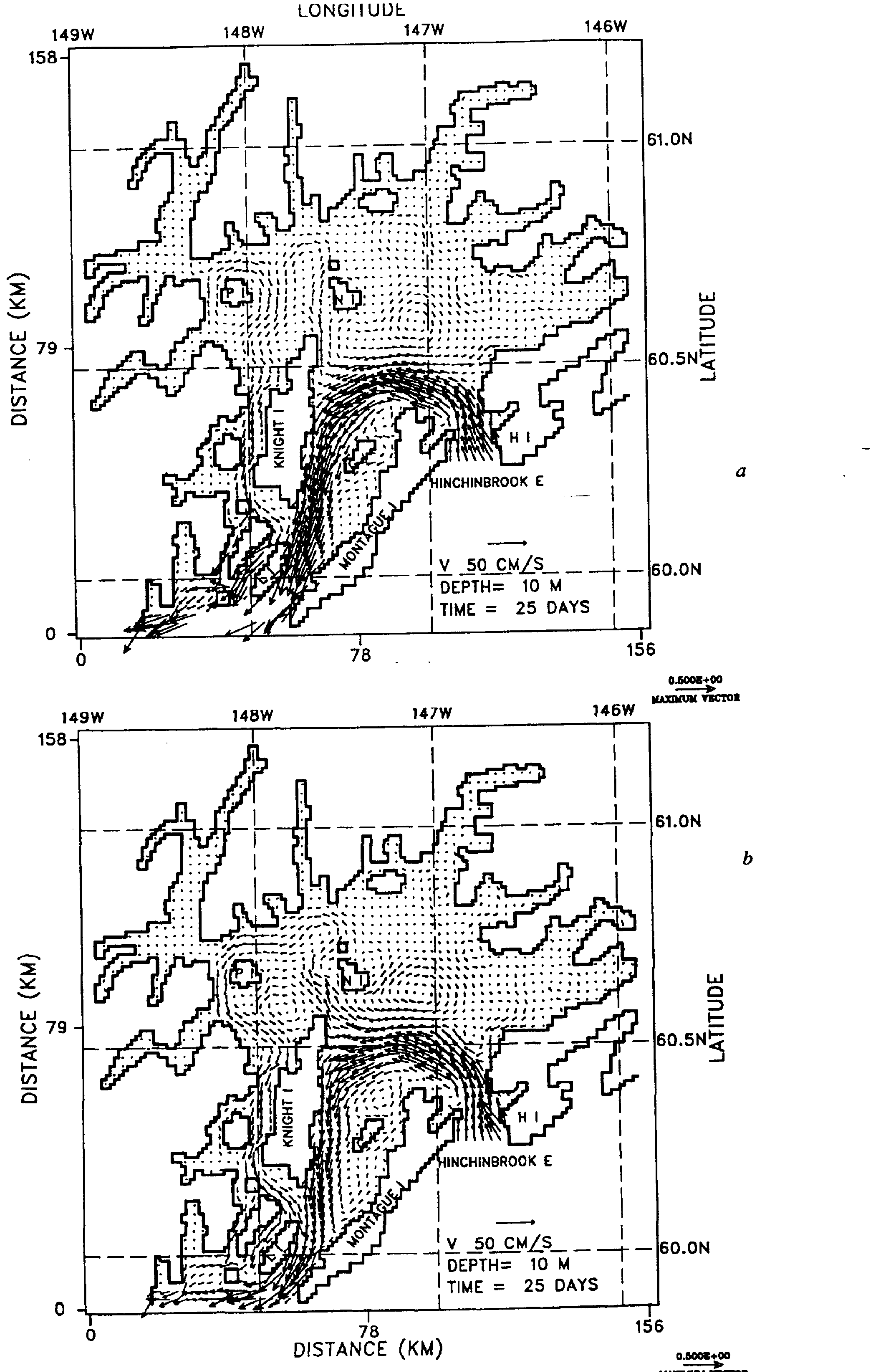


Figure 4 The 10m mean velocity field under forcing of the inflow/outflow of 0.3 Sv and eastward wind (a) and westward (b) wind of 7 ms^{-1}.

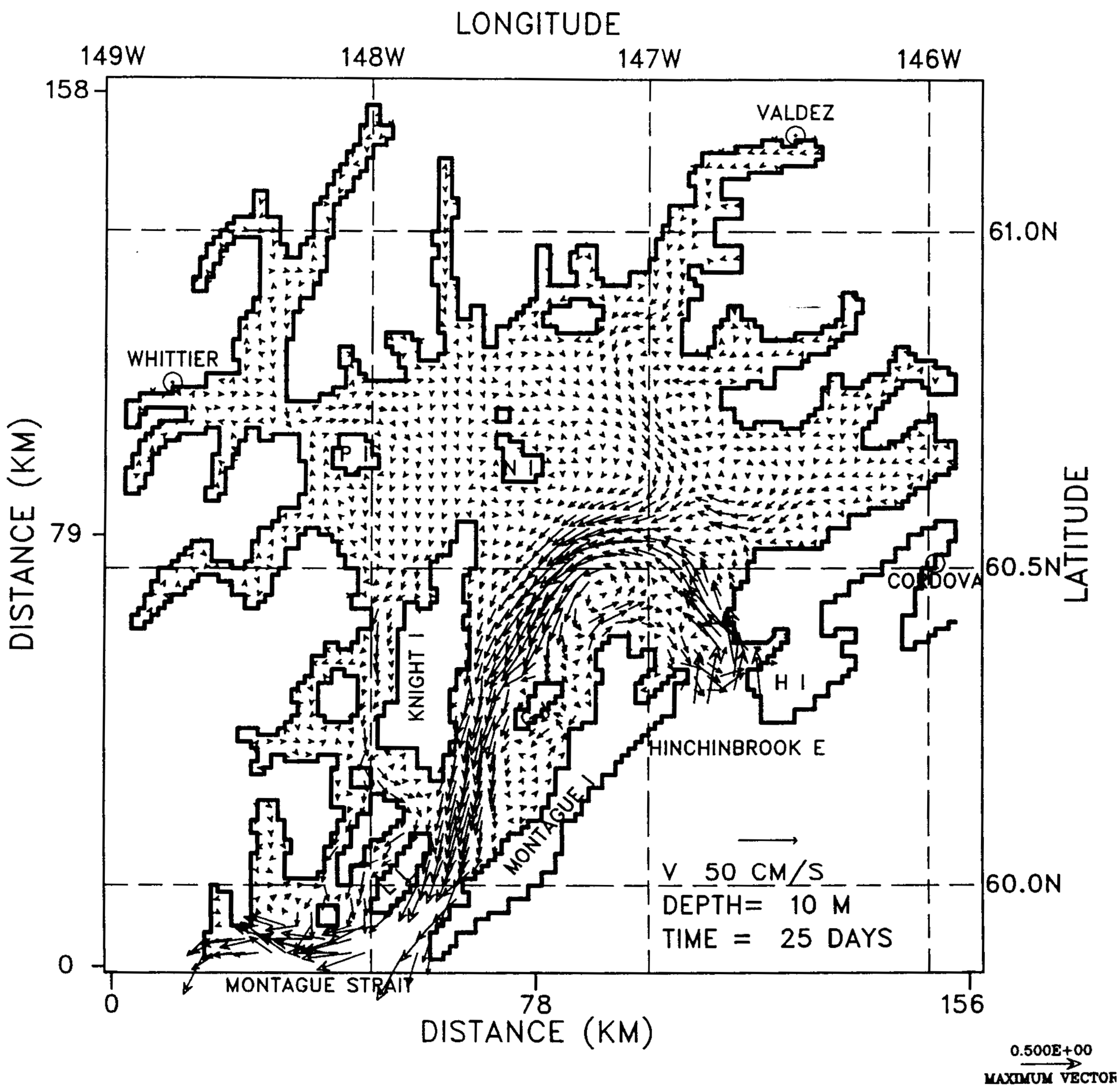
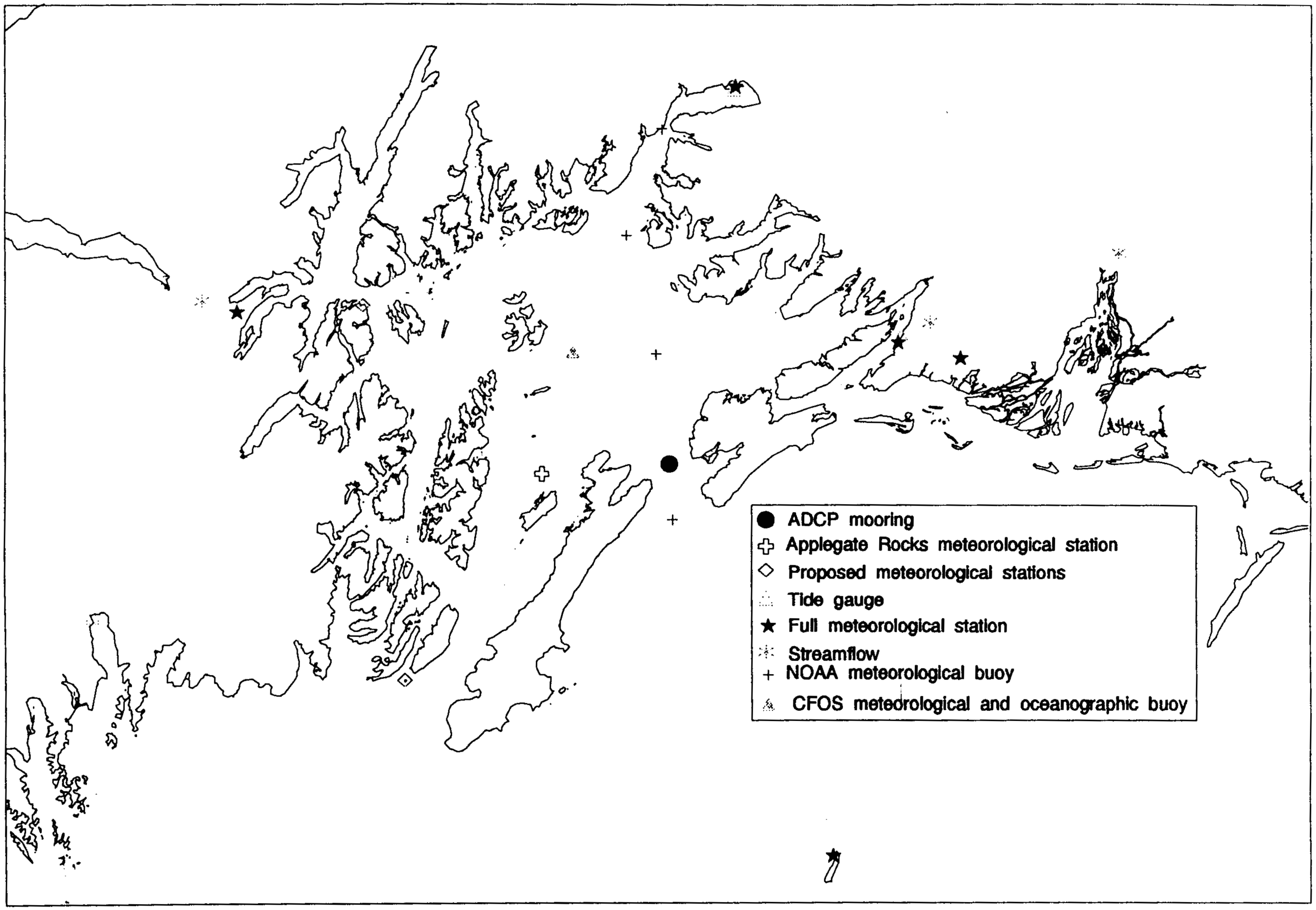


Figure 5 The 10m mean velocity field under forcing of the M_2 tide, inflow/outflow of 0.3 Sv, and eastward wind of 7 ms^{-1}.



Current and Proposed Climatological, Tidal, and Oceanographic Stations, Prince William Sound

Figure 6 The present proposed atmospheric, hydrologic, and oceanic observational stations.

Appendix 7

Finite Element Simulation of a Taxis Model for Population Interactions in 1D

FINITE ELEMENT SIMULATION OF A TAXIS MODEL FOR POPULATION INTERACTIONS IN 1D*

RICARDO H. NOCHETTO† AND SRIDHAR P. RAO‡

ABSTRACT. Predator prey interactions are central in the fields of ecology and biology. A taxis model, which incorporates fundamental processes such as foraging, reducing encounter with predators, and optimizing environmental conditions, is discretized with semi-implicit finite differences in time and mixed finite elements in space. The method exploits the special structure of advection, gradient of a taxis function, and yields a natural upwinding that works equally well in the entire range of advection and diffusion dominated flows. This is essential because either the flow character may vary with the trophic level at a given space-time or the advection/diffusion ratio may be subject to adjustment in the course of this study. The discrete problem is mass preserving, robust to disparate scales of diffusion and advection, and formally second order in space. It consists of a nonsymmetric tridiagonal matrix for each density, assembled from easily computable element contributions, which is solved via LU-decomposition at each time step.

1. Introduction. The taxis model consists of a bioenergetics model for growth coupled with a model for the dispersion of each population and the manner in which they coexist and interact. This yields a system of coupled nonlinear advection/diffusion PDEs

$$\partial_t u_i - \operatorname{div}\left(D_i \nabla u_i + \chi_i\, u_i \nabla \lambda_i(\mathbf{u})\right) = f_i \qquad \text{in } \Omega \tag{1}$$

for density u_i of the ith species and ODEs for size. Here Ω is a bounded domain in $\mathbf{R}^d$ with $d = 1,2,3$, constants $D_i, \chi_i > 0$ are to be adjusted so as to match measurements, and $\mathbf{u} = (u_i)_{i=1}^I$; preliminary details are contained in [7]. Function λ_i, so-called *taxis*, is responsible for movement and spatial distribution of organisms. Based on experimental, empirical, and theoretical considerations, taxis incorporates fundamental processes such as foraging (finding food), reducing the encounter with predators, and optimizing physical conditions (water temperature, light, salinity, etc). Prior related, but more restrictive, work includes models for bacteria chemotaxis by Keller and Segal and Kareiva and Odell [4,p.442;5,8].

After discretizing in time with semi-implicit backward differences and (uniform) time-step Δt

$$u_i^n - D_i \Delta t \operatorname{div}\left(\nabla u_i^n + \tfrac{\chi_i}{D_i} u_i^n \nabla \lambda_i(\mathbf{u}^{n-1})\right) = u_i^{n-1} + \Delta t f_i^{n-1} = g_i^{n-1}, \tag{2}$$

*This project is funded by the *Exxon Valdez* Oil Spill Trustee Council Restoration Project 95320

†Department of Mathematics, University of Maryland, College Park, MD 20742, USA
‡Prince William Sound Science Center, SEA Data Project, Cordova, AK 99574, USA

Typeset by AMS-TEX

the problem reduces to a space discretization of the system of uncoupled linear advection-diffusion PDE in (2). The Scharfetter-Gummel scheme is very popular to handle that in 1D [9], most notably for the advection dominated regime $D_i/\chi_i \ll 1$. Among the various attempts to generalize this method to 2D and 3D, we are interested in those due to Brezzi et al [2,3,6], because they uncover crucial upwinding and conservation properties by simply interpreting the 1D scheme as a combination of *exponential fitting* and *mixed* finite element methods [1,2,3,6]. Consider a single equation like (2) and set $\rho = u \ \exp(\lambda/\varepsilon)$ for $\varepsilon = D/\chi$. Then each PDE in (2) can be written equivalently as

$$\exp(-\lambda/\varepsilon)\rho - D\Delta t \ \mathrm{div}\left(\exp(-\lambda/\varepsilon)\nabla\rho\right) = D\Delta t g. \tag{3}$$

The idea is to discretize (3) with a method that accounts for rapid variations of $\exp(-\lambda/\varepsilon)$ and yield a positive definite M-matrix. For the method to be flux-preserving, that is to enforce continuity of the normal component of flux

$$\sigma = \nabla u + u\nabla(\lambda/\varepsilon) = \exp(-\lambda/\varepsilon)\nabla\rho \tag{4}$$

across interelement boundaries, mixed finite elements are adequate [1]. The method of [2,3,6] uses lowest order Raviart-Thomas mixed finite elements with Lagrange multipliers to relax the continuity constraint of σ. By static condensation, and proper scaling, the problem can be reduced to a *nonsymmetric* positive definite system for the (scaled) multipliers $\mathbf{w}$, which in turn approximate the original density u instead of ρ.

The rest of this report describes the method, the algorithm for local construction of matrices $\mathbf{M}_i$ and right-hand sides $\mathbf{G}_i$, and their assembly to produce $\mathbf{M}$ and $\mathbf{G}$ satisfying $\mathbf{M}\mathbf{w} = \mathbf{G}$.

2. Discretization. Since the discrete formulation is the same for every trophic level, from now on we focus on the *scalar* PDE (2). This second order elliptic self-adjoint PDE can be converted into a first order system by simply regarding both ρ and σ as independent variables as follows:

$$\sigma \ \exp(\psi) - \rho_x = 0 \tag{5}$$

$$\frac{\exp(-\psi)}{D\Delta t}\rho - \sigma_x = g, \tag{6}$$

where we have used the notation $\psi = \lambda/\varepsilon$. Equation (5) is just the constitutive relation between ρ and σ, whereas (6) coincides with (3). The advection dominated regime $\varepsilon \ll 1$ is of special interest.

