SUPPORT OF GULF OF MEXICO HYDRATE RESEARCH CONSORTIUM: ACTIVITIES TO SUPPORT ESTABLISHMENT OF A SEA FLOOR MONITORING STATION PROJECT

SEMIANNUAL TECHNICAL REPORT
1 June, 2005 THROUGH 30 SEPTEMBER, 2005

PREPARED BY THE MANAGEMENT TEAM,
CENTER FOR MARINE RESOURCES AND ENVIRONMENTAL TECHNOLOGY
220 OLD CHEMISTRY BUILDING, UNIVERSITY, MS 38677

MAY 18, 2006

DOE Award Number DE-FC26-02NT41628

This report was prepared with the support of the United States Department of Energy, under award No. DE-FC26-02NT41628. However, any opinions, findings, conclusions, or recommendations expressed herein are those of the authors and do not necessarily reflect the views of the DOE. DOE Award Number DE-FC26-02NT41628 is managed by the U.S. Department of Energy's National Energy Technology Laboratory.

FY03 Subcontractors:

Paul Higley, Specialty Devices, Inc., 1104 Summit Ave. #104, Plano, TX 75074
  Task 1: Continuation of Work on the Vertical Line Array

J. Robert Woolsey, Mississippi Mineral Resources Institute (MMRI) and center for Marine Resources and Environmental Technology (CMRET), 220 Old Chemistry Building, University of Mississippi, University, Mississippi, 38677
  Task 2: Construction of the Prototype Sea Floor Probe

Ralph Goodman, Department of Marine Sciences, University of Southern Mississippi, 1020 Balch Blvd., Stennis Space Center, MS 39529
  Task 3: Acoustic System for Monitoring Gas Hydrates

Vernon Asper, Department of Marine Sciences, University of Southern Mississippi, 1020 Balch Blvd., Stennis Space Center, MS 39529
  Task 4: Construction and Testing of an Electromagnetic Bubble Detector and Counter

Boris Mizaikoff, School of Chemistry and Biochemistry, Georgia Institute of Technology, Applied Sensors Laboratory, 770 State St, Atlanta, GA 30332
  Task 5: Mid-Infrared Sensor Systems for Continuous Methane Monitoring in Seawater

Angela Davis, AUGER Geophysical Services, School of Ocean Sciences, University of Wales, Bangor, Menai Bridge, Anglesey LL59 5EY, Bangor, Wales, UK
  Task 6: Seismo-Acoustic Characterization of Sea Floor Properties and Processes at the Hydrate Monitoring Station

FY04 Subcontractors:

Barrodale Computing Services, Ltd. Hut R, McKenzie Avenue, University of Victoria, Victoria, BC V8W 3W2 Canada
  Task 1: Data Management and Processing Software for the Sea-floor Monitoring Station

Bob A. Hardage, Bureau of Economic Geology, John A. and Katherine G. Jackson School of Geosciences, University of Texas at Austin, University Station, Box X, Austin, TX 78713
  Task 2: Applications of VSP Technology for Evaluation of Deep-Water Gas Hydrate Systems*

Jeffrey Chanton, Department of Oceanography, Florida State University, Tallahassee, FL 32306
  Task 3: Coupling of Continuous Geochemical and Sea-floor Acoustic Measurements

Rudy Rogers, Swalm School of Chemical Engineering, P.O. Box 9595, Mississippi State, MS 39762
  Task 4: Microbial Activity Related to Gas Hydrate Formation and Sea-floor Instabilities

* includes seismo-acoustic characterization of sea-floor properties and processes at the hydrate monitoring station until VSP data can be collected

DISCLAIMER

ABSTRACT

The Gulf of Mexico Hydrates Research Consortium (GOM-HRC) was established in 1999 to assemble leaders in gas hydrates research. The Consortium is administered by the Center for Marine Resources and Environmental Technology, CMRET, at the University of Mississippi. The primary objective of the group has been to design and emplace a remote monitoring station or sea floor observatory (MS/SFO) on the sea floor in the northern Gulf of Mexico by the year 2005, in an area where gas hydrates are known to be present at, or just below, the sea floor. This mission, although unavoidably delayed by hurricanes and other disturbances, necessitates assembling a station that will monitor physical and chemical parameters of the sea water and sea floor sediments on a more-or-less continuous basis over an extended period of time. Development of the station has always included the possibility of expanding its capabilities to include biological monitoring, as a means of assessing environmental health. This possibility has recently achieved reality via the National Institute for Undersea Science and Technology's (NIUST) solicitation for proposals for research to be conducted at the MS/SFO.

Establishment of the Consortium has succeeded in fulfilling the critical need to coordinate activities, avoid redundancies and communicate effectively among researchers in the arena of gas hydrates research. Complementary expertise, both scientific and technical, has been assembled to promote innovative research methods and construct necessary instrumentation. The observatory has achieved a microbial dimension in addition to the geophysical and geochemical components it had already included.

Initial components of the observatory, a probe that collects pore-fluid samples and another that records sea floor temperatures, were deployed in Mississippi Canyon 118 in May of 2005. Follow-up deployments, planned for fall 2005, have had to be postponed and the use of the vessel *M/V Ocean Quest* and its two manned submersibles sacrificed due to the catastrophic effects of Hurricane Katrina (and later, Rita) on the Gulf Coast. Every effort is being made to locate and retain the services of a replacement vessel and submersibles or Remotely Operated Vehicles (ROVs) but these efforts have been fruitless due to the demand for these resources in the tremendous recovery effort being made in the Gulf area. Station/observatory completion, anticipated for 2007, will likely be delayed by at least one year.

The seafloor monitoring station/observatory is funded approximately equally by three federal Agencies: Minerals Management Services (MMS) of the Department of the Interior (DOI), National Energy Technology Laboratory (NETL) of the Department of Energy (DOE), and the National Institute for Undersea Science and Technology (NIUST), an agency of the National Oceanographic and Atmospheric Administration (NOAA).

Subcontractors with FY03 funding fulfilled their technical reporting requirements in the previous report (41628R10). Only unresolved matching funds issues remain and

will be addressed in the report of the University of Mississippi's Office of Research and Sponsored Programs.

Noteworthy achievements funded with DOE's contributions to this multiagency effort include:

Progress on the Data Management and Processing Software for the Sea-floor Monitoring Station (Barrodale Computing Services Ltd. (BCS)):

- A software design for simulation, matched-field inversion, and real-time acoustic array monitoring for the MS/SFO has been completed. This design accommodates the ongoing observation of the hydrate-containing sub-bottom layer of the sea-floor by analyzing acoustic data generated by nearby sources of opportunity.
- A software system has been developed for simulation and matched-field inversion (MFI) of acoustic array data to provide a comprehensive and validated environment for investigating the application of MFI techniques to detect changes in the sub-bottom gas hydrate deposits under the sea floor in the region of the MS/SFO. The software developed provides a suite of components for generating synthetic data simulating shots and ship noise, transforming these data into a form where they can be used for matched field techniques, and applying these techniques to analyze the data.

Progress on the Applications of VSP Technology for Evaluation of Deep-Water Gas Hydrate Systems* at the University of Texas Bureau of Economic Geology's Exploration Geophysics Laboratory, EGL (seismo-acoustic characterization of sea-floor properties and processes at the hydrate monitoring station until VSP data can be collected):

- EGL scientists have developed a new concept for processing deep-water 4C OBC data that yields a significant improvement in the spatial resolution of P-P images.
- The above-mentioned development will facilitate use of industry standard seismo-acoustic data in conjunction with much higher resolution data than has been hitherto possible.

Progress on the Coupling of Continuous Geochemical and Sea-floor Acoustic Measurements:

- To help understand the spatial variability in geochemical indicators at a hydrate seep site and the role biogeochemical processes play in hydrate stability, nutrients, methane and sulfate concentrations and methane stable carbon isotopes were measured in 10 cores collected at MC 118.
- Core data have been extracted and analyzed from MC118 cores in an effort to characterize the site, geochemically.

Progress on the Microbial Activity Related to Gas Hydrate Formation and Seafloor Instabilities:

- o Sediments from two locations in the Mississippi Canyon, MC118 and MC798, were analyzed for propensity to form gas hydrates.
- o Hydrate formation rates and crystal initiation times were measured in the laboratory as a function of depth below sea-floor and as a function of lateral displacement. Trends were observed and documented.

Administration of the Monitoring Station/Sea-floor Observatory project consisted mostly of scheduling and arranging all aspects of the several deployment cruises required to get the station on the sea-floor and functioning. Due to the busy cruise schedule, a semiannual meeting was not scheduled but should take place toward the end of the year.

TABLE OF CONTENTS **PAGE**

LIST OF GRAPHICAL MATERIALS

Graphical materials used to illustrate reports can be found in the individual reports submitted by the subcontractors.

INTRODUCTION / PROJECT SUMMARY

The Gulf of Mexico-Hydrate Research Consortium (GOM-HRC) is in its fifth year of developing a sea floor station to monitor a mound where hydrates outcrop on the sea floor. The plan for the Monitoring Station/Sea Floor Observatory (MS/SFO) is that it be a multi-sensor station that provides more-or-less continuous monitoring of the near-seabed hydrocarbon system, within the hydrate stability zone (HSZ) of the northern Gulf of Mexico (GOM). The goal of the GOM-HRC, is to oversee the development and emplacement of such a facility to provide a better understanding of this complex hydrocarbon system, particularly hydrate formation and dissociation, fluid venting to the water column, and associated microbial and/or chemosynthetic communities. Models developed from these studies should provide a better understanding of gas hydrates and associated free gas as: 1) a geo-hazard to conventional deep oil and gas activities; 2) a future energy resource of considerable significance; and 3) a source of hydrocarbon gases, venting to the water column and eventually the atmosphere, with global climate implications.

The GOM-HRC initially received funding from the DOI Minerals Management Service (MMS) in FY1998. Funding from the DOE National Energy Technology Laboratory (NETL) began in FY2000 and from the Department of Commerce National Oceanographic and Atmospheric Administration's National Undersea Research Program (DOC NOAA-NURP) in 2002. Some ten industries and fifteen universities, the USGS and the US Navy, Naval Meteorology and Oceanography Command, Naval Research Laboratory and NOAA's National Data Buoy Center are involved at various levels of participation. Funded investigations include a range of physical, chemical, and, most recently, microbiological studies.

EXECUTIVE SUMMARY

A consortium has been assembled for the purpose of consolidating both laboratory and field efforts of leaders in gas hydrates research. The Consortium, established at and administered by the University of Mississippi's Center for Marine Resources and Environmental Technology (CMRET), has, as its primary objective, the design and emplacement of a remote monitoring station on the sea floor in the northern Gulf of Mexico by the year 2005. The primary purpose of the station is to monitor activity in an area where gas hydrates are known to be present at, or just below, the sea floor. In order to meet this goal, the Consortium has begun assembling a station that will monitor physical and chemical parameters of the sea water, sea floor sediments, and shallow subseafloor sediments on a more-or-less continuous basis over an extended period of time. Central to the establishment of the Consortium is the need to coordinate activities, avoid redundancies and promote effective and efficient communication among researchers in this growing area of research. Complementary expertise, both scientific and technical, has been assembled; collaborative research and coordinated research methods have grown out of the Consortium and design and construction of instrumentation for the sea-floor station is nearing completion.

The MS/SFO was designed to accommodate the possibility of expanding its capabilities to include biological monitoring. A portion of FY04 funding from the

Department of the Interior's Minerals Management Services was been directed toward this effort to support the study of chemosynthetic communities and their interactions with geologic processes. In addition, results will provide an assessment of environmental health in the area of the station. NOAA-NURP has, as a focal point, investigations of the effects of deep sea activities on world atmosphere and therefore, weather. In July of 2005, The National Institute for Undersea Science and Technology (NIUST) of NOAA-NURP Director made a portion of that agency's budget available *via* competitive grants to researchers with proven expertise in microbial research. A sea-floor microbial observatory is an objective of that agency and these sponsored projects will tie in with the MS/SFO in a move in that direction.

The centerpiece of the monitoring station, as originally conceived, is a series of vertical line arrays of sensors (VLAs), to be moored to the sea floor. Each VLA was to have extended approximately 200 meters from the sea-floor. Sensors in the VLAs include hydrophones to record water-borne acoustic energy (and measure sound speed in the lower water column), thermistors to measure water temperature, tilt meters to sense deviations from the vertical induced by water currents, and compasses to indicate the directions in which the deviations occur. During discussions among the members of the geophysical subgroup of the Consortium, it was discovered that the project may be better served if some vertical arrays are converted to horizontal line arrays (HLAs). The prospective horizontal water-bottom arrays, will consist of hydrophones and 3-component accelerometers and will be laid upon, and pressed into, the soft sediment of the sea-floor. They will be arranged into a cross so that they simulate two perpendicular arrays. Their deployment will be accomplished by means of a sea-floor sled designed to lay cable and deploy probes into shallow, unconsolidated sediments. This sled will also be used as a seismic source of compressional and shear waves for calibrating the subsurface seismo-acoustic array commissioned by the Joint Industries Program (JIP).

The prototype DOE-funded VLA has been completed and tested together with the associated data logging and processing systems. An Oceanographic Line Array (OLA) is ready to be equipped with any of a variety of geochemical sensors - thermistors, fluorometers, transmissometers, mass spectrometers, conductivity and current flow meters – and deployed at the observatory site, Mississippi Canyon 118 (MC118). Processing techniques continue to be developed for vertical array data by Consortium participants who are currently funded by the Minerals Management Service.

A Remotely Operated Vehicle (ROV) mateable connector system was designed and installed in the VLA Data Acquisition and Telemetry System (DATS) deployed in 2005. This improved design has been incorporated into the VLA and the OLA components of the observatory. Positioning sensors – including compass and tilt sensors – have been completed and tested. Pressure housings rated twice that of any anticipated deployment have been built and pressure tested.

In May, 2005, the Sea-Floor Probe (SFP) was used to retrieve core samples from MC118 as part of the effort to select sites appropriate for deployment of the geophysical and geochemical probes. The northwestern portion of the mound area defined on images recovered during a C&C autonomous underwater vehicle (AUV) survey April 30-May 2, 2005, was selected for probe deployments based on information from these cores. Both the pore-fluid array and the geophysical line array were deployed via SFP at MC118 in May, 2005.

Additional MS/SFO deployments, scheduled for September and October, 2005, have been delayed due to the devastation of the Mississippi Gulf Coast and environs by Hurricane Katrina and, to a lesser extent, the Louisiana Gulf Coast by Hurricane Rita. The immediate cause for delay was the removal of the *Ocean Quest*, the vessel that, with its two submersibles, was to have provided the platform from which many of the bottom-founded sensors would have been deployed and cable connections made. It would also have provided the visual survey needed to make optimal choices of deployment sites for station components. Every effort is being made to find a replacement vessel and/or vessels but most vessels in the Gulf are being used in the massive recovery effort that is underway in the region. In addition, damage to ship yards and various forms of infrastructure has been extensive. Work scheduling is largely guess work at this point and prospects of securing a vessel with submersibles or even ROV capability in the near future seem remote. It is now apparent that the station deployment date will have to be pushed forward by at least one year and perhaps more. The fall of 2006 is the soonest we can expect to gain access to the type of ship/vessel we will need to complete the deployments phase of the project.

In spite of the delays, project development continues:

Barrodale Computing Services Ltd. (BCS) has designed and developed a software system for simulation and matched-field inversion (MFI) of acoustic array data. The system, termed BCOMFI (**B**arrodale **CO**mputing **M**atched **F**ield **I**nversion), provides a comprehensive and validated environment for investigating the application of MFI techniques to detect changes in the sub-bottom gas hydrate deposits under the sea-floor in the region of the MS/SFO. This approach is based on the expectation that MFI analysis of acoustic array data originating from nearby sources of opportunity (passing ships) can be used to detect such changes. The basic principle is to derive geoacoustic models for sub-bottom regions of the station by applying MFI to data from calibration measurements, and then use these models to match with future data obtained from passing sources. The presence of a large mismatch would be taken as evidence of a change.

The BCOMFI software developed by BCS provides a suite of components for generating synthetic data simulating shots and ship noise, transforming these data into a form where they can be used for matched field techniques, and applying these techniques to analyze the data. Methods for simulating acoustic array data from shots or ship noise were developed first, and then used to support the development and validation of matched-field methods under controlled conditions. The result was that these methods are now in place and ready to be applied when real data from the array become available (expected in 2006).

Investigations have shown that standard P-P imaging of data acquired using 4C seafloor sensors does not produce the resolution of near-seafloor geology that is desired for this project. University of Texas Exploration Geophysics Laboratory (EGL) scientists have developed a new concept for processing deep-water 4C OBC data that yields a significant improvement in the spatial resolution of P-P images. The fundamental theory is that, in deep water, the large elevation difference between a sea-surface source and a seafloor sensor allows P-P data to be processed in much the same way as standard Vertical Seismic Profiling (VSP) data. Experimental activity this period focused on developing and testing software that creates high-resolution P-P images of near-seafloor geology from deep-water 4C OBC seismic data.

Methane and sulfate concentrations, methane stable carbon isotopes, and nutrients were measured in 10 cores collected at MC118 to help understand the spatial variability in geochemical indicators at a hydrate seep site and the role biogeochemical processes play in hydrate stability. Analyses of spatial variability of biogeochemical processes affecting hydrate stability at MC 118 lays the foundation for the hydrate seafloor observatory where hydrate stability is assessed through long term monitoring of both seismic events and biogeochemistry.

Nine of the 10 cores analyzed did not show any evidence of a seep site whereas core 9 did show evidence for having hit a hot spot. Sulfate concentrations were 28mM at the sediment water interface and decreased to near zero around 50cmbsf. Methane concentrations were very low at the sediment water interface, began to increase at the depth of no sulfate and remained around 3.5mM at 75cmbsf for the remainder of the core, to 450cmbsf. However, these concentrations are still considered minimums due to gas expansion during core retrieval. Carbon isotopic composition of the methane showed a subsurface minimum of -70‰ at 50cmbsf. Below this depth, the isotopic signature becomes more enriched in 13C to a value of -49.73 ± 1.11‰ (n=10), indicating a thermogenic source gas from deep below.

When induction times decrease, hydrates form more quickly. This is important when gases percolating through sediments have a limited residence time. After hydrate formation is initiated, higher formation rates result when conditions favor hydrate formation. Variations in ease of hydrate formation were observed at similar near-surface depths on the sea-floor of MC, possibly as a result of the variations in extent and depth of the sulfate zone, ascending with the sulfate zone in close proximity to gas hydrate outcrops.

In addition to the accomplishments of the subcontractors, administration of the Monitoring Station/Sea-floor Observatory project, including DOE-funded components has been more complex than in most funding cycles. Because we expected to sponsor and participate in several test and deployment cruises in the summer and fall of 2005, we did not schedule a semiannual meeting as we usually do for the September/October time period. Rather, we devoted much time and effort to arranging the cruises, reserving vessels, arranging schedules, participants, onboard space, facilities, etc. and even establishing a newsletter, “Cruise News” to keep would-be participants informed.

Unfortunately, these efforts have been fruitless, as all activities in the northern Gulf have been suspended in the wake of the incredible hurricane damage suffered there. Available vessels have all been drafted for service in recovery and research efforts put on "hold" until some semblance of order can be restored to the area. A semiannual meeting will take place later in the fall.

EXPERIMENTAL

Experiments are described in the individual reports submitted by the subcontractors and included in the "Results and Discussion" section, which follows.

RESULTS AND DISCUSSION

Results and discussion of those results are described in the individual reports submitted by the subcontractors. Reports from the subcontractors follow.

Software Design for Simulation, Matched-Field Inversion, and Monitoring for the Gulf of Mexico Hydrates Seafloor Observatory

Version 1.0

Barrodale Computing Services Ltd. (BCS)
Hut R, McKenzie Ave., University of Victoria
Victoria BC V8W 3W2 Canada
www.barrodale.com

April 26, 2005

Table of Contents

Executive Summary

This Report by Barrodale Computing Services Ltd. (BCS) for the University of Mississippi Gulf of Mexico Hydrate Research project provides a software design for simulation, matched-field inversion, and real-time acoustic array monitoring for the Gulf of Mexico Hydrates Seafloor Observatory. The main purpose of this monitoring station is to provide a system that will allow the ongoing observation of the hydrate-containing sub-bottom layer by analyzing acoustic data generated by nearby sources of opportunity. This acoustic energy will typically be produced in the 50 – 250 Hz band, and will be analyzed by matched-field inversion (MFI) using a number of discrete frequencies within this band.

MFI will be applied to estimate the geoacoustic environment in the region of the array. In this approach, the acoustic fields required for matching will be generated using the normal-mode acoustic propagation model ORCA. ORCA will be used to simulate both time-domain traces and frequency-domain data for a range-independent environment, and for performing MFI on this data. As a check, and to allow investigation of the effects of range dependence on the range-independent assumption, the parabolic equation code RAM (Range-dependent Acoustic Model) will also be used to simulate data for MFI using ORCA. In the proposed approach, it will be assumed, at least initially, that there are sectors or regions close to the array within which the environment may be approximated well by a range-independent model.

Following a description of some properties on the environment and the propagation modeling, designs for the proposed components of the software system are presented. The design for each component consists of:

- a flowchart of the computational stages involved and the associated data;
- a Graphical User Interface (GUI) for setting up and running the component;
- a pseudocode description of the algorithm used;
- a list and description of the modules (routines) required to implement the component.

The system components described in this report are as follows:

- ***Simulation of shot data***. This component will use the broadband option of ORCA to generate impulse responses at the array for a source at a specified location. These will be convolved with a wavelet to generate simulated traces for a shot. This simulation will provide synthetic shot data to test MFI calibration procedures.
- ***Simulation of ship data***. This component will use ORCA to generate synthetic frequency domain array data at selected frequencies for a source at a specified location. It will also provide for addition of noise from several noise models, and for cross-spectral matrix estimation. A further module in this component will allow the generation of data for a simple range-dependent environment using RAM. The synthetic data will be used for MFI and will

allow investigation of the characteristics of the inversion with respect to the parameters.

Preprocessing of shot data. This component will fast Fourier transform (FFT) the traces for the shots (simulated and/or real data) and will save the results for selected frequencies for use in MFI.

Preprocessing of ship data. This component will overlap, window, and FFT consecutive data segments, and will estimate cross-spectral matrices from this data for selected frequencies. The results will be used in MFI.

Ambiguity function generation. This component will allow the generation of 1D and 2D ambiguity functions using the Bartlett power processor as the matching function. This will allow investigation of how the function depends on the individual parameters, and pairs of parameters, and will assist in defining conditions for successful MFI.

Test bed for MFI investigations. This component will implement MFI, allowing the estimation of parameter values for a range-independent geoacoustic model, using a search-optimization approach. For the optimization, a gradient-based algorithm will be used, with numerical approximation of the derivatives. If time permits, a recently developed derivative-free method will also be evaluated. This test bed will allow the investigation and characterization of the MFI procedures and parameters, and will provide a framework for defining conditions for detecting large-scale changes in the sub-bottom layers.

Monitoring software. This component will implement a three-stage procedure for real-time analysis of the data. In the first stage, matches will be determined along a 1D grid of range values, for each of a number of sectors. In the second stage, these estimates will be used as initial values in a full optimization of all the significant parameters of the geoacoustic model for each sector. At some (third) stage in the future, the results of these optimizations should be analyzed for statistical significance based on a statistical model; note, however, that the actual development of such a model is outside the scope of the current project. Significant differences from the standard models (based on the calibration) could then be reported to the user.

Following the description of the above components, the data structures that will be required by these components are then described. Opportunities for parallel processing are then discussed, and an outline for a suitable hardware system is presented.

The software will be developed in IDL[1]. The reasons for this choice are that:

IDL provides a powerful environment that allows rapid software development, with an extensive mathematical routine library, as well as built-in visualization and GUI tools.

---

[1] Interactive Data Language, from Research Systems, Inc. (owned by ITT Industries - see www.rsinc.com)

IDL can call external executables written in C or Fortran.
BCS has existing MFI software written in IDL that will be adapted and extended for the present application.
BCS has recently developed software for running IDL applications in a distributed processor environment using the freely-available IDL Virtual Machine.

The designs and approaches proposed in this report will provide a comprehensive framework for examining the effectiveness of various MFI procedures for modeling the environment of the monitoring station, and for developing the software that will perform continuous monitoring of the real-time array data to be produced by the station.

Introduction

The Gulf of Mexico Hydrates Research Consortium and the Center for Marine Resources and Environmental Technologies are currently developing a multi-sensor Seafloor Observatory to be installed on the continental slope of the northern Gulf of Mexico. The aim of this station is to monitor and investigate the hydrocarbon system within the hydrate stability zone of the northern Gulf of Mexico, and to remotely observe changes in the physical and chemical parameters of gas hydrates. The intention has been to equip the station with a variety of sensors that would enable the determination of a steady-state description of physical, chemical and thermal conditions in its local environment, as well as to detect temporal changes of those conditions. Major components of this Seafloor Observatory are geochemical instruments, temperature sensors, accelerometers, and an array of hydrophones that will collect acoustic data.

Barrodale Computing Services Ltd. (BCS) has been contracted by the University of Mississippi to design and develop data management and processing software for this monitoring station. In two reports written during the initial stages of this project, BCS has characterized the data to be produced by the station[2], and has proposed a design for a data management and archiving system for these data[3]. In the present and remaining stages, BCS is required to design and develop software for simulating the data to be acquired by the vertical acoustic array of the station, and for analyzing these data using matched-field inversion (MFI) techniques. The ultimate aim of this analysis is to use acoustic energy emitted by sources of opportunity to monitor the sub-bottom layers in the region of the station, with the goal of detecting large-scale changes in the hydrate structures within these layers, should such changes occur. This report describes BCS's design for the simulation, MFI, and array data processing software to be developed to achieve this aim. This overall software system has been tentatively named BCOMFI, for Barrodale COmputing Matched-Field Inversion.

Our design for the *simulation software* includes components for generating time domain sensor traces from simulated shots as well as for computing frequency domain fields at the sensors for selected frequencies. The synthetic data will be generated using a range-independent environment representative of the region of the array. The ability to add noise to these data will also be provided. Also, preprocessing software will be written for transforming time domain data for both shots and passing ships into a form where it can be input to MFI. The major purpose of developing these simulations is to generate well-defined synthetic data for realistic source-array geometries and sub-bottom geoacoustic models of the site. These data will then be used to investigate the conditions under which changes in the sub-bottom layers could be detected by MFI.

---

[2] "Sensor and Data Characterization for the Gulf of Mexico Hydrates Monitoring Station" (Jan. 31, 2005)
[3] "Data Management Architecture Design for the Gulf of Mexico Hydrates Seafloor Observatory" (Feb. 28, 2005)

In our design for the *matched-field software*, the synthetic frequency domain data will be matched with replica data vectors generated for a test model, and the Bartlett power processor will be used to compute an ambiguity function for the match. Two basic approaches will be implemented for examining the ambiguity functions. The first approach will compute the ambiguity function as one or two parameters are varied along a regular 1D or 2D grid. Visualization and analysis of these results will allow the determination of parameter sensitivity and interdependence, and will provide an indication of the numbers of optima and other characteristics of the overall parameter space. The second approach will implement optimization techniques for matched-field inversion, where the matching function is optimized with respect to a selected set of parameters, which will include both the source-receiver geometry and the geoacoustic model. This software will then be used to determine the conditions under which the model and geometry can be estimated with reasonable confidence, and to use data obtained during calibration to generate "standard" range-independent models for each of several sectors centered at the array. It is further intended that the results of MFI runs will also provide a basis for the development of a statistical model that can be applied in change detection.[4]

The design for the *real-time data processing software* calls for a sector-based approach, to accommodate the fact that the environment is mildly range-dependent and a single range-independent model may not be suitable for all sectors. For each sector, the design involves three stages of processing. In the first stage, the matching function will be evaluated at the points of a 1D grid of range values, using the standard models based on calibration. In the second stage, the best matches in these grids will be used as starting points for full multi-parameter MFI using optimization techniques. In the third stage, systematic deviations in the resulting sub-bottom parameter estimates from the "standard" values will be analyzed for evidence of changes in the hydrate-containing layer.

In this report, the geoacoustic environment in the region of the proposed site is first described and a suggested parameterization of this environment is presented. Then the approaches to be taken for propagation modeling are outlined: it is proposed that the normal mode program ORCA will be used to generate the fields for a range-independent model, for both simulation and MFI, and the (much slower) parabolic equation program RAM (Range-dependent Acoustic Model) used will be used to simulate data for simple range-dependent non-elastic environments. The designs for the simulation, MFI and real-time processing components are then presented; each design includes a diagram of the overall process, an image of the proposed GUI, a statement of the algorithm, and a description of the associated program modules. The proposed data structures to be used by these components are then listed and their tags are described. The report concludes with a description of strategies for parallelizing

[4] Note, however, that the development of such a statistical model is outside the scope of the current project.

the real-time computations, and for the hardware environment for development and final implementation of the processing systems.

Geoacoustic Environment

Description

In order to formulate effective designs for the functionality of the software to be developed, it was necessary to obtain some information about the general geoacoustic environment in which the monitoring station is to be deployed. At a meeting of the Hydrates Research Consortium in November 2004, a desirable site for the station, in Mississippi Canyon Block 118, was identified by the attendees. Early in 2005, a final choice of the location within this block was made, and several seismic surveys were performed along four different tracks. These surveys, and associated data, indicated that:

- The depth at the site was about 875 m.
- There was a bottom slope of about 2 degrees, with depth increasing to the southeast.
- The sloping region immediately surrounding the site was reasonably planar, but there was a canyon running northwest-southeast about 1.5 km to the northeast, and a flatter plateau, which showed high acoustic reflectivity, about 2 km to the south.
- There was a strong sub-bottom reflection about 200 msec below the bottom reflection, with two or three weaker reflections above this region and occasional strong deeper reflections. The reflection at 200 msec was thought to represent the bottom of the hydrate stability zone.
- There was a strong sub-bottom domed feature 1.5 – 2 km to the southwest of the site.
- There were several bright spots in the seismic sections, also at about the 200 msec horizon. These might correspond to pockets of gas at the base of the hydrate stability zone.

These data clearly show that this environment is significantly range-dependent, both with respect to the water depth (approximately planar with a 2-degree slope) and the characteristics of the 200-msec horizon. Because of this range dependence, a key question is the extent to which this environment can be satisfactorily modeled by a range-independent propagation model such as ORCA. The approach taken in this project is to assume that there is some region close to the station within which the environment may be taken to be range-independent. This assumption both simplifies the parameterization and reduces the time required for matched-field inversion using optimization.

Based on the above considerations, we have formulated a design for a comprehensive software system that will allow generation of synthetic data for range-independent environments and analysis of synthetic and real data using matched-field inversion, again assuming a range-independent environment. The design for the analysis includes the ability to generate 1D and 2D ambiguity functions for specified parameters, and to perform parameter optimization. In addition, to allow further investigation of the effects of simple range dependence, the design also provides for the ability to generate synthetic data for a sloping-bottom range-dependent environment and to analyze these data using a range-independent model.

Parameterization

It is essential to define a geoacoustic parameterization that will be acceptably detailed and realistic for use in modeling and monitoring of the environment, yet be sufficiently simple and sparse to allow matched field inversion to be performed in a reasonable time frame. Based on the above description of the environment, a set of variables was defined that consisted of source-array geometry parameters, a single sediment layer that could have a gradient in its acoustic parameters, and a basement[5] layer with constant parameter values. The proposed parameters to be used (and optimized) for MFI of the environment are as follows:

Source depth. This will generally be close to the surface (0.5 – 5 m) for both shot data and sources of opportunity, but should be included for investigative purposes.

Source range. Matching is known to be strongly dependent on range. As noted above, it is anticipated that for reasonably close source ranges (say, 1 – 3 km) a range-independent propagation model may be used to estimate the source range.

Source bearing. For a vertical array in a range-independent environment, bearing is not resolvable (the fields are bearing-independent). However for a tilted array, bearing does have an effect and is potentially invertible.

Array tilt angle. Arrays will be tilted in practice, and it is known that this tilt has a large effect on the field matches. Optimization of this parameter (and the following parameter, i.e., tilt direction) will considerably enhance the matching. For the present, it will be assumed that the tilted array is linear.

Array tilt direction. As for previous parameter.

Water depth. Variations in water depth due to tides, and the presence of some degree of range dependence require that water depth be a parameter in the inversion.

Sediment thickness. It is expected that this will be a key parameter for detecting changes at the base of the hydrate stability zone.

Compressional sound speed at top of sediment*. Changes in this parameter (or the following parameter, or both) could be indicative of changes that occur within the hydrate stability zone.