Let $\mathcal{T}$ denote a partition of the spacial domain $[a, b]$ into intervals (or elements) $[x_i, x_{i+1}]$. Let τ and v be piecewise smooth function whose only discontinuities are located at the nodes x_i, and let τ vanish at a and b. Multiply (5) by τ, and (6) by v, and then integrate (5) by parts to arrive at

$$\sum_i \left(\int_{x_i}^{x_{i+1}} \exp(\psi)\sigma\tau + \int_{x_i}^{x_{i+1}} \rho\tau' + \rho(x_i)[\![\tau(x_i)]\!] \right) = 0 \tag{7}$$

$$\sum_i \int_{x_i}^{x_{i+1}} \left(\sigma_x v - \frac{\exp(-\psi)}{D\Delta t}\rho v \right) = -\sum_i \int_{x_i}^{x_{i+1}} g v, \tag{8}$$

where $[\![\tau(x_i)]\!] = \tau(x_i^+) - \tau(x_i^-)$ stands for the jump of τ across x_i. This is the so-called weak (or variational) formulation of the system (5-6). We stress that the natural boundary condition

$$\sigma = 0 \tag{9}$$

at a and b is implicit in the requirement that τ vanishes at a and b, because τ is an arbitrary member of the space of functions where we seek σ.

2.1 Mixed Finite Element Method. We have to choose appropriate discrete spaces $\mathcal{S}$ and $\mathcal{R}$ to approximate both σ and ρ. However $\mathcal{S}$ and $\mathcal{R}$ must be related to each other for the resulting scheme to be stable [1]. The simplest choice consists of discontinuous piecewise linear functions for $\mathcal{S}$, which vanish at a and b, and piecewise constant functions for $\mathcal{R}$. This choice, however, may not lead to a global M-matrix $\mathbf{M}$. Such a property is intimately related to oscillations, which may occur and should thus be avoided in the advection dominated regime. A simple modification based on [6], which preserves stability and prevents oscillations, introduces suitable piecewise quadratics in $\mathcal{S}$ depending on the flow direction in place of piecewise linears. The resulting scheme reads: seek $\sigma \in \mathcal{S}$, $\rho \in \mathcal{R}$, and $\omega \in \mathbf{R}^N$ (Lagrange multipliers) such that

$$\sum_i \left(\int_{x_i}^{x_{i+1}} \exp(\psi)\sigma\tau + \int_{x_i}^{x_{1+1}} \rho\tau' + \sum_i \omega_i [\![\tau(x_i)]\!] \right) = 0 \qquad \forall\, \tau \in \mathcal{S} \tag{10}$$

$$\sum_i \int_{x_i}^{x_{i+1}} \left(\sigma_x v - \frac{\exp(-\psi)}{D\Delta t} \rho v \right) = -\sum_i \int_{x_i}^{x_{i+1}} gv \qquad \forall\, v \in \mathcal{R} \tag{11}$$

$$\sum_i [\![\sigma(x_i)]\!] \mu_i = 0 \qquad \forall\, \mu = (\mu_i) \in \mathbf{R}^N. \tag{12}$$

Note that (12) enforces continuity of σ across nodes, namely $[\![\sigma(x_i)]\!] = 0$ for all i, as well as the vanishing of the boundary values $\sigma(a) = \sigma(b) = 0$. In matrix form, this system has the following structure

$$\begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{C} \\ \mathbf{B}^T & -\mathbf{D} & \mathbf{O} \\ \mathbf{C}^T & \mathbf{O} & \mathbf{O} \end{bmatrix} \cdot \begin{bmatrix} \sigma \\ \rho \\ \omega \end{bmatrix} = \begin{bmatrix} \mathbf{O} \\ \mathbf{F} \\ \mathbf{O} \end{bmatrix}. \tag{13}$$

This matrix is symmetric but indefinite. However $\mathbf{A}$ is symmetric block diagonal, and $\mathbf{D}$ is diagonal, thereby making elimination of σ and ρ by static condensation feasible at the element level. In fact, elimination of σ results in

$$\begin{bmatrix} \mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} & \mathbf{B}^T\mathbf{A}^{-1}\mathbf{C} \\ \mathbf{C}^T\mathbf{A}^{-1}\mathbf{B} & \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C} \end{bmatrix} \cdot \begin{bmatrix} \rho \\ \omega \end{bmatrix} = \begin{bmatrix} -\mathbf{F} \\ \mathbf{O} \end{bmatrix},$$

with $\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B}$ diagonal, and that of ρ yields the system $\mathbf{H}\omega = \mathbf{G}$ with

$$\mathbf{H} = \mathbf{C}^T\mathbf{A}^{-1}\mathbf{B}(\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} + \mathbf{D})^{-1}\mathbf{B}^T\mathbf{A}^{-1}\mathbf{C} - \mathbf{C}^T\mathbf{A}^{-1}\mathbf{C} \tag{14}$$

$$\mathbf{G} = -\mathbf{C}^T\mathbf{A}^{-1}\mathbf{B}(\mathbf{B}^T\mathbf{A}^{-1}\mathbf{B} + \mathbf{D})^{-1}\mathbf{F}. \tag{15}$$

Matrix **H** is a symmetric, positive definite M-matrix. The multipliers ω approximate the transformed variable ρ at nodes, which is obvious from (7) and (10). We can thus determine a nodal approximation for the density u by simply computing

$$w_i = \omega_i \ \exp(-\psi(x_i)). \tag{16}$$

In terms of the vector of scaled multipliers $\mathbf{w} = (w_i)$ the system reads $\mathbf{Mw} = \mathbf{G}$, with a global tridiagonal M-matrix **M** which is no longer symmetric. Matrix **M** can be assembled from element contributions and is insensitive to values of ε, the advection/diffusion ratio. This is explained below.

2.2 Element Computations. We define an average exponential $\bar{\psi}_i$ on the interval $[x_i, x_{i+1}]$ as follows ($h_i = x_{i+1} - x_i$):

$$\bar{\psi}_i = \log\Big(h_i^{-1} \int_{x_i}^{x_{i+1}} \exp(\psi(x))dx\Big).$$

The first equation in (13), namely $\mathbf{A}\sigma + \mathbf{B}\rho + \mathbf{C}\omega = \mathbf{O}$ can be restricted to the interval $[x_i, x_{i+1}]$. If we denote with $\mathbf{A}_i, \mathbf{B}_i$ and $\mathbf{C}_i$ the local matrices, the local system reads

$$\mathbf{A}_i \begin{bmatrix} \sigma_0^i \\ \sigma_1^i \end{bmatrix} + \mathbf{B}_i u_i + \mathbf{C}_i \begin{bmatrix} \omega_i \\ \omega_{i+1} \end{bmatrix} = \mathbf{O},$$

where

$$\mathbf{A}_i = h_i \ \exp(\bar{\psi}_i) \begin{bmatrix} 1 & 0 \\ 0 & \frac{2}{15} \end{bmatrix} \qquad \mathbf{B}_i = \begin{bmatrix} 0 \\ -1 \end{bmatrix}, \tag{17}$$

and

$$\begin{aligned} \psi(x_i) \geq \psi(x_{i+1}) \quad &\Longrightarrow \quad \mathbf{C}_i = \begin{bmatrix} 1 & -1 \\ 1 & 0 \end{bmatrix} \\ \psi(x_i) < \psi(x_{i+1}) \quad &\Longrightarrow \quad \mathbf{C}_i = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix}. \end{aligned} \tag{18}$$

The two different forms of matrix $\mathbf{C}_i$ correspond to the flow from left to right (first case) and the opposite situation. This is a natural built-in ***upwinding***.

The second equation in (13), namely $\mathbf{B}^T\sigma - \mathbf{D}\rho = \mathbf{F}$, when restricted to the interval $[x_i, x_{i+1}]$, can be written as

$$\mathbf{B}_i^T \begin{bmatrix} \sigma_0^i \\ \sigma_1^i \end{bmatrix} - \mathbf{D}_i\rho_i = F_i \qquad D_i = \frac{\exp(-\bar{\psi}_i)}{D\Delta t} h_i.$$

2.3 Assembly. The elimination process described above can be performed at the element level in that it is only the third equation in (13) that establishes a link between adjacent elements. We therefore produce an element matrix $\mathbf{H}_i \in \mathbf{R}^2 \times \mathbf{R}^2$ and vector $\mathbf{G}_i \in \mathbf{R}^2$ as follows:

$$\begin{aligned} \mathbf{H}_i &= \mathbf{C}_i^T\mathbf{A}_i^{-1}\mathbf{B}_i(\mathbf{B}_i^T\mathbf{A}_i^{-1}\mathbf{B}_i + \mathbf{D}_i)^{-1}\mathbf{B}_i^T\mathbf{A}_i^{-1}\mathbf{C}_i - \mathbf{C}_i^T\mathbf{A}_i^{-1}\mathbf{C}_i \\ \mathbf{G}_i &= -\mathbf{C}_i^T\mathbf{A}_i^{-1}\mathbf{B}_i(\mathbf{B}_i^T\mathbf{A}_i^{-1}\mathbf{B}_i + \mathbf{D}_i)^{-1}\mathbf{F}_i. \end{aligned}$$

Matrix $\mathbf{H}_i$ is a local stiffness matrix and $\mathbf{G}_i$ represents a local load vector. They have the following expressions depending on the flow direction:

$$\psi(x_i) \geq \psi(x_{i+1}) \Longrightarrow \mathbf{H}_i = \frac{1}{h_i \exp(\bar{\psi}_i)} \begin{bmatrix} \alpha_i & -1 \\ -1 & 1 \end{bmatrix} \quad \mathbf{G}_i = F_i\beta_i \begin{bmatrix} -1 \\ 0 \end{bmatrix}$$

$$\psi(x_i) < \psi(x_{i+1}) \Longrightarrow \mathbf{H}_i = \frac{1}{h_i \exp(\bar{\psi}_i)} \begin{bmatrix} 1 & -1 \\ -1 & \alpha_i \end{bmatrix} \quad \mathbf{G}_i = F_i\beta_i \begin{bmatrix} 0 \\ -1 \end{bmatrix}$$

where

$$\alpha_i = \frac{1 + \frac{17h_i^2}{15D\Delta t}}{1 + \frac{2h_i^2}{15D\Delta t}} \qquad \beta_i = \frac{1}{1 + \frac{2h_i^2}{15D\Delta t}}.$$

An obvious difficulty with these expressions is that they may easily yield over or underflow. If we replace the unknowns ω_i, which approximate $\rho(x_i)$, by the scaled variable w_i given by (16), then we get the following local quantities

$$\psi(x_i) \geq \psi(x_{i+1}) \Longrightarrow \mathbf{M}_i = \frac{1}{h_i} \begin{bmatrix} \exp(\psi_i - \bar{\psi}_i)\alpha_i & -\exp(\psi_{i+1} - \bar{\psi}_i) \\ -\exp(\psi_i - \bar{\psi}_i) & \exp(\psi_{i+1} - \bar{\psi}_i) \end{bmatrix} \tag{19}$$

$$\psi(x_i) < \psi(x_{i+1}) \Longrightarrow \mathbf{M}_i = \frac{1}{h_i} \begin{bmatrix} \exp(\psi_i - \bar{\psi}_i) & -\exp(\psi_{i+1} - \bar{\psi}_i) \\ -\exp(\psi_i - \bar{\psi}_i) & \exp(\psi_{i+1} - \bar{\psi}_i)\alpha_i \end{bmatrix}. \tag{20}$$

The global stiffness matrix $\mathbf{M}$ is obtained by assembling N local stiffness matrices $\mathbf{M}_i$ (N is the number of elements), which combine to form a tridiagonal matrix with negative off diagonals and a positive main diagonal (M-matrix).

2.4 Exponential Scaling. We now show that exponentials occurring in (19) and (20) can be computed safely regardless of the size of $\varepsilon = D/\chi$. We first observe that the taxis function $\lambda = \varepsilon\psi$ depends on previously computed densities, which are known at the nodes x_i. It is thus reasonable to think of ψ as a piecewise linear interpolant of the true scaled taxis function, thereby coinciding with it at x_i.

Computation of the scaled exponentials is dictated by the flow direction. We set

$$\Delta\psi_i = \psi(x_{i+1}) - \psi(x_i).$$

Hence

$$\Delta\psi_i \leq 0 \Longrightarrow \exp(\bar{\psi}_i) = \frac{\varepsilon}{\Delta\psi_i} \exp(\frac{\psi(x_i)}{\varepsilon})(\exp(\frac{\Delta\psi_i}{\varepsilon}) - 1)$$

$$\Delta\psi^i > 0 \Longrightarrow \exp(\bar{\psi}_i) = \frac{\varepsilon}{\Delta\psi_i} \exp(\frac{\psi(x_i)}{\varepsilon})(1 - \exp(\frac{-\Delta\psi_i}{\varepsilon}))$$

With these expressions at hand, we can now compute the exponentials in (19) and (20). In fact, if $\Delta\psi_i \leq 0$ we get

$$\exp(\psi_i - \bar{\psi}_i) = \frac{\Delta\psi_i}{\epsilon} \frac{1}{\exp(\frac{\Delta\psi_i}{\epsilon}) - 1}$$

$$\exp(\psi_{i+1} - \bar{\psi}_i) = \frac{\Delta\psi_i}{\epsilon} \exp(\frac{\Delta\psi_i}{\epsilon}) \frac{1}{\exp(\frac{\Delta\psi_i}{\epsilon}) - 1}.$$

Otherwise we obtain

$$\exp(\psi_i - \bar{\psi}_i) = \frac{\Delta\psi_i}{\epsilon}\exp(\frac{-\Delta\psi_i}{\epsilon})\frac{1}{1-\exp(\frac{-\Delta\psi_i}{\epsilon})}$$

$$\exp(\psi_{i+1} - \bar{\psi}_i) = \frac{\Delta\psi_i}{\epsilon}\frac{1}{1-\exp(-\frac{\Delta\psi_i}{\epsilon})}.$$

This method of computing the elements of $\mathbf{M}_i$ eliminates the concern of potential overflow, and shows its robustness with respect to ε.

3, Relation to Finite Differences. For uniform meshes $\mathcal{T}$, namely $h = h_i$ constant, the above mixed finite element method can be identified with a suitable upwind finite difference method. This interpretation is revealing in understanding the remarkable approximation and conservation properties of the proposed scheme, which is closely related to the Scharfetter-Gummel scheme for semiconductor device simulation [9].

4. Conclusions. The above approach exploits the special structure of advection due to the taxis, or loss function, via exponential fitting and mixed finite element methods. In doing so, the resulting discrete formulation accounts for both diffusion and advection dominated regimes through the use of inherent upwinding. This is essential in light of the expected high variability of population densities in complementary regions of the space-time domain, which could easily make both extreme regimes coexist for different trophic levels. This extra flexibility suits itself well to the needs of biological and ecological modeling, and is further enhanced by the possibility of using mesh refinement/coarsening strategies, a posteriori error estimation and adaptivity. All these features extend to multidimensional situations.

REFERENCES

1. F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer, New York, 1991.
2. F. Brezzi, L.D. Marini, and P. Pietra, *Méthodes des élément finis mxtes et schéma de Scharfetter-Gummel*, C.R. Acad. Sci. Paris Sér. I **305** (1987), 599–604.
3. F. Brezzi, L.D. Marini, and P. Pietra, *Two-dimensional exponential fitting and applications to drift-diffusion models*, SIAM J. Numer.Anal. **26** (1989), 1342–1355.
4. L. Edelstein-Keshet, *Mathematical Models in Biology*, The Random House/Birkhäuser, Mathematical Series, 1988.
5. P. Kareiva and G. Odell, *Swarms of predators exhibit "preytaxis" if individual predators use are-restricted search*, Amer. Naturalist **130** (1987), 233–270.
6. L.D. Marini and P. Pietra, *New mixed finite element schemes for current continuity equations*, COMPEL **9** (1990), 257–268.
7. D.M. Mason and E.V. Patrick., *A model for the space-time dependence of feeding for pelagic fish population*, Trans. American Fisheries Society **122** (1993.), 884–901.
8. J.D. Murray, *Mathematical Biology*, Springer-Verlag, New York, 1989.
9. D. Scharfetter and H. Gummel, *Large-signal analysis of a silicon Read diode oscillator*, IEEE Trans. Electronic devices **ED-16** (1969), 64–77.

Appendix 8

Alewife
A System for Modeling Population Interactions

Alewife

A System for Modeling Population Interactions

Sridhar Rao

Abstract

Alewife is the initial attempt at the design of a language oriented to the analysis of ecosystem behavior - in particular those elements dealing with population interactions.

The initial drive to design such a "language" came from the need to develop an adaptable system of parameter modification for the use in simulations on predator-prey interaction models. This ability to modify parameters quickly became the need to add new parameters as well.

With this adaptability becoming a strong need for the rigourous analysis of models and their relationships, a new approach was devised that was aimed at minimizing the coding and time needed to implement new model characteristics. Further evolution of alewife comes from its continued use and implementation in the real world.

This effort was motivated by the modeling problems being addressed in the Sound Ecosystem Assessment Project for Prince William Sound. The solution presented continues to evolve through this implementation with space and time dependent models for the physical and biological factors affecting the survivorship of juvenile fish. This project is funded by the *Exxon Valdez* Oil Spill Trustee Council Restoration Project 95320.

1.0 Introduction

The modeling process contains many different levels of development. Often the earlier stages are mathematically intensive and also very abstract in relation to the model as a whole. Formulation of the mathematical framework forms the first major step in modeling a system. Once a numerical analysis is complete, the parameters and relationships concerning the system can be applied. The integral formulation, and the use of variational methods to design a solution, is a broader and more general step that contains the actual model relationships and its uniqueness with regards to application.

A process can be developed in which a numerical solution (or formulation) can be designed and used with a set of model relationships. In doing so, different relationships can be added or modified without altering the underlying formulation. This will help to speed up the modeling process by eliminating the need to re-formulate a numerical analysis and help the modeler focus on aspects of the system to be studied.

The process can be expressed as shown in figure 1. The formulation and the model relationship form two parts of the mathematical solution. System parameters are then fed into this solution which results in data that is to be used for analysis. The formulation is the more abstract of the two and in most cases can be generalized for a certain set of modeling criteria. Model relationships are of primary interest to the scientist dealing directly with the system of study.

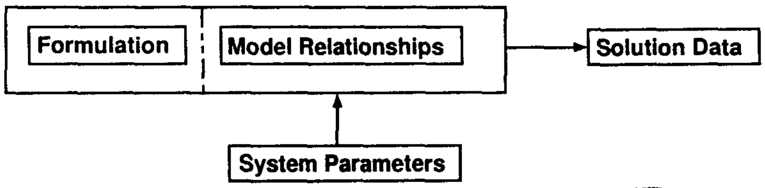


Figure 1: Components of The Modeling Process

Alewife separates these two parts of the solution and creates a standard set of formulations that can be linked with a set of model relationships. This newly linked solution can be used in conjunction with the model parameters to develop data for analysis.

For example, if one wishes to model a system of species using a finite element analysis, it can be separated into two problems: the discretized numerical analysis (formulation), and the population dynamics (model relationships). Using the approach of *alewife* one need only be concerned with the model relationships and how to integrate them into an f.e.m. solution. The numerical formulation will be encapsulated and ready for use from a library of formulations.

This example case formed the groundwork for *alewife* in work done on predator-prey interactions [1]. New attributes were constantly being added to the model relationships causing a large degree of recompile time and modifications to the source code. By separating the formulation and the relationships the addition of new dynamics to the model becomes a speedier process.

2.0 Elements

Alewife consists of various elements that are used to create a working solution called an *engine*. This engine is a compiled executable that takes in parameters and outputs data for the solution. Two main elements form the alewife engine: the *solver* and the *model relationships*.

The solver is a relocatable object file that is stored in a library of solvers and it represents the encapsulated numerical analysis. In the current distribution there is one solver which is a 1 dimensional formulation using a finite element analysis. The second section of the engine consists of the model relationships. These are scripts written by the modeler that are linked with the solver to produce an executable "engine" which is used to generate data.

3.0 Underlying Relationships

For the solver in this distribution the primary relationship that binds all these elements together is a basic advection diffusion form for the space-time behavior of multiple interacting population densities:

$$\frac{du}{dt} = f + D\Delta u + \chi\nabla \cdot u(\nabla\lambda). \tag{1}$$

In the above relationship u, f, and λ are vector valued functions and χ and D are constants. A vector component exists for each of the interacting trophic levels. The primary function of the solver is to solve for the function u which represents densities for populations.

The function λ forms the taxis relationship which is represented by the various model relationships. The forcing term f is also a model relationship and must be coded. Hence, λ and f are the two links provided by the solver to the code written by the modeler. Once these relationships are expressed, they can be joined with the solver to form an engine.

4.0 Modeling Fundamentals

To create a new engine requires a fundamental set of relationships, λ and f, which are to be designed by the modeler as mentioned. These two functions represent the temporal and spatial variations in the interactions between various populations. The function λ expresses the main biological factors and their influence on the densities in the current space-time iteration of the solution.

Due to the solution formulated in the numerical analysis, λ is actually a scaled value when returned. This scaling is related to the constants χ and D. The pre scaled function is the one that is actually used to produce biological results and is called ψ. The function f represents the forcing functions and must also be designated by the modeler.

Alewife "code" is written in a script which is interpreted by the compiler *aw*. *aw* then links the script with the solver and forms the engine which is used in conjunction with a set of model parameters and space time data to produce population dynamics data. ψ and f therefore form the two main modules that must be written in order to link with the solver. This modular approach can be taken to more levels below the top most ψ and f modules to form a series of relationships which allow for a high degree of model complexity.

4.1 Components

Now that we know what functions must be modeled in order to generate an engine, what components are at our disposal to model with? Since *Alewife* is meant primarily for the modeling of biological population interactions, a set of "basic variables" are defined for use:

```
Web
density
psi
tl
rhs
```

Web is the structure that contains variables and data related to the population densities. It contains the variables density and psi. The variable ***density*** refers to the density of the population that is currently being calculated and psi is the variable that contains the data that is produced from the ψ definitions. Since these are both contained in the structure of Web they are addressed accordingly:

```
Web.density Web.psi
```

The definition of the use of these variables is not yet complete. The values for density and psi are stored in the structure based on time (the current position in time n - and the previous time position n-1, etc.) and space (the solver described here uses a one-dimensional approach). Furthermore the structure Web itself is addressed based on the population or trophic level. Hence, the complete form for using these variables is:

```
Web[tl].density[i][0] Web[tl].density[i][1]

Web[tl].psi[i][0] Web[tl].psi[i][1]
```

The second counter deteremines how far back in time it is being indexed, with lower index values representing previous timesteps. Hence, timestep 0 is prior to 1. The variable tl is also an internal variable that is predifined and managed by *aw*. The variable *i* is also a reserved internal counter for use to represent the indexing through space. Since definitions of Web are only done through space, these definitons will always use i as a counter.

5.0 Coding Method and Example

To describe the method of coding and the syntax, it would be beneficial to begin with a simple example. The following piece of code will intialize all of the spatial elements of the web's density to 0.0 in both time steps (all code written in *alewife* follows a C like syntax except for when a module is declared or ended):

```
!!Module
init
Web[tl].density[i][0] = 0.0; Web[tl].density[i][1] = 0.0;
!!End
```

The first line of this code defines a new module. All *alewife* code consists of modules that are linked together in the processing stage. The next line after a module declare is the module's name, in this case *init*. Following the title is the actual logic for the module. These two lines set the values for the web's density elements for the trophic level tl to zero for all space. This code is automatically handled to iterate over all space and trophic levels. Specifically, there are five required modules for any *alewife* code:

```
init psi rhs post final
```

All of these modules are automatically iterated over space and trophic levels and time (if needed). The trophic levels that are handled can be selected from a data file that is fed to the compiled engine in the run time stage. The module psi is the module that encompasses all the biological relationships that are to be modeled. In essence it is psi that is the most crucial of all the modules. The next piece of code displays how all of the modules are tied together and incorporated into a file:

```
!!Components
lambda
exone
!!End
!!Module
lambda
lambda = (double) i * (1.0) / (N-1);
!!End
!!Module
exone
exone = exp(-1.0 * xx[i]);
!!End
!!Module
rhs
rhs = 0.0;
!!End
!!Module
psi
Web[tl].psi[i][0] = Web[tl].psi[i][1]; Web[tl].psi[i][1] =
Web[tl].psi[i][2];
call lambda print lambda Web[tl].psi[i][2] = lambda;
!!End
```

```
!!Module
init
call exone Web[tl].density[i][0] = exone; Web[tl].density[i][1]
= 0.0;
call lambda Web[tl].psi[i][0] = lambda; Web[tl].psi[i][1]
= lambda; Web[tl].psi[i][2] = lambda;
!!End
!!Module post
!!End
!!Module final
!!End
!!EndEnd
```

The above example shows a host of new terms and functions. Firstly, notice that the five required modules are all defined with the syntax:

```
!!Module
module name

!!End
```

However, prior to the first module declaration is a decalration *!!Components*. This declare states that the following list are the names of new modules that are to be declared and used later in a module declaration. Any new nonstandard module should be decalred here. These components are reserved for use in the final calculation of the value of psi, which is used in the solver to produce a solution when executed.

It is first crucial to recognize the main required modules and their definitions. These are done in the later section of the code. Looking at these definitions we see that they are assigning certain values to variables such as lambda or exone.

The module rhs is used to define the forcing terms previously mentioned as f. In this case there is no contribution here. The module *post* is used to do any post spatial calculations needed. This can be useful when certain trophic level relationships are dependent upon the newly attained solution in space. The module *final* is used for any calculations needed after the time iteration is complete (when all the trophic levels have been solved for in time).

When a new module is to be defined by the modeler, it is declared in the first section called the components section. Here, any new modules are named and then their logic will follow somewhere in the code. The fundamental modules, those modules which are required, are *not to be* defined in the components section.

The module lambda is first named in the components section and then the logic is written in the module definition for it. This example also shows lambda using other internally handled variables. When an assignment such as the follwing is made, it calls the module *lambda* and then does the proper assignment thereby returning the value to be assigned:

```
Web[tl].psi[i][1] = lambda;
```

Module definitions can be combined in any way to form a relationship for assignment purposes, hence the following is legal as long as all the modules are defined properly:

```
Web[tl].density[i][0] = foo + bar * (moo / goo);
```

This allows for the rapid modification of model definitons and behavior for the incorporation of new features. All of the above code is entered into the file *defpsi.x* which is one of two files that the *aw* processor takes in. The other is *parameters.x*:

```
!!Module
params

param tau1
param tau2
param tau3
param tau4

set tau4 = webparams->tau1 * (webparams->tau2 + webparams->tau3);
!!End
!!EndEnd
```

Again this file is similar in design in that it follows the standard module definition. The purpose of the parameters file is to define a series of parameters that can be used by any of the modules in defpsi for calculation. In this example four parameters are defined for use: *tau1, tau2, tau3, tau4*. These are all declared by the *params* command as shown. Parameters can *only* be defined within the params module. The last line of this module uses the set command which sets a particular parameter equal to other parameter values. This command is only valid in the params module as well. In this example tau4 is set to the sum of the other tau values times the first tau value. The addressing of parameters that have values is done by the notation:

webparams->*parameter name*

The definition of parameters is the most specialized case in the writing of *alewife* code. This definiton merely sets up the proper references to the engine to take in parameters from a data file that contains values for them. Those familiar with C will notice this as the standard pointer usage in structure elements.

This parameter file is fed to the engine upon execution. The data within it must match the assignment order of parameters in the parameters.x file, or else inconsistencies will arise in the calculated results and potential segmentation faults in the executable.

There are two data files, the space time data file and the web parameters data file. The space time data is data pertaining specifically to the engine object used as the solver. This object reads in these space time data and sets certain variables for use as indeces and such. In the examples above, a one dimensional finite element solver was used, and the indicies such as i and N are specific to that solver. When a solver is designed, a set of parameter definitions will be stated so that the proper files can be created. When looking at the engine specification one will notice the order and structure of the predefined parameters neccesary to design a new model.

6.0 Scope and Definition of Fundamental Variables

The variables that are defined (and hence reserved words in *alewife*) internally are called fundamental variables. These must have some definition in order for the model to make sense. Fundamental variables are in two categories: biological, and mathematical.

The mathematical fundamentals are specified by the type of engine one uses. The engine used here implements a finite element method in one dimension (for this solver the variables deal mainly with space time parameters). These fundamental variables are to be specified in the *engine specification* (the specification for this engine is located in the source distribution).

These variables are always linked to a solution of the biological side (in the case of this example, it is *psi*). Psi is defined with a given set of relationships (defined by the modeler) and combined into the vector that is fed to the solver for the current time step. The engine does all the handling of storage and recall of previous values and such.

The ultimate goal here is to have the smallest need to adjust and set engine internal variables, and to allow for the maximum effort of defining and coding to go to the biological modeling (although the design of this setup is general enough to potentially have other applications).

In review of the above example the fundamental variables (and their associated modules) are all automatically defined over a given space and time region. Hence when the statement:

```
call lambda
print lambda
Web[tl].psi[i][2] = lambda;
```

Is performed, the value of lambda is distributed over the all i (space) and will automatically be handled through time steps.

7.0 Engine Objects

Engine objects are the heart of the modeling process. These objects are precompiled and ready to link with the biological definitions designed by the modeler. The link between the solver and the defintion files (i.e. defpsi.x) are defined by the solver being used. The one addressed here is a 1d solver that uses finite elements to determine a solution to the advection diffusion relationship from section one.

The engine objects are located in the default library for aw, usually /usr/local/lib/aw. Currently there is only one available solver hence, *aw* does not have a solver choosing option.

The compiled and executable engine will also have a sepcified method of execution dealing with how it ingests data files. In the case of the current one dimensional solver it is done as such:

`kenobi 21:` **engine** *[webparams datafile] [spacetime datafile] [output datafile]*

The first two arguments specify the web and spacetime data respectively, and the final argument is the name of the data file to create for the solutions and any other values that require output. The engine for the 1d case defaults the following data to the ***output datafile***:

`Density` *time step trophic level density*

.

.

.

`Integral` *integral*

The number of density values printed per trophic levels is specified by the space time parameter pertaining to the number of spatial steps taken (for this solver it is the variable N).

7.1 Compiling an Engine

To actually "compile" an engine the files *defpsi.x parameters.x* and ***engine.o*** must exist. By typing *aw* the processor looks for these *.x files and builds a set of *.c files and a makefile that is then passed through gcc to produce the executable engine called *engine*.

The resulting engine can be executed by typing its name and as its arguments: the web parameter data, space time data, and a datafile name for the saving of data to (in that order).

> **kenobi 22: engine** *[web parameter filename] [space time parameter filename] [datafile]*

Figure 2 shows the process of taking multiple files and an engine object, and linking them through *aw* to form an executable engine. The final engine is then fed parameters to produce data that can be used for various applications such as visualization, and comparisons with real world data.

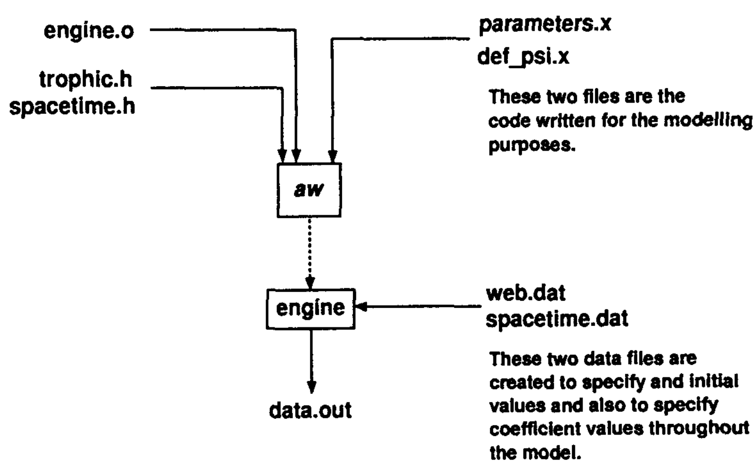


Figure 2: Process for Creating an Engine

7.2 Explanation of Engine Specification

The engine specification consists of four sections, the title, fundamental modules, definitions of structures and variables, and referencing.

The fundamental modules are those modules that *must* be defined when code is written to use this engine. Whatever is done within that module is up to the modeler.

The defined structures and variables are those which are available for the modeler's use in any of the modules. These are listed in their type form (i.e. double precision or integer etc.). The method in which these variables are used is explained in the referencing section.

The engine specification is the key to allowing the modeler knowledge on how the engine can be used in relation to any biological specifications he/she wishss to implement.

8.0 Distribution

The distribution is split into various subdirectories:

```
kenobi 23: ls
Makefile.bsd        docs/          gui/            src/
Makefile.solaris    examples/      interfaces/     testcase/
```

The directory aw contains the source for the compiler *aw*. Parameters is where some template parameter files for the space time and web data are for the 1d solver. These can be used as example building block files to create one's own parameter files. The directory testcase contains various examples for testing out the compiled binaries. Descriptions accompany the various examples. The docs directory is where documentation and help files in relation to this project are (this document should be found there). Interface is the location for the tcl/tk front end interface called the builder. This interface allows for a G.U.I. approach to making and running an engine (help files are located within the interface and in the docs directory). Finally the solver directory contains the source code for the one dimensional finite element solver that is compiled to build the *enigne.o* that is used for the linking process. The makefile should be modified to suit the needs of the installation and the system one is using and a make should be run. To install stuff in system wide accessible locations write privilages to /usr must exist for the user installing the binaries and other files. The examples directory contains expanded examples for use in observing diffusion advection behavior of the engine.

9.0 Conclusion

Alewife is a system designed to help speed up the modeling process for population interactions. By having a set of formulated solvers, one can easily select from this library and link to a modeled scenario. With the process of formulation taken care of, the modeling becomes more concentrated on the aspects of the system to be observed. When focus is shifted to the application, a more thoughtful approach can be given to the design of the underlying model relationships.