---

[5] The term “basement” is used here in a modeling, rather than geological, context.

*****An option will be provided in matched-field inversion to force the top and bottom parameters in the sediment to be equal (i.e., no gradient), to allow investigation of the effect of matching using a constant value when there is in fact a gradient in the layer.

Compressional sound speed at bottom of sediment*.* As for previous parameter.
Shear sound speed at top of sediment*.* Changes in this parameter (or the following parameter, or both) could also be indicative of changes that occur within the hydrate stability zone.
Shear sound speed at bottom of sediment*.* As for previous parameter.
Density at top of sediment*.* Although this parameter is generally quite insensitive with respect to matching, it should be included for investigative purposes.
Density at bottom of sediment*.* As for previous parameter.
Compressional sound speed in basement. Changes in this parameter could also reflect changes that occur at the base of the hydrate stability zone.
Shear sound speed in basement. As for previous parameter.
Density in basement. Although this parameter is generally quite insensitive with respect to matching, it should be included for investigative purposes.

Note that compressional and shear attenuations are not included as parameters since they are known to be very insensitive. Also, it is proposed that the sound-speed profile in the water column will not be included in the parameter set, but rather, that a generic, seasonally appropriate, sound-speed profile will be used. (In practice, this profile may be adjusted during monitoring to increase its accuracy based on temperature data provided by the thermistors embedded in the acoustic array.)

The above parameterization should be sufficient to describe the source-receiver geometry and geoacoustic environment while still being tractable in terms of matched-field inversion.

Propagation Modeling

The ORCA normal mode acoustic propagation model (developed by Evan Westwood[6]) will be the "engine" used to compute the acoustic fields required for simulation and matched-field-inversion. For a given ocean environment, specified by the sound-speed profile in the water column and a geoacoustic profile of the ocean bottom, ORCA finds the normal modes and computes the acoustic field at the sensors of an array. The model includes the effects of sound-speed gradients in the water and the bottom layers, shear waves in the bottom layers, steep-angle propagation represented by leaky modes, and attenuation in the bottom layers. It may be used to predict narrowband or broadband propagation. ORCA is unique among underwater acoustic propagation codes because it is largely automatic: the user does not need to guess at any convergence parameters such as depth- or range-sampling

---

[6] Westwood E.K., An efficient broadband normal-mode model for acoustoelastic ocean environments, J. Acoust. Soc. Am., **96**, 3352 (1994); Westwood E.K., Tindle C.T., and Chapman N.R., A normal mode model for acoustoelastic ocean environments, J. Acoust. Soc. Am., **100**, 3631-3645 (1996).

resolutions. It is also computationally efficient, typically requiring a few tenths of a second (on a 3 GHz Windows computer) to compute a propagation model at frequencies of interest in the present application.

ORCA is written in Fortran, requires several input files, and has recently been modified to produce an output file with the field values at the sensors. Since the simulation and MFI code are to be developed in IDL, a communication protocol and data transfer scheme between IDL and ORCA must be defined. It is proposed that initially the IDL process will write out files with the information required by ORCA, and will then spawn an ORCA process that will read in these files and produce an output file containing the fields. The IDL process will then read in this file and use the data in its internal computations.

If this approach proves to be too inefficient (i.e., if the time taken for the spawn is a substantial portion of the ORCA run time), then other methods will be investigated for more integrated communication between the processes. This may involve using the "call_external" feature of IDL, and modifying ORCA accordingly.

Despite its advantages and efficiency, ORCA is a range-independent propagation model and is not directly applicable to range-dependent environments. The environment in the array, however, is known to have significant range dependence, and it would be desirable to be able to investigate the effects of this at some stage of the project by simulating range-dependent data. While it might be possible to implement an adiabatic mode approximation using ORCA to achieve this, it would be preferable to implement a separate range-dependent code for performing simulations. In addition, such a code would provide an independent check on the correctness of the ORCA results, and to ensure that it is being used properly.

Accordingly, the RAM (Range-dependent Acoustic Model) parabolic equation model will be implemented to allow simulation of data in a simple range-dependent environment. This model, developed by Michael Collins[7], is also written in Fortran and so the same communication strategy as used for ORCA (i.e., IDL spawning a RAM process and reading in its output file) will be used for RAM.

Simulation of Shot Data

It is planned that immediately following the array installation, shots from a water-gun will be used to calibrate the site. This source will be towed behind a ship and detonated at shallow depth at regular intervals along several tracks. The hydrophones in the acoustic array will receive the data as individual traces, and

---

7 M.D. Collins, *Generalization of the split-step Pade solution*, J. Acoust. Soc. Am. **96**, 382-385 (1994);
M.D. Collins, R. J. Cederberg, D.B. King, and S.A. Chin-Bing, *Comparison of algorithms for solving parabolic wave equations*, J. Acoust. Soc. Am. **100**, 178-182 (1996).

this data will be analyzed by matched-field inversion to estimate the parameters described above. This data acquisition is expected to take place in the Fall of 2005.

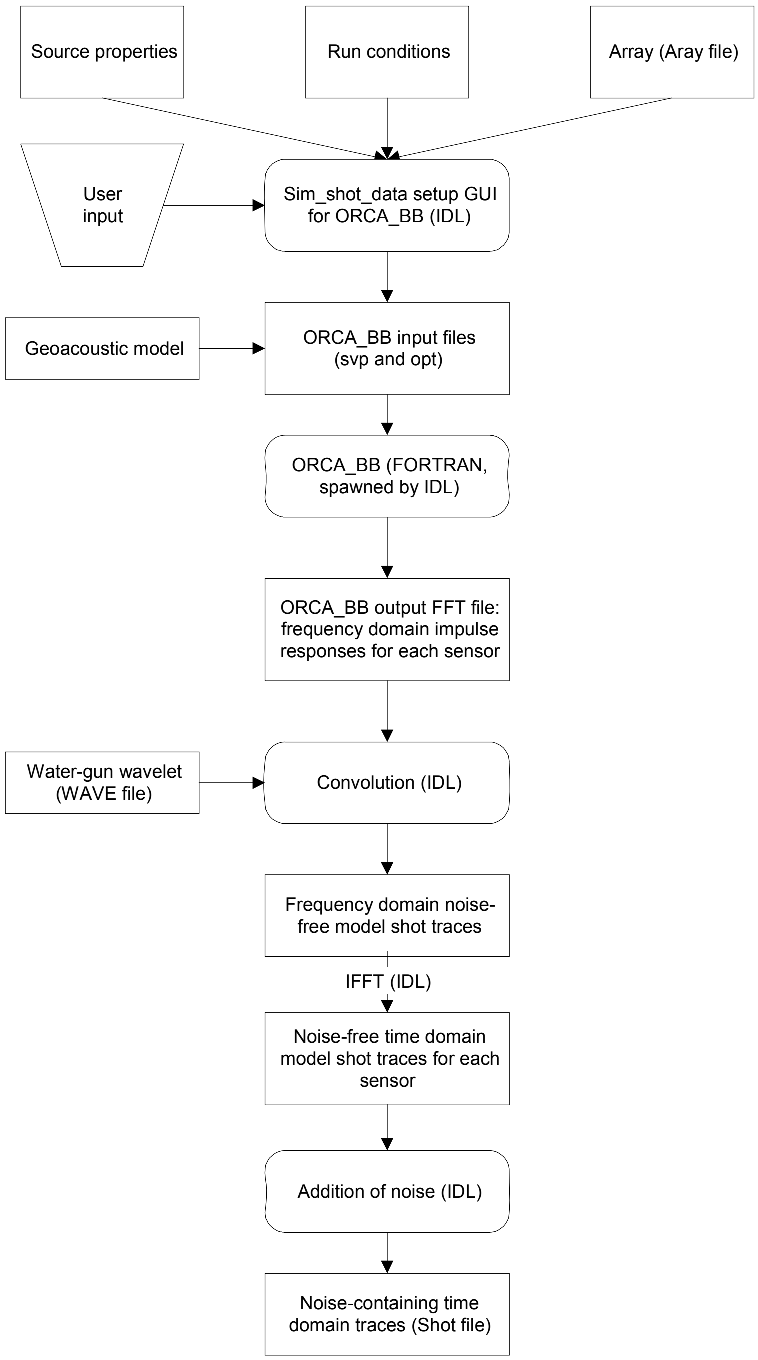
In order to develop and verify methods for MFI of the above shot data, it is essential to previously generate synthetic shot traces for specified source-array geometries and geoacoustic environments and use them to test the proposed MFI methods to be used. These synthetic data should consist of impulse responses (generated by propagation modeling) convolved with a representative source wavelet, with additive noise. The resulting model traces can then be FFT'd and selected frequencies used to perform and evaluate the subsequent MFI processing.

In the design described below for simulating shot data (the Sim_shot_data component of the BCOMFI system), the broadband option of ORCA will be used to generate the impulse responses. This option allows the user to specify a frequency band (e.g., 1 – 300 Hz), a sampling frequency, and a time window (or equivalently, a number of FFT points). ORCA will then generate the frequency domain impulse response for each discrete frequency in the specified band, and will output this result to a file. This FFT file will be read in and convolved with a specified (shot) waveform. The option to add Gaussian noise will be provided, and the resulting frequency domain traces will then be inverse FFT'd to yield the corresponding time domain traces.

Implementation of this software design for simulating shot data will allow the methods for processing the real shot data to be developed and tested in advance of the real data becoming available.

Overall Design

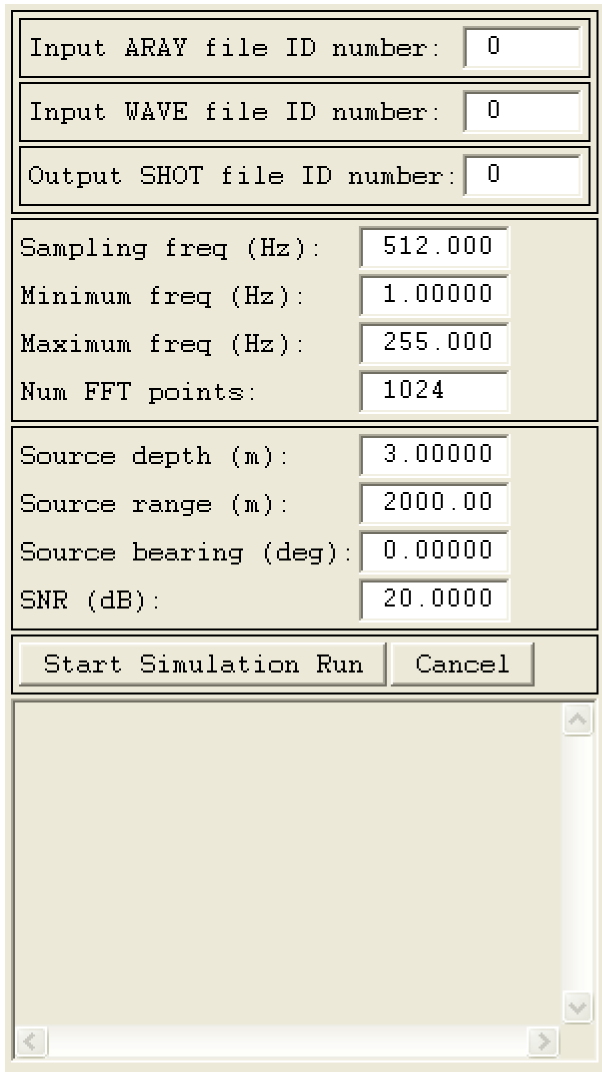
In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Shot Data Simulation: Sim_shot_data

Graphical User Interface

The proposed GUI for Sim_shot_data is illustrated in the following diagram.



The first three items allow the user to specify the input files (Aray and Wave) to be used for the run, and the Shot file to contain the results.
The next four items allow the user to specify the conditions for the broadband ORCA run.
The following four items (in the “Source” box) allow the user to specify the source-receiver geometry to be used in the run and the (power) signal-to-noise ratio (dB) for each trace.
The “Start Simulation Run” and “Cancel” buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for simulating the shot traces is outlined in the following pseudocode:

Set up an svp (sound velocity profile) file for ORCA to use as the geoacoustic environmental model.
Set up an Aray file to contain the array to be used in the simulation and a Wave file to contain the wavelet.
Using the GUI, specify the files to be used as the input array (Aray) and wavelet (Wave) files and the output Shot file, and specify the run conditions for the source and ORCA broadband.
Check that the run conditions are consistent.
Apply a tilt to the array.
Generate an array geometry file and an ORCA options file.
Spawn a process to perform an ORCA broadband run:
    ORCA process:
    For each frequency:
        Compute the complex fields that would be observed at each sensor for a source at the specified position.
    Output an ORCA FFT file containing the complex fields for the impulse responses for each of the sensors.
Read in the FFT file generated by ORCA.
Read in and FFT the wavelet from the specified Wave file.
Generate time domain traces: For each sensor:
    Reflect/conjugate the FFT about the Nyquist frequency to obtain the Fourier transform of the impulse response.
    Convolve the wavelet with the impulse response.
    Inverse FFT the result to give a time domain trace.
    Add Gaussian noise at specified signal-to-noise ratio.
Output the traces to a Shot file.

Modules

The following are the IDL modules that will be required to implement the Sim_shot_data component of the BCOMFI system.

Sim_shot_data. Sets up the GUI for specification of the files and input conditions.
Set_sim_shot_data. Initializes default values for the fields of the GUI.
Sim_shot_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Check_sim_shot_data. Checks for existence of files and consistency of the data entered by the user.
Run_sim_shot_data. Performs a run using the input data specified by the user, i.e., calling ORCA broadband, processing the resulting FFT file to convolve the impulse response with a wavelet, generating traces for each sensor of the array and adding noise to the traces.

Read_orca_fft_file. Reads in the frequency domain impulse responses for each sensor of the array.
Gen_array_geom_file. Uses the data in the Aray file to generate an array geometry file for use by ORCA.
Gen_bb_opt_file. Uses the data input by the user to generate an options file for ORCA to perform a run using the broadband option.

In addition, an executable ORCA program is required which will input the specified data and run conditions and output the FFT file.

The Sim_shot_data component uses the Aray, Wave, and Shot data structures. A description of these structures and their tags is given in a later section

Simulation of Ship Data

The main purpose of the monitoring station is to provide a system that will allow the ongoing observation of the hydrate-containing sub-bottom layer by analyzing acoustic data generated by nearby sources of opportunity. This acoustic energy will typically be produced in the 50 – 250 Hz band, and will be analyzed by MFI using a number of discrete frequencies within this band.

As with the shot data, it is essential to have a software component that will generate simulated data in order to test the correctness and performance of the MFI algorithms being developed. These synthetic data will consist of complex signal vectors or cross-spectral matrices at several user-selected frequencies, each of which can optionally contain noise corresponding to various models. Since there may be more than one actual source, and these sources will generally be moving, the simulation will also include the ability to model and generate data for at least two moving sources.

As mentioned above, the main thrust of the simulations will be to generate data for a range-independent environment. For this application, ORCA (in standard rather than broadband mode) will be used to generate the fields at the sensors that would be observed for a simulated moving source. However, in view of the fact that the environment has a significant range-dependence, it would be of considerable interest to also be able to simulate data for a simple range-dependent environment, and analyze this data with MFI using a range-independent model (ORCA). The range-dependent parabolic equation (PE) code RAM provides the ability to generate data for range-dependent environments, and is freely available over the web. This RAM code will be obtained and incorporated into the software system for generating synthetic data. The availability of two entirely separate propagation modeling codes will also provide a consistency/validity check by allowing comparison of the fields that are produced by the two models.

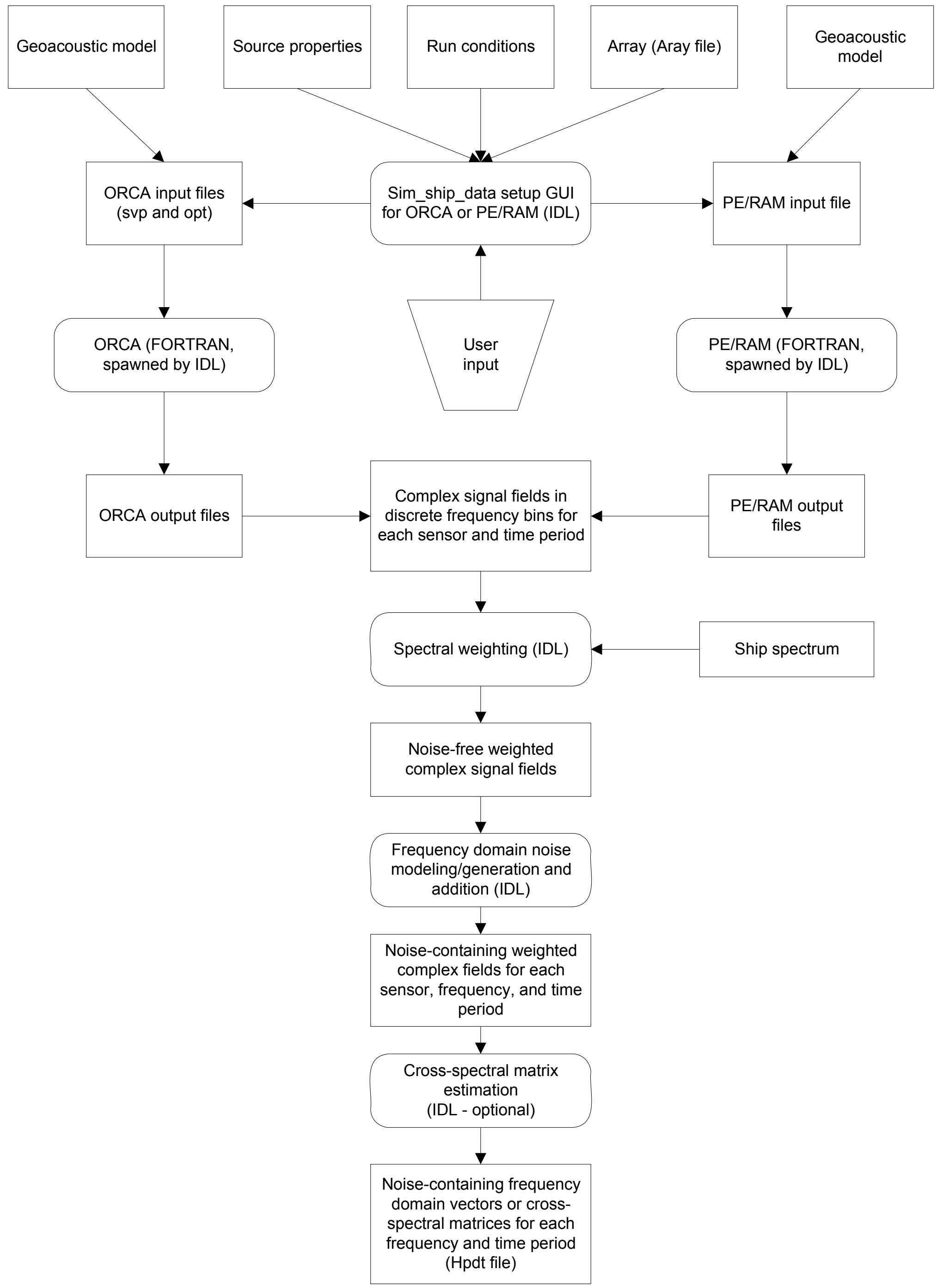
In the design described in this section for simulating ship data, both ORCA and RAM options will be used to generate acoustic fields at multiple frequencies for moving sources. There will also be options provided for adding noise and performing cross-spectral estimation.

Implementation of this software design for simulating ship data will allow the methods for processing the real ship data to be developed and tested in advance of the real data becoming available.

Finally, it is noted here that the above simulation scheme is based largely in the frequency domain (but with time domain simulation of source motion). Since the actual data that will be generated by the array will be time domain, it may be desirable at some point to simulate time domain ship data. This could be done by using the algorithm to generate shot data, and randomizing the phase appropriately prior to inverse FFT. At each frequency a random phase perturbation could be generated and applied to each of the sensors. This would result in data that is coherent in space, but not in frequency, which is a customary assumption in MFI.

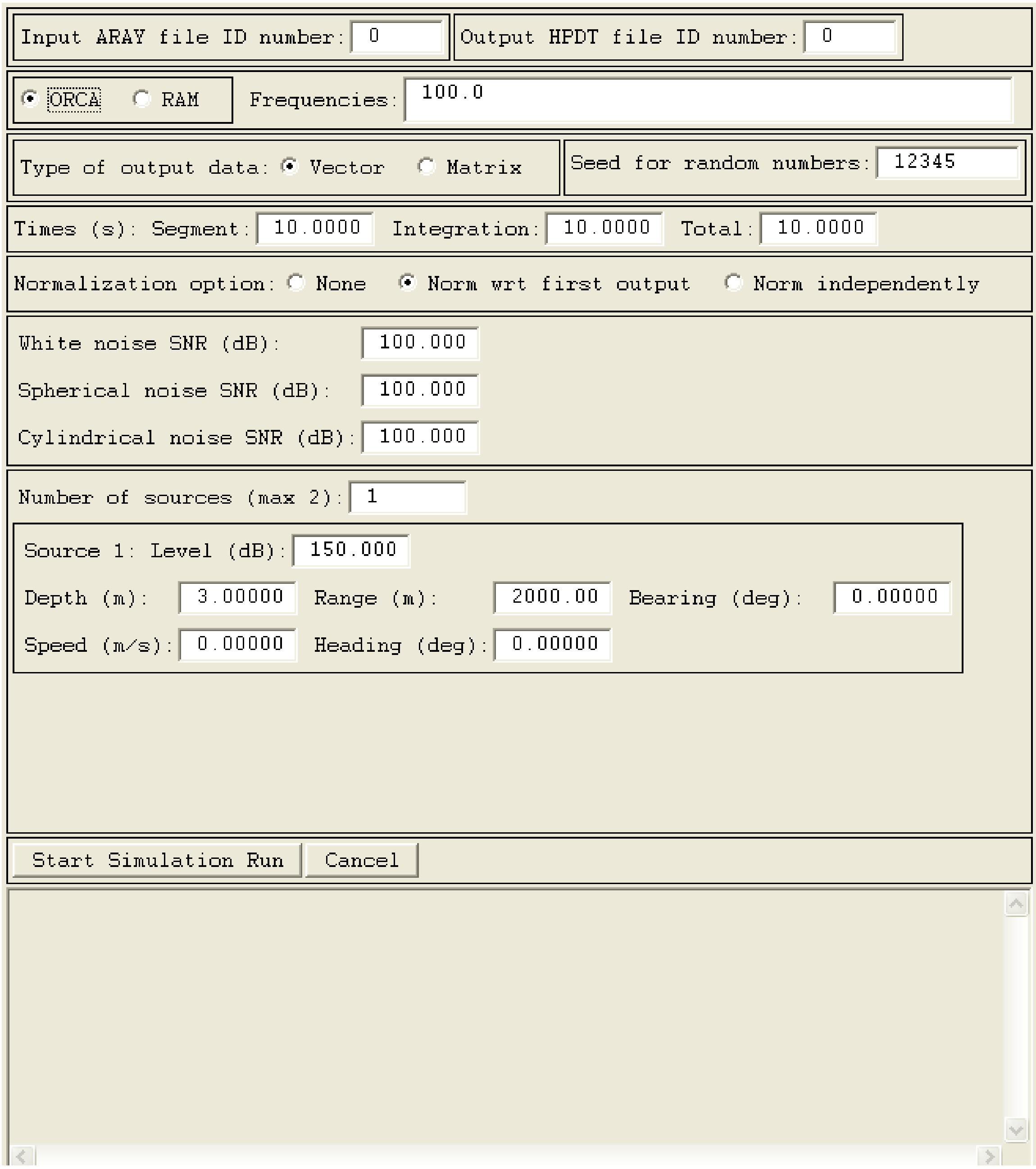
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Ship Data Simulation: Sim_ship_data

Graphical User Interface



The first items in the top row allow the user to specify the input Aray file to be used for the run, and the Hpdt file to contain the results.
The second row allows the user to indicate whether ORCA or RAM is to be used, and to specify the frequencies for which the data are to be generated.
The third row allows the user to specify whether the data are to be output as vectors or cross-spectral matrices, and to enter a seed for the random

number sequence to be used in generating noise realizations. The latter allows repeated runs involving noise to be reproducible.

The fourth row provides for specification of the segment time, the integration time (the time over which cross-spectral matrix estimation takes place), and the total time (which controls how many vectors/matrices are computed).

The fifth row provides options for normalizing the output.

The sixth row provides options for adding noise of specified signal-to-noise ratio from three different distributions: white, spherical, and cylindrical.

The seventh row allows one or two sources to be specified.

The eighth row (with the “Source 1” label) allows the user to specify the source level and the source-receiver geometry for Source 1 to be used in the run.

The ninth row (if present) allows the user to specify the source level and the source-receiver geometry for Source 2 to be used in the run.

The “Start Simulation Run” and “Cancel” buttons are self-explanatory.

The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for simulating the ship data is outlined in the following pseudocode:

```
Set up an svp (sound velocity profile) file for ORCA and/or RAM, to use as
    the geoacoustic environmental model.
Set up an Aray file to contain the array to be used in the simulation.
Using the GUI, specify the input Aray file and the output Hpdt file, and
    select the run conditions.
Check that the run conditions are consistent
Apply a tilt to the array.
For ORCA, generate an array geometry file and an options file.
For each frequency specified:
    Generate and Cholesky decompose the noise matrices for white,
    spherical, cylindrical, and modal noise at that frequency.
For each integration time:
    For each segment time:
        For each source:
            Compute new source position
            For each frequency:
                Run ORCA or RAM to generate a signal vector.
                Scale the signal vector and randomize the phase.
            For each frequency:
                Generate a random noise vector for the array sensors using
                    the noise cross-spectral matrix for that frequency and
                    array.
                Add noise vector to signal vector.
            For each frequency:
```

Accumulate cross-spectral matrices if this option chosen.
Normalize matrices if this option specified.
Output the vector/matrix data to an Hpdt file.

Modules

The following are the IDL modules that will be required to implement the Sim_ship_data component of the BCOMFI system.

Sim_ship_data. Sets up the GUI for specification of the files and input conditions.
Set_sim_ship_data. Initializes default values for the fields of the GUI.
Sim_ship_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Check_sim_ship_data. Checks for existence of files and consistency of the data entered by the user.
Run_sim_ship_data. Performs a run using the input data specified by the user, i.e., calling ORCA or RAM, adding noise, performing cross-spectral matrix estimation if specified, and writing the data to an Hpdt file.
Gen_sim_orca_sv. Sets up files for an ORCA run and then calls ORCA to simulate a signal vector at an array for a particular source, array, and frequency.
Gen_ram_sv. Sets up files for a RAM run and then calls RAM to simulate a signal vector at an array for a particular source, array, and frequency.
Gen_wn_matrix. Generates a white noise matrix.
Gen_sn_matrix. Generates a spherical noise matrix.
Gen_cn_matrix. Generates a cylindrical noise matrix.
Chol_matrix. Performs a Cholesky decomposition on the input matrix.
Gen_nv. Generates an estimated noise vector, based on the Cholesky decomposition.
Gen_array_geom_file. Uses the data in the “Aray” file to generate an array geometry file for use by ORCA.
Gen_cw_opt_file. Generates an options file for the ORCA run in cw mode.
Gen_ram_file. Generates an input data file for RAM.
Write_hpdt. Writes out the Hpdt data structure to an Hpdt file.

In addition, executable ORCA and RAM programs are required which will input the specified data and run conditions and output the complex fields to be used as signal vectors.

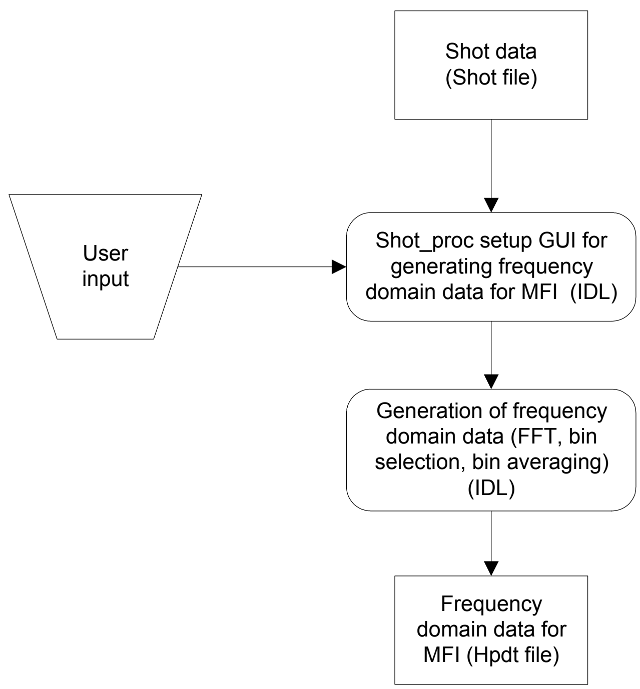
The Sim_ship_data component uses the Aray and Hpdt data structures. A description of these structures and their tags is given in a later section.

Preprocessing of Shot Data

The data for each shot will be processed separately into a form amenable to MFI analysis. The strategy for processing the shot data will be to FFT the traces for the sensors and form signal vectors at user-specified frequencies. These vectors, which will be coherent in frequency as well as space, will be output to an Hpdt file, which will in turn be used as input for the ambiguity function and MFI components of BCOMFI.

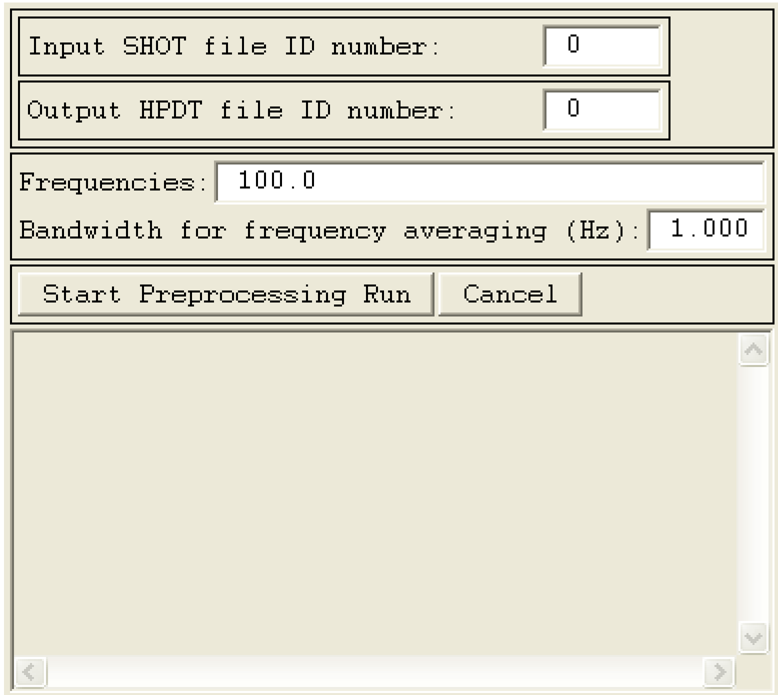
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Shot Data Preprocessing: Prep_Shot_Data

Graphical User Interface



The first items in the top row allow the user to specify the input Shot file to be used for the run, and the output Hpdt file to contain the results.

The next two lines allow the user to specify the frequencies at which the FFT'd shot data are to be retained and the bandwidth within which frequency averaging centered at these frequencies is to be performed.

The "Start Preprocessing Run" and "Cancel" buttons are self-explanatory.

The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for processing the shot data is outlined in the following pseudocode:

Specify the input Shot file and output Hpdt file.
Select the frequencies at which signal vectors are to be computed.
For each shot:
Decimate the data to reduce its effective sampling rate (optional).
FFT the traces for each sensor.
For each selected frequency:
Perform frequency averaging within a band of bins centered at the selected frequency (optional).
Form a signal vector from the values in the FFT bins for each hydrophone.
Write the signal vectors to an Hpdt file.

Modules

Prep_shot_data. Sets up the GUI for specification of the files and input conditions.
Set_prep_shot_data. Initializes default values for the fields of the GUI.
Prep_shot_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Check_prep_shot_data. Checks for existence of files and consistency of the data entered by the user.
Run_prep_shot_data. Performs a run using the input data specified by the user, i.e., reading in and FFTing the shot data, and writing the signal vector data to an Hpdt file.