References

[1] Mason, D.M., and E.V. Patrick. *A model for the space-time dependence of feeding for pelagic fish population*, Transactions of the American Fisheries Society 122(5):884-901, 1993.

Appendix 9

Diffusion-Taxis Model for Distribution, Mortality and Growth of Multiple Interacting Populations

Diffusion-taxis Model for Distribution, Mortality, and Growth of Multiple Interacting Populations:

The Submodels

Vincent Patrick Doran M. Mason

Abstract

The purpose of this note is to describe the current collection of submodels used in the initial trials of the diffusion-taxis model. A second purpose it to begin to develop a suitable notation.

Construction of the loss function

A loss function is used to model the methods by which physical conditions and biological processes within and between individuals determine the time evolution of the spatial distribution of populations. Specifically, the loss function consists of a choice of how to combine sensory responses of a standardized individual into a single scalar that reflects the net relative dislike of that individual for each point in space and time. Figure 1 is a flow chart for the construction of the loss function currently used to describe juvenile fish in western Prince William Sound along with the other fish populations interacting with those juveniles. It shows each of the physical and biological variables used, with the latter including population density and the size of individuals. The various sources for these space-time dependent variables is indicated. Figure 1 indicates that the loss function is used in a set of partial differential equations. These are described next.

The spatial distribution of freely swimming organisms (nekton) is modelled using equations of the form

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= \operatorname{div}(D\operatorname{grad}u + \chi u \operatorname{grad}\lambda) + f \quad && in\ \Omega \\
u &= g && on\ \Gamma_0 \subset \partial\Omega \\
D\frac{\partial u}{\partial n} + \chi u \frac{\partial u}{\partial n} &= 0 && on\ \Gamma_1 = \partial\Omega - \Gamma_0
\end{aligned}
\tag{1}
$$

where f is the reaction term describing mortality and regeneration; the first term in the divergence is the diffusion term describing dispersal by diffusive-like random motions; and the second term in the divergence is an advection term describing directed motions with a drift velocity given by the gradient of a function. The function λ in (1) is the loss function. Note that it is the gradient of the loss function and not the loss function itself that is of consequence. That is, the rate of change and the direction of change of the loss function is the manner by which the loss function influences distributions. In particular, the change of the loss function by the addition of any constant has no effect. However, multiplication by a constant does.

The boundary conditions are typically the zero-flux boundary conditions—the second of the two boundary conditions shown. The first boundary condition is shown for completeness. In exceptional cases there can be the need to specifiy the value of the density on the boundary rather than the flux across the boundary.

The equations are properly interpreted as a set of simultaneous equations. That is, u is a vector valued density function, with each component representing a different tophic level. In this case, there is a different diffusion constant, taxis constant, reaction term, and loss function for each trophic level. It is this set of equations that is solved in the *Alewife* software.

In (1) the function λ is to be a function of space and time. On the other hand, just prior to (1) the loss was described as a weighting of responses to physical and biological variables. For brevity let the physical and biological variables collectively be referred to as ***habitat variables***. Then the composition of funtions

$$\text{space} \times \text{time} \longrightarrow \{\text{habitat variables}\} \longrightarrow \{\text{responses}\} \longrightarrow \text{loss}$$

specifies how to assign a value $\lambda(x,t) = \text{loss}\,(x,t)$ for each point x in space and time t. To identify explicitly trophic level in this composition a trophic level index TL is added

$$\text{space} \times \text{time} \longrightarrow \{\text{habitat variables[TL]}\} \longrightarrow \{\text{responses[TL]}\} \longrightarrow \text{loss[TL]} \;.$$

The solutions to (1) represent the consequences of each individual in each trophic level being aware at the present time t of the loss[TL]$\,(x,t)$ for the position x that it currently inhabits relative to the loss[TL]$\,(x',t)$ for nearby positions x'. An absolute measure of loss is not needed. Nearby individuals for each trophic level then disperse in the direction that most quickly reduces loss[TL] and at a speed proportional to magnitude of the gradient of loss[TL].

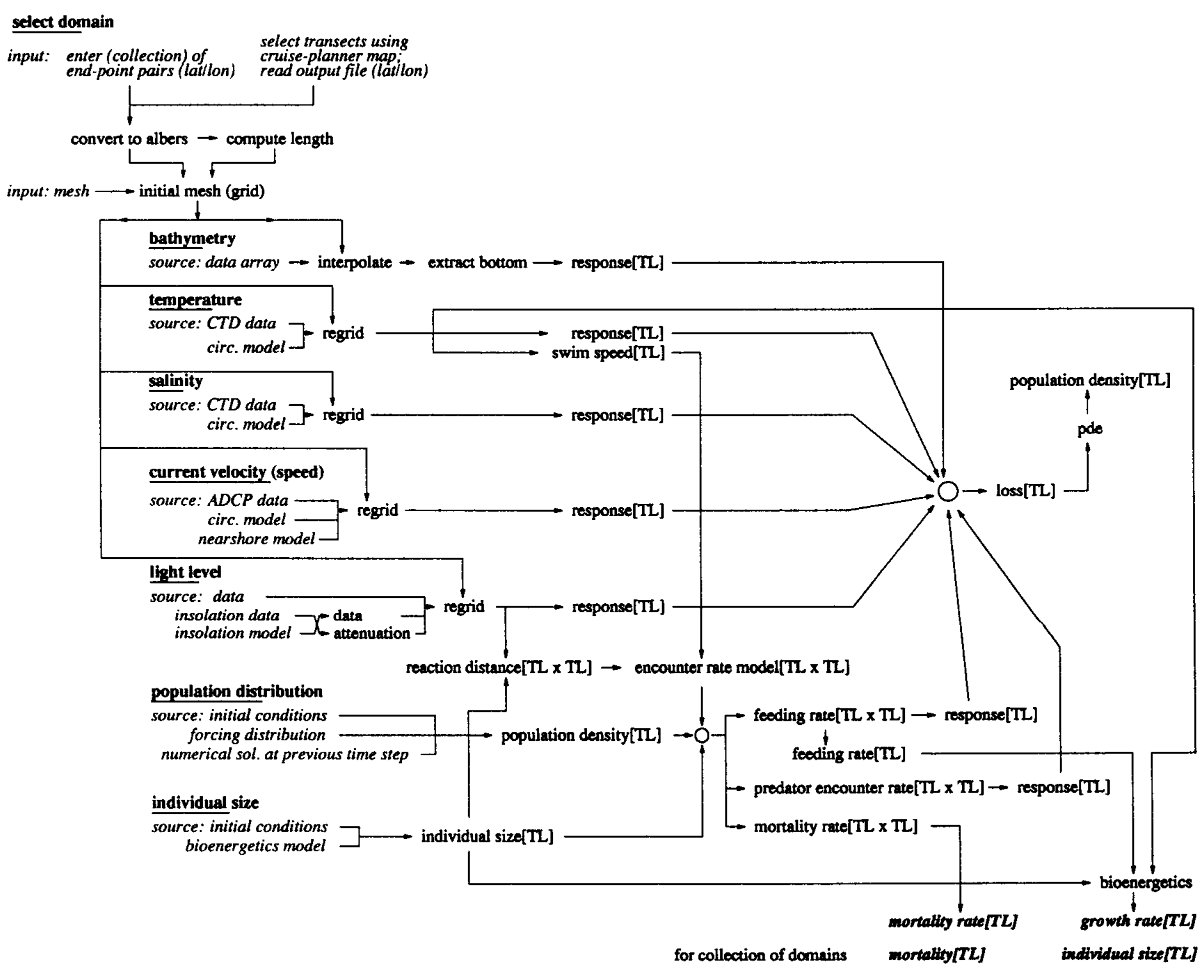


Figure 1 Construction of the loss function for pink salmon fry recruitment model.

In the following sections each of the data sources and sub-models used in the construction of the loss function is described. Each section refers to a node in the flow chart of Figure 1.

Selection of the domains

The first versions of the two- and three-dimension fish models is not ready.

One-dimensional versions of the fish model have been used for the initial developement. The space-time domain for the one-dimensional model is simply a single line and a time interval. The flow-chart in Figure 1 begins with select domain since the use of the one-dimensional model requires a selection of domain location and orientation.

Cruise planner

The cruise planner provides a convenient means to select and display one-dimensional domains for use with the fish model. Pink salmon fry and macrozooplankton occupy the near-surface regions of the water column during the time period of interest. For a zeroth-order approximation these upper regions are assumed to be homogeneous vertically and in the along-shore direction. It is assumed that most of the variation in physical structure and population distribution occurs in the "cross-channel" direction. With these assumptions the one-dimensional model can be used to simulate fry growth and mortality due to a given set of cross channel physical and biological conditions.

An expanded view of the region and of the domains is shown in Figure 2 using the zoom capability and postscript output of the cruise planner.

Length and time

Each of the domains is used separately with the one-dimensional model. For each domain only the length and the time interval is needed. For the alewife interface these data are entered in the Time and Space Parameters table in units of meters and seconds. For initial development a 1 km domain has been used. The tables ask for a beginning and ending position (although unnecessary) and the variables for these are X_A and X_B in meters. The time interval shown corresponds to 4 hr in units of seconds.

Mesh

For alewife the time steps and the spatial mesh are hard coded and a change requires a recompilation. (See CONST.h) The simulations shown here use 250 spatial nodes—a 4 m grid for a 1000 m domain—and 96 time steps—1 hr time steps for a 4 da simulation.

Bathymetry

The variable of bottom depth has two roles in the one-dimensional model. First, a basic issue for pink salmon fry is their reported tendency to prefer nearshore regions. This is in contrast to juvenile gadids that are more likely found in large aggregations off-shore in relatively open water. Although the behavior of salmon fry is often characterized in terms of distance from shore, the variable distance from shore is not likely one for which there is any perception. Consequently the mechanisms for this regulatation of their distribution must be sought amoung other variables. One of these is bottom depth. Once again it would seem that bottom depth itself is not perceived. There are however, two variables that can be perceived and are linked to bottom depth. Thesea are light level and benthic food sources. The bottoms in many nearshore regions are rocky, relatively smooth, and reflective. This bottom source of reflected light is readily detected. Second there are reports of significant benthic feeding for fry, and the illumination level

Figure 2 Cruise Planner map with typical model domains

at the bottom can be a factor for this feeding. For this one-dimensional model study the bottom depth itself is used as a surrogate variable for these potential factors and details for either light or bottom feeding are not made explicit. Conversely, there are populations that tend not to occur nearshore. For these also bottom depth is used as a habitat factor.

A second role for bathymetry is a means for including in the one-dimensional model euphasids. The rise of these animals in the water column during night means that their distribution tends to occur over water of sufficient depth. For these alternative prey the population is treated as forcing, time varying, and with a fixed spatial distribution—that is, their mortality is not included and the model does not attempt to conserve their numbers or account for their dispersal.

Data

The SEA database has a gridded bathymetry dataset for Prince William Sound in Albers 50 154 equal area projection, with a uniform 60 m grid. The bathymetry along the model domain is to be interpolated from this data set after converting the cruise planner domain data to Albers.

Simulation

For immediate trial the bathymetry is simulated with a functional form. The alewife model has all of these physical simulation functions hard-coded and changes require recompiliing. The bathymetry function is in def_lam.c. For example, $B_{max} = 350\,\text{m}$ and for s the position between X_A and X_B, the bottom depth $B(s)$ is specified very simply by

$$B(s) = B_{max} \sin\left(\frac{\pi(s - X_A)}{X_B - X_A}\right).$$

Responses

The response must be specified for each trophic level. In alewife Version 3 four trophic levels are allowed, with the first (i.e., TL = 1) strickly forcing. For current trials this trophic level is assigned to macrozooplankton, for this population distribution is assumed to be regulated as much by advection as by mortality and responses. Hence, a response specification is needed for TL = 2, 3, and 4.

alternative prey: juvenile gadids For these a response is constructed such that as bottom depth $B(s)$ decreases below $B_{onset}[\text{TL}] = 30\,\text{m}$ then the response Res[TL] gets less favorable (loss increases), with the response varying between 0 and 1.

$$\text{Res}_{\text{bath}}[\text{TL}](s) = \left(\frac{\max\{(B_{onset}[\text{TL}] - B(s)), 0\}}{B_{onset}[\text{TL}]}\right)^2$$

pink salmon fry For fry a candidate response that switches more rapidly is

$$\text{Res}_{\text{bath}}[\text{TL}](s) = \left(1 - \exp\left(\frac{-B(s)^2}{2B_{onset}[\text{TL}]^2}\right)\right)^2.$$

A reasonable value in this case as well is $B_{onset}[\text{TL}] = 30\,\text{m}$.

pollock The same response as for the juvenile gadid alternative prey is used for pollock.

The simulation for the bathymetry and the plots for the two responses are shown in Figure 3.

Temperature

Presently temperature is not used explicity for the one-dimensional model. Swimming speed is specified as an input variable rather than computed using temperature.

Data

The use of measured temperatures requires extensive work to formulate gridded data with sufficiently fine mesh. A better but more difficult solution is to use temperature data that has been assimilated using the ocean circulation model.

Responses

Functional forms have been developed to provide a response that defines a preferred temperature range. This response becomes much more important for spatial domains that extend through the water column.

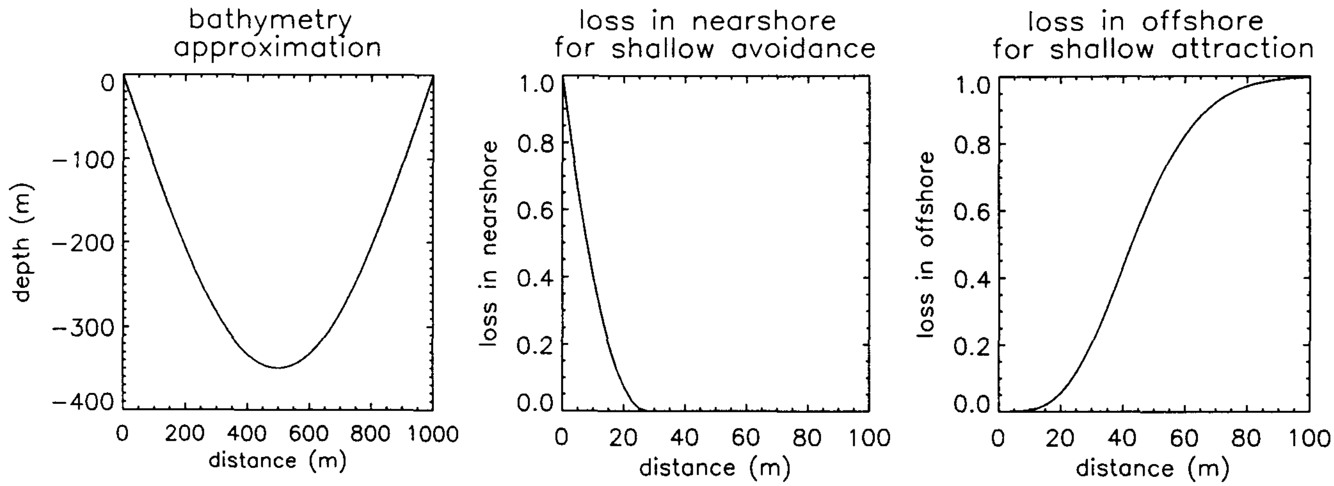


Figure 3 Simulation of bathymetry and plots of loss in the nearshore.

Salinity

Of those species being modelled only the plankton have strong responses to salinity within the range of salinity variation found in the sound. Because the plankton are hard coded and forcing in alewife Version 3, no salinity data, simulation, or response has been implemented.

Current velocity

The magnitudes and the spatial structures of the time varying current velocities, especially the tidal velocities, are hypothesized as significant factors in the spatial distribution of pink salmon fry. To incorporate this variable the time varying magnitude of the currents along the one-dimensional domain are needed.

Data

Soon time varying circulation data on 1.2 km grids will be available from the ocean circulation simulations. At most the data will need regridding for use with the fish model mesh. In addition some alternative approaches will be needed to estimate the current velocities in the near-shore that are not adequately characterized by the circulation model.

Simulations

For the immediate needs the cross-channel structure of the current has been represented by an approximating functional form. One such approximation is the following. Let $V_{max} = .30\,\mathrm{m/sec}$ be the maximum speed. The parameter $\sigma_V = 150\,\mathrm{m}$ is used to set the rate at which the current speed $V(s)$ decreases as s approaches the shore, that is, as s approaches either X_A or X_B.

$$V(s) = V_{max}\left(1 - \exp\left(\frac{-(s - X_A)^2}{2\sigma_V^2}\right)\right)\left(1 - \exp\left(\frac{-(s - X_B)^2}{2\sigma_V^2}\right)\right).$$

Responses

In the current form the responses $\mathrm{Res}_{\mathrm{curr}}[\mathrm{TL}]$ for all except pink salmon fry are neglected.

pink salmon fry For fry the square of the magnitude of the velocity is used for the response,

$$\text{Res}_{\text{curr}}[\text{TL}](s) = V^2(s) \ .$$

Normalization is not appropriate here for it would eliminate the comparison between different domains (lines). The specification of $V_{max} = .30\,\text{m/sec}$ is for a single domain.

The current speed as a function of distance along the domain is plotted in Figure 4 along with a plot for the loss in the nearshore region.

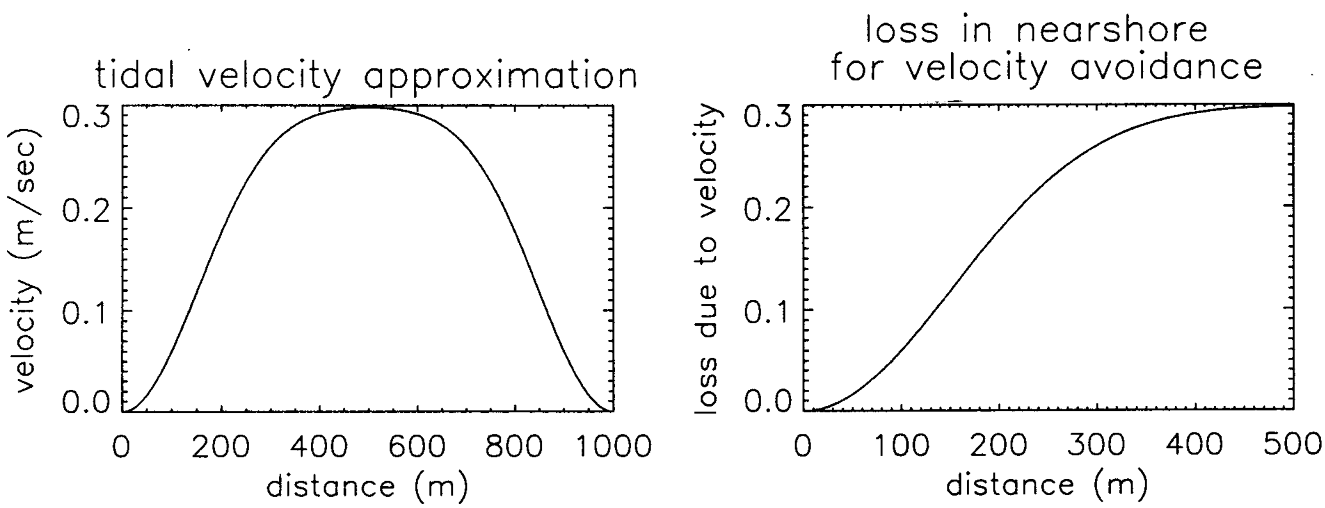


Figure 4

Light level

The light level is driven by the isolation at the surface and the attenuation through the water column. For the one-dimensional study an intermediate value must be used since the depth dimension is not used. However, there is still the diel change in insolation to be accommodated. For the present a functional approximation is used until better time and space data for insolation is available.

Insolation

A time periodic function is used to approximate the diel light cycle. This function is given in the subroutine e2_0.c of the alewife code. The function is somewhat cumbersome for sufficient parameters were included to adjust the insolation range and offset in order to simulate the long days (or nights) in Prince William Sound. A plot of the diel light cycle under this approximation is shown in Figure 5. Both linear and logarithmic plots are shown.

Attenuation

The insolation used here is defined by the fractional power of an affine transformation of a *cosine*. First, let

$$a = 1.03\,,$$
$$s^* = 4.8\,,$$
$$s^+ = 3.0\,.$$

The illumination during a 24 hr (or 86400 sec) May day, with maximum illumination at $t = 0$, is approximated by

$$s^+ + s^* \left[0.25 + a + (0.75 - a)\,cos\left(\frac{2\pi t}{86400}\right)\right]^{0.135}.$$

Responses

For the present study no direct response to the light level is used. All responses are due to the effects of the light level on the reaction distance.

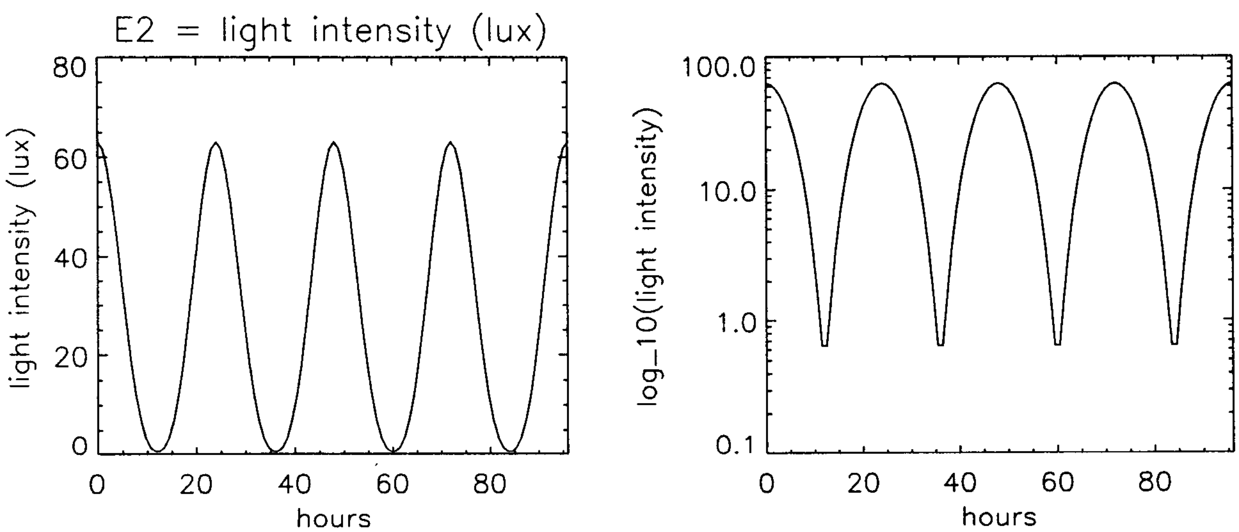


Figure 5

Individual size

This begins the biological components of the model.

Initial conditions

The parameters for individual mean lengths and biomass are shown in the "Feeding and Risk" table.

Mixing assumptions

The values for length and biomass are the initial conditions for the model run. In addition, during the relatively short time intervals of a simulation the individuals are assumed to undergo small changes in size and the effect of this change can be neglected. The net change in size due to feeding or lack thereof during several such intervals is computed at the end of the intervals and the size is changed by use of a bioenergetics model.

Each tropic level is assumed to consist of individuals of common size. If a set of sizes is needed then addtitional tropic levels are used. This is because the use of a size continuum in a single population changes the form of the model to an integral equation. An assumption is needed to have a single population grow at a common rate. It is assumed here that within the distributed population there is sufficient mixing among all individuals such that on average each has the same consumption.

Some work has been done to move to a more complete model that will include size ranges and also time varying satiation within the population.

Reaction distance

The *reaction distance* is the maximum distance at which an individual of the observing trophic level *obs* can detect an individual of the target population *tar*. A functional model [ref] with minimal complexity is used. Specifically, there are four independent variables: *obs*, *tar*, light level, and the size of the detected individual. This dependence is shown in the flow-chart in Figure 1. Reaction distance differs from the foregoing variables in that it does not depend explicitly on space and time.

Model

The function used to model the reaction distance is

$$R[obs, tar](I, s) = R_{max}[obs, tar](s)\left(1 - e^{-\beta[obs]\ \max\{(I - I_0[obs]), 0\}}\right),$$

where s is the size of the target individual in *tar*, I is the light level, and . $R_{max}[obs, tar](s)$ is the maximum reaction distance for the detection of an individual of size s regardless of light level. (In Version 3 $R_{max}[obs, tar](s)$ is entered manually in the tables.) It is assumed that nothing is detected by individuals in *obs* for light levels less than $I_0[obs]$, and that the single parameter $\beta[obs]$ suffices to characterize reaction distance for $I > I_0$.

The trophic level names *obs* and *tar* avoid introducing into the relationship those activities associated with the names predator and prey. This reaction distance function applies to the case of an individual prey as observer scanning for predators as targets as well as to the case of a predator as observer and prey as targets.

Swimming speed

For the current development a single average swimming speed is used for each trophic level. For the present trials wherein temperature does not vary within a domain the swimming speed values are manually entered in the tables.

Encounter rate

The *encounter rate* means the rate at which an individual observer of trophic level *obs* has individuals of trophic level *tar* enter the spatial region about the observer defined by the observer's reaction distance $R[obs, tar]$ for targets.

Models

The Gerritsen-Strikler encounter rate model assumes

- all individuals in *tar* are moving at the same "average" swimming speed $\bar{v}_{tar}$, with all directions equally likely
- the reaction distance for the observer does not depend upon viewing angle

Let v_{obs} denote the observer swimming speed and u_{tar} the density of the individuals in *tar*. With the above assumptions and the specification of which of v_{obs} and $\bar{v}_{tar}$ is greater the encounter rate can be computed.

$$\mathcal{E}[obs, tar](v_{obs}, \bar{v}_{tar}, u_{tar}, I, s) = \frac{\pi u_{tar} R[obs, tar](I, s)}{3} \frac{\min(\bar{v}_{tar}, v_{obs})^2 + 3\max(\bar{v}_{tar}, v_{obs})^2}{\max(\bar{v}_{tar}, v_{obs})}$$

Feeding rate

The model used for feeding rate is similar in form to a "carrying capacity" model. The feeding rate model uses two parameters to set the feeding rate limits—handling time and "waiting time." Waiting time addresses the feeding rate regulation associated with the hunger. Handling time is used in a slightly extended sense: the time required to complete the physical process of both capture and consumption.

All of the foregoing models applied to all trophic levels. The predator-prey models involve selection among trophic levels. Let $N + 1$ be the total number of trophic levels in the model, and let each trophic level be assigned an integer identifierTL in the sequence $0, 1, \ldots, N$. The index name TL is used whenever there is no need to identify both a trophic level and a process. The index variable $\mathbf{P}$, $\mathbf{P} = 0, 1, \ldots, N$, is used to refer to a trophic level functioning as predator. The index variable $\mathbf{f}$, $\mathbf{f} = 0, 1, \ldots, N$, is used to refer to a trophic level functioning as prey or forage.

Models

Prey For each trophic level $\mathbf{P}$ there is a subset of the $N + 1$ trophic levels that are prey. To provide a notation for this, let K be the function defined by $\mathrm{K}(\mathbf{P}, \mathbf{f}) = 1$ if $\mathbf{f}$ is a prey trophic level for $\mathbf{P}$, and $\mathrm{K}(\mathbf{P}, \mathbf{f}) = 0$ otherwise.

Handling time Let $h(\mathbf{P}, \mathbf{f})$ be the handling time, with $\mathbf{P}$ and $\mathbf{f}$ as above.

Waiting time The feeding rate is known to vary with gut fullness and "hunger." The feeding rate for fry that have been starved and then introduced to prey is initially high then drops to a rate that maintains a constant level of gut fullness [ref]. This study for fry did not address the issue of whether the gut fullness also is initially higher during the first feeding burst. However, feeding in bursts that fill the gut is a known phenomena for other predators.

A separate submodel has been formulated to provide the time varying feeding associated with the above phenomena. The current model does not incorporate this submodel because of the additional complexity. The submodel and the extension to incorporate it are described later and will be implemented in the near future. For now a "mean" waiting time is used.

To accommodate the adaptation of feeding for motivation let $\xi(\mathbf{P}, \mathbf{f})$ be the waiting time that follows consumption of a prey item from $\mathbf{f}$.

Prey preference For the present trials no preferences are assumed. That is, the selection of prey is assumed to be proportional to encounter rates.

Feeding capacity model The model for the feeding rate of a predator in **P** is

$$\varphi[\mathbf{P}] = \frac{\sum_{\mathbf{f}=0}^{N} \mathrm{K}(\mathbf{P},\mathbf{f})\,\mathcal{E}[\mathbf{P},\mathbf{f}]\,w(\mathbf{f})}{1+\sum_{\mathbf{f}=0}^{N} \mathrm{K}(\mathbf{P},\mathbf{f})\,\mathcal{E}[\mathbf{P},\mathbf{f}]\,(h(\mathbf{P},\mathbf{f})+\xi(\mathbf{P},\mathbf{f}))},$$

where $w(\mathbf{f})$ is the weight of a prey item in **f**. Properly $\xi(\mathbf{P},\mathbf{f})$ should depend upon gut fullness and a variable associated with hunger. It is this dependence that is omitted here. However, even in the more complete form it seems adequate to assume that $\xi(\mathbf{P},\mathbf{f})$ and $h(\mathbf{P},\mathbf{f})$ have a similar dependence on the weight of the prey item so that $\xi(\mathbf{P},\mathbf{f})/h(\mathbf{P},\mathbf{f})$ is independent of **f**. With this assumption define

$$\xi^*(\mathbf{P}) = \frac{\xi(\mathbf{P},\mathbf{f})}{h(\mathbf{P},\mathbf{f})}.$$

It is this variable $\xi^*(\mathbf{P})$ that is needed to account for gut dependent feeding rates. With this the feeding rate is

$$\varphi[\mathbf{P}] = \frac{\sum_{\mathbf{f}=0}^{N} \mathrm{K}(\mathbf{P},\mathbf{f})\,\mathcal{E}[\mathbf{P},\mathbf{f}]\,w(\mathbf{f})}{1+\sum_{\mathbf{f}=0}^{N} \mathrm{K}(\mathbf{P},\mathbf{f})\,\mathcal{E}[\mathbf{P},\mathbf{f}]\,h(\mathbf{P},\mathbf{f})(1+\xi^*(\mathbf{P}))}$$

Maximum feeding rate If the encounter rate $\mathcal{E}[\mathbf{P},\mathbf{f}_0]$ for a specific prey type $\mathbf{f}_0$ gets large, then the feeding rate approaches

$$\frac{w(\mathbf{f}_0)}{h(\mathbf{P},\mathbf{f}_0)(1+\xi^*(\mathbf{P}))}.$$

An upper bound for feeding rate $\varphi_{max}[\mathbf{P}]$ is then

$$\varphi_{max}[\mathbf{P}] = \frac{1}{(1+\xi^*(\mathbf{P}))} \max_{\mathbf{f}} \frac{w(\mathbf{f})}{h(\mathbf{P},\mathbf{f})}.$$

If gut fullness is included then a further upper bound is the maximum evacuation rate.

Responses

The response is simply that which maximizes feeding. Since response is defined in terms of loss we use

$$\mathrm{Res}_{\mathrm{feed}}[\mathrm{TL}] = \left(\frac{\varphi[\mathrm{TL}]-\varphi_{max}[\mathrm{TL}]}{\varphi_{max}[\mathrm{TL}]}\right)^2$$

Predator encounter rate

The last of the components is simple. It is the previous encounter rate.

Models

For a given prey trophic level **f** the total rate of encounters with predators for **f** is

$$\mathcal{E}_{\mathrm{Pred}}[\mathbf{f}] = \sum_{\mathbf{P}=0}^{N} \mathrm{K}(\mathbf{P},\mathbf{f})\,\mathcal{E}[\mathbf{f},\mathbf{P}].$$

Note that the order of the trophic level indeces in K is predator, prey; the order in $\mathcal{E}$ is observer, target. In the summation over predators it is prey **f** that is the observer and **P** is the target.

Responses

$$\mathrm{Res}_{\mathrm{pred}}[\mathrm{TL}] = \mathcal{E}_{\mathrm{Pred}}[\mathrm{TL}]^2$$

Loss

In the foregoing responses have been specified for bathymetry, current velocity, feeding rate, and encounter rate with predators. The loss for a given TL is the linear combination (i.e., weighted sum) of the responses.

BIOPHYSICAL PLANKTON MODELING
A Component of SEADATA, 95320-J

Dr. David L. Eslinger
Institute of Marine Science
University of Alaska Fairbanks

The SEADATA modeling effort is aimed at achieving three broad goals: 1) creation of a three-dimensional physical current field model; 2) creation of a three-dimensional biophysical model of plankton dynamics, which is coupled to the flow field produced by 1); and 3) creation of nekton models of pink salmon and Pacific Herring. The plankton modeling is being carried out at the University of Alaska Fairbanks. The 3-D plankton model is being built in a structured, step-wise fashion. As a first step, and as a deliverable for FY-95, we stated we would produce an annual one-dimensional coupled biophysical model by the end of FY-95. This model was to include diatoms, flagellates, large *Neocalanus* copepods, and smaller *Pseudocalanus* copepods. We are happy to report this model has been constructed, and is producing very reasonable results. We have presented a portion of these results at two national meetings: the American Association for the Advancement of Science Arctic Science Meeting in Fairbanks, AK, September, 1995; and the American Geophysicists Union/American Society of Limnology and Oceanography Ocean Sciences Meeting in San Diego, CA, February, 1996. These results are presented below.

Description of the model

The coupled biophysical model used in this study is based upon the spring diatom bloom model of Eslinger and Iverson (1996). Their physical model was a 39-layer, one-dimensional mixed-layer model based on the model of Pollard, *et al.* (1973) as modified by Thompson (1976). Meteorological forcing (wind mixing, solar heating, ocean-atmosphere heat fluxes) is applied at the surface and the water column mixes downward until there is a balance between the kinetic energy available for additional mixing and the potential energy cost of overcoming the existing stratification. This balance is determined by examining the Fruode number of the mixed layer. A full description of the 1-D physical model can be found in Eslinger (1990). The biological portion of the Eslinger and Iverson model (the EI model) was designed to simulate spring bloom dynamics over the southeastern Bering Sea shelf. Therefore, many of the biological dynamic processes were simplified. The EI model included nitrogen dynamics, but assumed that all forms of nitrogen are equal. Therefore, nutrient dynamics were modeled using a nitrogen pool containing total nitrate, nitrite and ammonium. Diatoms were the only phytoplankton included in the model, and zooplankton were included only in the sense that there was a grazing loss to the phytoplankton. The grazing loss was a constant fraction of daily production. The EI model was run only for the late winter and early spring period for which it was valid, however, for that period, the highly vertically resolved model reproduced the upper water column phytoplankton, nutrient, and temperature fields with excellent accuracy, Figure BPM-1.