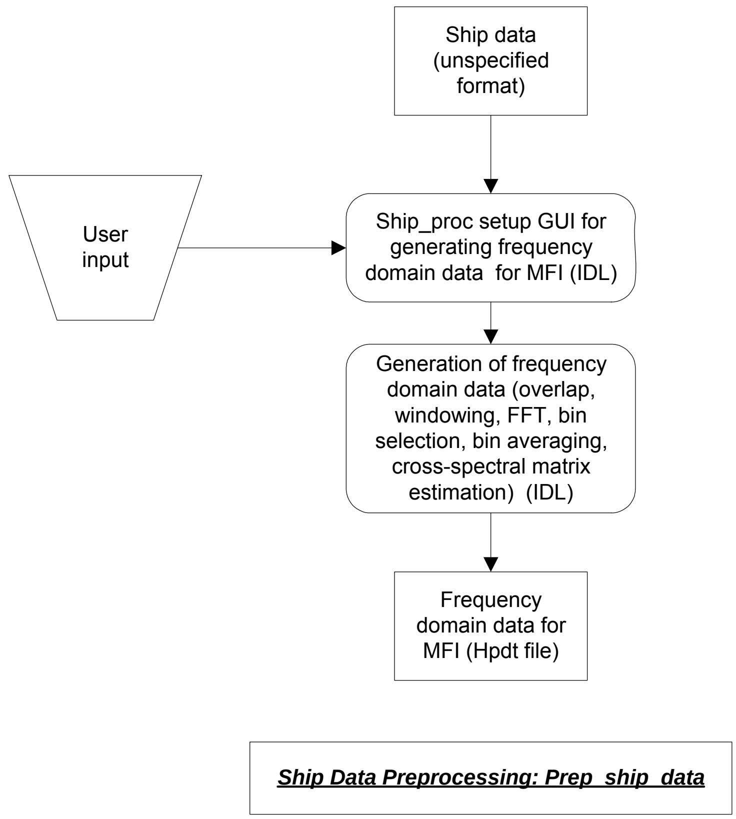
Preprocessing of Ship Data

While the shot data will be obtained and analyzed as a series of discrete traces, the ship noise data will be continuous in character and will be analyzed differently. Because there may be more than one source, and to reduce the effects of noise, it is desirable to process the data for a certain period of time to form cross-spectral matrices at selected frequencies. Ideally, this time should be long enough to permit reasonable estimation of these matrices, but sufficiently short to still allow stationary methods to be used in MFI.

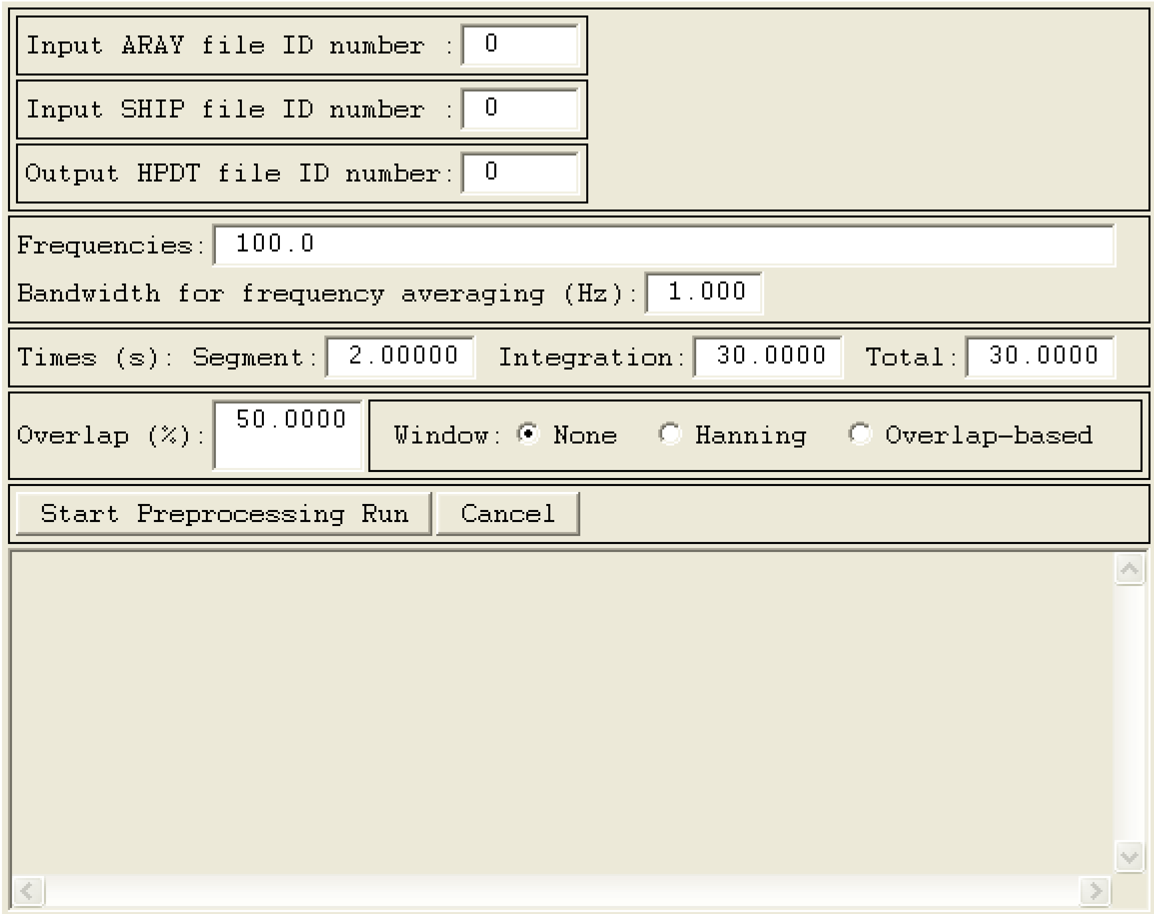
The measured time domain ship data will be processed as a series of (possibly overlapping and windowed) segments to yield cross-spectral matrices for MFI analysis. The strategy for processing the ship data will be to FFT the sensor traces for each time segment, possibly overlapping the time segments by 50% and applying a window. Signal vector data at user-specified frequencies will then be obtained from this data and used to form cross-spectral matrices, which will constitute the input to MFI.

Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Graphical User Interface



The items in the top row allow the user to specify the input Aray file and Ship data file to be used for the run, and the Hpdt file to contain the results.
The next two lines allow the user to specify the frequencies at which the FFT'd ship data are to be retained and the bandwidth within which frequency averaging centered at these frequencies is to be performed.
The third row provides for specification of the segment time, the integration time (the time over which cross-spectral matrix estimation takes place), and the total time (which controls how many vectors/matrices are computed).
The fourth row allows the user to specify the overlap to use for the time series to be FFT'd and the window to use prior to performing the FFT.
The "Start Preprocessing Run" and "Cancel" buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for preprocessing the ship data is outlined in the following pseudocode:

Specify the input data file and the output Hpdt file.
Select the frequencies at which signal vectors are to be computed.
Repeat for each integration time:
    For each sensor:
        Obtain the data for the segment (optional overlap).
        Window the data.
        FFT the data.
    For each frequency:
        Form a signal vector for the segment.
        Take the outer product of the signal vector and accumulate the cross-spectral matrix for that frequency.
    Output the cross-spectral matrices to an Hpdt file.

Modules

Prep_ship_data. Sets up the GUI for specification of the files and input conditions.

Set_prep_ship_data. Initializes default values for the fields of the GUI.

Prep_ship_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.

Check_prep_ship_data. Checks for existence of files and consistency of the data entered by the user.

Run_prep_ship_data. Performs a run using the input data specified by the user, i.e., reading in, overlapping, windowing and FFTing the ship data, performing cross-spectral matrix estimation, and writing the cross-spectral matrices to an Hpdt file.

Ambiguity Function Generation

In investigations of MFI, it is very useful to be able to examine how the ambiguity function (the matching function used as an objective function in MFI) depends on the individual parameters, and, sometimes, groups of parameters. To allow the visualization of this behavior, a component of BCOMFI has been designed that will allow the generation of 1D and 2D ambiguity functions (higher-dimension grids are too time-consuming to generate and more difficult to visualize). The user can then use display software to examine the characteristics of these functions, including dynamic range, peak widths, presence of multiple optima, parameter sensitivity, and, in the case of 2D ambiguity functions, parameter interdependency. This last item is of particular significance, since it can lead to ill-posed MFI problems and inconsistent results in parameter estimation.

Visualization of the ambiguity functions for the parameter space can assist in the interpretation of such results.

The Bartlett power processor will be used to compute the matching (ambiguity) function between the measured data and the replica vectors generated using ORCA. For multi-frequency measured data in vector form, this processor is defined as

$$B(\mathbf{r}) = \sum_{j=1}^{N} w_j \frac{|\mathbf{r}_j^* \mathbf{m}_j|^2}{|\mathbf{r}_j|^2 |\mathbf{m}_j|^2},$$

where $\mathbf{r}_j$ is the replica vector at the jth frequency, $\mathbf{m}_j$ is the measured vector, N is the number of frequencies, and w_j is a weight ($\sum w_j = 1$). For measured data in matrix form, the processor is

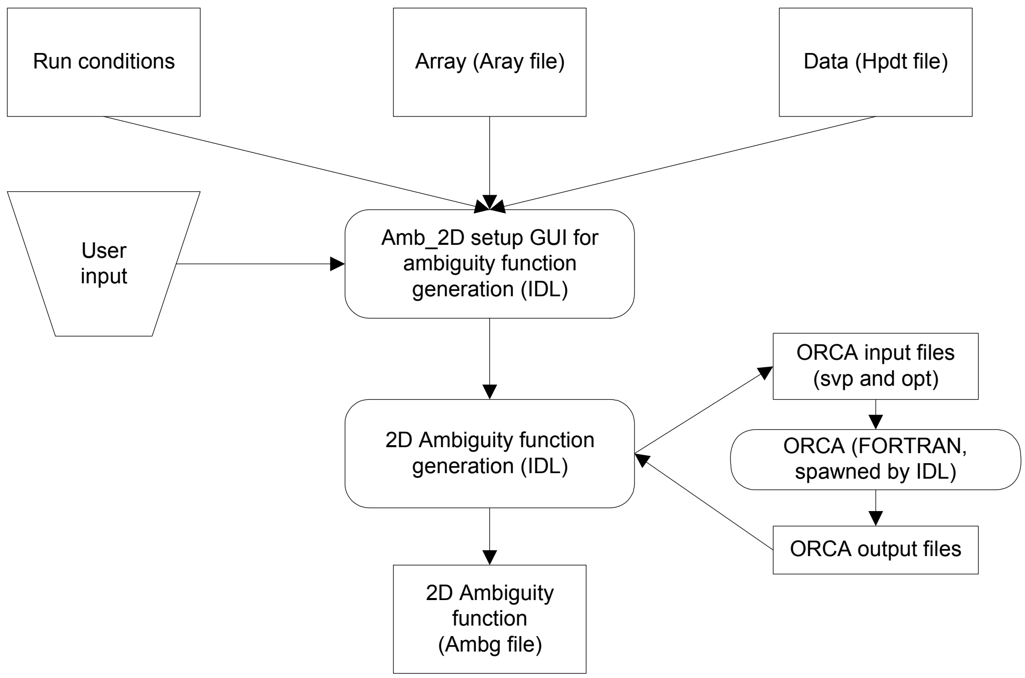
$$B(\mathbf{r}) = \sum_{j=1}^{N} w_j \frac{|\mathbf{r}_j^* \mathbf{M}_j \mathbf{r}_j|^2}{|\mathbf{r}_j|^2 \|\mathbf{M}_j\|},$$

where ||**M**|| is the spectral norm of **M**.

This section describes the proposed design of the ambiguity function generation component, using the 2D case as an example; the 1D version is just a simplified version of the 2D design.

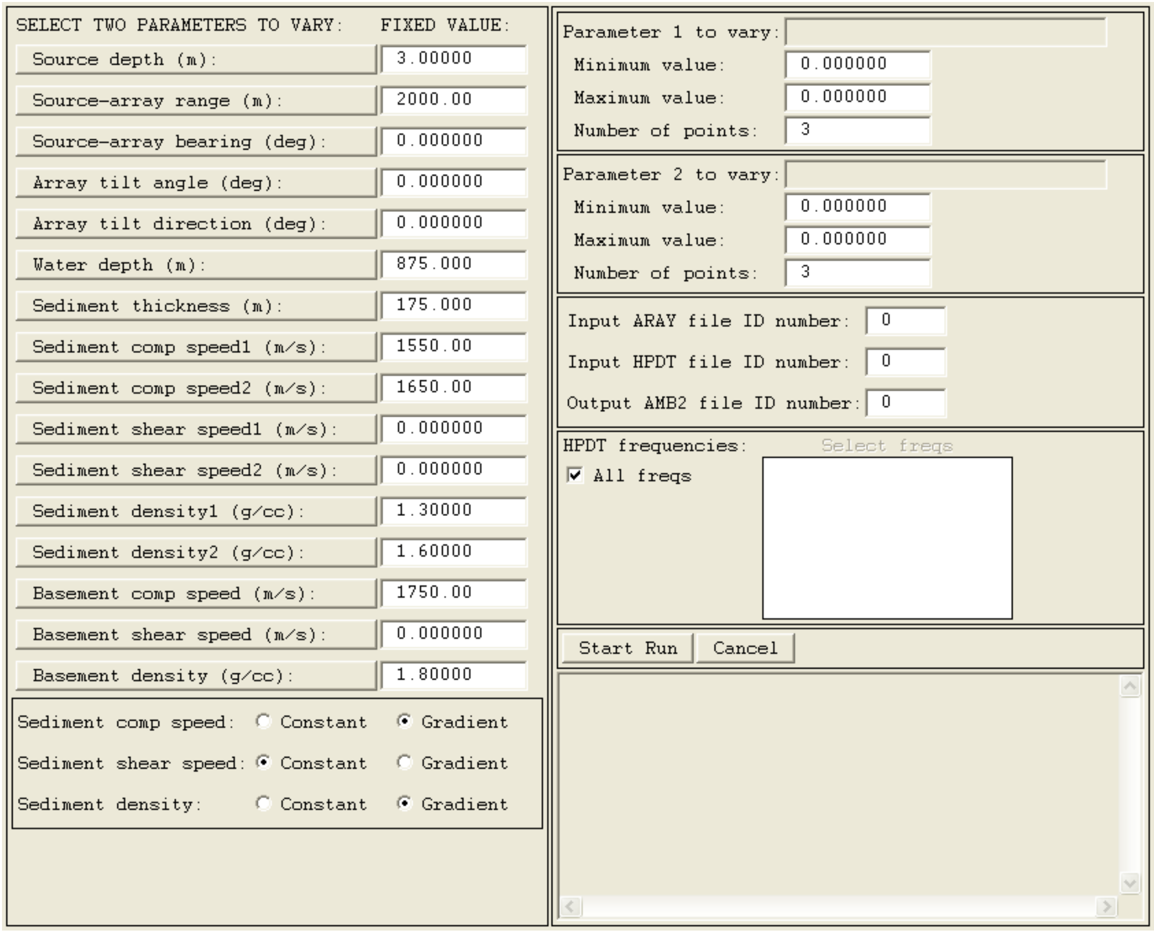
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



2D Ambiguity Function: Amb_2d

Graphical User Interface



The block on the top left lists the 16 parameters that can be varied to generate the ambiguity function, and provides default values for those parameters. Pressing on the button containing a parameter name chooses that parameter as one of the two to be varied and populates one of the “Parameter 1 to vary” or “Parameter 2 to vary” fields in the blocks at the top right.

The block on the bottom left provides an option for the user to force the indicated parameter pairs for the top (e.g., density1) and bottom (e.g., density2) of the sediment layer to be the same.

The top two blocks on the right allow specification of the domain over which the parameters are to be varied, and the number of points in each dimension.

The third block on the right allows the user to specify the input files (Aray and Hpdt) to be used for the run, and the Ambg file to contain the results.

The fourth block on the right allows the user to view the frequencies of the data in the Hpdt file and select a subset of these to use in computing the multi-frequency ambiguity function. An option to select all the frequencies in the data is also provided.

The “Start Run” and “Cancel” buttons are self-explanatory.

The large empty area at the bottom right provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for generating the 2D ambiguity function is outlined in the following pseudocode:

Specify the input Aray file and the output Hpdt file, and specify the run conditions.
Check that the run conditions are consistent.
Apply a tilt to the array.
For each value of parameter 1:
For value of parameter 2:
For each frequency selected:
Apply tilt values to the array positions.
Generate an array geometry file for ORCA.
Generate an opt file for ORCA.
Generate an svp file for ORCA.
Spawn an ORCA process to generate a replica vector.
Compute the Bartlett power of the match for that frequency.
Sum the weighted Bartlett powers for the frequencies to give the ambiguity function for (parameter 1, parameter 2).
Output the ambiguity function to an Ambg file.

Modules

Amb_2d. Sets up the GUI for specification of the files and input conditions.

Set_amb_2d. Initializes default values for the fields of the GUI.

Amb_2d_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.

Check_amb_2d_parms. Checks for existence of files and consistency of the data entered by the user.

Run_amb_2d. Performs a run using the input data specified by the user, i.e., sets up the values of the parameters and computes the matches at these values.

Gen_mfi_orca_sv. Sets up and performs an ORCA run to generate a signal (replica) vector based on the run conditions.

Gen_array_geom_file. Uses the data in the Aray file to generate an array geometry file for use by ORCA.

Gen_cw_opt_file. Uses the data input by the user to generate an options file for ORCA to perform a run using the continuous wave option..

Gen_svp_file. Generates an svp file for ORCA, based on the parameter values.

Bartlett. Computes the output of the Bartlett power processor, for either vector or matrix data.

Matched-Field Inversion Test Bed

The purpose of this component is to provide a test bed for investigation of MFI techniques and determine which approaches will be most effective for detecting changes within the hydrate stability zone. Using synthetic data generated by the above schemes, the MFI test bed will allow investigation of the following questions, for example:

- What are the relative sensitivities, peak widths, oscillations, and dynamic ranges of the matching function with respect to each of the 16 parameters?
- How do the above parameter characteristics vary with range and frequency?
- Can we estimate the numbers of optima in the entire search region?
- Can we reliably ignore density optimization in the matching?
- What is the effect of some of the parameters, including the sound-speed profile in the water column, being in error?
- What is the effect of allowing a gradient in the sediment when the parameter is constant, or forcing it to be constant when there is in fact a gradient?
- What is the effect of multiple sediment layers in the data above the hydrate stability zone, when matching is done with only one layer?
- What is the effect of additional layers below the hydrate stability zone, when the matching is done with only one layer?
- What is the effect of noise of various types, including modal noise from distant shipping?
- What is the effect of multiple sources and source motion?
- What is the effect of gas at the lower interface?
- Under what conditions is regularization required to obtain consistent inversion results?
- Under what conditions and for what parameters can we expect to detect significant changes that would indicate alterations in the hydrate-containing layer? That is, how substantial would changes have to be before they could be reliably detected?
- How do the answers to the above questions change when we move to a range-dependent environment?

The test bed will also provide an environment for the analysis of real data (both shot and ship) with the intended result of defining a standard model, or perhaps a sector-dependent set of models, that represent the environment in the region of the array. These models will then be used as a basis in the real-time monitoring software described in the next section.

In designing the test bed, we considered that MFI is a nonlinear process that is generally approached using optimization techniques that repeatedly solve the forward problem for varying sets of parameters until a suitable good match to the data is obtained. MFI optimization approaches must be able to deal with the following challenges:

- There are typically 5 – 20 parameters, and it is required to be able to optimize any or all combinations of these.
- There are generally multiple local optima present in the parameter space.
- It is desirable to restrict the domain of the parameters, usually by bounds constraints.
- The sensitivities of the parameters can be very different (by a factor of 100 or more).
- The parameters will need to be scaled for optimization.
- Derivatives are unavailable, except by numerical approximation.
- Certain parameters can be interdependent or "correlated", leading to ill-posed problems where widely different choices of parameters can give almost identical locally optimal matches.
- There is a possibility of discontinuities in the matching function, particularly if the propagation algorithms do not always successfully converge/complete (as has been observed to be the case with ORCA under some conditions).

The approach that we propose in implementing optimization in the MFI test bed is to use a search/gradient-optimization technique. This method involves an initial search stage in which the parameter space is sampled, and a second optimization stage in which each of a specified number of the best matches found in the search stage is used as a starting point for optimization. Various algorithms can be used for the optimizations, including the Nelder-Mead simplex method (which does not require derivatives) and quasi-Newton methods (for which the required derivatives can be approximated numerically). The advantages of the search/gradient-optimization approach are the ability to obtain multiple estimates of the same or different optima, and the relatively small number of function evaluations required (e.g., 500 for search space sampling and 2000 for 10 optimizations). While a potential drawback of this method it that the estimates are local, a sufficiently comprehensive search state will increase the likelihood that one of the optima is in fact global. For reasons of efficiency, it is planned that the quasi-Newton variant of the method will be provided in the MFI test bed. If time permits, a new and very promising derivative-free optimization will be evaluated and incorporated into the test bed.

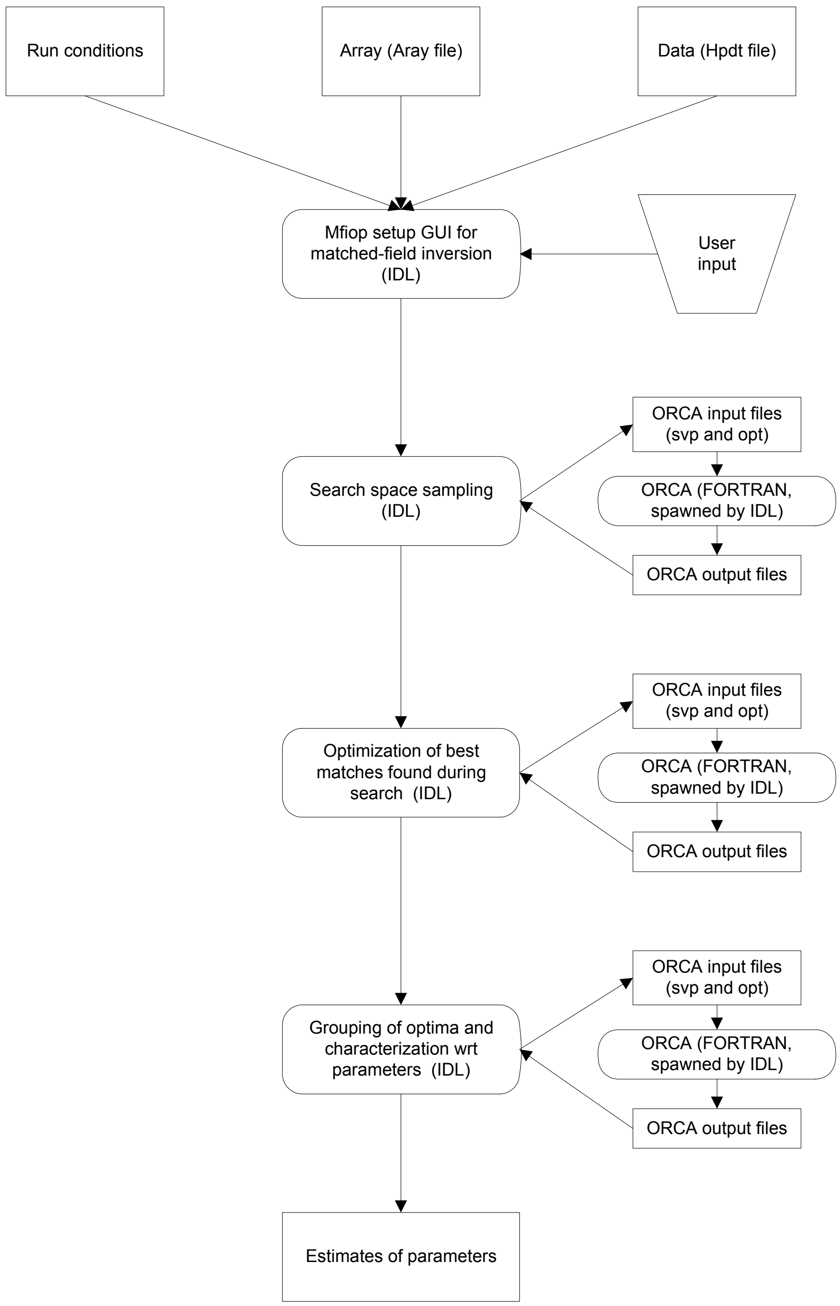
It is also planned that the ability to perform certain post-optimization analysis will be provided in the test bed. This will include grouping of multiple estimates, and characterization of the parameter space in the region of the optimum, e.g., for each parameter axis estimating the peak width, sensitivity, number of extrema, and dynamic range.

The ultimate result of the investigations conducted using the test bed will be to compute range-independent geoacoustic models of the environment in the region of the monitoring station. Based on the data acquired during the calibration stage, it is likely that these models will be sector-dependent; i.e., a different model may be derived by MFI for each of several sectors centered at the array.

These models will then be used as standards in the monitoring / change detection software application described in the next section.

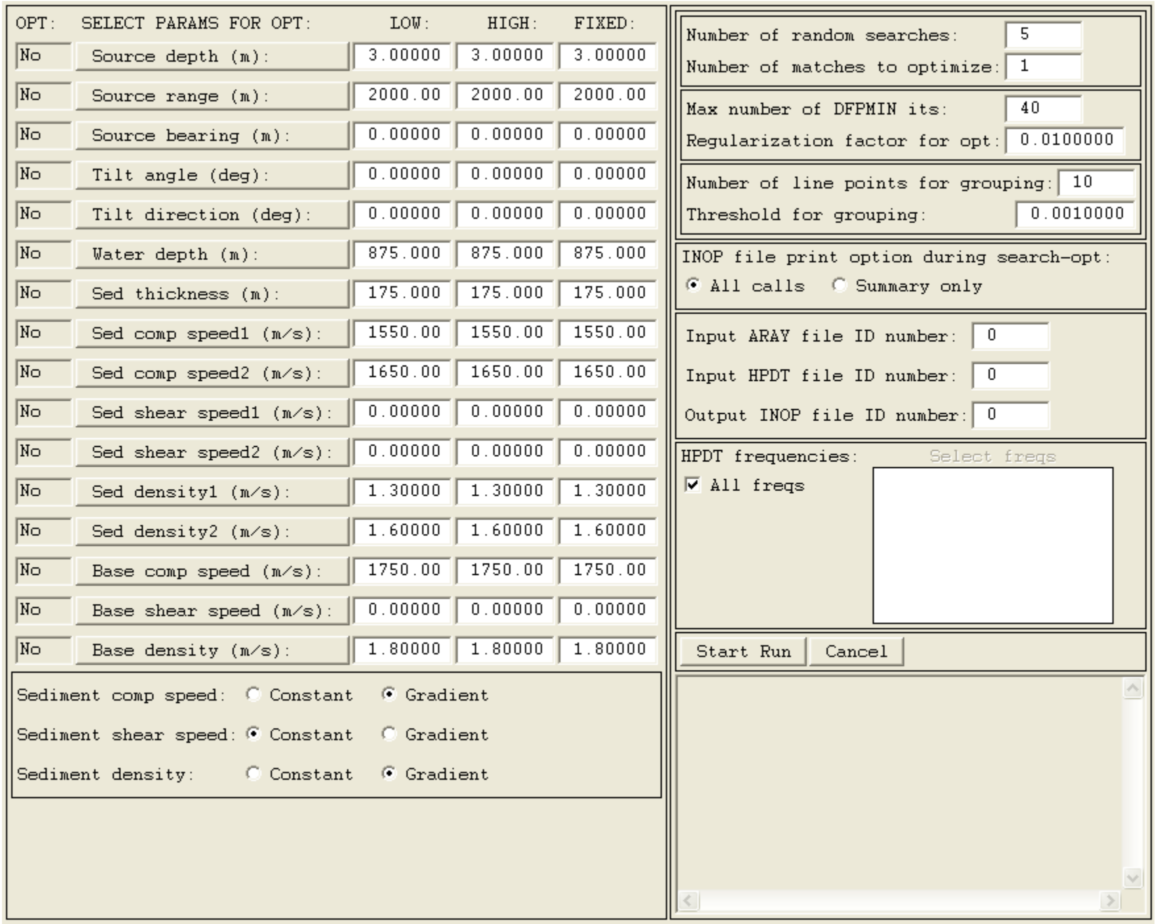
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Matched-Field Inversion: Mfiop

Graphical User Interface



The block on the top left lists the 16 parameters that can be chosen for optimization and provides bounds and default values for those parameters. Pressing on the button containing the parameter name chooses that parameter as one which is to be optimized.

The block on the bottom left provides an option for the user to force the indicated parameter pairs for the top (e.g., density1) and bottom (e.g., density2) of the sediment layer to be the same.

The top block on the right allows the user to specify the conditions for the search stage, i.e., the number of random samples of the parameter space and the number of best matches to optimize.

The second block on the right allows specification of convergence and regularization factors for the optimization.

The third block on the right allows the user to specify conditions for the grouping analysis of the multiple peaks.

The fourth block on the right provides print verbosity options.

The fifth block on the right allows the user to specify the input files (Aray and Hpdt) to be used for the run, and the Mfop file to contain the results.

The sixth block on the right allows the user to view the frequencies of the data in the Hpdt file and select a subset of these to use in computing the multi-frequency ambiguity function. An option to select all the frequencies in the data is also provided.

The "Start Run" and "Cancel" buttons are self-explanatory.

The large empty area at the bottom right provides a field for status messages to be displayed.

Algorithm

The algorithm to be used to perform matched-field inversion using the search/gradient optimization method is outlined in the following pseudocode:

Specify the input Aray file and the output Hpdt file, and specify the run conditions, including those parameters that are to be optimized and their bounds.

Check that the run conditions are consistent.

Apply a tilt to the array.

Generate specified number of random (or Latin square) samples of the search space and rank the results in order of the best matches.

For each of a specified number of the best matches:

Using the current best match as an initial estimate, call the optimization function to optimize the parameters (see objective function below for more details).

Identify those optima which are estimates of the same peak and combine them to form unique estimates.

Perform post-optimization analysis on the best optimum, including peak width, etc., estimates for each parameter.

Write results to an Mfin file.

The algorithm for computing the objective function, given a set of parameters provided by the optimization algorithm, is as follows:

For each frequency selected:

Apply tilt values to the array positions.

Generate an array geometry file for ORCA based on the input parameters.

Generate an opt file for ORCA based on the input parameters.

Generate an svp file for ORCA based on the input parameters .

Spawn an ORCA process to generate a replica vector.

Compute the Bartlett power of the match for that frequency.

Sum the weighted Bartlett powers for the frequencies to give the ambiguity function for the input parameters.

Modules

Mfiop. Sets up the GUI for specification of the files and input conditions.
Set_mfiop. Initializes default values for the fields of the GUI.
Mfiop_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Check_mfiop_parms. Checks for existence of files and consistency of the data entered by the user.
Run_mfiop. Performs an MFI run using the input data specified by the user, i.e., performs the initial search, optimizes the objective function for multiple starting estimates and performs the grouping and peak analysis.
Mfi_func. Evaluates the objective function to be minimized in MFI. This is a composite function consisting of the sum of the Bartlett processor and a penalty function if any parameter exceeds its bounds.
Mfi_dfunc. Estimates the gradient of the objective function using central differences.
Gen_mfi_orca_sv. Sets up and performs an ORCA run to generate a signal (replica) vector based on the run conditions.
Gen_array_geom_file. Uses the data in the "Aray" file to generate an array geometry file for use by ORCA.
Gen_cw_opt_file. Uses the data input by the user to generate an options file for ORCA to perform a run using the continuous wave option..
Gen_svp_file. Generates an svp file for ORCA, based on the parameter values.
Gen_grp. Groups multiple optima corresponding to the same peak.
Analyze_peak. Performs post-processing on the best optimum to estimate peak width, sensitivity, etc., for each parameter.
Bartlett. Computes the output of the Bartlett power processor, for either vector or matrix data..

Real-Time Monitoring Software

The aim of the software for real-time monitoring is to use an array to continuously measure the acoustic field that is produced by nearby sources of opportunity, and to analyze this data to detect changes in the hydrate-containing sub-bottom layers. The array data will be generated (and stored) continuously, and will be processed according to the design below. At this stage, this design is necessarily preliminary, and it is expected that it will be revised and refined based on results and experience that will be obtained in the later stages of this project, i.e., after experimentation using the simulation and MFI test bed software systems described above.

In this connection, it should be borne in mind that the initial model proposed for the sub-bottom (i.e., a single range-independent sediment layer with gradients) is an approximation of the true structure, which will inevitably be more complicated.

Although it is anticipated that this model will give reasonable results for MFI, it is certainly possible that more complex models will be required to provide a sufficiently good fit to the data.