The EI model formed an excellent base for constructing an annual model of upper water column physics, phytoplankton, zooplankton, and nutrient dynamics in Prince William Sound, Alaska. As we modified the EI model to be valid over annual time periods, we increased the number and complexity of the chemical and biological processes included in the model. We have added ammonium and silicon dynamics; a flagellate component; and three types of zooplankton: large *Neocalanus*-type copepods, smaller *Pseudocalanus*-type copepods, and euphausiids. There is in addition, an unspecified carnivorous nekton component which preys upon the zooplankton. Details of the model are given below.

Model Domain and Forcing Variables:

The physical domain is the upper 100 meters of a significantly deeper water column. Therefore, there are no bottom boundary layer or tidally mixed layer effects. These processes can be simulated by the model, however, they are not included in the present runs. Vertical grid resolution is 2 meters (a 50 layer model), and the time step is two hours. The model was run to examine interannual and spatial variability. When run in an interannual mode, the model was run beginning in late February or early March, depending on the availability of forcing data, and was run through approximately the middle of November for 1993, 1994, and 1995. Although the model was run for the greater part of the year, we will limit the remainder of this report to the spring and summer periods, when the planktonic dynamics are greatest. Meteorological forcing data were obtained from the Cooperative Fisheries and Oceanographic Studies (CFOS) moored buoy system initiated by Dr. Ted Cooney. For 1995, buoy data were unavailable for the early portion of the year, so meteorological data from a National Weather Service (NWS) unmanned station on Middleton Island, Alaska, was used. Wind speeds and air temperatures are shown in Figures BPM-2 and BPM-3, respectively. The spatial variability analysis was performed using 1995 forcing data from two NWS stations: the Middleton Island, AK NWS station, and a NWS station located at Whittier, AK. Wind speeds and air temperatures for the spatial analysis are shown in Figures BPM-4 and BPM-5, respectively. Insolation data required by the model was simulated using the radiation model of Frouin *et al.* (1989).

Phytoplankton Dynamics:

Diatoms and flagellates were included in the model. Maximum possible daily growth rate of both species was determined by temperature (Eppley, 1972), and could be reduced by light or nutrient limitation. Nitrate, ammonium, and silicon were considered as potentially biologically limiting nutrients, and nutrient uptake rate was assumed to follow a Michaelis-Menten relationship (Dugdale, 1967). Ammonium inhibition of nitrate uptake was included (Wroblewski, 1977). Both phytoplankton species competed for the nitrogen nutrients; silicon was utilized only by the diatoms. Photosynthesis was calculated as a function of light intensity, with a possibility of photoinhibition (Platt *et al.,* 1980). See Eslinger and Iverson (1996) for the actual implementation scheme for light or nutrient limitation.

Zooplankton Dynamics:

Three types of zooplankton were included in the model: *Neocalanus*, and *Pseudocalanus* type copepods, and euphausiids. Euphausiids are generally of minor importance in the zooplankton dynamics in PWS, and will not be included in the simulation results presented here. The two types of calanoid copepods differ in two significant ways: the adult *Neocalanus* are much larger than the *Pseudocalanus*, and the two types have significantly different reproductive strategies. The *Neocalanus*-type copepods undergo a dramatic ontogenetic migration, descending in late summer as stage copepidite V (hereafter, CV's) to a depth of 200-400 meters, where they overwinter. The following spring they mature, reproduce and die. The eggs hatch at depth and the nauplii begin to ascend towards the surface, which they reach at about the time they mature to the CI stage, and generally prior to the spring phytoplankton bloom. They feed and grow in the surface waters for approximately 65-75 days, after which they begin to descend again, as CV's (Fulton, 1973). In contrast, the *Pseudocalanus*-type copepods spend their entire life cycle in the upper water column, and overwinter as adult, fertilized females. They must feed on the spring phytoplankton bloom to begin reproducing, and can reproduce up to 10 times at approximately 5 day intervals (Corkett and McLaren, 1978). Actual embryonic and inter-molt timing was a function of temperature. The above scenarios are representative of the two broad categories of calanoid copepods which dominate the zooplankton biomass of PWS. The life history descriptions given are representative and are a simplification for the purposes of creating this model. For the present set of simulation results, specific life stage and reproductive aspects of *Pseudocalanus* life history were simplified. Total *Pseudocalanus* biomass was modeled, with no attempt made at keeping track of life stage. This introduces two errors into the model: 1) **individual weight-specific** parameters, *e.g.* grazing rate, are constant for all life stages; and 2) egg biomass is included in total *Pseudocalanus* biomass when calculating **population biomass-specific** effects, *e.g.* total phytoplankton biomass consumed by zooplankton grazing. The first assumption is the most critical and is being addressed in current work. The second assumption is less important because, although egg **numbers** may be high at times, total egg **biomass** is never large compared to copepidite biomass.

In the model, *Neocalanus* arrive in the surface (enter the model domain), as three groups of CI's, spaced over a 30 day time period, with the middle group containing one-half of the total biomass the other two groups containing one-quarter total biomass each. *Neocalanus* dynamics include modified Ivlev-type grazing (Ivlev, 1945; Magley, 1990) on both diatoms and flagellates; maturation, fecal pellet production, excretion of ammonium, and natural mortality (6%/day). With the simplification of the *Pseudocalanus* life history simulation, the *Pseudocalanus* dynamics are the same. Actual rates of the various parameters differed between the two zooplankton types.

Results

The model results showed that small differences in the meteorological forcing over a few critical weeks early in the spring phytoplankton bloom could create order of magnitude variations in the standing stock of zooplankton later in the summer. These small changes had similar effects when they occurred at a single location due to interannual variation in meteorological conditions, and when they occurred at different locations in the same year due to small horizontal gradients in meteorological conditions.

Interannual Variability

The model was run using forcing data for 1993, 1994, and 1995. The model simulations began with identical initial temperature and nutrient fields and initial concentrations of phytoplankton and zooplankton. Simulated phytoplankton chlorophyll concentrations are shown in Figure BPM-6a. Also shown is CFOS buoy fluorescence, which is representative of total phytoplankton chlorophyll. In 1993, the model simulated the timing, magnitude, and duration of the spring phytoplankton bloom extremely well. The bloom was fairly short in duration, ~15 days, and maximum chlorophyll concentrations were reached approximately 5 days after the onset of the bloom. The brief, intense phytoplankton bloom quickly stripped nutrients form the surface layer, and chlorophyll concentrations decreased rapidly. Zooplankton were unable to take full advantage of the brief phytoplankton increase. Biomass of the first *Neocalanus* group increased, while that of the second and third groups, which had less **total** biomass, increased much less. *Pseudocalanus* biomass remained fairly low, even after the phytoplankton bloom (Figure BPM-7a).

In 1994, the model bloom occurred ~5 days earlier than the bloom observed in the fluorescence data. The magnitude of the true bloom was underestimated in the modeled bloom. In both the model results and field data, the initial chlorophyll increase of the 1994 bloom occurred at approximately the same time as it had in 1993. However, in 1994, a strong cooling event occurred in the middle of the bloom, which led to a more protracted phytoplankton bloom (Figure BPM-3b). Zooplankton were able to take full advantage of this longer bloom period, and both *Neocalanus* and *Pseudocalanus* biomass increased substantially, with maximum biomass levels of approximately 1.0 and 0.75 GC wet-weight/m^{-3} respectively occurring in the model Figure BPM-7a.

In the 1995 simulation, the initial chlorophyll increase also began near day 90, but the increase to maximum chlorophyll concentrations took approximately 25 days, due to periodic strong wind mixing events, *c.f.* Figures BPM-2c and 6c. The relatively long, slow phytoplankton bloom allowed the zooplankton to efficiently utilize the primary production. Maximum daily *Neocalanus* biomass was higher in 1995 than in prior years, and *Pseudocalanus* biomass values approached those reached in 1994 (Fig. BPM-7c).

Although CFOS buoy data was not available for comparison with model chlorophyll, we did compare model results with discrete field samples from our 1995 cruises. In Figure BPM-8, phytoplankton and zooplankton biomass simulated in the model (the lines) agree very well with field data.

Spatial Variability

For the analysis of spatial variability, simulations were run using meteorological forcing data from Whittier, AK, located in northwestern Prince William Sound, and from Middleton Island, AK, located at the edge of the continental shelf, south of Prince William Sound (PWS). The Middleton Island data is assumed to be representative of the conditions over the Gulf of Alaska (GOA) shelf, *i.e.*, the "river" which may impact the Sound. The model simulations presented below all began with identical initial concentrations of phytoplankton and zooplankton. Initial temperature and nutrient fields were determined from field data. Figure BPM-8 shows the results of the spatial-variability model runs. CFOS buoy data is not available for this set of simulations.

The phytoplankton bloom began near day, at both locations, however, it was much slower for the GOA location. The maximum spring bloom chlorophyll concentrations were very similar between the two years, reaching peak concentrations of ~20 mg Chl m^{-3}, Figure BPM-9. Zooplankton biomass differed greatly between locations, with maximum values higher by a factor of approximately four for both *Neocalanus* and *Pseudocalanus*.

Discussion

In both interannual and spatial simulations, meteorological factors were responsible for both the timing of the initiation of the spring phytoplankton bloom, and the **character or nature** of the bloom. By character, we mean whether the bloom was a brief, intense event, or whether it was a more protracted event with a slower increase to, and duration of, maximal chlorophyll values. In cases when the bloom was brief and intense, *i.e.*, the 1993 and the NW PWS simulations, winds calmed, and remained relatively calm, for approximately 10 days (Figs BPM-2-5). This allowed a strong thermocline to develop, and the phytoplankton community responded with a rapid increase in biomass. This increase soon stripped the near surface, stratified layer of nutrients. and the phytoplankton spring bloom ceased (Figs BPM-6,9). Continued production the near-surface layer was driven by recycling through the zooplankton. Zooplankton, whose grazing is a function of biomass, could not take full advantage of these brief intense blooms. Therefore, zooplankton biomass, and *Pseudocalanus* biomass in particular, remained relatively low (Figs BPM 7, 9).

In contrast, when the initial stratification of the water column was periodically interrupted during the bloom period, by either convective cooling, as in 1994 (*c.f.* Fig. BPM-2b, 3b, and 6b), or intermittent strong wind mixing, as in 1995 and the GOA simulations (*c.f.* Fig. BPM-2c, 3c, and 6c; and 4a, 5a, and 9a), the phytoplankton bloom occurred over a longer period of time and was mixed deeper into the upper water column. This deeper mixing caused more nutrients to become available to phytoplankton in the euphotic zone and lead to a longer phytoplankton bloom, with more total primary production. The more gradual bloom gave zooplankton time to increase their biomass at a rate more similar to that of the phytoplankton production. This, in turn, lead to zooplankton biomasses as much as an order of magnitude greater than those found when the phytoplankton bloom was brief and shallow.

This aspect of the system leads to difference in export of fixed organic carbon to the aphotic zone and/or benthos which are summarized in Table 1.

Table 1. Primary production and flux from surface waters

Simulation	Total New Primary Production	% PP Sinking from Surface	Absolute Flux
1993	42.9 GC/m^2	34.8	14.9 GC/m^2
1994	54.9 GC/m^2	26.8	14.7 GC/m^2
1995	51.9 gC/m^2	31.8	16.5 GC/m^2
GOA Shelf	51 GC/m^2	33.6	17.5 GC/m^2
NW PWS	35 GC/m^2	56.6	19.6 GC/m^2

When the phytoplankton spring bloom was short and intense, *i.e.*, in the 1993 and NW PWS simulations, the total primary production during the bloom was minimal. However, because of the poor coupling between the phytoplankton and zooplankton populations, the **percent** of the new production which was exported out of the near-surface waters was maximal. The net effect of these two factors is to increase the absolute amount of primary production exported from the surface layers. This is dramatically illustrated in the spatial simulations, where, although total NW PWS new production was only 69% that of the GOA shelf, the absolute flux of organic carbon from the NW PWS surface layer was 112% that of the GOA shelf.

Conclusions

Plankton dynamics are a highly non-linear, but deterministic, pathway through which variations in the physical environment can be passed up the food web to both nekton and benthic organisms. Differences in the physical environment, and in meteorological forcing in particular, over a few weeks in early spring, can lead to order of magnitude differences in the upper water column zooplankton biomass throughout the summer. The potential effects of meteorological variations can be simulated, and eventually predicted, using coupled biophysical numerical models. Differences in the zooplankton community throughout the summer can be extremely importance to the planktivorous juvenile pink salmon and Pacific herring populations. We suggest that the models presented here are a critical component in understanding the dynamics of these species. In addition, plankton dynamics determine the partitioning of pelagic primary production between the upper water column and the benthos. Therefore, the type of model presented here will have value to other studies examining trophic structures in Prince William Sound, in particular EVOS groups such as the APEX and NVP programs.

References

Corkett, C.J. and I.A. McLaren, 1978, The biology of *Pseudocalanus. Adv. Mar. Biol.,* 15:1-23.

Dugdale, R.C., 1967, Nutrient limitation in the sea: dynamics, identification, and significance, *Limnology and Oceanography*, 12:685-695.

Eslinger, D.L., 1990, *The Effects of Convective and Wind-driven Mixing on Springtime Phytoplankton Dynamics as Simulated by a Mixed-layer Model.* Ph.D. Dissertation, Florida State University, Tallahassee, FL.

Eslinger, D.L. and R.L. Iverson, 1996, The effects of convective and wind mixing on springtime phytoplankton dynamics in the southeastern Bering Sea shelf. *Accepted, Continental Shelf Research.*

Fulton, J., 1973, Some aspects of the life history of *Calanus plumchrus* in the Strait of Georgia, *J. Fish. Res. Bd. Canada*, 30:811-815.

Frouin, R., D.W. Lingner, C. Gautier, K.S. Baker, and R.C. Smith, 1989, A simple analytical formula to compute clear sky total and photosynthetically available solar irradiance at the ocean surface, *Journal of Geophysical Research*, 94:9731-9742.

Ivlev, V.S., 1945, The biological productivity of waters, *Usp. Sovrem. Biol.*, 19:98-120.

Magley, W.C., 1990, *A Phytoplankton-Zooplankton Model of the Middle and Outer Shelf Domains of the Southeast Bering Sea Shelf During Spring Bloom Conditions*, Ph.D. Dissertation, Florida State University, 1990.

Platt, T., C.L. Gallegos, and W.G. Harrison, 1980, Photoinhibition of photosynthesis in natural assemblages of marine phytoplankton, *Journal of Marine Research*, 38:687-701.

Pollard, R.T., P.B. Rhines, and R.O.R.Y. Thompson, 1973, The deepening of the wind-mixed layer. *Geophysical Fluid Dynamics,* 4:381-404.

Thompson, R. O. R. Y., 1976, Climatological numerical models of the surface mixed layer of the ocean, *Journal of Physical Oceanography*, 6:496-503.

Wroblewski, J.S., 1977, A model of phytoplankton plume formation during variable Oregon upwelling, *J. Marine Res.*, 35:357-394.

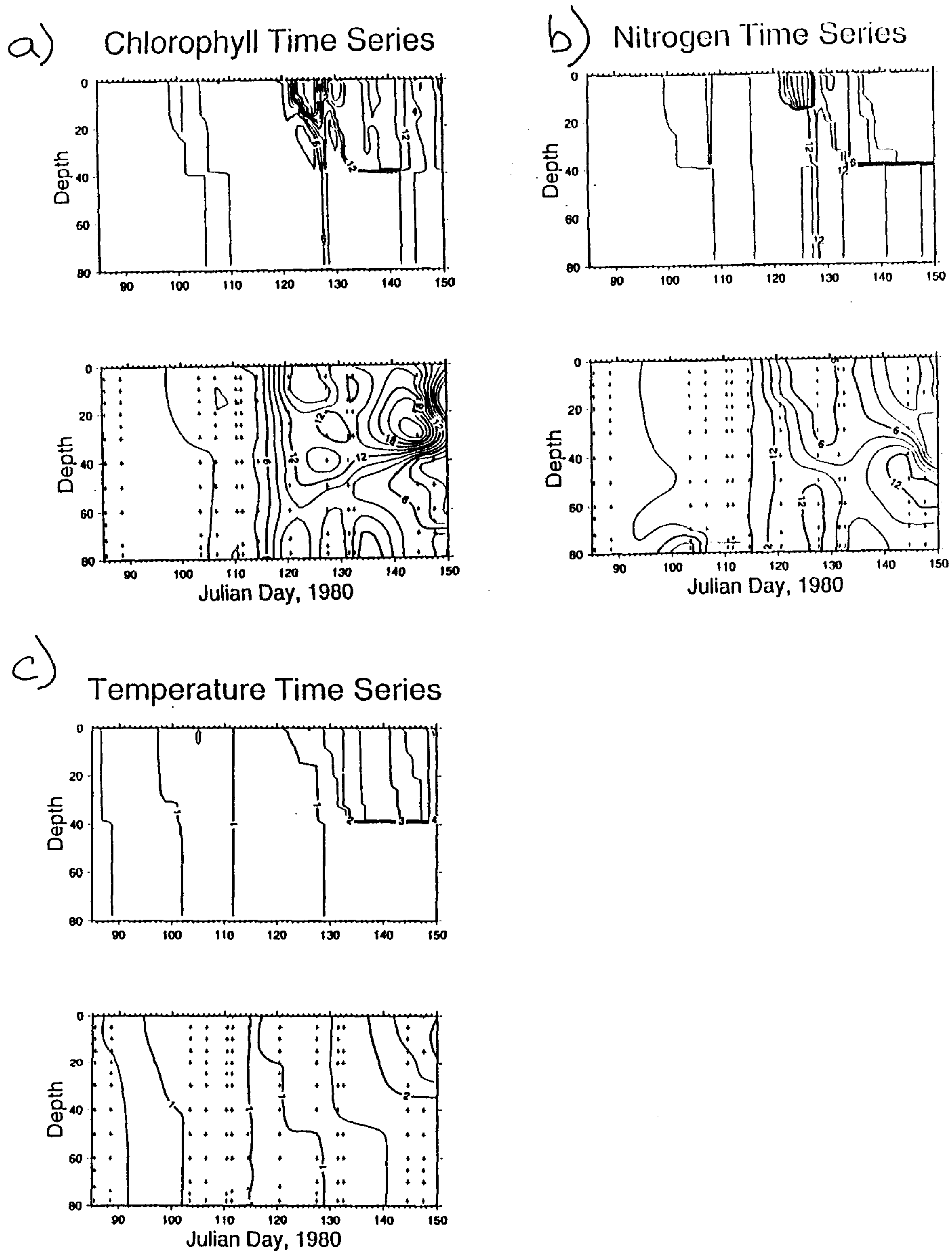


Figure BPM-1. Comparisons of one-dimensional models with field data. a) Depth-time contour of model results (top) and field data (bottom) for chlorophyll concentrations (mg m^{-3}) (Eslinger and Iverson, 1996) in the SEBS; b) as a) but for total nitrogen (μg-at l^{-1}); c) as a) but for temperature (C).

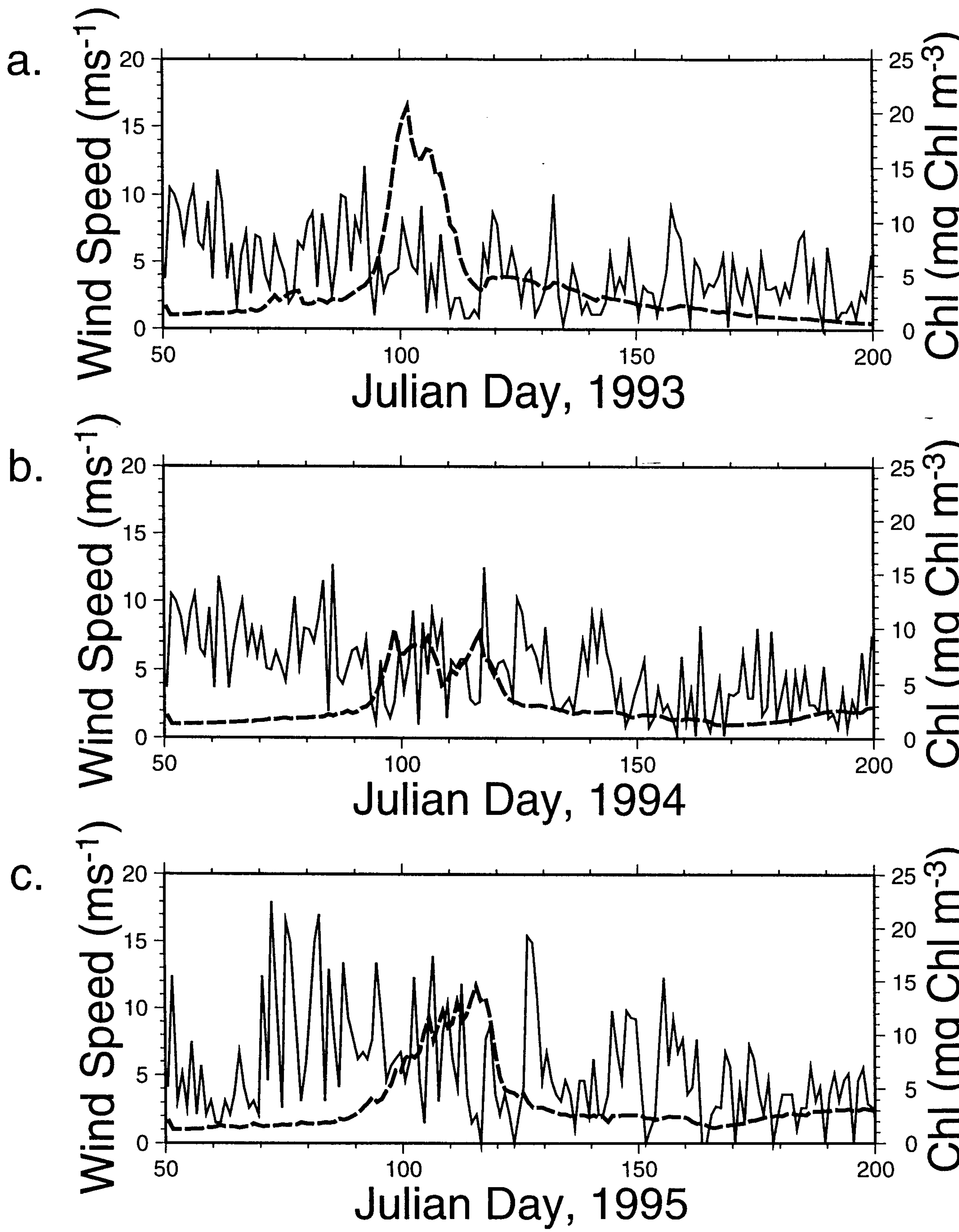


Figure BPM-2. Wind speed and model simulated chlorophyll concentrations for a) 1993, b) 1994, c) 1995.

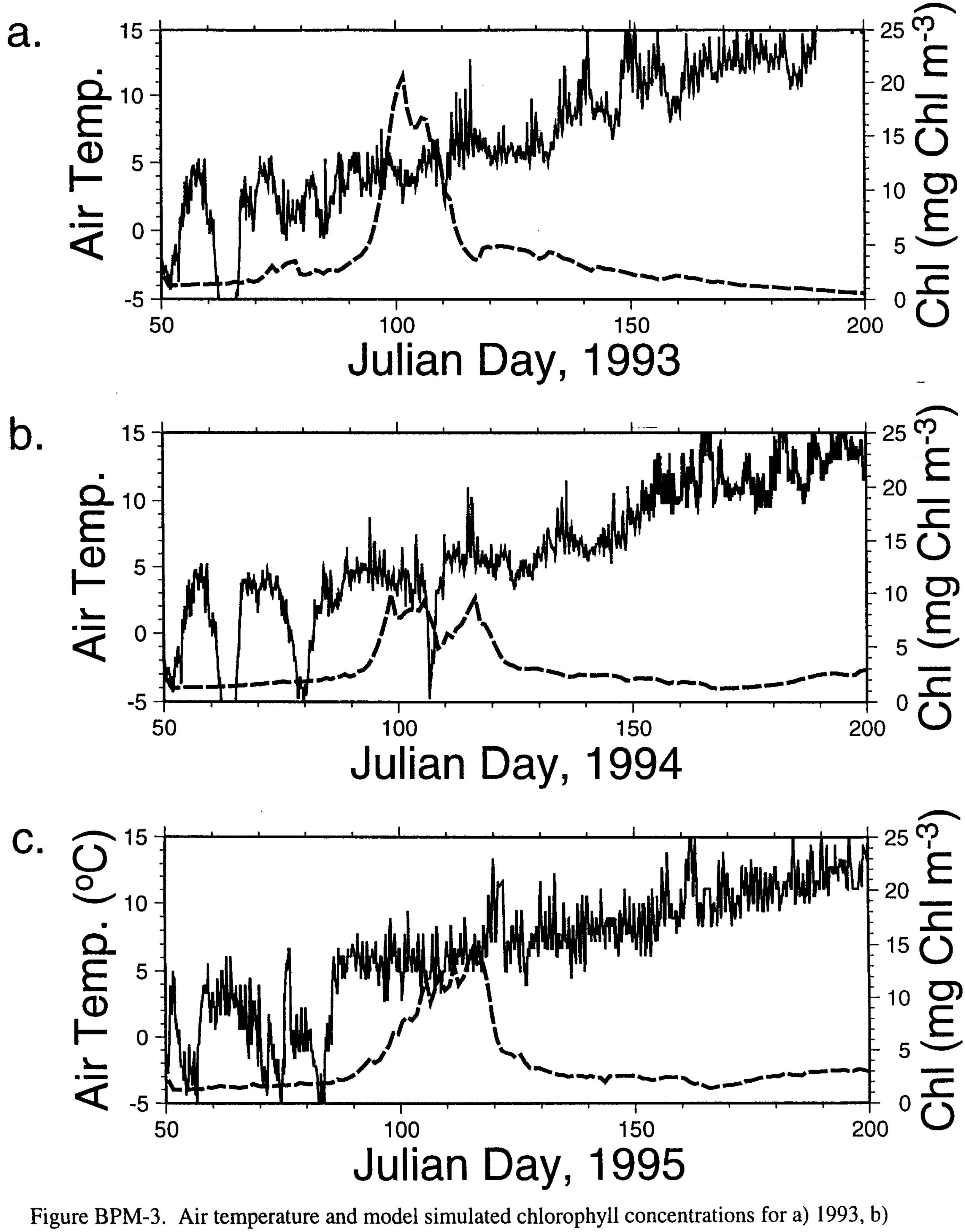


Figure BPM-3. Air temperature and model simulated chlorophyll concentrations for a) 1993, b) 1994, c) 1995.

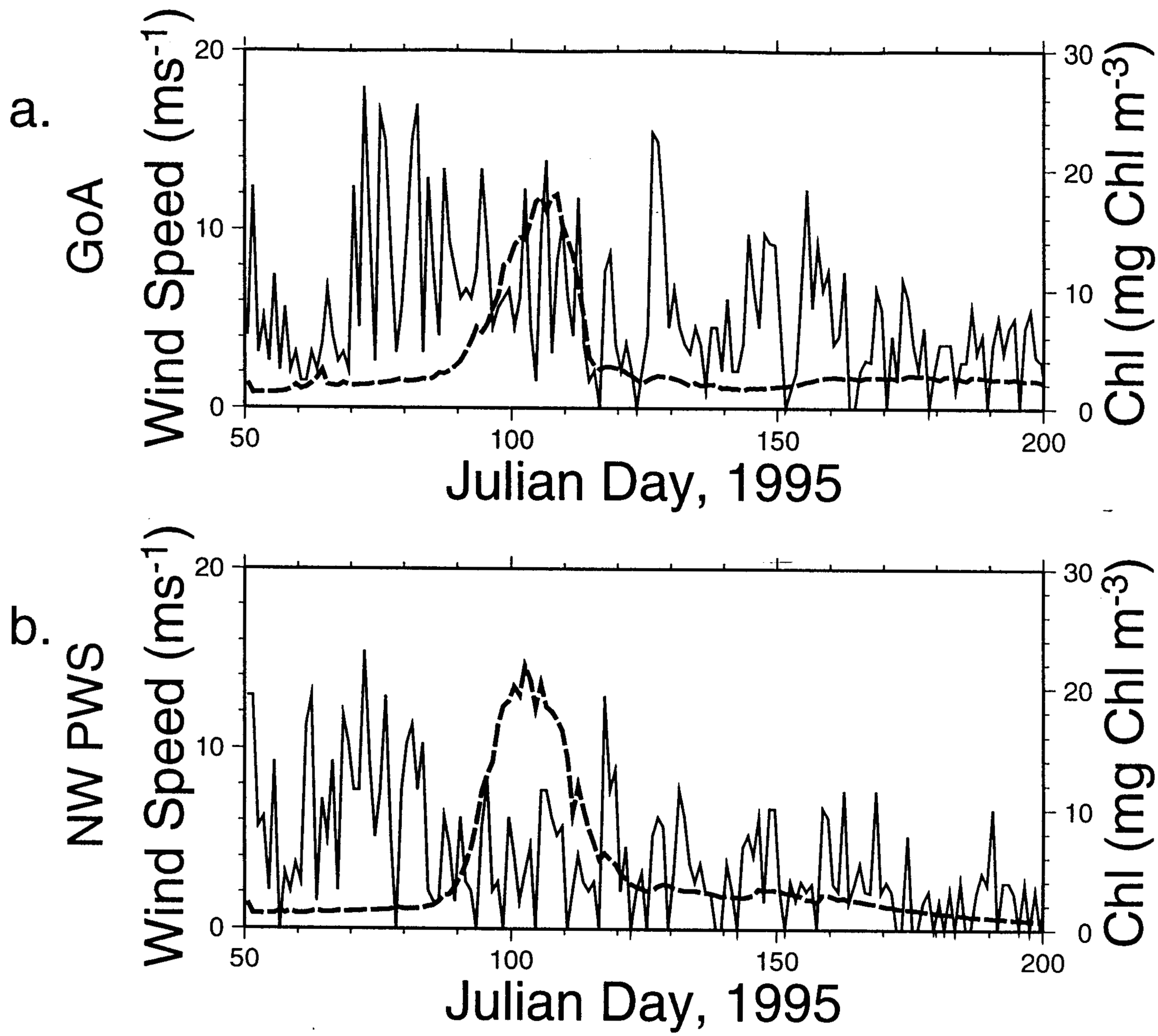


Figure BPM-4. Wind speed and model simulated chlorophyll concentrations for a) Gulf of Alaska shelf, and b) northwestern Prince William Sound, AK.

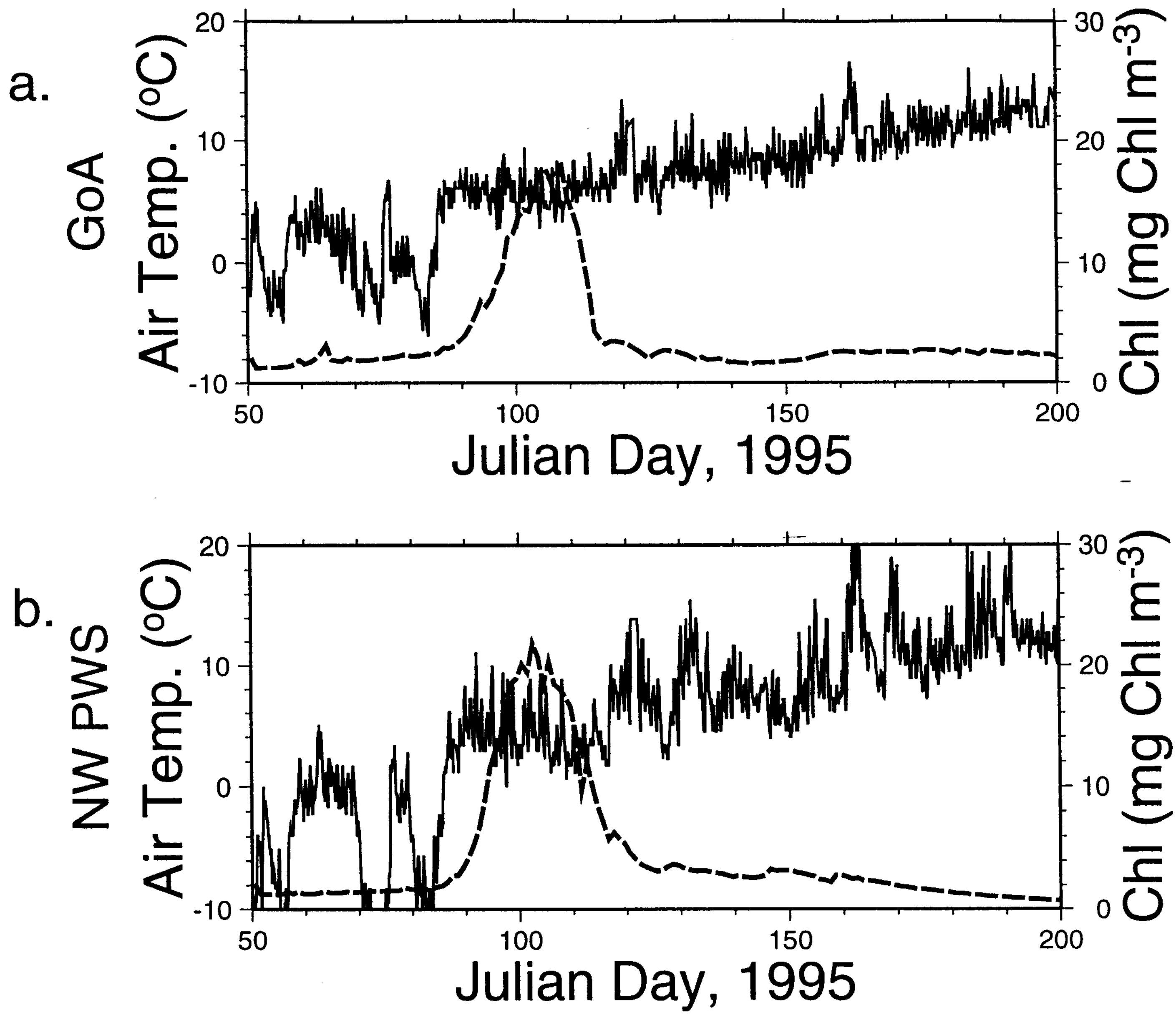


Figure BPM-5. Air temperature and model simulated chlorophyll concentrations for a) Gulf of Alaska shelf, and b) northwestern Prince William Sound, AK.