The basic approach to the real-time monitoring will be to first divide the local region into a number of sectors centered on the array, and use the initial calibration data to derive several standard range-independent geoacoustic models of the local region – one for each sector. Then, as a ship passes nearby and the acoustic intensity reaches a certain threshold, the real-time data will be processed to generate cross-spectral matrices, and these matrices will be analyzed by MFI in a three-stage procedure.

In the first stage, matches will be determined along a 1D grid of the geometric parameter range. This will be done for each sector using the standard geoacoustic models for the sectors. Replica vectors will be generated for each range on the grid, using the sensor locations predicted by the orientation sensors along the array. This process will result in the localization of the source in range (provided it is sufficiently close).

In the second stage, these estimates will be used as initial values in a full optimization of all the significant parameters of the geoacoustic model for each sector. The result will be a set of optimized geoacoustic models of the sub-bottom sediment and basement layers.

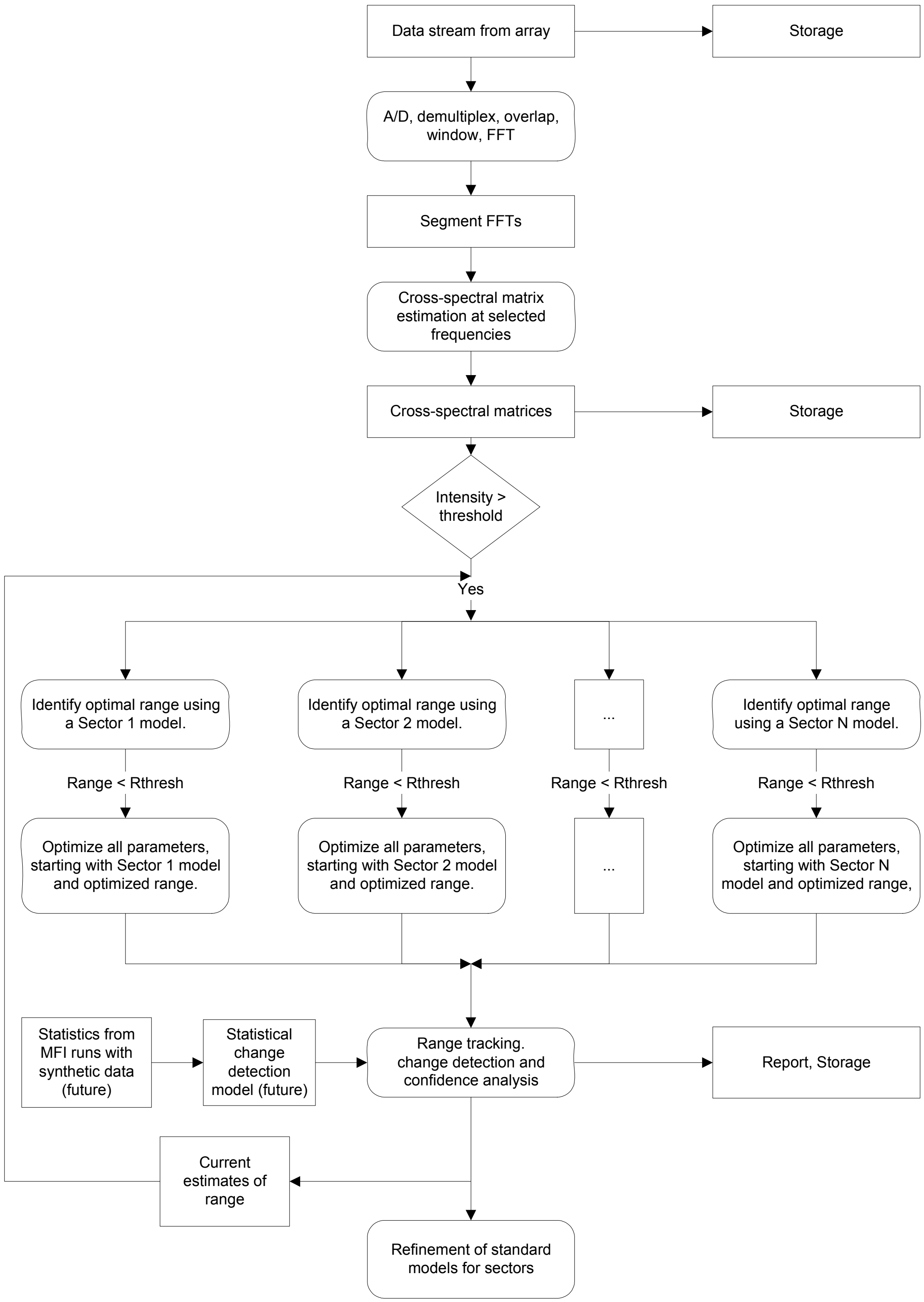
In the (future) third stage, the results of these optimizations will be analyzed for statistical significance based on a statistical model to be devised based on the results of simulation and MFI.[8] If differences between the optimized results and the standard models for the sectors are found to be consistently statistically significant, then this will be taken as evidence for a possible change in the sub-bottom hydrate-bearing structures.

---

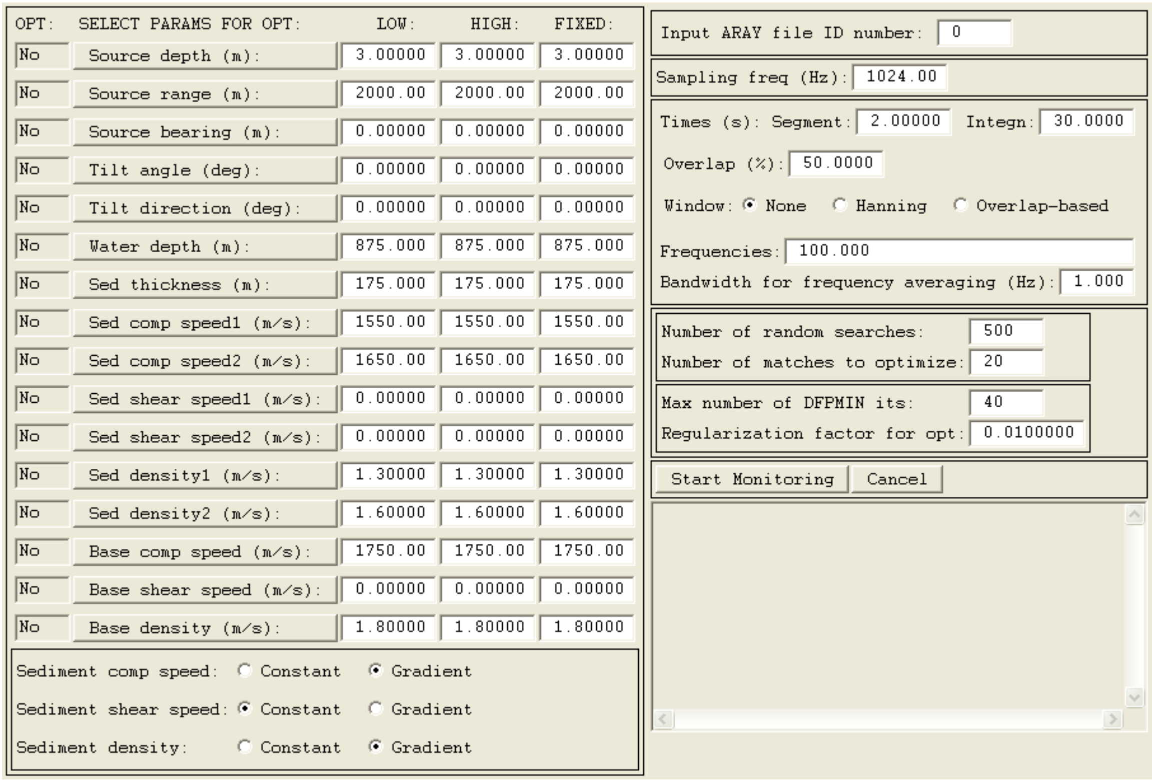
8 Recall that this development is outside the scope of the current project.

Overall Design

In this design, data and/or files are indicated by rectangles and processes are represented by rounded rectangles.



Graphical User Interface



The block on the top left lists the 16 parameters that can be chosen for optimization and provides bounds and default values for those parameters. Pressing on the button containing the parameter names chooses that parameter as one which is to be optimized.

The block on the bottom left provides an option for the user to force the indicated parameter pairs for the top (e.g., density1) and bottom (e.g., density2) of the sediment layer to be the same.

The top block on the right allows the user to specify the input Aray file to be used for the monitoring.

The second row on the right allows the user to specify the sampling frequencies for the data.

The third row on the right provides for specification of the processing conditions, i.e., the segment time, the integration time (the time over which cross-spectral matrix estimation takes place), the overlap, the type of window, and the frequencies for which the data are to be generated, and the bandwidth within which frequency averaging centered at these frequencies is to be performed.

The fourth block on the right allows the user to specify the conditions for the search stage, i.e., the number of random samples of the parameter space and the number of best matches to optimize.

The fifth block on the right allows specification of convergence and regularization factors for the optimization.

The “Start Monitoring” and “Cancel” buttons are self-explanatory.

The large empty area at the bottom right provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for performing the processing of real-time data monitoring is outlined in the following pseudocode:

Specify the geoacoustic models to be used as standard models for each sector.
Select the frequencies at which matching is to be done (i.e., signal replica vectors are to be computed).
Repeat indefinitely:
  Repeat for each integration time:
  For each sensor:
    Obtain the next segment of data from the array (e.g., 2 sec time), with optional overlap.
    Window the data.
    FFT the data.
  For each frequency:
    Form a data vector with the sensor fields for the segment.
    Take the outer product of the data vector and accumulate the cross-spectral matrix for that frequency.
  Output the cross-spectral matrices to file.
  For each sector:
    For each selected frequency:
      Evaluate match (ambiguity function) at each point of a 1D range grid, using sensor locations predicted from the orientation sensor data.
    Combine matches for the selected frequencies and identify the optimum range.
  If range match found, then, for each sector:
    Optimize full or partial set of parameters using the optimum range, and the standard geoacoustic model (or the optimized geoacoustic parameters from the previous integration time) as initial estimates. Alternatively, use standard values as initial estimates for the geoacoustic parameters.
    Test whether the estimates for the geoacoustic parameters are different from the standard values. Include this information in the report of the results of the real-time analysis.

Modules

Monitor_data. Sets up the GUI for specification of the files and input conditions.

Set_monitor_data. Initializes default values for the fields of the GUI.

Monitor_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.

Check_monitor_data. Checks for existence of files and consistency of the data entered by the user.

Run_monitor_data. Starts the real-time data continuous monitoring process as outlined in the Algorithm section above.

Gen_range_match. Computes the ambiguity function for the 1D grid of range values that applies to a particular sector.

Mfi_func. Evaluates the objective function to be minimized in MFI optimization. This is a composite function consisting of the sum of the Bartlett processor and a penalty function if any parameter exceeds its bounds.

Mfi_dfunc. Estimates the gradient of the objective function using central differences.

Gen_mfi_orca_sv. Sets up and performs an ORCA run to generate a signal (replica) vector based on the run conditions.

Gen_array_geom_file. Uses the data in the "Aray" file to generate an array geometry file for use by ORCA.

Gen_cw_opt_file. Generates an options file for ORCA to perform a run using the continuous wave option..

Gen_svp_file. Generates an svp file for ORCA, based on the parameter values.

Bartlett. Computes the output of the Bartlett power processor, for either vector or matrix data..

Gen_model_stats. Performs statistical tests on the parameter values obtained from optimization to determine whether it is likely that the estimated geoacoustic model differs significantly from the standard model for some sector.

Data Structures

We note here that these structures also correspond to files, such that a file with a particular four-character prefix (e.g., "aray0023.dat") would hold a data structure of the same name (an "aray" structure).

Ambg

The Ambg structure will hold the data for the 1D and 2D ambiguity functions, and will contain the following tags:

num_param. The number of parameters to vary (1 or 2).

param1. The first parameter to vary.

init1. The lower limit of the domain to vary param1.

final1. The upper limit of the domain to vary param1.

incr1. The increment for varying param1.
num1. The number of points for varying param1.
param2. The second parameter to vary (for 2D only).
init2. The lower limit of the domain to vary param2 (for 2D only).
final2. The upper limit of the domain to vary param2 (for 2D only).
incr2. The increment for varying param2 (for 2D only).
num2. The number of points for varying param2 (for 2D only).
sed_comp_speed_constant. A flag indicating whether the compressional speed in the sediment is to be held constant.
sed_shear_speed_constant. A flag indicating whether the shear speed in the sediment is to be held constant.
sed_density_constant. A flag indicating whether the density in the sediment is to be held constant.
hpdt_id. The ID number for the input Hpdt file containing the data for MFI.
aray_id. The ID number for the input Aray file containing the array geometry.
ambg_id. The ID number for the output Ambg file to contain the ambiguity function.
all_freq. A flag indicating whether all the frequencies in the Hpdt file are to be used for matching.
num_freq. The number of frequencies in the Hpdt file that are to be used for matching.
freq. The frequencies in the Hpdt file that are to be used for matching.
use_freq. A flag indicating which of the frequencies in the Hpdt file (in order) are to be used for matching.
ambg_data. A vector or 2D array containing the ambiguity function.

Aray

The Aray structure will hold the data for the array, and will contain the following tags:

description. A text description of the array type of characteristics.
num_sens. The number of sensors in the array.
tilt_angle. The tilt angle from vertical (degrees).
tilt_direction. The direction in which the array is tilted (degrees true).
sens_x. The *x*-coordinates of the sensors (before tilt is applied).
sens_y. The *y*-coordinates of the sensors (before tilt is applied).
sens_z. The *z*-coordinates of the sensors (before tilt is applied).
tether_z. The tether depths of the sensors (allowing the application of tilt).

Hpdt

The Hpdt structure will hold the simulated and real frequency domain data to be used for input to MFI, and will contain the following tags:

hpdt_id. The ID number for the Hpdt file containing the data for MFI.
aray_id. The ID number for the Aray file containing the array geometry.
num_sens. The number of sensors in the array from which the data were obtained.
num_freq. The number of frequencies at which data are present.
freq. A vector containing the actual frequencies at which data are present.
data_type. A string with a value of either "vector" or "matrix".
seed. A seed for the random number generator (used for noise and phase randomization).
num_seg_int. The number of segments in one integration time.
num_int_tot. The total number of integration times.
seg_time. The time for a single data segment.
int_time. The integration time for the data.
tot_time. The total time for the full set of data.
norm. An indicator of the type of normalization applied to the data (0 = none; 1 = norm with respect to first data set; 2 = norm independently).
wn_level_db. The white noise level at a sensor.
sn_level_db. The spherical noise level at a sensor.
cn_level_db. The cylindrical noise level at a sensor.
num_source. The number of sources (max 2).
source_level_db. The source intensity levels (dB re 1 μPa at 1 m).
source_depth. The depth(s) of the source(s).
source_range. The range(s) of the source(s).
source_bearing. The bearing(s) of the source(s).
source_speed. The speed(s) of the source(s).
source_heading. The heading(s) of the source(s)
mdata. A multi-dimensional array containing the complex fields for the sensors, frequencies, and times.

Mfop

The Mfop structure will hold the conditions, for and the results of, an MFI run, and will contain the following tags:

sed_comp_speed_constant. A flag indicating whether the compressional speed in the sediment is to be held constant.
sed_shear_speed_constant. A flag indicating whether the shear speed in the sediment is to be held constant.
sed_density_constant. A flag indicating whether the density in the sediment is to be held constant.
hpdt_id. The ID number for the input Hpdt file containing the data for MFI.
aray_id. The ID number for the input Aray file containing the array geometry.
mfop_id. The ID number for the output Mfop file to contain the results of the MFI.
all_freq. A flag indicating whether all the frequencies in the Hpdt file are to be used for matching.

num_freq. The number of frequencies in the Hpdt file that are to be used for matching.
freq. The frequencies in the Hpdt file that are to be used for matching.
use_freq. A flag indicating which of the frequencies in the Hpdt file (in order) are to be used for matching.
print_all. A flag indicating the verbosity of the output to screen and the Mfop file.
optimize. An *M*-element vector of flags indicating whether the *m*th parameter is to be optimized.
lower. An *M*-element vector containing the lower bounds for the parameters to be optimized (and the fixed values for those that are not to be optimized).
upper. An *M*-element vector containing the upper bounds for the parameters to be optimized (and the fixed values for those that are not to be optimized).
num_search. The number of random samples of parameters to use in the search stage of the algorithm.
num_best. The number of best matches from the search stage that are to be optimized during the optimization stage of the MFI.
num_its. The maximum number of iterations for the DFPMIN (Davidon-Fletcher-Powell) optimization algorithm to perform.
reg_factor. The regularization factor to apply in computing the objective function.
num_inter. The number of intermediate points on a hyperspace line between two optima along which to evaluate the objective function in order to detect whether the optima are estimates of the same peak and should be grouped.
thresh_group. The threshold used to determine whether two optima (minima) are estimates of the same peak. If the objective functions at the two optima differ by this amount or more, or if any point on the above line connecting the optima is more that this amount greater than the optimum value at the endpoints, the optima are taken to be different and are not grouped.

Shot

The Shot structure will hold the traces for a single shot, and will contain the following tags:

aray_id. The ID number for the Aray file containing the array geometry.
wave_id. The ID number for the Wave file containing the wavelet.
shot_id. The ID number for the Shot file containing the traces for the shots.
num_sens. The number of sensors in the array.
samp_freq. The sampling frequency for the traces.
min_freq. The minimum frequency in the band for which ORCA broadband should compute a normal mode model (must be positive).
max_freq. The maximum frequency in the band for which ORCA broadband should compute a normal mode model (must be less than samp_freq).
num_fft. The number of points in the FFT to be performed on the frequency domain data, and which corresponds to the number of points in the traces

(power of 2). Care should be taken to make this large enough to prevent wrap-around artifacts in the traces.
source_level_db. The source intensity level (dB re 1 μPa at 1 m).
source_depth. The depth of the source.
source_range. The range of the source.
source_bearing. The bearing of the source.
trace_data. A 2D (num_fft x num_sens) array containing the traces for the sensors.

Wave

The Wave structure will hold an acoustic wavelet, and will contain the following tags:
num. The number of elements in the wavelet.
samp_freq. The sampling frequency for the wavelet points.
wdata. A vector containing the elements of the wavelet.

Parallelization

To process the data during the monitoring, it is proposed that parallelization will be implemented at several levels, including:

The initial processing. This would include overlapping, windowing, FFT, and estimating cross-spectral matrices at selected frequencies. The data for each channel (i.e., sensor) could be transferred to a separate processor, and sampled frequencies be returned after the FFT to the master, which would then perform cross-spectral matrix estimation.
The 1D range grid evaluation. This would require *MN* processors, where *M* is the number of frequencies and *N* is the number of sectors. For each integration time, the appropriate cross-spectral matrix would be transferred, and the processor would then evaluate the matching function at the positions in the grid. The grid results would be returned to the master processor for frequency averaging within each sector.
The optimization. It is anticipated that there will be about 30 – 60 seconds available for performing an optimization (this corresponds to the integration time to form the quasi-stationary cross-spectral matrices). As the time to generate an ORCA model is a few tenths of a second (depending on frequency), it will be necessary to perform the optimizations for the various sectors in parallel, and also to perform the ORCA computations for the frequencies in parallel.

The parallelization of the initial processing and of the 1D range grid evaluation could be done either by installing the IDL Virtual Machine (IDLVM) on each processor and performing the computations using IDL, or by writing special-purpose C or Fortran code and installing that on the processors. To implement the parallelization of the optimization processing, an ORCA executable would be installed on each of the processors. Coordination of the master and slave processes could be done either using an existing IDL-based data transfer utility written by BCS or by writing special-purpose C code.

Hardware Environment

The initial development (simulation and MFI test bed software components) will be done on a Windows computer running IDL and Fortran. The development of the monitoring software component will be performed on a Windows computer connected to a LAN or cluster for the implementation and testing of the parallel computations. Each node in this cluster will have IDLVM and ORCA executables installed.

The final implementation could involve either a Windows or Linux cluster. It is proposed that the final choice should be made based on the experience to be obtained with the Windows cluster to be used during development, and an analysis of any issues encountered with that environment.

Software for Simulation and Matched-Field Inversion for the Gulf of Mexico Hydrates Seafloor Observatory

Version 1.0

Barrodale Computing Services Ltd. (BCS)
Hut R, McKenzie Ave., University of Victoria
Victoria BC V8W 3W2 Canada
www.barrodale.com

September 30, 2005

Table of Contents

Executive Summary

This report describes a software system, developed by Barrodale Computing Services Ltd. (BCS) under contract to the University of Mississippi, for simulation and matched-field inversion (MFI) of acoustic array data.

The purpose of the system, termed BCOMFI[9], is to provide a comprehensive and validated environment for investigating the application of MFI techniques to detect changes in the sub-bottom gas hydrate deposits under the sea floor in the region of the monitoring station of the Gulf of Mexico. This approach is based on the expectation that MFI analysis of acoustic array data originating from nearby sources of opportunity (passing ships) can be used to detect such changes. The basic principle is to derive geoacoustic models for the current sub-bottom regions of the station by applying MFI to data from calibration measurements, and then use these models to match with future data obtained from passing sources. The presence of a large mismatch would be taken as evidence of a change.

The BCOMFI software developed by BCS provides a suite of components for generating synthetic data simulating shots and ship noise, transforming these data into a form where they can be used for matched field techniques, and applying these techniques to analyze the data. Methods for simulating acoustic array data from shots or ship noise were developed first, and then used to support the development and validation of matched field methods under controlled conditions. The result was that these methods are now in place and ready to be applied when real data from the array become available (expected in 2006).

The software was developed in IDL[10]. The reasons for this choice were that:

- IDL provides a powerful environment that allows rapid software development, with an extensive mathematical library, as well as built-in visualization and GUI tools.
- BCS has extensive experience, accumulated over several years, in writing MFI software modules in IDL.
- IDL provides a cost-free run-time environment, the IDL Virtual Machine (IDL VM), which allows users to run IDL executables, including the present BCOMFI software, without having to purchase an IDL license.

The software also makes use of two Fortran programs, ORCA[11] and RAM[12], as

---

[9] for **B**arrodale **CO**mputing **M**atched **F**ield **I**nversion

[10] Interactive Data Language, from Research Systems, Inc. (owned by ITT Industries - see www.rsinc.com)

[11] Westwood E.K., *An efficient broadband normal-mode model for acoustoelastic ocean environments*, J. Acoust. Soc. Am., **96**, 3352 (1994); Westwood E.K., Tindle C.T., and Chapman N.R., *A normal mode model for acoustoelastic ocean environments*, J. Acoust. Soc. Am., **100**, 3631-3645 (1996).

[12] M.D. Collins, *Generalization of the split-step Pade solution*, J. Acoust. Soc. Am. **96**, 382-385 (1994);
M.D. Collins, R. J. Cederberg, D.B. King, and S.A. Chin-Bing, *Comparison of algorithms for solving parabolic wave equations*, J. Acoust. Soc. Am. **100**, 178-182 (1996).

engines for the acoustic propagation modeling computations. ORCA is a normal-mode code for range-independent environments, and RAM is a parabolic equation program that can be used for range-dependent environments. The use of two acoustic propagation codes provided versatility in the modeling and also furnished a means of checking and verifying the results of the simulations in range-independent environments.

The report is organized as follows:

- The background and context for the software development are outlined.
- Some properties of the environment in the region of the station are summarized, and the selected parameterization of this environment (which involves 23 geometric and geoacoustic parameters, with up to three layers) for matched field analysis is described.
- The methods used in the propagation modeling are presented.
- The installation and use of the software system are described in detail, including the files involved and the functionality of all of the components for simulation, ambiguity function generation, MFI and display. The description for each component includes:
  - a flowchart of the computational stages involved and the associated data;
  - a Graphical User Interface (GUI) for setting up and running the component;
  - a pseudocode description of the algorithm used;
  - a guide to using the component.
- The implementation of an option in BCOMFI that supports parallel processing for matched field computations is also described.
- The internal data structures and modules in the BCOMFI software are then described in two Appendices.

The BCOMFI software components described in this report are as follows:

- ***Simulation of shot data***. This component uses the broadband option of ORCA to generate impulse responses at the array for a source at a specified location. These are convolved with a wavelet to generate simulated traces for a shot. This simulation provides synthetic shot data to test MFI procedures.
- ***Simulation of ship data***. This component uses ORCA or RAM to generate synthetic frequency domain array data at selected frequencies for one or two moving sources. It also provides for addition of noise from several noise models, and for cross-spectral matrix estimation.
- ***Conversion of shot to ship data***. This component uses phase randomization to convert shot traces to simulated ship noise time series with the same spectra as the shots.
- ***Conversion of real data to shot or ship data***. This component converts real array data in p16 format to an internal format used for storing shot or ship data.

Preprocessing of shot data. This component performs a fast Fourier transform (FFT) on the traces for the shots and saves the cross-spectral matrices for selected frequencies for use in MFI.

Preprocessing of ship data. This component overlaps, windows, and FFTs consecutive data segments, and estimates cross-spectral matrices from this data for selected frequencies for use in MFI.

Ambiguity function generation. This component generates 1D, 2D or 3D ambiguity functions using the Bartlett power processor as the matching function. This allows investigation of how the function depends on the parameters, and can assist in defining appropriate conditions (e.g., parameter subsets) for performing MFI.

MFI. This component implements a global search / local optimization approach for MFI, allowing the estimation of parameter values for a range-independent geoacoustic model consisting of up to 23 parameters.

Visualization. This component allows the shot and ship time series data, the ambiguity functions, and the results of MFI to be displayed.

The BCOMFI software provides a comprehensive framework for examining the effectiveness of various MFI procedures for modeling the environment in the region of the monitoring station, and for future development of the actual software that will perform continuous monitoring of the real-time array data to be produced by the station.

Introduction

The Gulf of Mexico Hydrates Research Consortium and the Center for Marine Resources and Environmental Technologies (CMRET) at the University of Mississippi are currently developing a multi-sensor Seafloor Observatory to be installed on the continental slope of the northern Gulf of Mexico. The aim of this station is to monitor and investigate the hydrocarbon system within the hydrate stability zone of the northern Gulf of Mexico, and to remotely observe changes in the physical and chemical parameters of gas hydrates. A key component of the monitoring is a vertical array of hydrophones which will record data that will be analyzed and interpreted using matched field techniques. The ultimate aim of this analysis is to use acoustic energy emitted by passing ships to monitor the sub-bottom layers in the region of the station, with the goal of detecting the occurrence of large-scale changes in the hydrate structures within these layers.

Barrodale Computing Services Ltd. (BCS) has been contracted by the University of Mississippi to design and develop data management and processing software for this monitoring station. In four reports written during the earlier stages of this project, BCS has characterized the data to be produced by the station[13], proposed a design for a data management and archiving system for these data[14], formulated a design for a software system for simulating the data to be acquired by the vertical acoustic array of the station and analyzing these data using matched-field inversion (MFI) techniques[15], and, following implementation of this design, provided a User Guide to the initial version of the software[16]. The present report provides a full description of the implementation (for which the language was IDL) and use of the final version of the software, which has been termed BCOMFI, for **B**arrodale **CO**mputing **M**atched-**F**ield **I**nversion.

The simulation and conversion software component of BCOMFI provides several types of functionality. First, since the real data will consist of both time-limited shots and continuous data from nearby moving ships, the software has been implemented to simulate both shot traces and ambient noise data streams that would be observed at the sensors of an array. It also provides the ability to preprocess these data into a form where the resulting complex pressure fields can be used for MFI. In addition, to provide for greater flexibility in modeling correlated noise, and to promote efficiency, the software allows direct generation of frequency domain data (i.e., complex pressure fields) at selected frequencies, to serve as input to the MFI algorithms. These complex "measured" data can be generated in the form of either a signal vector **m** or a cross-spectral matrix $\mathbf{M} = \sum_j \mathbf{m}_j \mathbf{m}_j^*$,

[13] "Sensor and Data Characterization for the Gulf of Mexico Hydrates Monitoring Station" (Jan. 31, 2005).
[14] "Data Management Architecture Design for the Gulf of Mexico Hydrates Seafloor Observatory" (Feb. 28, 2005).
[15] "Software Design for Simulation, Matched-Field Inversion, and Monitoring for the Gulf of Mexico Hydrates Seafloor Observatory" (April 26, 2005).
[16] "BCOMFI Software User Guide" (August 12, 2005).

where (*) denotes the conjugate transpose operation. By using appropriate acoustic propagation models, the complex fields can be computed for both range-independent environments and for constant-slope bathymetry environments. A further component of BCOMFI allows real data "P16" files, which will be produced by the hydrophone array, to be read in and converted to a format where the data can then be analyzed by MFI.

The ambiguity function generation and MFI software components of BCOMFI allow "measured" frequency domain data to be matched with replica data vectors generated from a test model. This range-independent model includes 23 geometric and geoacoustic parameters, and can provide for up to three separate layers over a halfspace. The Bartlett power processor is used as a quantitative measure of the match between replica vectors generated using the model and the "measured" data. The software provides options for generating 1D, 2D, and 3D ambiguity functions at the points of a regular grid. This functionality allows the investigation of parameter sensitivity and correlation, and can provide an indication of the numbers of optima and other characteristics of the overall parameter space. MFI has been implemented using a global search, local optimization approach.[17] In the search stage, matches given by random choices of selected parameter values are obtained; then the best matches are chosen as starting points and for each of these, the matching function is optimized. This software allows investigation of issues such as the conditions where the model and geometry can be estimated with reasonable confidence, and could also be applied to use data obtained during calibration to generate "standard" range-independent models.

An additional software component of BCOMFI provides the ability to display the results of the simulation and processing. It allows visualization of the simulated time domain data, the ambiguity functions, and the results of the MFI processing.

The capabilities and use of the software are described in detail below. Very generally, the BCOMFI software consists of two basic types of code:

- ***Two Fortran executable modules for generating the acoustic fields***. These "computational engines" are: ORCA, a normal-mode code for range-independent environments, and RAM, a parabolic equation program that can be used for range-dependent environments. Most of the simulation computations, and all of the current ambiguity function and MFI computations, involve using ORCA, which is generally more than an order of magnitude faster than RAM. In the simulation components, RAM is used for generating data for an environment with constant slope. Note that the availability of two entirely separate propagation models provides a consistency check by allowing comparison of the fields that are produced by the two models.
- ***IDL code for the front end and the back end***. This software provides the environment for setting up GUIs, controlling the simulation and MFI, calling the computational engines, processing the computed fields, and displaying the

[17] Zala, C. A. and Ozard, J. M. *Estimation of geoacoustic parameters from narrowband data using a search-optimization technique*, J. Comput. Acoust. **6**, 223–243 (1998).

results. The software has been developed in such a way that it can be run using the IDL Virtual Machine (IDL VM – available at no cost from RSI) and therefore does not require an IDL Development Environment. The actual code has been delivered in the form of a save file (analogous to an executable file) that can run under the IDL VM.

In this report, the geoacoustic environment in the region of the proposed site is first described and the parameterization of this environment is outlined. Then the approaches for propagation modeling are outlined: the normal mode program ORCA is used to generate the fields for a range-independent model, for both simulation and MFI, and the (slower) parabolic equation program RAM (Range-dependent Acoustic Model) is used to simulate data for simple range-dependent non-elastic environments. The design and use of the simulation and MFI components are then described: each description includes a diagram of the overall process, an image and explanation of the GUI, a statement of the algorithm, and a description of how to use the component. Our implementation of a parallel processing option for matched field computations is also described, and the use of the display components is outlined. A computing environment to support this processing is then proposed. Finally, two Appendices describe the data structures and tags used in the software and detail the modular structure of each of the components.

Geoacoustic Environment

Description

In order to formulate effective designs for the functionality of the software to be developed, it was necessary to obtain some information about the general geoacoustic environment in which the monitoring station is to be deployed. At a meeting of the Hydrates Research Consortium in November 2004, a desirable site for the station was identified by the attendees, in Mississippi Canyon Block 118. Early in 2005, a final choice of the location within this block was made, and several seismic surveys were performed along four different tracks. These surveys, and associated data, indicated that:

- The depth at the site was about 875 m.
- There was a bottom slope up to about 2 degrees, with depth increasing to the southeast.
- The sloping region immediately surrounding the site was reasonably planar, but there was a canyon running northwest-southeast about 1.5 km to the northeast, and a flatter plateau, which showed high acoustic reflectivity, about 2 km to the south.
- There was a strong sub-bottom reflection about 250 msec below the bottom reflection, with two or three weaker reflections above this region and occasional strong deeper reflections. It was suggested that the reflection at 250 msec could possibly represent the bottom of the hydrate stability zone.
- There was a strong sub-bottom domed feature 1.5 – 2 km to the southwest of the site.

There were several bright spots in the seismic sections, also at about the 250 msec horizon. These were thought to correspond to pockets of gas.

These data clearly showed that this environment is significantly range-dependent, both with respect to the water depth (approximately planar with a maximum slope of about 2°) and the characteristics of the 250-msec reflection. Because of this range dependence, a key question is the extent to which this environment can be satisfactorily modeled by a range-independent propagation model such as ORCA. The initial approach taken in this project was to assume that there is some region close to the station within which the environment may be taken to be approximately range-independent. This assumption both simplifies the parameterization and reduces the time required for matched-field inversion using optimization.

Based on the above considerations, we have designed and implemented a comprehensive software system that allows generation of synthetic data for range-independent environments and analysis of synthetic and real data using matched-field inversion, assuming a range-independent environment. The implementation includes the ability to generate 1D, 2D and 3D ambiguity functions for specified parameters, and to perform parameter optimization. In addition, to allow further investigation of the effects of simple range dependence, the design also provides for the ability to generate synthetic data for a sloping-bottom range-dependent environment and to analyze these data using a range-independent model.

Parameterization

It was essential to define a geoacoustic and geometric parameterization that is acceptably realistic for use in range-independent modeling of the environment, yet is sufficiently simple to allow matched field inversion to be performed in a reasonable time frame. Based on the above description of the environment, a set of variables was defined that consisted of source-array geometry parameters, up to three sediment layers that could each have a gradient in their acoustic parameters, and a basement[18] halfspace with constant parameter values. The proposed 23-parameter model to be used and optimized for MFI of the environment is as follows:

Water depth. Variations in water depth due to tides, and the presence of some degree of range dependence require that water depth be a parameter in the inversion.

Source depth. This will generally be close to the surface (0.5 – 5 m) for both shot data and sources of opportunity, but should be included for investigative purposes.

Source range. Matching is known to be strongly dependent on range. As noted above, it is anticipated that for reasonably close source ranges (say, 1 – 3 km) a range-independent propagation model may be used to estimate the source range.

[18] The term “basement” is used here in a modeling, rather than geological, context.

Source bearing. For a vertical array in a range-independent environment, bearing is not resolvable (the fields are bearing-independent). However for a tilted array, bearing does have an effect and is potentially invertible.

Array tilt angle. Arrays will be tilted in practice, and it is known that this tilt has a large effect on the field matches. Optimization of this parameter (and the following parameter, i.e., tilt direction) could significantly enhance the matching. For the present, it will be assumed that the tilted array is linear.

Array tilt direction. As for previous parameter.

Layer parameters. These consist of five parameters for each of a prespecified number of layers (0 – 3: note that the number of layers is not optimizable):

- ***Layer thickness***. It is expected that this will be a useful parameter for detecting changes at the base of the hydrate stability zone.
- ***Compressional sound speed at layer top***. Changes in this parameter (or the following parameter, or both) could be indicative of changes that occur within the hydrate stability zone.
- ***Compressional sound speed at layer bottom***. As for previous parameter.
- ***Density at layer top***. Although this parameter is generally quite insensitive with respect to matching, it has been included for investigative purposes.
- ***Density at layer bottom***. As for previous parameter.

Compressional sound speed in basement halfspace. Changes in this parameter could also reflect changes that occur at the base of the hydrate stability zone.

Density in basement halfspace. Although this parameter is generally quite insensitive with respect to matching, it has been included for investigative purposes.

An option was provided in the matched-field applications to force the top and bottom compressional sound speed and density parameters in the layers to be equal (i.e., no gradient), so as to allow investigation of the effect of matching using a constant value when there is in fact a gradient in the layer. Note that shear sound speed and compressional and shear attenuations are not included as parameters at present.

Also, the sound-speed profile in the water column was not included in the parameter set, but rather, a means has been provided of specifying this data in an input file. This file could contain a generic, seasonally appropriate, sound-speed profile to be used in the matched field applications.

The above parameterization was designed to give considerable flexibility in the modeling while still being tractable in terms of matched-field inversion.

It is expected that the most useful optimizations will involve the geometric parameters water depth, source-array range and the thicknesses and sound speeds of the first one or two layers. The densities are not expected to have a substantial effect on the matches, but are provided nonetheless as a research tool.

Propagation Modeling

The ORCA normal mode acoustic propagation model is the “engine” used to compute the acoustic fields required for simulation and matched-field-inversion. For a given ocean environment, specified by the sound-speed profile in the water column and a geoacoustic profile of the ocean bottom, ORCA finds the normal modes and computes the acoustic field at the sensors of an array. The model includes the effects of sound-speed gradients in the water and the bottom layers, shear waves in the bottom layers, steep-angle propagation represented by leaky modes, and attenuation in the bottom layers. It may be used to predict narrowband or broadband propagation. ORCA is unique among underwater acoustic propagation codes because it is largely automatic: the user does not need to guess at any convergence parameters such as depth- or range-sampling resolutions. It is also computationally efficient, typically requiring a few tenths of a second (on a 3 GHz Windows computer) to compute a propagation model at frequencies of interest in the present application.

ORCA is written in Fortran, requires several input files, and was modified to produce an output file with the field values at the sensors. Since the simulation and MFI code are in IDL, a simple communication protocol and data transfer scheme between IDL and ORCA was defined. In this scheme, the IDL process writes out files with the information required by ORCA, and then spawns an ORCA process that reads in these files and produces an output file containing the complex field values. The IDL process then reads in this file and uses the data in its internal computations.

Despite its advantages and efficiency, ORCA is a range-independent propagation model and is not directly applicable to range-dependent environments. The environment in the array, however, is known to have significant range dependence, and it is desirable to be able to investigate the effects of this at some stage of the project by simulating range-dependent data. While it might be possible to implement an adiabatic mode approximation using ORCA to achieve this, it was deemed preferable to implement a separate range-dependent code for performing such simulations. In addition, such a code provided the opportunity to perform a validation check on the correctness of the ORCA results.

Accordingly, the RAM (Range-dependent Acoustic Model) parabolic equation model was implemented, and modified as required to integrate it into the BCOMFI system, to allow simulation of data in a simple range-dependent environment. This model is also written in Fortran and so the same general communication strategy as used for ORCA (i.e., IDL spawning a RAM process and reading in the output file) was used for RAM.

Installing the BCOMFI Software

The BCOMFI software that runs under the IDL VM is provided in the form of the zipfile “bcomfi.zip”. It sets up an appropriate directory structure and contains various data files as well as the software. To install this software, simply unzip the file into a directory of your choosing. The following directory structure and files will be produced:

\Bin. This contains the Windows version of the executable files for ORCA, RAM, and the orca server enabling parallel processing of matched field computations, i.e.,

orca.exe
rampe.exe
orcaserver.exe

\Data. This contains a set of files of the form xxxxnnnnn.ttt, where xxxx is a four-character file prefix, nnnnn is a five-character identification (ID) number, and ttt is either “dat” for a user-editable data file or “sav” for an IDL save file. These files are used to provide input and to store the results. (Certain of them may be overwritten during the processing, if the ID numbers are not changed during GUI setup.) See below for more detailed descriptions of these files, which correspond to IDL data structures used internally by the BCOMFI system.

\Doc. This contains User Guides for ORCA (PDF file) and RAM (PostScript file), and the present BCOMFI report.

\Inputs. This contains input files used by ORCA. In particular, the files sim_shot_svp and sim_ship_svp must be present in this directory; these files can be edited to modify the sound speed profile to be used by ORCA and/or RAM during the simulation. The other file types are generated during runs of the BCOMFI simulation system.

\Linux. This contains the Linux version of the executable files for ORCA and the orca server enabling parallel processing of matched field computations.

\Outputs. This contains files used by ORCA for output.

\P16_data. This contains sample ship and shot real data in p16 format in its subdirectories \ship and \shot.

\Ramout. This is initially empty but will contain files output by RAM during a simulation run where RAM is used.

\Source. This contains the files “bcomfi.sav”, “ramgeo.in”, and “servers.dat”, and will contain some input files for ORCA (orca_in) and/or RAM (ramgeo.dat) that are required for, or generated during, runs.

Installing the IDL VM and Launching BCOMFI

The IDL Virtual Machine can be obtained from the RSI web site. To download the IDL VM, go to http://www.rsinc.com/idlvm/, press the “Download IDL VM” option, enter the required registration information, check the “Windows” option, and follow the instructions. When the download is complete, run the install script. This will install the IDL VM on your computer and create an icon on the desktop.

The IDL VM allows you to locate an IDL save (.sav) file – in this case, bcomfi.sav – and run it. Pressing the IDL VM icon will first display a splash screen; you must click on this screen to proceed. When you have done so, a new file browser screen will appear. Navigate to the directory containing the bcomfi.sav file and select this file by double-clicking or selecting it into the "File name:" field. The file browser will then disappear and the BCOMFI program will be launched.

For convenience, it is a good idea to edit the "Properties" field of the IDL VM screen icon, select the "Shortcut" tab, and set the "Start in:" field to the directory containing the bcomfi.sav file. This will allow fast launching of the BCOMFI system.

The BCOMFI system consists of an initial GUI (the "Main Menu") that allows selection of a number of processing tasks. Pressing a button to select a desired simulation, conversion, preprocessing, or ambiguity function/MFI task brings up another GUI containing a number of fields or conditions that you can specify for the run. The simulation / conversion / preprocessing GUIs have been set up so that they can be run using default values, and each run will result in a data file being produced. The ambiguity function and search/optimization GUIs require user specification of parameters before a run can be launched. The contents of certain of these data files can be visualized or viewed using the appropriate button in the initial GUI, as described later in this report.

Setting up for Parallel Processing

To allow parallel processing of multi-frequency matched field computations on a network of Windows or Linux computers, the following additional installation steps are needed.

First, after installing the IDL VM on the master computer, copy the files "idl_tools_nodelay.dll" and "idl_tools_nodelay.dlm" from the \bin directory created above to the C:\rsi\idlXX\bin\bin.x86 directory on the client computer, where "XX" is the corresponding IDL version number (e.g., idl62).

Then, for each (remote) Windows computer in the network:

Create a directory C:\Program Files\Orca.
Copy the two executables "orcaserver.exe" and "orca.exe" and the batch file "orca.bat" from the \bin directory created above into the newly created directory on the remote computer.

Then, for each (remote) Linux computer in the network:

Copy the executables "orcaserver" and "orca90" from the \linux directory created above to /usr/local/bin on the remote computer. You will need root permission to do this.

Copy the tar file "fortranlib.tar.gz" from the \linux directory created above to /usr/local/lib on the remote computer. You will need root permission to do this.
cd to /usr/local/lib on the remote computer, and run the two commands "gunzip fortranlib.tar.gz" and "tar xf fortranlib.tar".
Make sure that the directory /usr/local/lib is in included in the value of the LD_LIBRARY_PATH environment variable for the session running orcaserver on the remote machine. To set LD_LIBRARY_PATH in the C shell (csh), type "setenv LD_LIBRARY_PATH ${LD_LIBRARY_PATH}:/usr/local/lib". To set this environment variable in the bash shell, type
" export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/usr/local/lib".
To check the value of LD_LIBRARY_PATH, type
"echo $LD_LIBRARY_PATH". You may wish to include the setenv or export command in your account's .cshrc (for C shell) or .bashrc (for bash) file so that it does not need to be entered each time orcaserver is run.

To actually run the server on either Windows or Linux remote computers, cd to the directory containing the orcaserver executable on the remote computer and enter "orcaserver *N*", where *N* is the remote port number to be used (e.g., "orcaserver 1501"). These orcaserver processes must be running on the remote servers for parallel processing to be enabled. Note that the port numbers can be arbitrarily chosen but must correspond to those selected in the matched field interfaces described in later sections. Also note that the same port numbers can be used with the different computers, providing the numbers in the orcaserver commands on the remote computers agree with the entries in the interfaces.

Overview of BCOMFI Functionality

Processing Components

The BCOMFI system consists of 16 separate components for simulation, data conversion, preprocessing, ambiguity function generation, MFI, and display. These components, which correspond to the initial GUI, are listed and briefly described in this section, while the following section gives some representative examples of processing tasks that may be accomplished using these components in a sequence. In the following descriptions and in later sections, "TD" denotes time domain, while "FD" denotes frequency domain.

Simulate TD shot data. This component uses ORCA to generate the impulse responses that would be observed at the sensors of an array for the specified geoacoustic environment and source-array geometry. The resulting traces are written to a Shot file, which can then be viewed or processed into a Ship file or an Hpdt file.

Simulate ship FD hpdt data. This component uses ORCA or RAM to generate frequency domain data at selected frequencies that would be observed at the sensors of an array for the specified geoacoustic environment and source-array geometry (with one or two moving sources). Noise of various types can be added to this data, and cross-spectral matrices can be generated from sequential segments. The resulting complex data are written to an Hpdt file, which can be used for input to ambiguity function generation and MFI.

Convert TD shot data to TD ship data. This component takes the impulse responses at the sensors in a Shot file and applies a separate phase randomization at each frequency (but not between sensors) in order to generate simulated time domain ship data with the same spectra as the traces in the Shot file. The resulting traces are written to a Ship file, which can then be viewed or processed into an Hpdt file.

Convert p16 data to shot or ship data. This component reads in real data files in p16 format (a binary integer format in which the data from the actual hydrophone arrays are collected and stored) and converts the data in each file to Shot or Ship format, as specified by the user. These can then be processed into Hpdt files.

Preprocess TD shot data to FD hpdt data. This component reads in a Shot file, FFTs the impulse responses for the sensors, forms “measured” cross-spectral matrices within a specified band for each of a number of selected frequencies, and writes the result to an Hpdt file.

Preprocess TD ship data to FD hpdt data. This component reads in a Ship file, FFTs the trace data according to the specified segmentation, accumulates “measured” cross-spectral matrices within a specified band for a number of selected frequencies, and writes the results to an Hpdt file.

Generate 1D ambiguity function. This component inputs an Hpdt file and generates an ambiguity function (matching function) as a single parameter of the 23-parameter model is varied between specified limits, with fixed values for the other parameters. The result is written to an Am1d file.

Generate 2D ambiguity function. This component inputs an Hpdt file and generates an ambiguity function (matching function) as two parameters of the 23-parameter model are varied between specified limits, with fixed values for the other parameters. The result is written to an Am2d file.

Generate 3D ambiguity function. This component inputs an Hpdt file and generates an ambiguity function (matching function) as three parameters of the 23-parameter model are varied between specified limits, with fixed values for the other parameters. The result is written to an Am3d file.

Perform MFI search-optimization. This component inputs an Hpdt file and performs matched field inversion of selected parameters of the 23-parameter model using a global search / local optimization technique with multiple starting points optimized until convergence. The estimates are grouped to reduce nonuniqueness, and 1D ambiguity functions of the specified parameters passing through the best estimate are computed to characterize the parameter space. The results are written to an Mfop file.

Display shot data. This component inputs a Shot file and displays the traces for the sensors in one waterfall plot and their corresponding spectra in another.

Display ship data. This component inputs a Ship file and displays the traces for the sensors in one waterfall plot and their corresponding spectra in another.

Display 1D ambiguity function. This component reads in an Am1d file and displays the 1D ambiguity function, using IDL's iplot component of its itools suite of interactive visualization utilities.

Display 2D ambiguity function. This component reads in an Am2d file and displays the 2D ambiguity function, using IDL's isurface or iimage components of its itools suite of interactive visualization utilities.

Display 3D ambiguity function. This component reads in an Am3d file and displays the 3D ambiguity function, using IDL's ivolume component of its itools suite of interactive visualization utilities or its slicer3 volume visualization tool.

Display MFI search-optimization results. This component reads in an Mfop file and allows visualization of:

- the matching functions computed during the search stage;
- the evolution of the objective function and parameters during any of the multiple convergences;
- the 1D ambiguity functions at the best optimum found.

It also displays text information describing the groups of optima found and the characteristics of the parameter space at the best optimum.

Examples of Processing using BCOMFI Components

Some examples of how the above processing components of BCOMFI can be used to perform some representative tasks are given in this section. Note that, for brevity, it is assumed that the files required for the runs (e.g., for ORCA and RAM) have been set up. (See later sections for details on which files are involved in the runs and how they should be set up.)

Determine the impulse response for specified conditions

- Run the *Simulate TD shot data* component to generate a Shot file.
- View this file using the *Display shot data* component.

Check the correspondence of ORCA and RAM

- Run the *Simulate ship FD hpdt data* component, selecting the ORCA option, to generate Hpdt file A.

Run the *Simulate ship FD hpdt data* component, selecting the RAM option, to generate Hpdt file B.

Run the *Generate 1D ambiguity function* component for selected parameters, using Hpdt file A, and producing Am1d file C. Also note the numerical values in the text log window of this component.

Run the *Generate 1D ambiguity function* component for the same selected parameters, using Hpdt file B, and producing Am1d file D. Also note the numerical values in the text log window of this component.

Compare the text values from the two runs. Also use the *Display 1D ambiguity function* component (twice) to view plots of the data in files C and D. Verify the closeness of the results.

Check that simulated shot and ship data produce the same matching as direct generation of hpdt data

Run the *Simulate TD shot data* component to generate a Shot file.

Run the *Convert TD shot data to TD ship data* component to input the Shot file and output a corresponding Ship file.

Run the *Preprocess TD shot data to FD hpdt data* component on the Shot file to produce Hpdt file A.

Run the *Preprocess TD ship data to FD hpdt data* component on the Ship file to produce Hpdt file B.

Run the *Simulate ship FD hpdt data* component under equivalent conditions to produce Hpdt file C.

Run the *Generate 1D ambiguity function* component for selected parameters, using Hpdt files A, B and C, and producing Am1d files D, E, and F. Compare the numerical values in the log window as above, and use the *Display 1D ambiguity function* component to view plots of the data in files D, E and F. Verify the closeness of the results.

Perform MFI on p16 data obtained for several shots

Run the *Convert p16 data to shot or ship data* component, selecting the shot option to generate Shot files from the selected input files.

For each Shot file, run the *Preprocess TD shot data to FD hpdt data* component to generate an Hpdt file.

For each Hpdt file, run the *Perform MFI search-optimization* component to generate an Mfop file (this is a time-consuming process).

View the results of the inversions using the *Display MFI search/optimization results* component.

Data Files

BCOMFI Data Files

The files in the "\data" directory have names of the form "xxxxnnnnn.ttt" and contain input and output data that correspond to internal data structures used by the programs. Files of type ".dat" are user-editable while those of type ".sav" are non-editable IDL save files. A brief description of the data files and their use is given here.

Am1dnnnnn.sav. This file is produced by the "Generate 1D ambiguity function" process. It contains the conditions used for generating the 1D ambiguity function (i.e., the values for fixed parameters, and the values for the parameter varied) as well as the results of the run (i.e., the matches for each value of the varied parameter, computed using the Bartlett processor). The ambiguity function can then be visualized by pressing the "Display 1D ambiguity function" button in the Main Menu interface.

Am2dnnnnn.sav. This file is produced by the "Generate 2D ambiguity function" process. It contains the conditions used for generating the 2D ambiguity function (i.e., the values for fixed parameters, and the values for the two parameters varied) as well as the results of the run (i.e., the matches for each pair of varied parameters, computed using the Bartlett processor). The 2D ambiguity function can then be visualized by pressing the "Display 2D ambiguity function" button in the Main Menu interface.

Am3dnnnnn.sav. This file is produced by the "Generate 3D ambiguity function" process. It contains the conditions used for generating the 3D ambiguity function (i.e., the values for fixed parameters, and the values for the three parameters varied) as well as the results of the run (i.e., the matches for each triplet of varied parameters, computed using the Bartlett processor). The 3D ambiguity function can then be visualized by pressing the "Display 3D ambiguity function" button in the Main Menu interface.

Araynnnnn.dat. This input file contains the specification for the array to be used in the simulations. Note that the arrays are used by the simulation, ambiguity function and MFI options, but not by the conversion, preprocessing, and display options. These text files are user-editable – see the example in \data\aray00000.dat.

Grpsnnnnn.dat. This file is produced by the "Display MFI search/optimization result" process, and is derived from data in an Mfop file. It contains the run conditions and a summary of the results of the grouping and peak analysis stages of a run of the MFI search/optimization. It contains the main results of the MFI run, in the form of a list of the various grouped convergences (optima) that were found, as well as the statistical properties for each group, when the group contained two or more convergences to the same optimum.

Hpdtnnnnn.sav. This file is produced by the "Simulate ship FD hpdt data", "Preprocess TD shot data to FD hpdt data", and "Preprocess TD ship data to FD hpdt data" processes. It contains the conditions used for simulation and the complex field vectors or cross-spectral matrices for the array at user-selected frequencies. It can also contain a sequence of such datasets, at each of a number of times as specified by the user. Hpdt files contain the acoustic field data that will be used for MFI.

Mfopnnnnn.sav. This file is produced by the "Perform MFI search/optimization" process. It contains values used in the setup of the search/optimization, information about the fixed and varied parameters, the results for the global search stage, the results of each separate local optimization, grouping of the optima, and 1D ambiguity functions through the best optimum found. This information can be viewed or visualized by pressing the "Display MFI search/optimization results" button in the Main Menu interface.

Shipnnnnn.sav. This file is produced by the "Convert TD shot data to FD ship data" option and can also be used for input by the "Preprocess TD ship data to FD hpdt data" option. It contains the conditions used for simulation and the simulated time domain ship data traces (generated from shot traces). These traces can be visualized by pressing the "Display ship data" button in the Main Menu interface.

Shotnnnnn.sav. This file is produced by the "Simulate TD shot data" option and can also be used for input by the "Convert TD shot data to FD ship data" and "Preprocess TD shot data to FD hpdt data" processes. It contains the conditions used for simulation and the simulated time domain shot data traces. These traces can be visualized by pressing the "Display shot data" button in the Main Menu interface.

Wavennnnn.dat. This input file contains the specification for the wavelet to be used in the generation of shot traces by the "Simulate TD shot data" option. These text files are user-editable – see the example in \data\wave00000.dat.

Wsspnnnnn.dat. This input file contains the water-column sound speed profile to be used by the ambiguity function generation and search/optimization MFI procedures. The data format consists of the number of points in the profile, and the (depth, sound speed) pairs defining the profile, one pair per line. These text files are user-editable – see the example in \data\wssp00000.dat.

ORCA Data Files

\Source Directory

Orca_in. Produced automatically by all runs involving ORCA. Contains the "_opt" and "_svp" files to use for the run.
Orca_status. Produced automatically by all runs involving ORCA. Contains the exit status of the ORCA run.

\Inputs Directory

Sim_shot_svp. <u>Must be provided by the user</u> prior to an ORCA run simulating TD shot data, and contain the sound velocity profile (water and bottom

layers/halfspace) for the environment. An example file is provided in \inputs\sim_shot_svp.
Sim_ship_svp. Must be provided by the user prior to an ORCA run simulating ship FD hpdt data, and contain the sound velocity profile (water and bottom layers/halfspace) for the environment. An example file is provided in \inputs\sim_ship_svp.
Mfi_svp. Produced automatically by all ORCA runs involving ambiguity function generation or search/optimization.

Sim_shot_opt. Produced automatically by runs involving simulating TD shot data. Contains options for the ORCA run.
Sim_ship_opt. Produced automatically by runs involving simulating ship FD hpdt data using ORCA. Contains options for the ORCA run.
Mfi_opt. Produced automatically by all runs involving ambiguity function generation or search/optimization. Contains options for the ORCA run.

Sim_shot_array_geom. Produced automatically by runs involving simulating TD shot data. Contains the array geometry for the ORCA run.
Sim_ship_array_geom. Produced automatically by runs involving simulating ship FD hpdt data. Contains the array geometry for the ORCA run.
Mfi_array_geom. Produced automatically by runs involving ambiguity function generation or search/optimization. Contains the array geometry for the ORCA run.

\Outputs Directory

Sim_field.dat. Produced automatically by ORCA runs involving simulating ship FD hpdt data. Contains the complex-valued fields at the sensors of the array, which are then read in by the IDL code.
Mfi_field.dat. Produced automatically by ORCA runs involving ambiguity function generation or search/optimization. Contains the complex-valued fields at the sensors of the array, which are then read in by the IDL code.

Sim_modes. Produced automatically by ORCA runs involving simulating ship FD hpdt data. Contains the properties of the modes (not used).
Mfi_modes. Produced automatically by ORCA runs involving ambiguity function generation or search/optimization. Contains the properties of the modes (not used).

Sim_out. Produced automatically by ORCA runs involving simulating ship FD hpdt data. Contains the input data for the run and a summary of the number of modes and timing data (not used).
Mfi_out. Produced automatically by ORCA runs involving ambiguity function generation or search/optimization. Contains the input data for the run and a summary of the number of modes and timing data (not used).

Sim_shot_fft. Produced automatically by ORCA runs involving simulating TD shot data. Contains a Fortran binary version of the FFT file output by ORCA for the run (not used).
Sim_shot_fft.dat. Produced automatically by ORCA runs involving simulating TD shot data. Contains ASCII output constituting the FFT file output by ORCA for the run, which is then input by the BCOMFI code that generates the impulse response.

RAM Data Files

\Source Directory

Ramgeo.dat. Must be provided by the user prior to a RAM run simulating ship FD hpdt data, and contain the sound velocity profile (water and bottom layers/halfspace) for the environment in the same format as that in the input file ramgeo.in. Ramgeo.dat acts as a source and template for the setting of parameters other than frequency, source range and source depth (which are set during BCOMFI simulation), and allows the automated generation of ramgeo.in files during the RAM runs. An example file is provided in \source\ramgeo.dat.

Ramgeo.in. Produced automatically just prior to each RAM run simulating ship FD hpdt data and used as input for the RAM run.

\Ramout Directory

Field.dat. Produced automatically by RAM runs. Contains a Fortran binary version of the complex fields computed during the RAM run (not used).
Field.out. Produced automatically by RAM runs. Contains an ASCII version of the complex fields computed during the RAM run, which is then used by BCOMFI to compute the fields at the sensors of the array.
Tl.grid. Produced automatically by RAM runs. Contains a Fortran binary version of the transmission loss grid computed during the RAM run (not used).
Tl.line. Produced automatically by RAM runs. Contains an ASCII version of the transmission loss along a constant-depth line computed during the RAM run (not used).

P16 Data Files

These files, which will contain real data from the array, must have names of the form pnnnnn.p16, where nnnnn is a five-character identification (ID) number. However, they can be placed in any user-accessible directory, as they are specified interactively during a run using a special purpose “pickfile” dialog box. Example files are provided in the directories \p16_data\ship and \p16_data\shot.

Servers Data File

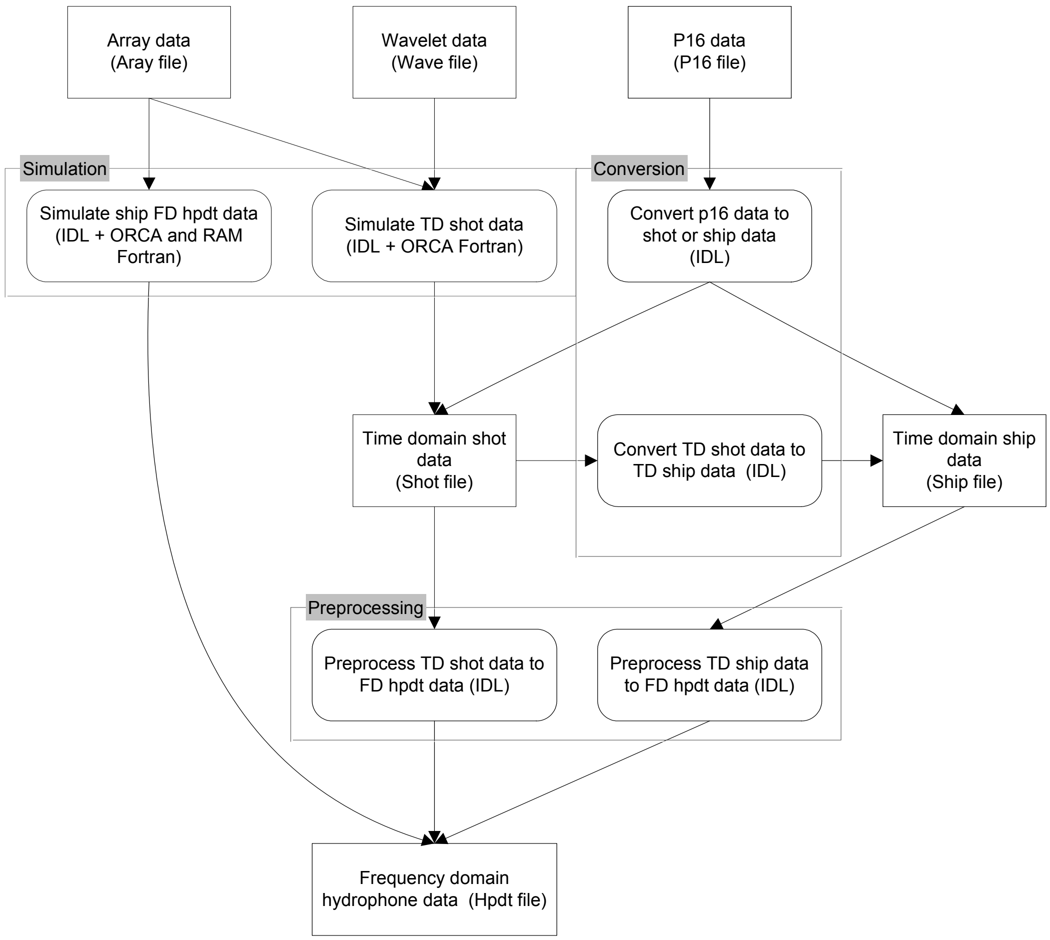
The file “servers.dat” must be present in the \source directory in order to run the matched field computations. It contains the default IP addresses and ports for each server process to be run on the remote computers, and can be edited to correspond to the distributed environment to be used. If the parallel processing option is not selected, the information in this file is simply ignored. An example of the file structure is provided in the \source directory of the software delivery.

BCOMFI Software: Main Menu

The Main Menu provides an interface for invoking any of the functional components of the BCOMFI system listed in the Processing Components section above, i.e., the simulation, conversion, preprocessing, ambiguity function generation, MFI, and display components. Note that, in general, the result of using the simulation, conversion and preprocessing components is ultimately an Hpdt file. This Hpdt file is then used as input to the ambiguity function generation and MFI components.

Overall Design: Simulation, Conversion and Preprocessing Components

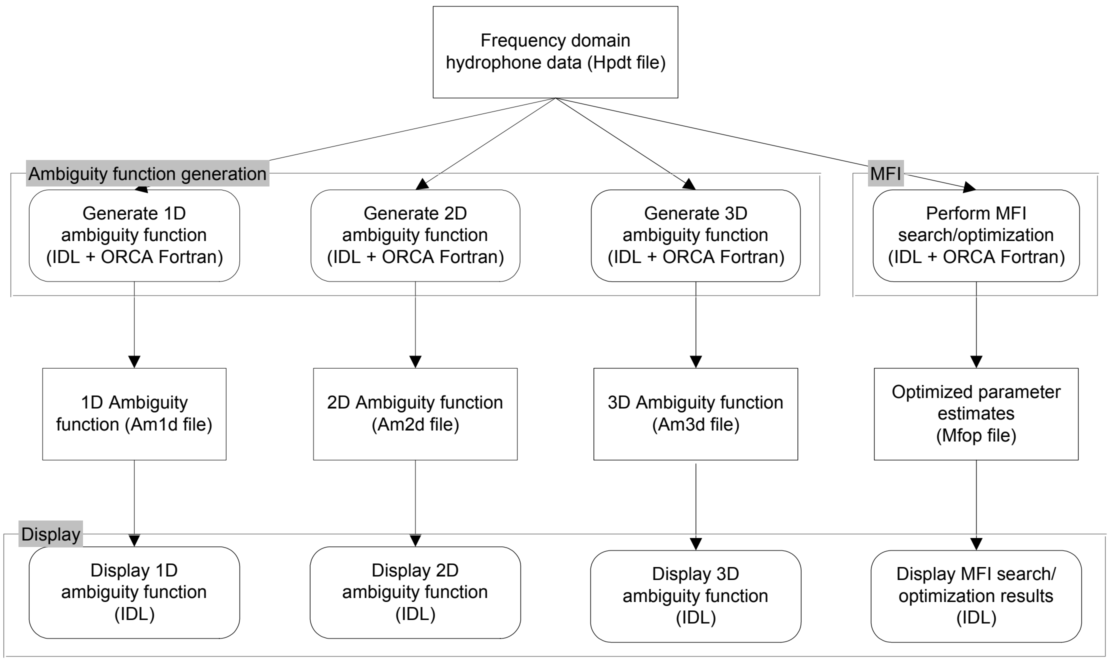
In this design, data and/or files are indicated by rectangles and processes are represented by rounded rectangles.