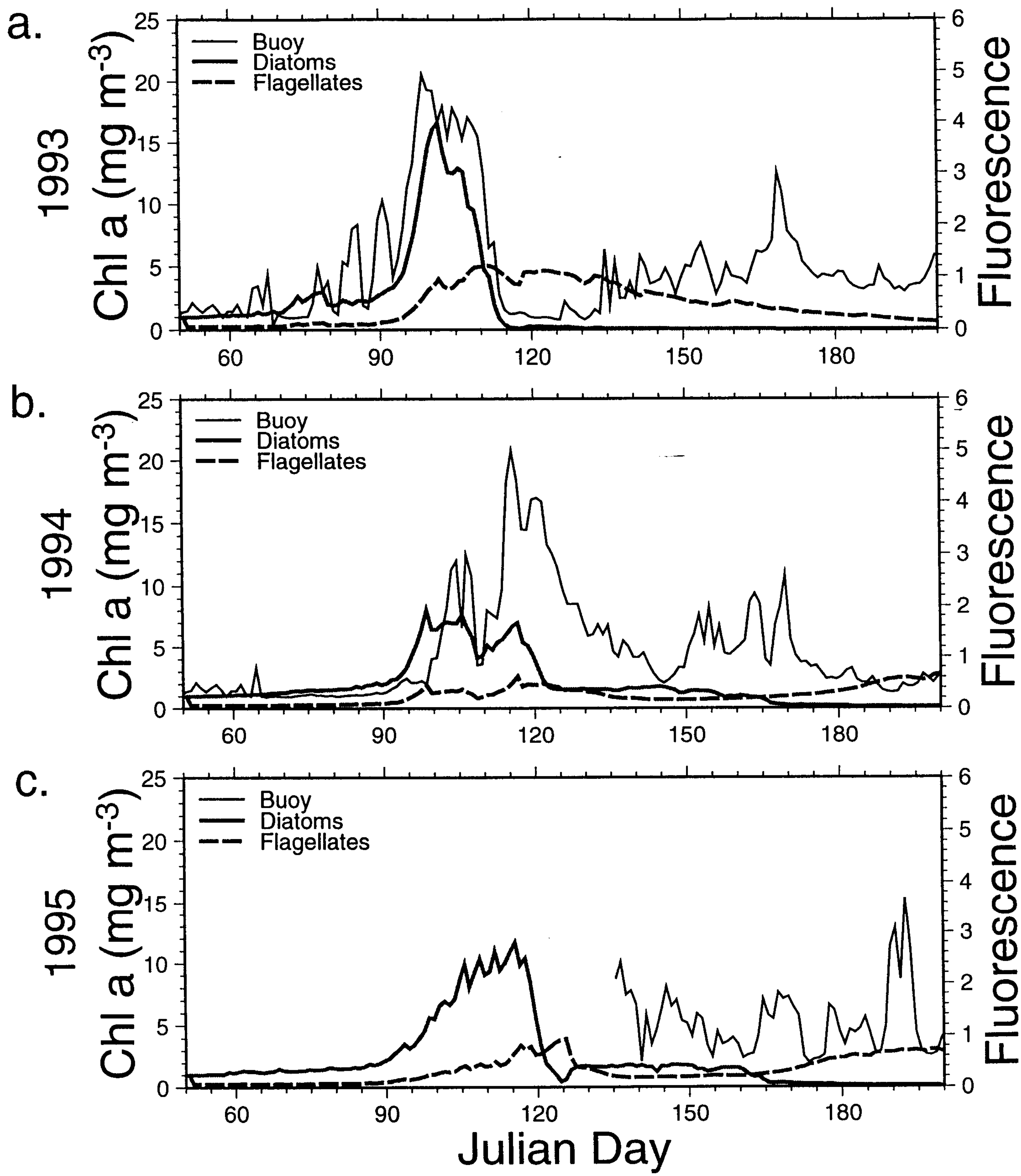


Figure BPM-6. Phytoplankton field data and model simulation results for a) 1993, b) 1994, c) 1995. The thin line is fluorescence measured at the CFOS buoy, the thick line is simulated diatom chlorophyll concentrations and the thick dashed line is simulated flagellate concentrations.

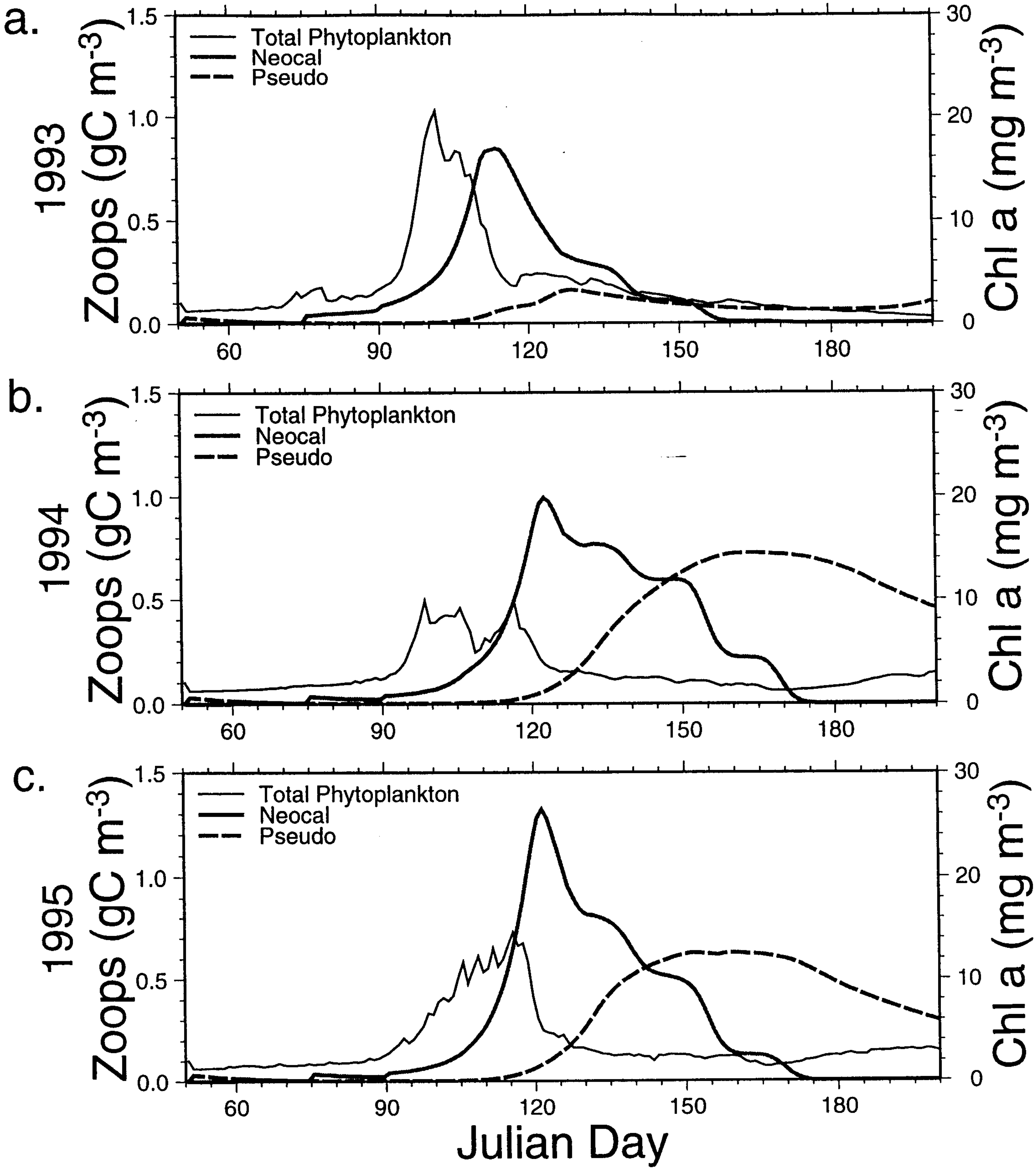


Figure BPM-7. Zooplankton model simulation results for a) 1993, b) 1994, c) 1995. The thin line is total simulated phytoplankton chlorophyll, shown for reference; the thick line is simulated *Neocalanus* wet weight; and the thick dashed line is simulated *Pseudocalanus* wet weight.

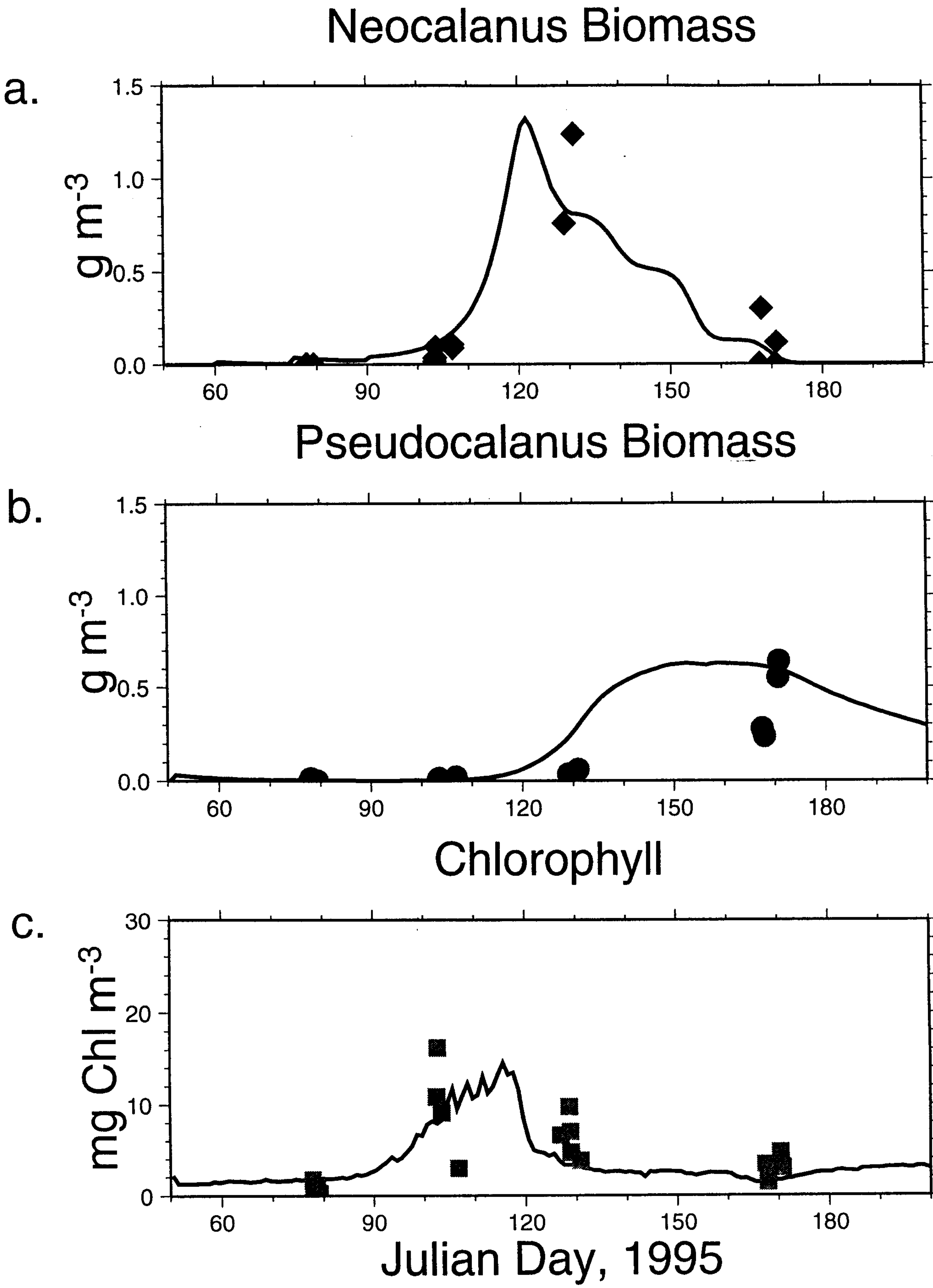


Figure BPM-8. Zooplankton and phytoplankton field data and model simulation results for 1995. The thick lines are model results and the symbols are field data from four cruises made in 1995. a) *Neocalanus* wet weight, b) *Pseudocalanus* wet weight, and c) total phytoplankton chlorophyll.

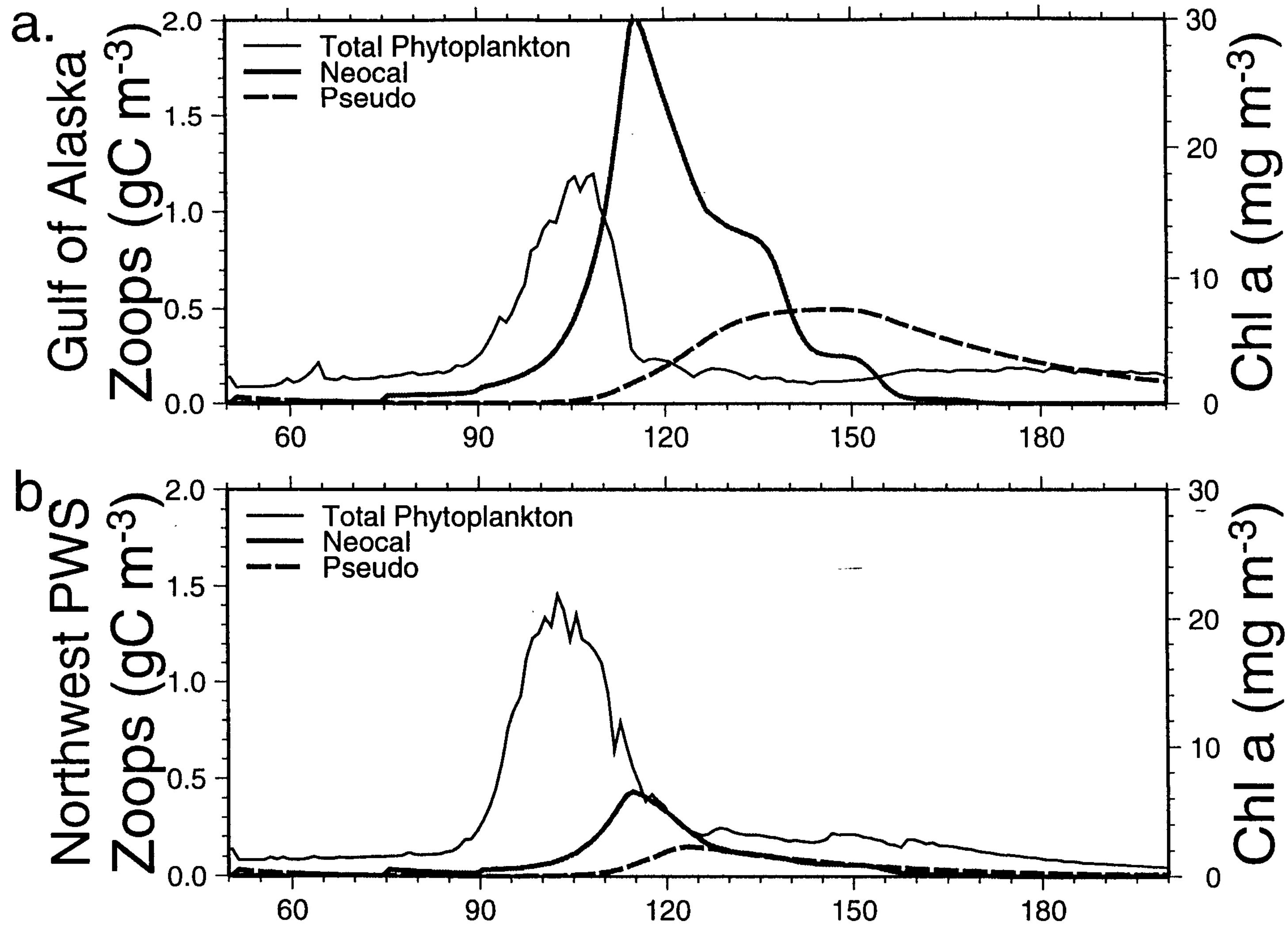


Figure BPM-9. Zooplankton model simulation results for a) Gulf of Alaska shelf, and b) northwestern Prince William Sound, AK. The thin line is total simulated phytoplankton chlorophyll, shown for reference; the thick line is simulated *Neocalanus* wet weight; and the thick dashed line is simulated *Pseudocalanus* wet weight.