Overview of BCOMFI Simulation, Conversion and Preprocessing Components

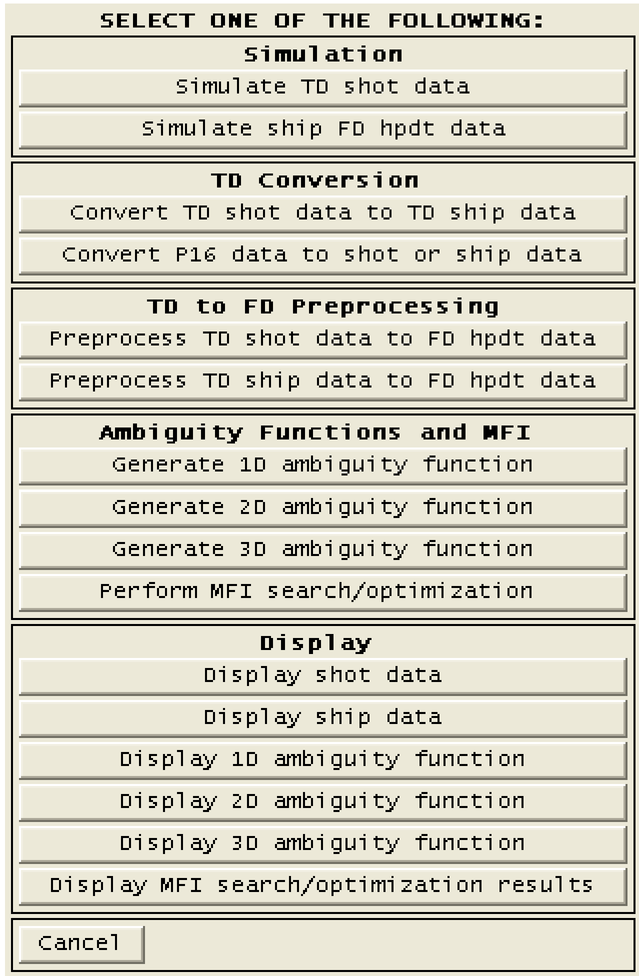
Overall Design: Ambiguity Function Generation, MFI and Display Components

In this design, data and/or files are indicated by rectangles and processes are represented by rounded rectangles.



__Overview of BCOMFI Ambiguity Function Generation, MFI and Display Components__

Graphical User Interface



Pressing one of the buttons (other than Cancel) brings up another interface (described below) that allows the specification of the conditions for a run of that selected component of BCOMFI.

Simulation of Time Domain Shot Data

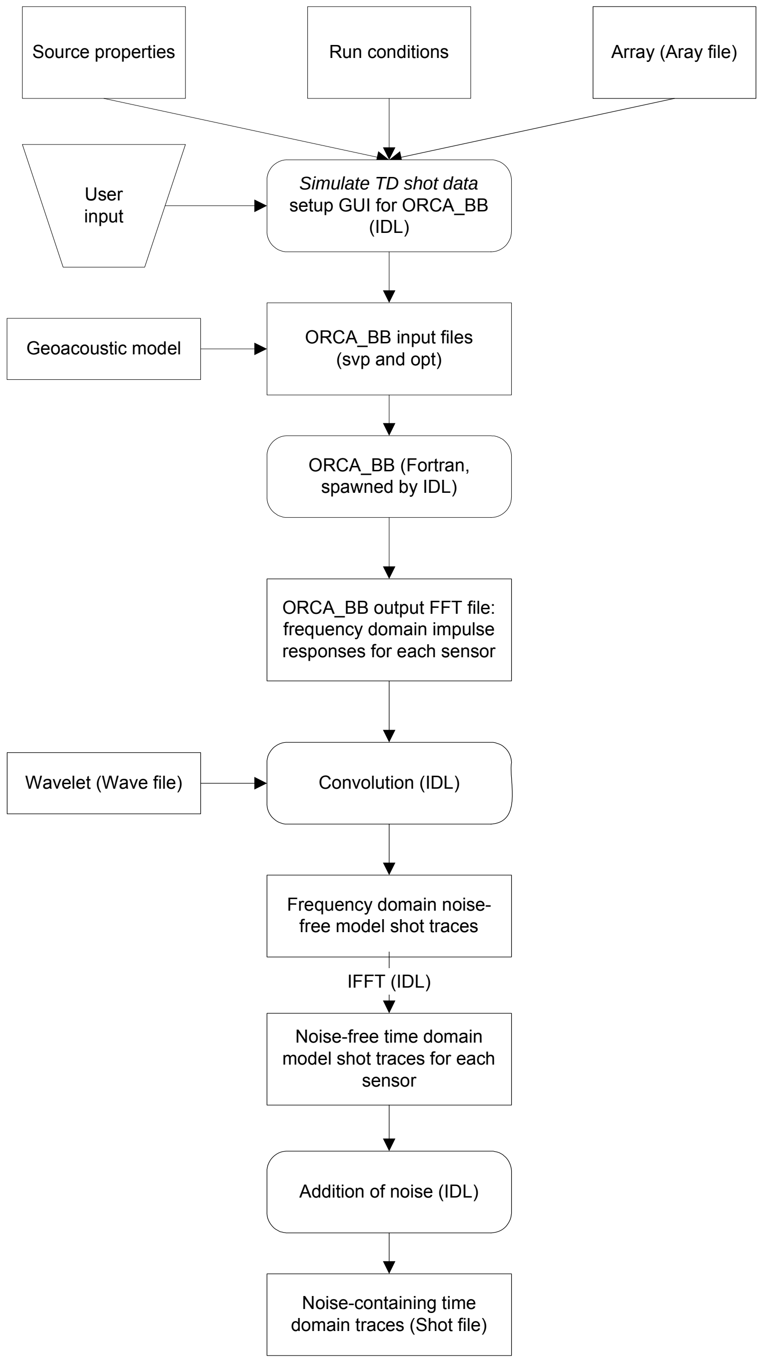
In order to develop and verify methods for MFI of the shot data to be obtained, it was essential to first be able to generate synthetic shot traces for specified source-array geometries and geoacoustic environments and use them to test the ambiguity function generation and MFI methods. These synthetic data consist of impulse responses (generated by propagation modeling) convolved with a representative source wavelet, with additive noise. The resulting model traces can then be FFT'd and selected frequencies used to perform and evaluate the subsequent MFI processing.

In the *Simulate TD shot data* component of the BCOMFI system, the broadband option of ORCA is used to generate the impulse responses. This option allows the user to specify a frequency band (e.g., 1 – 250 Hz), a sampling frequency, and a time window (or equivalently, a number of FFT points). ORCA then generates the frequency domain impulse response for each discrete frequency in the specified band, and outputs this result to an FFT file. This file is then read in and convolved with a specified (shot) waveform. An option to add Gaussian noise is provided, and the resulting frequency domain traces are then inverse FFT'd to yield the corresponding time domain traces.

Implementation of this software for simulating shot data allowed the methods for processing the real shot data to be developed and tested in advance of the real data becoming available.

Overall Design

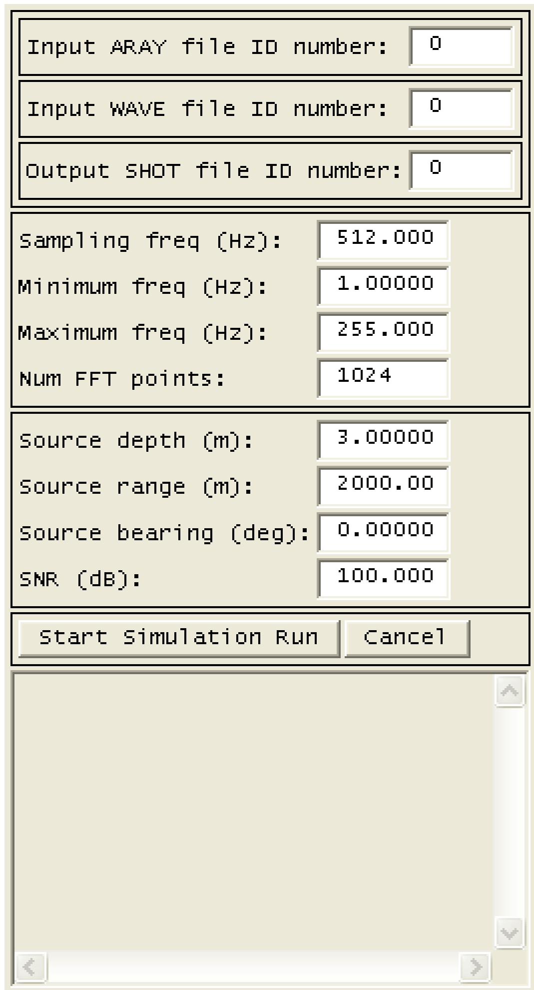
In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Simulate TD shot data

Graphical User Interface

The GUI for *Simulate TD shot data* is illustrated in the following diagram.



The first three items allow the user to specify the input files (Aray and Wave) to be used for the run, and the Shot file to contain the results.
The next four items allow the user to specify the conditions for the broadband ORCA run.
The following four items (in the “Source” box) allow the user to specify the source-receiver geometry to be used in the run and the (power) signal-to-noise ratio (dB) for each trace.
The “Start Simulation Run” and “Cancel” buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for simulating the shot traces is outlined in the following pseudocode:

Set up an svp (sound velocity profile) file for ORCA to use as the geoacoustic environmental model.
Set up an Aray file to contain the array to be used in the simulation and a Wave file to contain the wavelet.
Using the GUI, specify the files to be used as the input array (Aray) and wavelet (Wave) files and the output Shot file, and specify the run conditions for the source and ORCA broadband.
Check that the run conditions are consistent.
Apply a tilt to the array.
Generate an array geometry file and an ORCA options file.
Spawn a process to perform an ORCA broadband run:
 ORCA process:
 For each frequency:
  Compute the complex fields that would be observed at each sensor for a source at the specified position.
 Output an ORCA FFT file containing the complex fields for the impulse responses for each of the sensors.
Read in the FFT file generated by ORCA.
Read in and FFT the wavelet from the specified Wave file.
Generate time domain traces: For each sensor:
 Reflect/conjugate the FFT about the Nyquist frequency to obtain the Fourier transform of the impulse response.
 Convolve the wavelet with the impulse response.
 Inverse FFT the result to give a time domain trace.
 Add Gaussian noise at specified signal-to-noise ratio.
Output the traces to a Shot file.

Using the Component

Press the “Simulate TD shot data” button to bring up the GUI for generating time domain traces that would be observed at the sensors of an array for a particular source-array geometry, geoacoustic model and source wavelet. The process uses the broadband option of ORCA to generate, for each of the sensors, the complex fields at each frequency within the specified band that would be observed on the basis of the geometry and geoacoustic model. These complex vectors are written to an FFT file by ORCA, and this file is then read in and the sensor data are multiplied with the Fourier transform of the wavelet. The result is inverse Fourier transformed and written to a Shot save file.

To run this model, first ensure that the file “sim_shot_svp” exists in the \data directory noted above, and that it contains the desired sound velocity profile in

ORCA format. Then specify the following information in the GUI:

the ID number of the Aray file containing the array to be used (sensor (*x, y, z*) points, tether depth for the *z* points, and tilt data);
the ID number of the Wave file containing the wavelet to be used;
the ID number of the output Shot file that will be produced;
the sampling frequency, which should be at least twice the maximum frequency specified below;
the minimum frequency in the band for which the complex fields are to be generated by ORCA;
the maximum frequency in the band for which the complex fields are to be generated by ORCA (this should be at most half the sampling frequency);
the number of FFT points; this should be long enough so that the entire trace is contained within the time window (i.e., number of FFT points ÷ sampling frequency); otherwise the traces may wrap around;
the source depth and its range and bearing with respect to the array;
the SNR (signal-to-noise ratio) for addition of white noise to the traces; note that for SNR ≥ 100, no noise is added.

When the above have been specified, press the "Start Simulation Run" button to perform the simulation. This will likely take several minutes. The progress of the ORCA broadband run can be monitored by viewing the DOS / Command Prompt window generated by the spawning process by which IDL invokes ORCA.

Simulation of Frequency Domain Ship Data

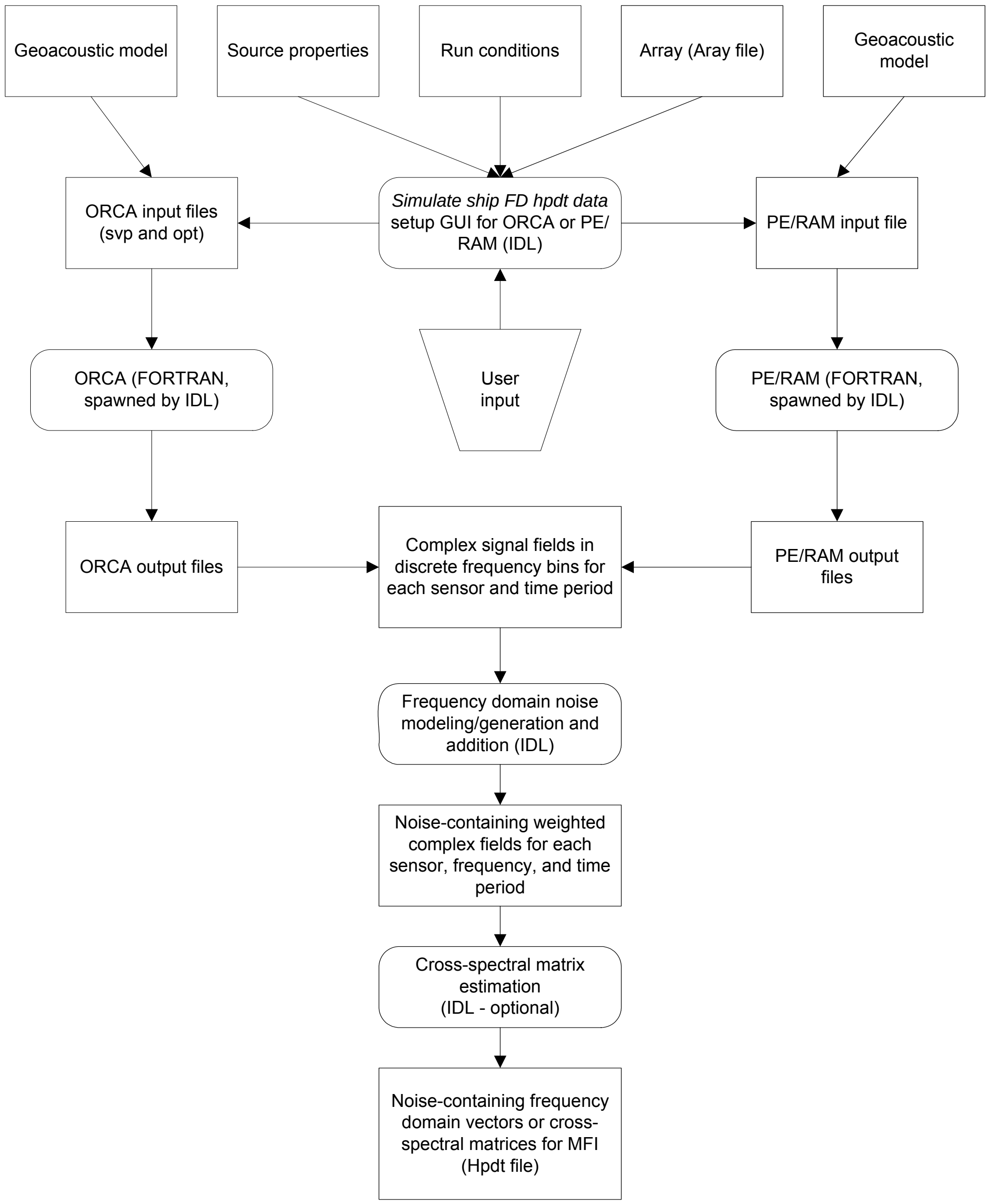
As with the shot data, it was essential to have a software component that could generate simulated data in order to test the correctness and performance of the MFI algorithms during development. These synthetic data consist of complex signal vectors or cross-spectral matrices at several user-selected frequencies, each of which can optionally contain noise corresponding to various models. Since there may be more than one actual source, and these sources will generally be moving, the simulation also includes the ability to model and generate data for two moving sources.

As noted above, the main application for the simulations is to generate data for a range-independent environment. For this application, ORCA (in standard rather than broadband mode) is used to generate the fields at the sensors that would be observed for a simulated moving source. However, in view of the fact that the environment has a significant range-dependence, it was also of considerable interest to be able to simulate data for a simple range-dependent environment, and analyze this data with MFI using a range-independent model (ORCA). The range-dependent parabolic equation (PE) code RAM provides the ability to generate data for range-dependent environments, and is freely available over the web. This RAM code was obtained and incorporated into the software system for generating synthetic data. The availability of two entirely separate propagation modeling codes provided a consistency/validity check by allowing comparison of the fields that are produced by the two models for a range-independent environment.

Hence, both ORCA and RAM options were implemented to generate acoustic fields at multiple frequencies for moving sources. Options were also provided for adding noise and performing cross-spectral matrix estimation. This implementation allowed the methods for processing the real ship data to be developed and tested in advance of the real data being available.

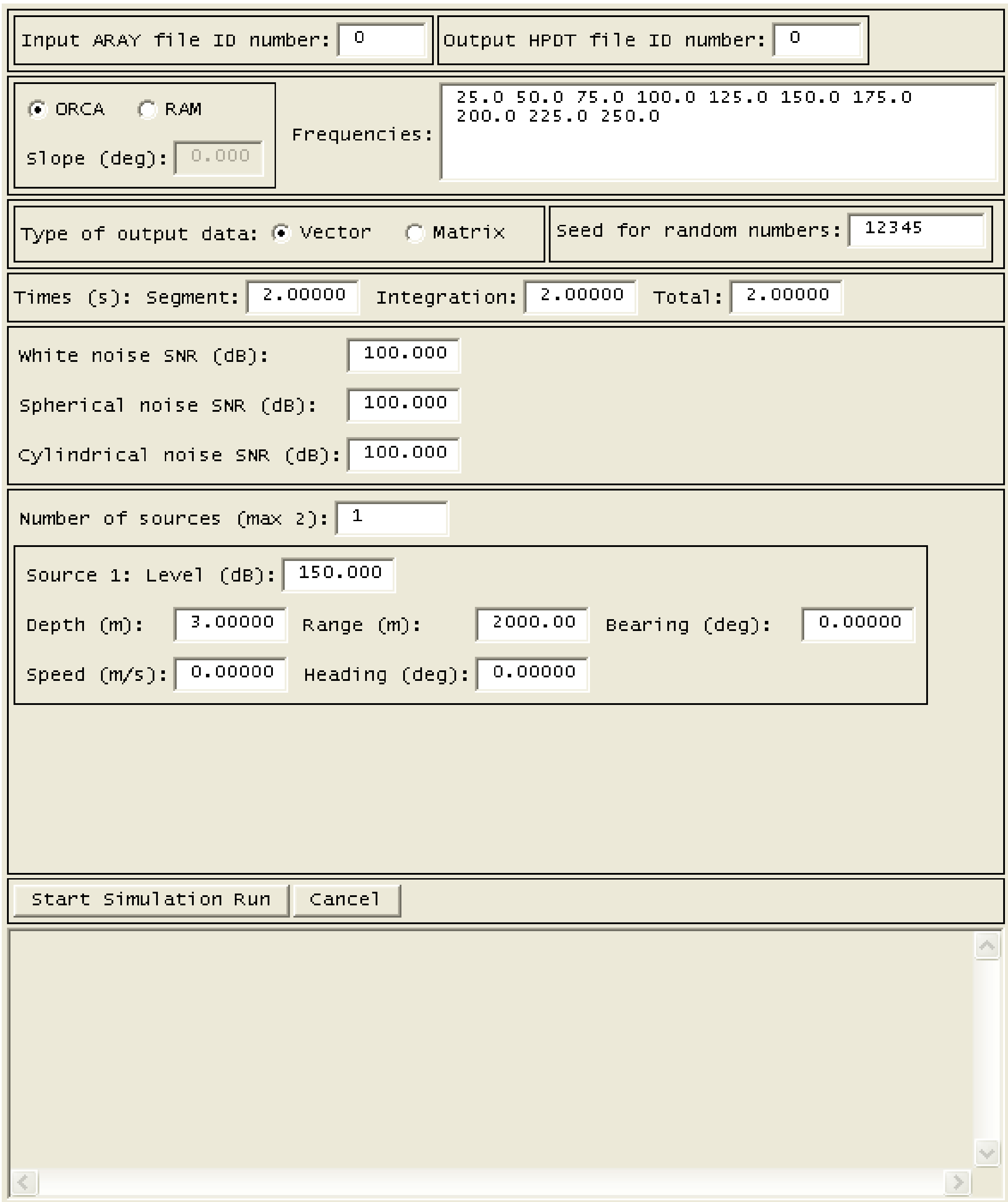
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Simulate ship FD hpdt data

Graphical User Interface



The first items in the top row allow the user to specify the input Aray file to be used for the run, and the Hpdt file to contain the results.

The second row allows the user to indicate whether ORCA or RAM is to be used, and to specify the frequencies for which the data are to be generated.
The third row allows the user to specify whether the data are to be output as vectors or cross-spectral matrices, and to enter a seed for the random number sequence to be used in generating noise realizations. The latter allows repeated runs involving noise to be reproducible.
The fourth row provides for specification of the segment time, the integration time (the time over which cross-spectral matrix estimation takes place), and the total time (which controls how many vectors/matrices are computed).
The fifth row provides options for adding noise of specified signal-to-noise ratio from three different distributions: white, spherical, and cylindrical.
The sixth row allows one or two sources to be specified.
The seventh row (with the "Source 1" label) allows the user to specify the source level and the source-receiver geometry for Source 1 to be used in the run.
The eighth row (if present) allows the user to specify the source level and the source-receiver geometry for Source 2 to be used in the run.
The "Start Simulation Run" and "Cancel" buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for simulating the ship data is outlined in the following pseudocode:

Set up an svp (sound velocity profile) file for ORCA and/or RAM, to use as the geoacoustic environmental model.
Set up an Aray file to contain the array to be used in the simulation.
Using the GUI, specify the input Aray file and the output Hpdt file, and select the run conditions.
Check that the run conditions are consistent.
Apply a tilt to the array.
For ORCA, generate an array geometry file and an options file.
For each frequency specified:
- *Generate and Cholesky decompose the noise matrices for white, spherical, cylindrical, and modal noise at that frequency.*

For each integration time:
- *For each segment time:*
  - *For each source:*
    - *Compute new source position.*
    - *For each frequency:*
      - *Run ORCA or RAM to generate a signal vector.*
      - *Scale the signal vector, generate a complex constant and use it to randomize the phase across the array, for that frequency.*
  - *For each frequency:*

Generate a random noise vector for the array sensors using the noise cross-spectral matrix for that frequency and array.
Add noise vector to signal vector.
For each frequency:
Accumulate cross-spectral matrices if this option is chosen.
Output the vector/matrix data to an Hpdt file.

Using the Component

Press the “Simulate ship FD hpdt data” button to bring up the GUI for generating data vectors or cross-spectral matrices, at selected frequencies, that would be observed at the sensors of an array for a particular source-array geometry (where the source can be moving) and geoacoustic model. For a moving source, a quasi-stationary assumption is used, so that the cross-spectral matrices are formed by averaging a number of stationary matrices computed for the source at points along a linear track. The vector or matrix data generated by the run are written to an Hpdt file for input to MFI.

The process uses ORCA or RAM to generate frequency domain pressure field values for each of the sensors, at the selected frequencies. If ORCA is chosen, the geoacoustic model is range-independent. If the RAM option is chosen, a constant slope may be specified for the bathymetry between the source and the array; also, if RAM is chosen, an image of the 2D output transmission loss is displayed at the end of the run for each frequency. Note that runs with RAM are much more time-intensive than ORCA runs.

To run this model, first ensure that, if ORCA is to be used, the file “sim_ship_svp” exists in the \data directory noted above, and that it contains the desired sound velocity profile in ORCA format. If RAM is to be used, ensure that the file “ramgeo.dat” exists in the \source directory and contains the information for the run conditions and the sound velocity profile in RAM format. Note that the parameters that can be influenced by information provided in the GUI are indicated by uppercase labels (e.g, FREQ, ZS) in ramgeo.dat. Also note that the water depth at the array is defined by the first ZB term in ramgeo.dat.

Then specify the following information in the GUI:

the ID number of the Aray file containing the array to be used (sensor (*x, y, z*) points, tether depth for the *z* points, and tilt data);
the ID number of the output Hpdt file that will be produced;
whether ORCA or RAM is to be used, and, if RAM, the slope between the source and the array; positive slopes indicate that the depth increases from the array toward the source;
the frequencies at which the data are to be generated (the default is ten frequencies, evenly spaced between 25 and 250 Hz);
whether the output data are to be generated in the form of vectors or cross-spectral matrices;

a seed for simulated noise generation;
values for segment time (data length for a single FFT), integration time (time within which outer-product matrices are averaged to accumulate cross-spectral matrices), and total time;
SNRs for three ocean noise types: white, spherical and cylindrical; note that for SNR ≥ 100, no noise of that type is added;
the number of sources: currently restricted to 1 or 2; if the latter is chosen, a second region for source specification will appear below;
the characteristics of the source(s): level (dB re 1 μPa at 1 m), depth, range and bearing with respect to the array, speed and heading.

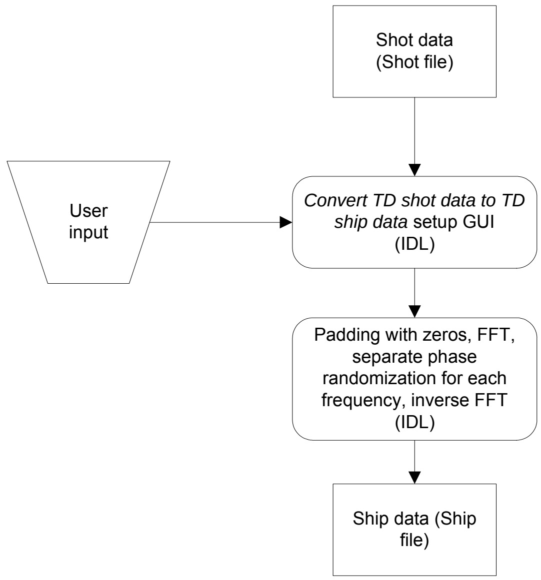
When the above have been specified, press the "Start Simulation Run" button to perform
the simulation. The time taken will depend on the frequencies chosen and on the propagation model: for ORCA, the computations will likely take several seconds, while for RAM, they may take several minutes.

Conversion of Time Domain Shot Data to Time Domain Ship Data

This functionality was developed in order to allow the generation of time domain ship data from previously-generated synthetic shot data for a specified environment. These ship data could then be used as input to a process that would use these data to generate Hpdt files for input to MFI. This software component provided a way of validating the processing of ship data for MFI in advance of the real ship data from the monitoring station being available.

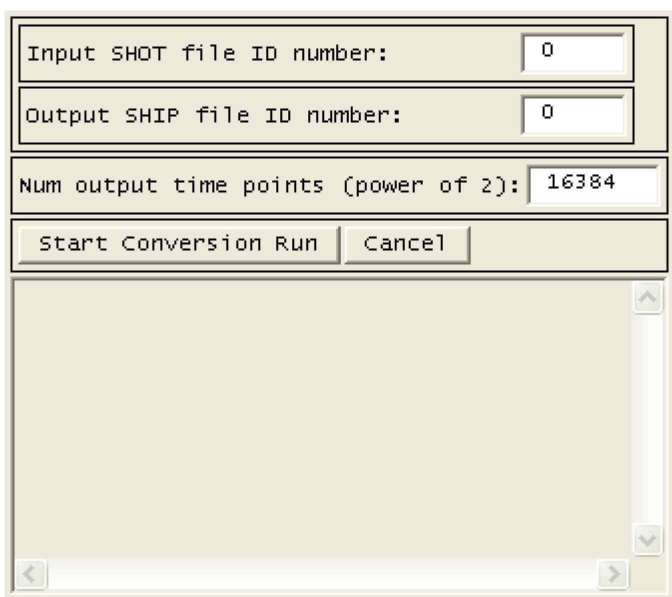
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Convert TD shot data to TD ship data

Graphical User Interface



The first two rows allow the user to specify the input Shot file to be used for the run, and the Ship file to contain the results.

The next row allows the specification of the number of points in the output time series for each sensor. This should be at least the number of input points in the Shot file and should also be a power of 2 (otherwise, it will be set to a power of 2).
The "Start Conversion Run" and "Cancel" buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for simulating the ship data is outlined in the following pseudocode:

Set up a Shot file to contain the array to be used in the simulation.
Using the GUI, specify the input Aray file and the output Hpdt file, and select the run conditions.
Check that the run conditions are consistent.
For each sensor:
Pad the trace with zeroes and FFT.
For each frequency:
Generate a random complex constant and multiply it with the frequency bin value for each sensor to randomize the phase.
For each sensor:
Inverse FFT the trace.
Output the result to a Ship file.

Using the Component

Press the "Convert TD shot data to FD ship data" button to bring up the GUI for converting shot traces to the type of data that would correspond more closely to that being collected for ships of opportunity. These data have the same spectra, but are different in that shot traces are coherent in frequency while the ship data are not; both data types are, of course, spatially coherent. In addition, the shot data are confined to a small region of time, while the ship data can go on for an arbitrary length of time. The conversion process involves reading in the shot traces, padding with zeros to increase their length, and Fourier transforming. Then phase randomization is applied to each frequency bin by generating a random rotation and applying the same rotation to each sensor of the array. The result is then inverse transformed to yield the simulated ship data, which is written to a Ship file.

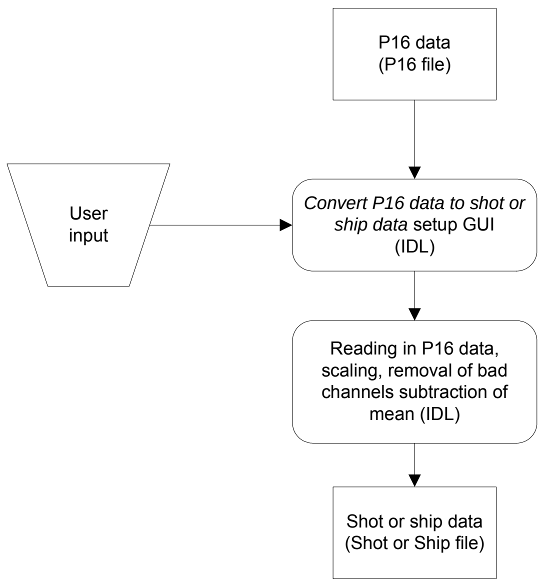
To run this conversion, specify:
the ID number of the Shot file containing the array to be converted;
the ID number of the output Ship file that will be produced;
the number of time points in the output traces (power of 2).

When the above have been specified, press the “Start Conversion Run” button to perform the conversion. The process should be almost instantaneous.

Conversion of Time Domain P16 Real Data to Time Domain Shot or Ship Data

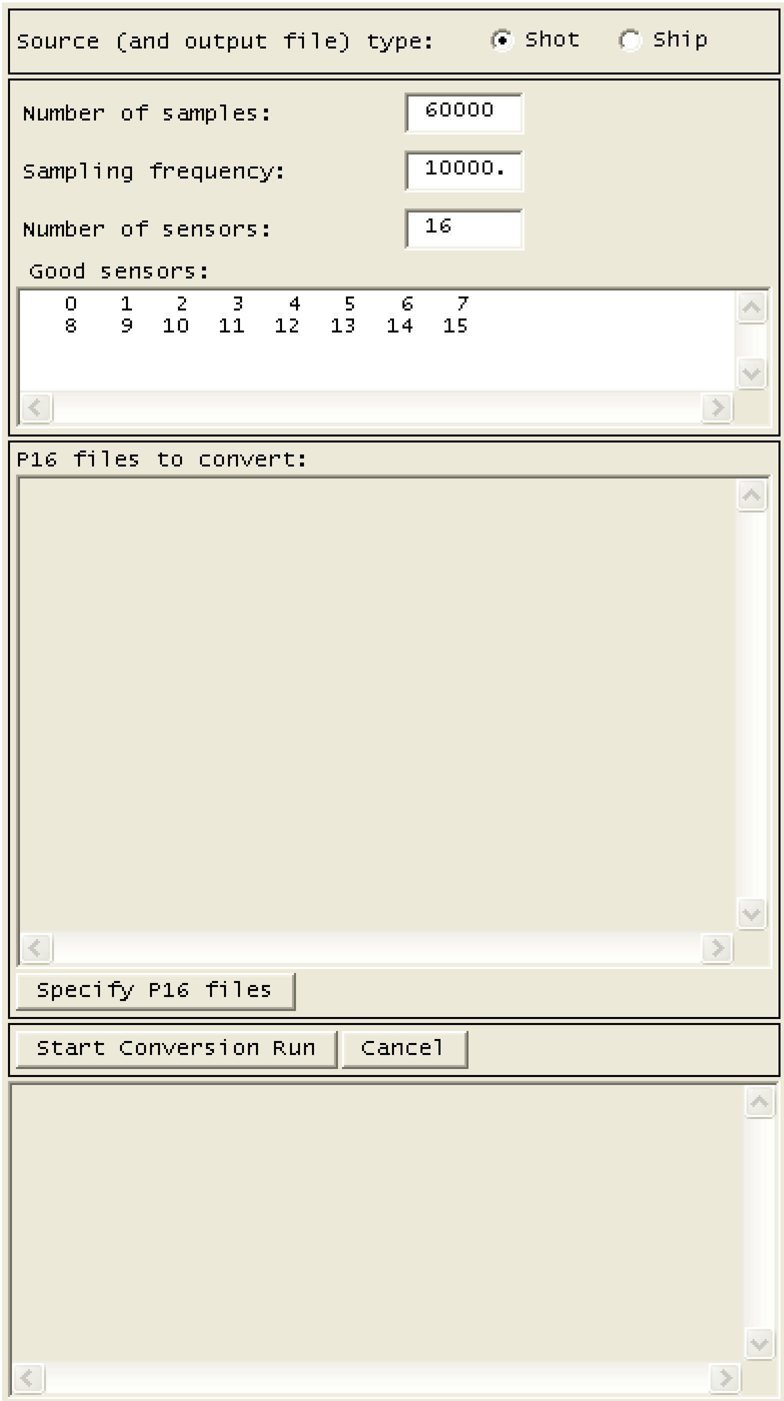
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Convert P16 data to shot or ship data

Graphical User Interface



The first row allows selection of the source and output file type: either Shot or Ship. This should be matched to the type of input data in the P16 file(s).
The second row allows the user to specify the number of time points that have been recorded for each sensor in the input P16 file(s).
The third row allows specification of the sampling rate (Hz).

The fourth row specifies the number of sensors in the input data.
The fifth row allows the specification of which sensors are "good" and which should be retained in the output file(s).
The next large initially empty area provides a field for displaying the P16 files selected for conversion.
The "Specify P16 files" button brings up a dialog box that allows the user to specify which file(s) are to be converted from P16 to Shot or Ship files.
The "Start Conversion Run" and "Cancel" buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for simulating the ship data is outlined in the following pseudocode:

Using the GUI, specify the input P16 file(s) and select the run conditions.
Check that the run conditions are consistent.
For each P16 file:
    Read the binary data.
    Scale the data to lie between –1 and 1.
    Remove bad channels from the data.
    For each trace:
        Subtract the mean.
    Output the traces to a Ship or Shot file with the same ID as the P16 file.

Using the Component

Press the "Convert P16 data to shot or ship data" button to bring up the GUI for converting real array data in P16 format to shot or Ship files with the same ID number as the P16 files.

To run this conversion, specify:
the type of data in the P16 files to be converted – either ship or shot;
the number of time points in the output traces (power of 2), the sampling frequency, and number of sensors in the array;
the "good" sensors, for which the data are to be retained;
the P16 files to be converted, by pressing the "Specify P16 files" button, navigating to the directory containing those files, and selecting the desired ones – note that they should correspond to the specified source type (shot or ship); the selected files will be displayed in the text window above the button;

When the above have been specified, press the "Start Conversion Run" button to perform the conversion. The process should take only a few seconds.

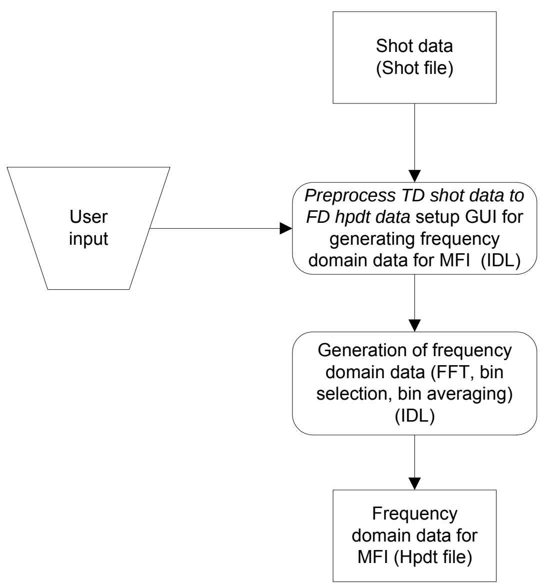
Note that only the “good” sensors data will appear in the output data files. Hence, you will have to create a corresponding Aray file to match these sensors before performing ambiguity function generation or MFI.

Preprocessing of Time Domain Shot Data to Frequency Domain Hpdt Data

This component allows the data for each shot to be processed into a form amenable to MFI analysis. The strategy for processing the shot data is to FFT the traces for the sensors and form cross-spectral matrices at user-specified frequencies, with the option for averaging within a small frequency band at each of these frequencies. These matrices are output to an Hpdt file, which can be used as input for the ambiguity function and MFI components of BCOMFI.

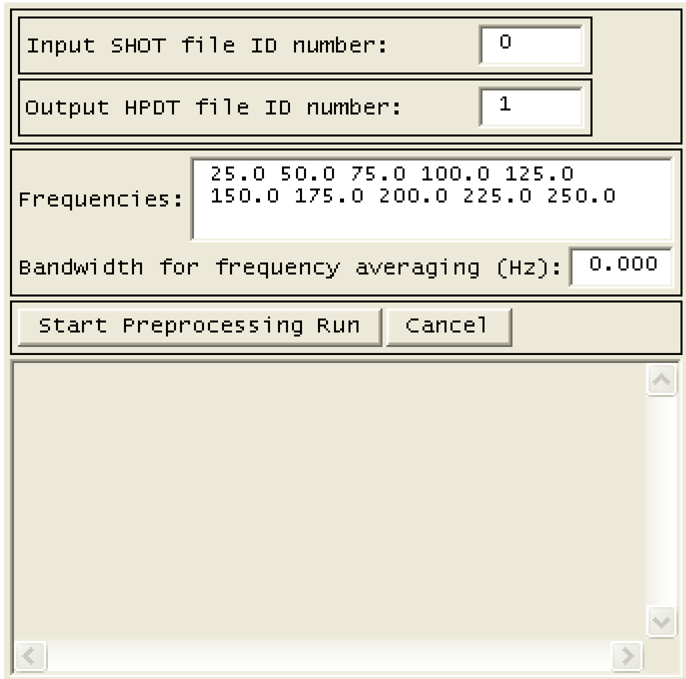
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Preprocess TD shot data to FD hpdt data

Graphical User Interface



The first items in the top row allow the user to specify the input Shot file to be used for the run, and the output Hpdt file to contain the results.
The next two lines allow the user to specify the frequencies at which the FFT'd shot data are to be retained and the bandwidth within which frequency averaging centered at these frequencies is to be performed.
The "Start Preprocessing Run" and "Cancel" buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for processing the shot data is outlined in the following pseudocode:

Specify the input Shot file and output Hpdt file.
Select the frequencies at which signal vectors are to be computed.
For each shot:
    FFT the traces for each sensor.
For each selected frequency:
    Form a frequency-averaged cross spectral matrix by accumulating the outer products of vectors within a specified band of frequencies centered at the selected frequency (optional).
Write matrices to an Hpdt file.

Using the Component

Press the “Preprocess TD shot data to FD hpdt data” button to bring up the GUI for preprocessing shot traces to hpdt form, where they can be used as input to MFI. The preprocessing involves Fourier transforming the traces, and then, for each selected frequency, forming an averaged cross-spectral matrix corresponding to the sensors of the array. The resulting data are written to an Hpdt file.

To run this conversion, specify:

- the ID number of the Shot file containing the array to be converted;
- the ID number of the output Hpdt file that will be produced;
- the frequencies at which the complex fields are to be generated from the shot traces;
- the bandwidth to use for averaging the cross-spectral matrices at each frequency.

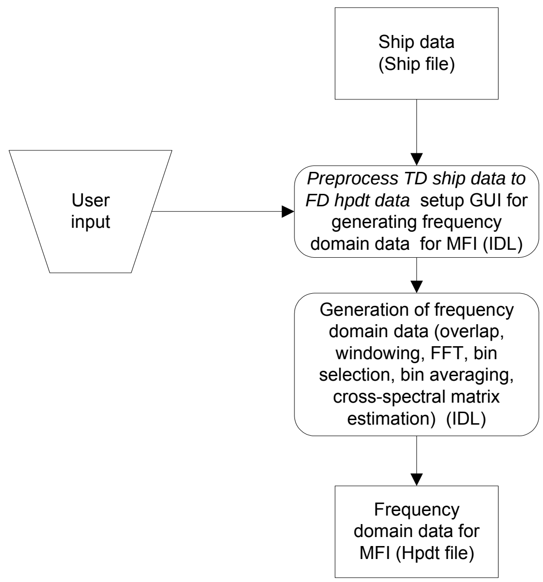
When the above have been specified, press the “Start Preprocessing Run” button to perform the preprocessing. The process should be almost instantaneous.

Preprocessing of Time Domain Ship Data to Frequency Domain Hpdt data

The time domain ship data are processed as a series of (possibly overlapping and windowed) segments to yield cross-spectral matrices for MFI analysis. The strategy for processing the ship data is to segment the time series, optionally overlap the time segments by 50% and apply a window (e.g., Hanning), and then FFT the sensor data for each segment. The data at user-specified frequencies will then be used to form cross-spectral matrices, which will be output to an Hpdt file for MFI.

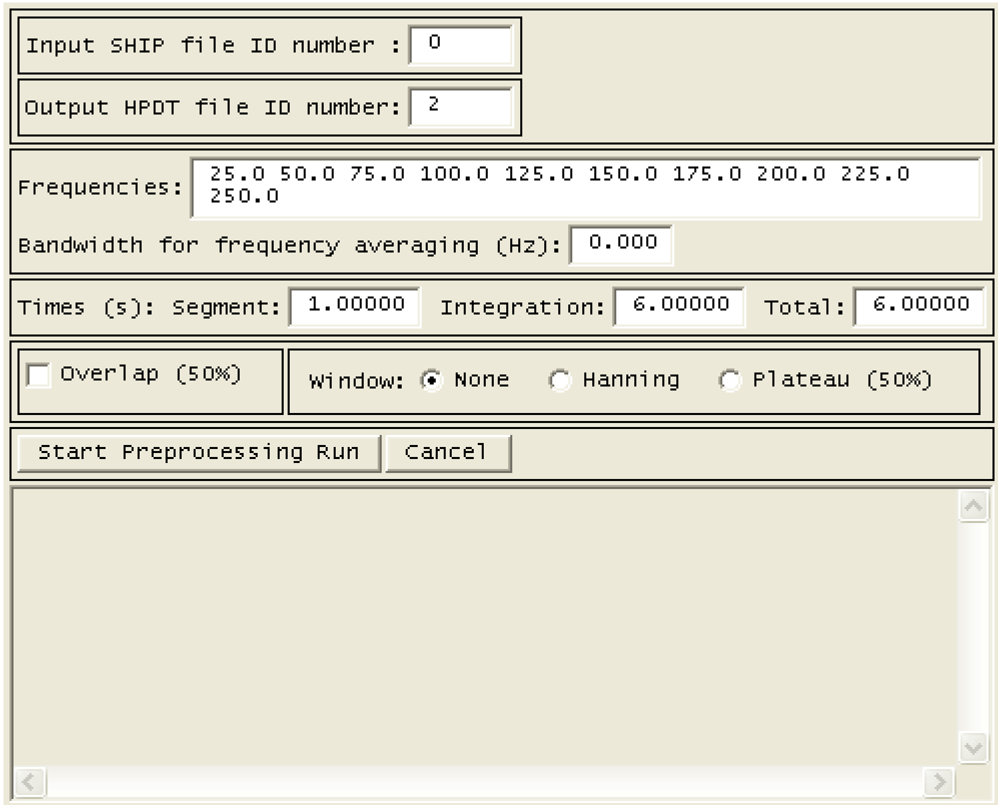
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Preprocess TD ship data to FD hpdt data

Graphical User Interface



The items in the top row allow the user to specify the input Ship file to be used for the run, and the Hpdt file to contain the results.
The next two lines allow the user to specify the frequencies at which the FFT'd ship data are to be retained and the bandwidth within which frequency averaging centered at these frequencies is to be performed.
The third row provides for specification of the segment time, the integration time (the time over which cross-spectral matrix estimation takes place), and the total time (which controls how many vectors/matrices are computed). Note that at present only the data for the first integration time can be analyzed by MFI.
The fourth row allows the user to specify the overlap to use for the time series to be FFT'd and the window to use prior to performing the FFT.
The "Start Preprocessing Run" and "Cancel" buttons are self-explanatory.
The large empty area at the bottom provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for preprocessing the ship data is outlined in the following pseudocode:

Specify the input Ship file and the output Hpdt file.
Select the frequencies at which signal vectors are to be computed.
Repeat for each integration time:
  For each sensor:
    Obtain the data for the segment (optional overlap).
    Window the data.
    FFT the data.
  For each frequency:
    Form a frequency-averaged cross spectral matrix by accumulating the outer products of vectors within a specified band of frequencies centered at the selected frequency (optional).
  Output the cross-spectral matrices to an Hpdt file.

Using the Component

Press the “Preprocess TD ship data to FD hpdt data” button to bring up the GUI for preprocessing ship traces to hpdt form, where they can be used as input to MFI. The preprocessing involves Fourier transforming sequential segments of the traces, and accumulating cross-spectral matrices at selected frequencies. Options are provided for overlapping the segments by 50%, and for windowing the segments prior to Fourier transformation for each selected frequency. The transformed data are used to form a sequence of complex cross-spectral matrices representing the fields observed at the sensors of the array. An option is also provided in the interface to average the outer products of sensor field vectors for frequency bins within a window centered on each of the chosen frequencies. The matrices generated by the preprocessing are written to an Hpdt file.

To run this conversion, specify:

- the ID number of the Shot file containing the array to be converted;
- the ID number of the output Hpdt file that will be produced;
- the frequencies at which the complex fields are to be generated from the shot traces;
- the bandwidth within which the outer products of the complex field data are to be averaged for formation of the cross-spectral matrices;
- values for segment time (data length for a single FFT), integration time (time within which outer-product matrices are averaged to accumulate cross-spectral matrices), and total time;
- whether the segments are to be overlapped by 50%;
- the window, if any, to be applied to the traces before FFT.

When the above have been specified, press the “Start Preprocessing Run” button to perform the preprocessing. The process should be very rapid.

Matched Field Techniques: Brief Overview

Matched field processing and matched field inversion are array signal processing methods which search over a parameter space for unknown model parameters by matching a replica field vector computed using a parameter-based model with "measured" data. In the present context, the replica data are computed using acoustic propagation modeling to generate the fields at an array that would be observed for a particular realization of source-array geometry and geoacoustic model. The "measured" data are obtained from an array of hydrophones or from propagation modeling using a reference source-array geometry and geoacoustic model. The replica data are matched with the "measured" data using a power processor, with high processor output indicating good matches. A maximum processor value can indicate a reasonable correspondence between the model parameters and the actual structure of the environment.

The normalized Bartlett power processor is used here to compute the matching (ambiguity) function between the measured data and the replica vectors generated using ORCA. For a measured data vector **m** and a replica vector **r**(**p**) computed for parameter set **p** at a single frequency**,** the Bartlett processor is defined as

$$B(\mathbf{r}(\mathbf{p}),\mathbf{m}) = \frac{|\,\mathbf{r}(\mathbf{p})^{*}\mathbf{m}\,|^{2}}{|\,\mathbf{r}(\mathbf{p})\,|^{2}|\,\mathbf{m}\,|^{2}}\,.$$

For a measured data cross-spectral matrix **M,** it is defined as

$$B(\mathbf{r}(\mathbf{p}),\mathbf{M}) = \frac{|\,\mathbf{r}(\mathbf{p})^{*}\mathbf{M}\mathbf{r}(\mathbf{p})\,|}{|\,\mathbf{r}(\mathbf{p})\,|^{2}\|\,\mathbf{M}\,\|}\,,$$

where ||**M**|| is the spectral norm of the cross-spectral matrix **M**.

For multi-frequency measured data, a processor that combines the Bartlett outputs is defined as

$$C(\mathbf{p}) = \sum_{j=1}^{N} w_j B(\mathbf{r}_j(\mathbf{p}),\mathbf{m}_j) \quad \text{or} \quad C(\mathbf{p}) = \sum_{j=1}^{N} w_j B(\mathbf{r}_j(\mathbf{p}),\mathbf{M}_j)$$

for vector and cross-spectral matrix data, respectively, where $\mathbf{r}_j(\mathbf{p})$ is the replica vector for parameter set **p** at the *j*th frequency, $\mathbf{m}_j$ is the measured vector at the *j*th frequency, *N* is the number of frequencies, and w_j is a weight ($\sum w_j = 1$).

Matched field techniques always involve ambiguity, in that many local maxima may be present in the parameter space of interest, and these matches can often be comparable to those obtained using the true parameter set. Also, matches can be very insensitive to some parameters, or two parameters may be highly correlated, giving rise to other forms of ambiguity. The presence of noise introduces additional ambiguity. Hence, the ability to visualize the ambiguity in matched field techniques provides valuable insight into the characteristics of the parameter space and the validity of estimation of the geometric or geoacoustic model parameters. Based on these considerations, functionality has been provided in BCOMFI to allow the computation and visualization of ambiguity functions of up to three dimensions. The use of these ambiguity functions is described in the following section.

Matched field inversion is routinely accomplished by repeated forward modeling to generate replicas and matching the fields with the "measured" data. The use of optimization methods allows the parameters to be adjusted so as to achieve the best fit between the replica and "measured" data. Since there can be many local optima in the parameter space, it is essential to have an approach that has both global and local aspects. The approach to MFI taken in BCOMFI is to have an initial global search stage followed by local optimization of each of a number of the best matches found during the search stage. This approach is described in the Matched-Field Techniques: Inversion section below.

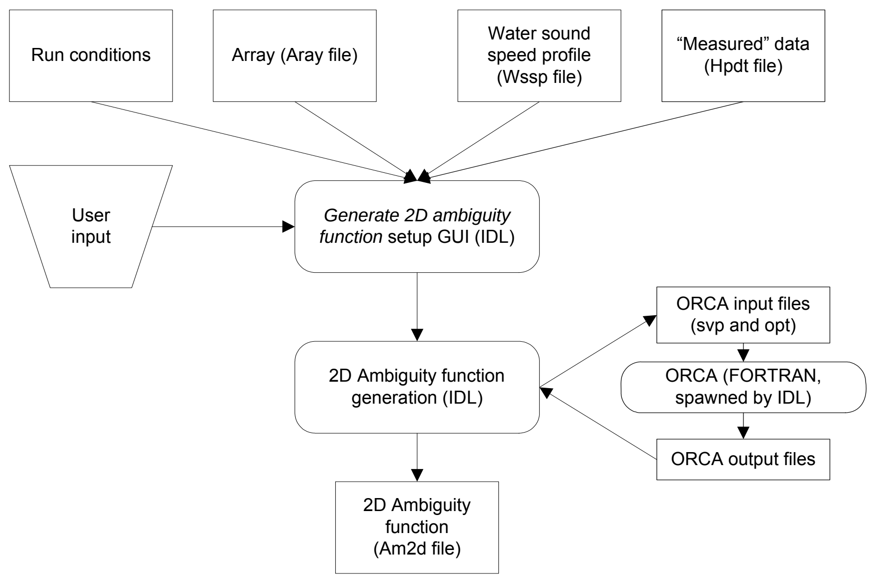
Matched Field Techniques: Ambiguity Function Generation

In investigations involving MFI, it is useful to be able to examine how the ambiguity function (the matching function used as an objective function in MFI) depends on the individual parameters, and, sometimes, groups of parameters. To allow the visualization of this behavior, a component of BCOMFI has been provided that allows the generation of 1D, 2D, and 3D ambiguity functions (higher-dimension grids are too time-consuming to generate and more difficult to visualize). The user can then use IDL display software to examine the characteristics of these functions, including dynamic range, peak widths, presence of multiple optima, parameter sensitivity, and, in the case of 2D or 3D ambiguity functions, parameter interdependency. This last item is of particular significance, since it can lead to ill-posed MFI problems and inconsistent results in parameter estimation. Visualization of the ambiguity functions for the parameter space can assist in the interpretation of such results.

This section describes the ambiguity function generation component, using the 2D case as an example; the 1D and 3D versions are analogous to the 2D version.

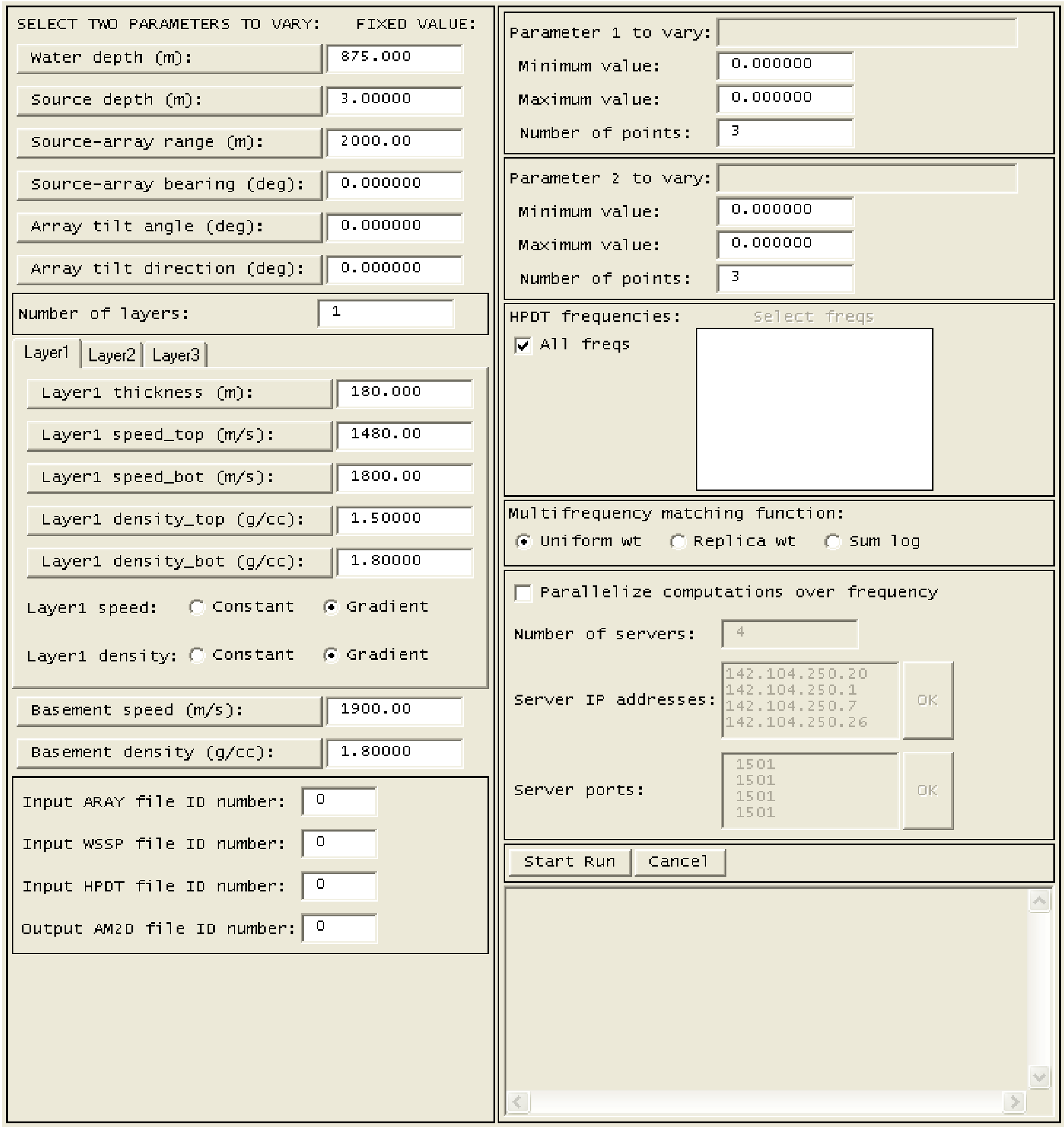
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Generate 2D ambiguity function

Graphical User Interface



The block (including tabs) on the top left lists the 23 parameters that can be varied to generate the ambiguity function, and provides default fixed values for those parameters. Pressing on the button containing a parameter name chooses that parameter as one of the two to be varied and populates one of the "Parameter 1 to vary" or "Parameter 2 to vary" fields in the blocks at the top right.

The block under the tabs provides an option for the user to force the indicated parameter pairs for the top (e.g., density1) and bottom (e.g., density2) of the corresponding layer to be the same.

The lower block on the left allows the user to specify the input files (Aray, Wssp, and Hpdt) to be used for the run, and the Am2d file to contain the results.

The top two blocks on the right allow specification of the domain over which the parameters are to be varied, and the number of points in each dimension.

The third block on the right allows the user to view the frequencies of the data in the Hpdt file and select a subset of these to use in computing the multi-frequency ambiguity function. An option to select all the frequencies in the data is also provided.

The fourth block on the right provides options for how the matches at the multiple frequencies are to be combined in the overall match. For the uniform and replica options, the Bartlett outputs at the frequencies are weighted uniformly or according to the squared modulus of the replica vector, respectively. For the Sum log option, the expression $\sum_{j=1}^{N} \log(1 - B(\mathbf{r}_j(\mathbf{p}), \mathbf{m}_j))$ (or $\sum_{j=1}^{N} \log(1 - B(\mathbf{r}_j(\mathbf{p}), \mathbf{M}_j))$ for matrix data) is used.

The fifth block on the right allows for parallel processing over frequency, with the server IP addresses and ports of the remote computers specified in the text boxes.

The “Start Run” and “Cancel” buttons are self-explanatory.

The large empty area at the bottom right provides a field for status messages to be displayed.

Algorithm

The algorithm to be used for generating the 2D ambiguity function is outlined in the following pseudocode:

Specify the input Aray file and the output Hpdt file, and specify the run conditions.
Check that the run conditions are consistent.
Apply a tilt to the array.
For each value of parameter 1:
  For value of parameter 2:
    For each frequency selected (optionally parallelized):
      Apply tilt values to the array positions.
      Generate an array geometry file for ORCA.
      Generate an opt file for ORCA.
      Generate an svp file for ORCA.
      Spawn an ORCA process to generate a replica vector.
      Compute the Bartlett power of the match for that frequency.
    Sum the weighted Bartlett powers for the frequencies to give the ambiguity function for (parameter 1, parameter 2).
Output the ambiguity function to an Am2d file.

Using the Component

Press the “Generate 2D ambiguity function” button to bring up the GUI for computing a 2D ambiguity function. This will compute a 2D array of Bartlett matches for pairs of selected parameters being varied, with fixed values for the other parameters. To generate the 2D ambiguity function, do the following:

- Select the parameters to vary, by clicking on the corresponding text buttons containing the names in the left column; these parameter names will then appear in the two text fields on the top right.
- Specify the values to use for the fixed parameters in the editable fields on the left portion of the interface. To change the number of layers (which can be from 0 up to 3) specify this number in the “Number of layers” field. To access the parameters in a layer other than the present layer, click on one of the tabs: Layer1, Layer2, Layer3.
- Ensure that radio buttons are selected to enable or disable gradients for sound speed and density in the corresponding layer.
- Specify the upper and lower limits for the two parameters to vary, and the number of points to evaluate in each parameter dimension.
- Specify ID numbers for the input Aray, Wssp, and Hpdt files, and the output Am2d file to contain the computed ambiguity function.
- Specify the frequencies in the Hpdt file to use in the matching. Checking the “All freqs” checkbox uses all the frequencies in the file. Unchecking the box enables the option to select one or more of the individual frequencies in that file. Note that the Hpdt file with the specified ID number must exist in order to be able to use this option.
- Specify the option to use for combining the frequencies in the final matching function.
- If the computations are to be run in parallel over frequency, check the “Parallelize computations over frequency” checkbox and ensure that the numbers of servers, the IP addresses and the ports specified in the text boxes are correct. If you edit these fields, note that you must press the respective “OK” button for the changes to take effect. (If you wish to change the default values, edit the file “servers.dat” in the \source directory.) Also ensure that orcaserver processes are running on each of the computers selected (see the Installing the BCOMFI Software section above for information on installing and running these processes on Windows and Linux computers).
- Press the “Start run” button.

The result of the run will be an Am2d file, which can be viewed using the “Display 2D ambiguity function” option on the Main Menu interface.

Matched-Field Techniques: Inversion
Background

The purpose of the MFI component is two-fold:

to provide a test bed for investigation of MFI techniques, and
to allow the analysis of real data by means of MFI.

The aim of the MFI software component of BCOMFI is to provide an environment for modeling and MFI using synthetic data and various parameterizations. Using this software, studies can then be performed to determine which approaches will be effective for detecting changes within the hydrate stability zone. Using synthetic data generated by the above components, the software can allow investigation of the following questions, for example:

What are the relative sensitivities, peak widths, oscillations, and dynamic ranges of the matching function with respect to each of the 23 model parameters?
How do the above parameter characteristics vary with range and frequency?
Can we estimate the numbers of optima in the entire search region?
Under what conditions can we ignore density optimization in the matching?
What is the effect of errors in one or more of the fixed parameters?
What is the effect of allowing a gradient in the sediment when the parameter is constant, or forcing it to be constant when there is in fact a gradient?
What is the effect of multiple sediment layers in the data above the hydrate stability zone, when the matching model contains fewer layers?
What is the effect of additional layers below the hydrate stability zone, when the matching model contains fewer layers?
What is the effect of noise of various types, both uncorrelated and correlated?
What is the effect of multiple sources and source motion on the matching?
What is the effect of a gas layer at an interface?
When is regularization required to obtain consistent inversion results?
Under what conditions and for what parameters can we expect to detect significant changes that would indicate alterations in the hydrate-containing layer? That is, how substantial would changes have to be before they could be reliably detected?
How do the answers to the above questions change when we move to a range-dependent environment?

The MFI component also provides an environment for the future analysis of real data (both shot and ship) with the intended result of defining a standard model, or perhaps a sector-dependent set of models, that provide a reasonable representation of the environment in the region of the array. These models might then be used as a basis for real-time monitoring.

In designing and implementing the MFI component, we considered that MFI is a nonlinear process that is generally approached using optimization techniques that repeatedly solve the forward problem for varying sets of parameters until a suitable good match to the data is obtained. MFI optimization approaches must be able to deal with the following challenges:

- There are typically 5 – 25 parameters, and it is required to be able to optimize any or all combinations of these.
- There are generally multiple local optima present in the parameter space.
- It is desirable to restrict the domain of the parameters, usually by bounds constraints.
- The sensitivities of the parameters can be very different (by a factor of 100 or more).
- The parameters need to be scaled for optimization.
- Derivatives are unavailable except by numerical approximation.
- Certain parameters can be correlated, leading to ill-posed problems.
- There is a possibility of discontinuities in the matching function, particularly if the propagation algorithms do not always successfully converge/complete (as has been observed to be the case with ORCA under some conditions).

MFI has been implemented using a global search / local optimization approach. The advantages of this approach are the ability to obtain multiple estimates of the various optima, and the moderate number of function evaluations required (e.g., 2000 for search space sampling and 3000 for 10 optimizations). While a potential drawback of this method is that the optimized estimates are local, a sufficiently comprehensive search stage will increase the likelihood that one of the optima is, in fact, global.

The ultimate result of the investigations conducted using the MFI component will be to compute effective range-independent geoacoustic models of the environment in the region of the monitoring station. Based on the data acquired during the calibration stage, it is likely that these models will be region-dependent; i.e., a different model may be derived by MFI for each of a number of regions in the general area. One possible approach to monitoring would be to use such models as standards and then look for mismatches between these models and the array data obtained from passing ships.

Implementation

The overall method implemented for MFI involves the following stages:

A global search stage. In this stage, sets of values for the parameters to be varied are generated that are randomly distributed between the lower and upper bounds for those parameters. ORCA is then used to generate a replica vector for this parameter set, and the matching function is then computed. This process is repeated a specified number of times to provide a sampling of the overall search space.

A local optimization stage. In this stage, each of a specified number of the best matches found in the global search stage is used as a starting point for optimization of the parameters. Here the objective function $F(\mathbf{p})$ of the parameter set **p** to be minimized is:

$$F(\mathbf{p}) = 1 - C(\mathbf{p}) + P(\mathbf{p}) + R(\mathbf{p}),$$

where $C(\mathbf{p})$ is the processor output, $P(\mathbf{p})$ is a penalty function for values of the parameters outside their bounds, and $R(\mathbf{p})$ is a regularization function.

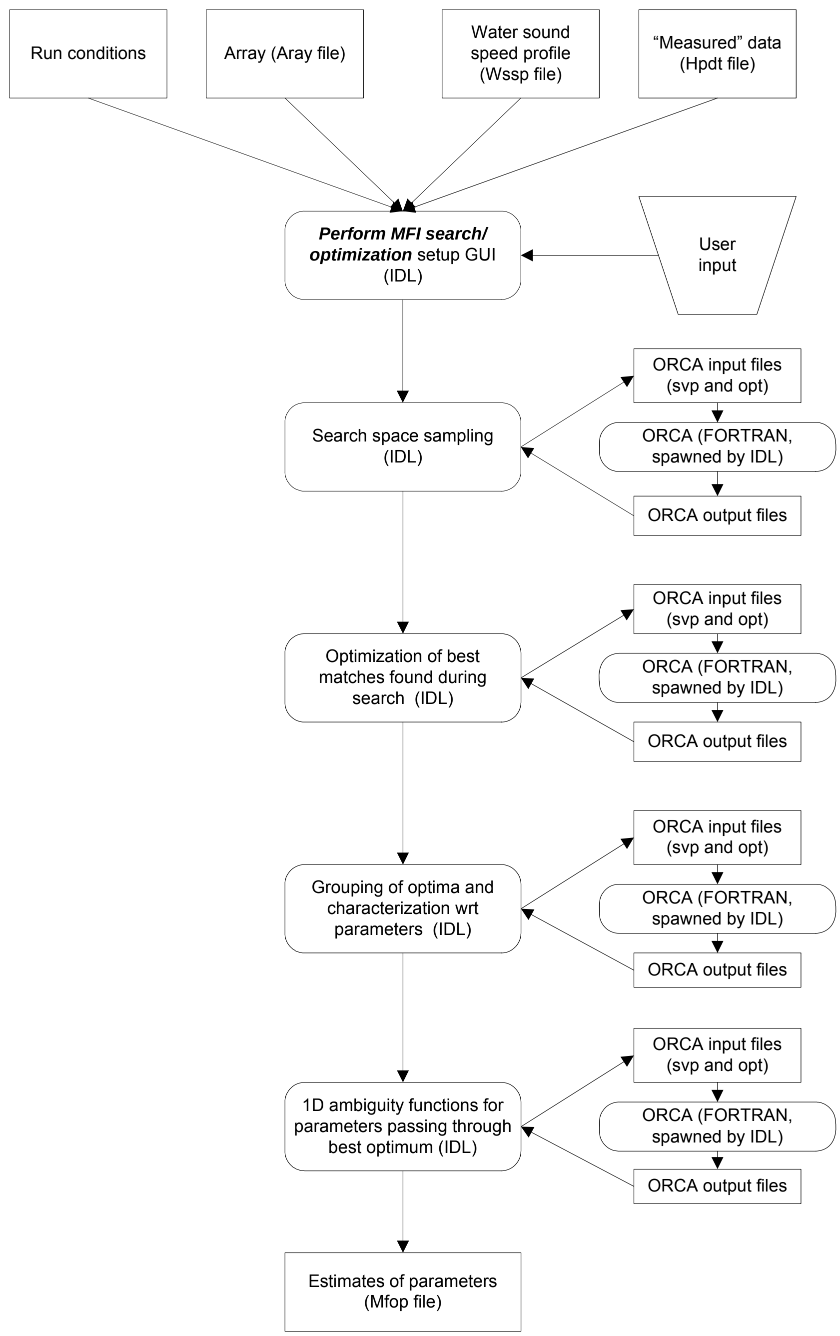
The IDL routine DFPMIN (a quasi-Newton method for which the required derivatives can be approximated numerically by central differences) is used to perform the optimizations. The result is a set of independent estimates of optimized parameter values.

A grouping stage. In this stage, the optima are analyzed and multiple estimates of the same optimum are grouped together. This is done by starting with the optimum with the lowest objective function value and determining other optima with sufficiently similar values for the function value. The objective function is evaluated along a line in the parameter space connecting the optima and they are grouped if the objective function along this line does not exceed a specified threshold.

A parameter characterization stage. In this stage, the best optimum is identified and the objective function is evaluated independently for each varied parameter along a line spanning the region between its lower and upper bounds and intersecting the peak. This provides an estimate of the sensitivities of the individual parameters, and an indication of the peak width, dynamic range, and oscillations of the function with respect to each parameter.

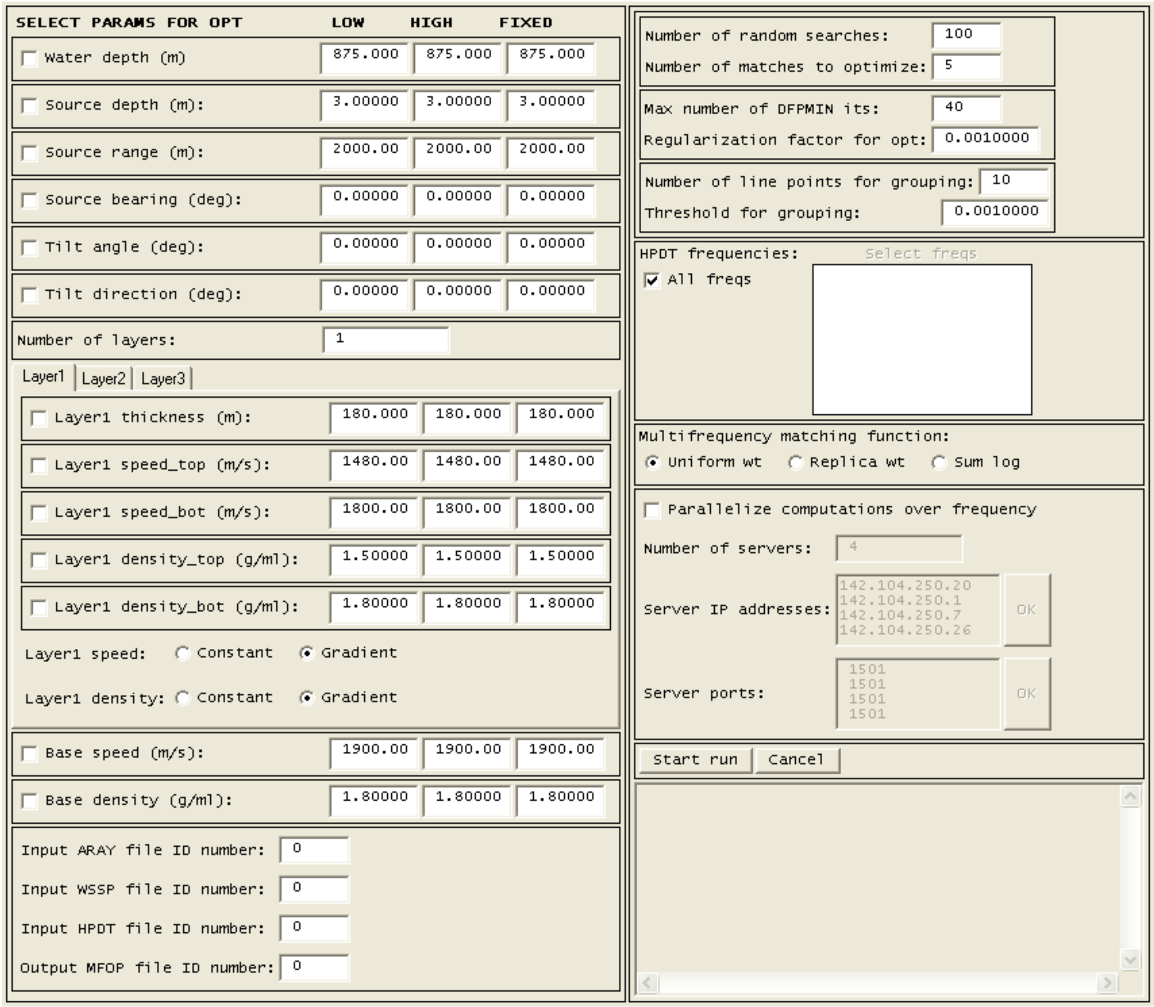
Overall Design

In this design, data and/or files are indicated by rectangles, processes are represented by rounded rectangles, and user input is denoted by a trapezoid.



Processing for
Perform MFI search/optimization

Graphical User Interface



The block on the top left (including tabs) lists the 23 parameters that can be chosen for optimization and provides bounds and default fixed values for those parameters. The checkboxes are used to select those parameters which are to be optimized. The lower and upper bounds for selected parameters can then be set using the "LOW" and "HIGH" fields for the parameter (in which case the "FIXED" value is ignored). Note that the "Layer*n* speed" and "Layer*n* density" buttons under the tabs provide an option for the user to force the indicated parameter pairs for the top and bottom of the corresponding layer *n* to be the same.

The lower block on the left allows the user to specify the input files (Aray, Wssp, and Hpdt) to be used for the run, and the Mfop file to contain the results.

The top block on the right allows the user to specify the conditions for the search stage, i.e., the number of random samples of the parameter space and the number of best matches to optimize.

The second block on the right allows specification of convergence and regularization factors for the optimization.

The third block on the right allows the user to specify conditions for the grouping analysis of the multiple peaks.

The fourth block on the right allows the user to view the frequencies of the data in the Hpdt file and select a subset of these to use in computing the multi-frequency ambiguity function. An option to select all the frequencies in the data is also provided.

The fifth block on the right provides options for how the matches at the multiple frequencies are to be combined in the overall match. For the uniform and replica options, the Bartlett outputs at the frequencies are weighted uniformly or according to the squared modulus of the replica vector, respectively. For the Sum log option, the expression $\sum_{j=1}^{N} \log(1 - B(\mathbf{r}_j(\mathbf{p}), \mathbf{m}_j))$ (or $\sum_{j=1}^{N} \log(1 - B(\mathbf{r}_j(\mathbf{p}), \mathbf{M}_j))$ for matrix data) is used.

The sixth block on the right allows for parallel processing over frequency, with the server IP addresses and ports of the remote computers specified in the text boxes.

The “Start Run” and “Cancel” buttons are self-explanatory.

The large empty area at the bottom right provides a field for status messages to be displayed.

Algorithm

The algorithm to be used to perform matched-field inversion using the search/gradient optimization method is outlined in the following pseudocode:

Specify the input Aray and Wssp files and the output Hpdt file, and specify the run conditions, including those parameters that are to be optimized and their bounds.
Check that the run conditions are consistent.
Generate specified number of random samples of the search space and rank the results in order of the best matches.
For each of a specified number of the best matches:
    Using the current best match as an initial estimate, call the optimization function to optimize the parameters.
Identify those converged optima which are estimates of the same peak and group them together to form unique estimates.
Generate 1D ambiguity functions passing through the best optimum for each parameter, and estimate the characteristics of the parameters (number of peaks, peak width, etc.).
Write results to an Mfop file.

The algorithm for computing the objective function, given a set of parameters provided by the optimization algorithm, is as follows:

For each frequency selected (optionally parallelized):
- *Apply tilt values to the array positions.*
- *Generate an array geometry file for ORCA based on the input parameters.*
- *Generate an opt file for ORCA based on the input parameters.*
- *Generate an svp file for ORCA based on the input parameters .*
- *Spawn an ORCA process to generate a replica vector.*
- *Compute the Bartlett power of the match for that frequency.*

Combine the weighted Bartlett powers for the frequencies to give the ambiguity function for the input parameters.

Apply penalty function for parameters outside the bounds, and add a (small) regularization function term which increases quadratically with distance from the midpoint of the parameter range.

Using the Component

To perform MFI using the search/optimization procedure, press the "Perform MFI search/optimization" button on the Main Menu to bring up the GUI. Then do the following:

- Select the parameters to optimize, by clicking on the checkboxes beside the names in the left column.
- Specify the lower and upper bounds for each parameter to be varied in the corresponding text fields (under columns LOW and HIGH) for that parameter.
- To change the number of layers (which can be from 0 up to 3), specify this number in the "Number of layers" field. To access the parameters in a layer other than the present layer, click on one of the tabs: Layer1, Layer2, Layer3.
- Specify the values to use for the fixed parameters in the corresponding fields for those parameters (under column FIXED).
- Ensure that radio buttons are selected to enable or disable gradients for sound speed and density in the corresponding layer.
- Specify ID numbers for the input Aray, Wssp, and Hpdt files, and the output Mfop file to contain the results of the run.
- Specify the number of random samples to be generated during the global search stage, and the number of best matches found during this stage to optimize in the next stage.
- Specify the maximum number of iterations for the IDL optimization algorithm DFPMIN, and the regularization scaling factor for the quadratic function used in the regularization.
- Specify the number of points along a line connecting two possibly different optima when performing the grouping, and the threshold for the grouping.
- Specify the frequencies in the Hpdt file to use in the matching. Checking the "All freqs" checkbox uses all the frequencies in the file. Unchecking the box enables the option to select one or more of the individual frequencies in that file.

Note that the Hpdt file with the specified ID number must exist in order to be able to use this option.
Specify the option to use for combining the frequencies in the final matching function.
If the computations are to be run in parallel over frequency, check the "Parallelize computations over frequency" checkbox and ensure that the numbers of servers, the IP addresses and the ports specified in the text boxes are correct. If you edit these fields, note that you must press the respective "OK" button for the changes to take effect. (If you wish to change the default values, edit the file "servers.dat" in the \source directory.) Also ensure that orcaserver processes are running on each of the computers selected (see the Installing the BCOMFI Software section above for information on installing and running these processes on Windows and Linux computers).
Press the "Start run" button. During each stage, graphical information (objective function and parameter values) will be plotted to the screen every 20^{th} function call. This provides a convenient way of monitoring the progress of the search, optimization, and other stages.

The result of the run will be an Mfop file, which can be viewed using the "Display MFI search/optimization results" option on the Main Menu interface.

Matched Field Techniques: Parallelization

Matched-field methods are computationally intensive, and to reduce the times involved it is necessary to develop implementations of these techniques where the computations are parallelized in some way. Parallelization is advantageous both for the computation of ambiguity functions (particularly in the case of higher dimensions) and especially for MFI runs, which can involve many thousands of function evaluations.

Since the basic unit of computation here is a run of ORCA at a single frequency, a parallelization option for BCOMFI was developed in which the computations are distributed among processors according to frequency. This approach also allows convenient separation of client and server processes, with IDL code running on the client and C/Fortran code running on the server. The parallelization was implemented as follows:

An IDL client sockets routine was written that used the IP addresses of a number of servers (on single or multi-processor computers) and an array containing the names of the input and output files for ORCA at each of a number of frequencies.
An ORCA executable module was produced and installed on each of a number of server processors.
A C server routine was written that accepted a set of files for a single frequency from a client, performed an ORCA run using these files, and transferred the

output file containing complex field values back to the client. Note that the server processes should not be run on the machine that is running the client. The IDL code for the matched field computations was modified to provide an option for parallelized multi-frequency processing. In this implementation, the input files for ORCA at all these frequencies are generated, and the IDL client sockets routine is called. This routine controls the transfer of the files to the servers and handles the receiving of the resulting output files. It continually assigns tasks to servers as they become available, until the fields at all selected frequencies have been computed. The IDL code then inputs the complex field data and performs the matching.

The implementation was designed to be general, in that it allowed distribution of N tasks (i.e., runs at a single frequency) among M processors. The servers were implemented in both Windows and Linux, with ORCA executables compiled for both environments.

BCOMFI Display Components

The BCOMFI display functions allow the viewing of the simulated shot and ship traces, the ambiguity functions, and the results of MFI search/optimization. Certain of these applications (e.g., plots of shot and ship traces) have been implemented using IDL direct graphics, while others make use of the IDL "itools" interfaces (which include iplot, isurface, iimage and ivolume). These interfaces provide useful functionality that is not available in direct graphics (e.g., copy and paste), at the cost of speed and memory requirements (for which reason it was not used in plotting traces from Shot and Ship files). Components of the itools interfaces are used in displaying the 1D, 2D and 3D ambiguity functions, with an additional special-purpose IDL slicer tool being provided for 3D volume visualization, while direct graphics are used in display of the search/optimization results. (Note that for the latter, the screen resolution should be set to at least 1280 by 1024 to view the full extent of some of the plots.) The itools interfaces are designed for a high level of user interaction, and some effort should be made to become familiar with their functionality (a tutorial is available at the RSI website http://www.rsinc.com/idl/idl_itools.asp).

Displaying Time Domain Shot Data Traces

Press the "Display shot data" button to plot the traces in a particular Shot file. A file browser dialog box will appear, which will display all files of the type "shotnnnnn.sav", where nnnnn is an ID number. After you select the desired file, the traces in that file, and their corresponding amplitude spectra, will be displayed in separate windows as waterfall plots.

Displaying Time Domain Ship Data Traces

Press the “Display ship data” button to plot the data streams (actually, long traces) in a particular Ship file. A file browser dialog box will appear, which will display all files of the type “shipnnnnn.sav”, where nnnnn is an ID number. After you select the desired file, the data streams in that file, and their corresponding amplitude spectra, will be displayed in separate windows as waterfall plots.

Displaying 1D Ambiguity Functions

Press the “Display 1D ambiguity function” button to plot a 1D ambiguity function. A file browser dialog box will appear, which will display all files of the type “am1dnnnnn.sav”, where nnnnn is an ID number. After you select the desired file, there may be a momentary delay while itools loads, and an itools splash screen may appear and vanish. The plot will then be displayed in an iplot window.

Displaying 2D Ambiguity Functions

Press the “Display 2D ambiguity function” button to plot a 2D ambiguity function. A file browser dialog box will appear, which will display all files of the type “am2dnnnnn.sav”, where nnnnn is an ID number. After you select the desired file, a setup GUI will then appear, in which you can specify the conditions for the display. First, choose whether you want the display as a surface (which can be rotated in 3D) or as an image. You then have the option of resizing the 2D function before display (which you will usually want to do if the Image option is chosen). If you have chosen the Surface option, you can then specify the shading algorithm to be used (either flat or Gouraud). If you have chosen the Resize option, you can also specify the interpolation scheme (bilinear or cubic).

Once you have specified the conditions, press the Go button. There may be a momentary delay while itools loads, and an itools splash screen may appear and vanish. If the Surface option was chosen, the ambiguity function will presently be displayed in an isurface window. You can use the Rotate button on this window to rotate and examine the surface in 3D. If the Image option was chosen, the function will be displayed as an iimage window. Note that this plot does not display and label the axes, but these may be added using options in the itools interface.

Displaying 3D Ambiguity Functions

Press the “Display 3D ambiguity function” button to plot a 3D ambiguity function. A file browser dialog box will appear, which will display all files of the type “am3dnnnnn.sav”, where nnnnn is an ID number. After you select the desired file, there may be a momentary delay while itools loads, and an itools splash screen may appear and vanish. Then two separate interfaces will be displayed.

One of these is the ivolume tool, which displays a bounding box with labeled axes, and allows rendering using the button on the right side, and image plane and isosurface generation using the Operations Volume menu item. Note that you may sometimes have to invoke the Edit Select All menu item to enable this functionality. You can use the Rotate button on this window to rotate and examine the isosurface in 3D.

The second interface is the 3D Data Visualizer (Slicer3) interface, which provides a different interface for slice, isosurface, and projection operations through its Mode droplist. Note that you may need to use the Tools Erase menu item to clear the display area of existing images before displaying a new one. Also note that while the ivolume interface has the capability of copying and pasting the display, the Slicer3 interface does not, and so to save a Slicer3 image, a screen dump utility would have to be used.

Displaying Results of MFI

Press the “Display MFI search/optimization results” button to show several different types of results from an MFI search/optimization run. A file browser dialog box will appear, which will display all files of the type “mfopnnnnn.sav”, where nnnnn is an ID number. After you select the desired file, a Setup GUI will then appear, from which you may select the following options for display:

Search. This option plots the successive values of mismatch during the random global search stage. Note that, since mismatch is plotted, the best matches are those with the *lowest* values.

Optimization convergence. This option plots the evolution of the objective function value and the corresponding parameters during the course of an optimization. It is used in conjunction with the droplist at the bottom of the GUI, which specifies which one of the *M* best matches found during the search stage is to be viewed. (For example, if 0 is chosen, the optimization course for the best match is shown; if 1 is chosen, the course for the second best match is chosen, and so on.) The plots are displayed as objective function, or parameter value, versus the number of function calls.

1D ambiguity functions at optimum. This option plots the Bartlett matches obtained as each separate parameter is varied from its lower to its upper bound along a line passing though the “best” optimum (i.e., that with the lowest objective function value). It provides an indication of the sensitivity and ambiguity associated with each parameter.

Show groups found. This displays a text window containing information about the conditions for the run and the results of grouping the various optima found. Each group is displayed separately, along with the number of optima (convergences) in that group, the mean function value and parameter values for the group, and, where possible, the standard deviations of the multiple estimates for that group. The results of the peak analysis for the 1D ambiguity functions computed for the best group are then listed separately for each parameter. These include the number of peaks and extrema in the function, the dynamic range (on a scale from 0 to 1), the peak width in both relative units (where 1.0 represents the range between lower and upper bounds) and absolute (physical) units, and the sensitivity (i.e., the RMS variation of the function value between adjacent points along the line).

Print groups found. This sends the information displayed in the above text window to the default printer.

When you have selected one of the above options, press the Go button. The program will then perform the specified action.

Computing Environment

The BCOMFI development was done on a Windows XP computer running the IDL Development Environment (IDL DE), and on which executables of the Fortran programs ORCA and RAM were prepared and installed.

The BCOMFI software delivered in the zipfile “bcomfi.sav” requires a Windows computer on which the IDL VM has been installed. The BCOMFI executable runs under the IDL VM and calls the ORCA and RAM executables provided with the delivery.

The parallelized option for matched field computations is designed for the client process to run on a Windows computer, but to make use of ORCA servers running on either Windows or Linux computers to perform the propagation modeling for the matched field computations. (Note that RAM is currently not involved in the matched field components.) Hence, this parallelized option can be run with a client Windows computer containing the IDL VM and a network of server computers – either Windows or Linux – on which ORCA server processes are running. Note that the server processes should not be run on the machine that is running the client.

Hence, the computing environment recommended for using BCOMFI is:

> a Windows computer on which the IDL VM is installed, which allows the simulation, conversion, preprocessing, matched field, and display components of BCOMFI to be run;
> optionally, a network of Windows and Linux computers with ORCA servers installed and running, which allows the matched field components (ambiguity function generation and MFI) to be efficiently run.

Appendix A: Data Structures

We note here that the BCOMFI data structures described in this section also correspond to files, such that a file with a particular five-character prefix (e.g., aray00023.dat) would hold a data structure of the same name (an Aray structure).

Am2d

The Am2d structure is used as an example of structures used in ambiguity function generation (the Am1d and Am3d structures are analogous). The structure contains the following tags:

param1. The first parameter to vary.
init1. The lower limit of the domain to vary param1.
final1. The upper limit of the domain to vary param1.
incr1. The increment for varying param1.
num1. The number of points for varying param1.
param2. The second parameter to vary (for 2D only).
init2. The lower limit of the domain to vary param2 (for 2D only).
final2. The upper limit of the domain to vary param2 (for 2D only).
incr2. The increment for varying param2 (for 2D only).
num2. The number of points for varying param2 (for 2D only).
layer1_speed_constant. A flag indicating whether the compressional speed in layer 1 is to be held constant.
layer1_density_constant. A flag indicating whether the density in layer 1 is to be held constant.
layer2_speed_constant. A flag indicating whether the compressional speed in layer 2 is to be held constant.
layer2_density_constant. A flag indicating whether the density in layer 2 is to be held constant.
layer3_speed_constant. A flag indicating whether the compressional speed in layer 3 is to be held constant.
layer3_density_constant. A flag indicating whether the density in layer 3 is to be held constant.
aray_id. The ID number for the input Aray file containing the array geometry.
wssp_id. The ID number for the input Wssp file containing the water column sound speed profile.
hpdt_id. The ID number for the input Hpdt file containing the data for MFI.
am2d_id. The ID number for the output Am2d file to contain the ambiguity function.
all_freq. A flag indicating whether all the frequencies in the Hpdt file are to be used for matching.
num_freq. The number of frequencies in the Hpdt file that are to be used for matching.
freq. The frequencies in the Hpdt file that are to be used for matching.
use_freq. A flag indicating which of the frequencies in the Hpdt file (in order) are to be used for matching.

match_fun. A string indicating the scheme for combining the results at multiple frequencies.
matches. A 3D array containing the individual matches at each frequency and pair of parameter values.
amb_fun_incoh. A 2D array containing the ambiguity function computed using the standard Bartlett processor (i.e., incoherent with frequency).

Aray

The Aray structure holds the data for the array, and contains the following tags:

description. A text description of the array type of characteristics.
num_sens. The number of sensors in the array.
tilt_angle. The tilt angle from vertical (degrees).
tilt_direction. The direction in which the array is tilted (degrees true).
sens_x. The *x*-coordinates of the sensors (before tilt is applied).
sens_y. The *y*-coordinates of the sensors (before tilt is applied).
sens_z. The *z*-coordinates of the sensors (before tilt is applied).
tether_z. The tether depths of the sensors (allowing the application of tilt).

Hpdt

The Hpdt structure holds the simulated and real frequency domain data to be used for input to MFI, and contains the following tags:

hpdt_id. The ID number for the Hpdt file containing the data for MFI.
aray_id. The ID number for the Aray file containing the array geometry.
num_sens. The number of sensors in the array from which the data were obtained.
prop_model. A string containing the propagation model to be used ('ram' or 'orca').
slope. The bottom slope in degrees, if RAM is to be used (with positive slopes the depth increases from the array to the source).
num_freq. The number of frequencies at which data are present.
freq. A vector containing the actual frequencies at which data are present.
data_type. A string with a value of either "vector" or "matrix".
seed. A seed for the random number generator (used for noise and phase randomization).
num_seg_int. The number of segments in one integration time.
num_int_tot. The total number of integration times.
seg_time. The time for a single data segment.
int_time. The integration time for the data.
tot_time. The total time for the full set of data.
wn_level_db. The white noise level at a sensor.
sn_level_db. The spherical noise level at a sensor.

cn_level_db. The cylindrical noise level at a sensor.
num_source. The number of sources (max 2).
source_level_db. The source intensity levels (dB re 1 μPa at 1 m).
source_depth. The depth(s) of the source(s).
source_range. The range(s) of the source(s).
source_bearing. The bearing(s) of the source(s).
source_speed. The speed(s) of the source(s).
source_heading. The heading(s) of the source(s)
fdata. A multi-dimensional array containing the complex fields for the sensors, frequencies, and times.

Mfop

The Mfop structure holds the conditions for, and the results of, an MFI run; it is made up of six other structures, as follows:

Ctrl

The Ctrl structure holds the basic conditions for an MFI run, and contains the following tags:

layer1_speed_constant. A flag indicating whether the compressional speed in layer 1 is to be held constant.
layer1_density_constant. A flag indicating whether the density in layer 1 is to be held constant.
layer2_speed_constant. A flag indicating whether the compressional speed in layer 2 is to be held constant.
layer2_density_constant. A flag indicating whether the density in layer 2 is to be held constant.
layer3_speed_constant. A flag indicating whether the compressional speed in layer 3 is to be held constant.
layer3_density_constant. A flag indicating whether the density in layer 3 is to be held constant.
aray_id. The ID number for the input Aray file containing the array geometry.
wssp_id. The ID number for the input Wssp file containing the water column sound speed profile.
hpdt_id. The ID number for the input Hpdt file containing the data for MFI.
mfop_id. The ID number for the output Mfop file to contain the results of the MFI.
all_freq. A flag indicating whether all the frequencies in the Hpdt file are to be used for matching.
num_freq. The number of frequencies in the Hpdt file that are to be used for matching.
freq. The frequencies in the Hpdt file that are to be used for matching.

use_freq. A flag indicating which of the frequencies in the Hpdt file (in order) are to be used for matching.
match_fun. A string indicating the scheme for combining the results at multiple frequencies.
num_search. The number of random samples of parameters to use in the search stage of the algorithm.
num_best. The number of best matches from the search stage that are to be optimized during the optimization stage of the MFI.
num_its. The maximum number of iterations for the DFPMIN (Davidon-Fletcher-Powell) optimization algorithm to perform.
reg_factor. The regularization factor to apply in computing the objective function.
num_inter. The number of intermediate points on a hyperspace line between two optima along which to evaluate the objective function in order to detect whether the optima are estimates of the same peak and should be grouped.
thresh_group. The threshold used to determine whether two optima (minima) are estimates of the same peak. If the objective functions at the two optima differ by this amount or more, or if any point on the above line connecting the optima is more than this amount greater than the optimum value at the endpoints, the optima are taken to be different and are not grouped.
num_params. The total number of parameters in the overall model here, 23).

Opt

Opt is a vector of structures characterizing how each parameter is to be involved in the optimization and giving information about the widgets for the GUI. Each structure of Opt contains the following tags:

wid_yesno. The ID of the IDL widget for the checkbox in the GUI.
wid_lower. The ID for the IDL widget for the lower bound field in the GUI.
wid_upper. The ID for the IDL widget for the upper bound field in the GUI.
wid_fixed. The ID for the IDL widget for the fixed field in the GUI.
optimize. An flags indicating whether this parameter is to be optimized.
lower. The lower bound for this parameter, if it is to be optimized.
upper. The upper bound for this parameter, if it is to be optimized.
fixed. The fixed value for this parameter, if it is not to be optimized

Search

The Search structure contains the results of the global search stage of MFI, and has the following tags:

f. A vector containing the function values for the searches.
x. A 2D array containing the parameter values used for the searches.

Optim

The Optim structure contains the results of all the optimizations done during the optimization stage of MFI, and has the following tags:

num_best. The number of best matches from the search stage that were optimized.

p_f. A vector of pointers; the *i*th pointer points to a vector containing the function values that were computed during the course of the *i*th optimization.

p_x. A vector of pointers; the *i*th pointer points to a 2D array containing the parameter values used during the course of the *i*th optimization.

f_min. A vector containing the optimized function values for the optimizations.

x_min. A 2D array containing the parameters corresponding to the optimized function values for the optimizations.

Groups

The Groups structure contains the results of the grouping analysis and has the following tags:

num_group. The number of different groups found.

num_in_group. A vector containing the number of convergences (equivalent optima) in each group.

p_f. A vector of pointers; the *i*th pointer points to a vector containing the function values for the individual optima in *i*th group.

p_x. A vector of pointers; the *i*th pointer points to a 2D array containing the parameter values for the individual optima in the *i*th group.

x_mean. A 2D array containing the means for the parameters in the groups.

x_sd. A 2D array containing the standard deviations for the parameters in the groups.

x_corr. A 3D array containing the correlation matrices for the parameters in the groups.

x_prob. A 3D array containing the significance levels of the correlations in the correlation matrices for the parameters in the groups.

Peak

The Peak structure contains the results of the 1D ambiguity function for each parameter passing through the best optimum found, and has the following tags:

num_peak. A vector containing the number of peaks (local maxima) in the function for each parameter.

num_extrema. A vector containing the number of extrema (local minima and maxima) in the function for each parameter.

dymanic_range. A vector containing the dynamic range (maximum – minimum) in the function for each parameter.

width. A vector containing the estimated width of the peak at half height for each parameter.

sens_rms. A vector containing the estimated sensitivity (RMS difference of the adjacent function points) for each parameter.

Ship

The Ship structure holds the traces for a single shot, and contains the following tags:

aray_id. The ID number for the Aray file containing the array geometry.
shot_id. The ID number for the Shot file containing the traces for the shots.
ship_id. The ID number for this Ship file.
num_sens. The number of sensors in the array.
num_time_pt. The number of samples in each time series in the structure.
samp_freq. The sampling frequency for the traces.
source_depth. The depth of the source.
source_range. The range of the source.
source_bearing. The bearing of the source.
tdata. A 2D array containing the time series for the sensors.

Shot

The Shot structure holds the traces for a single shot, and contains the following tags:

aray_id. The ID number for the Aray file containing the array geometry.
wave_id. The ID number for the Wave file containing the wavelet.
shot_id. The ID number for the Shot file containing the traces for the shots.
num_sens. The number of sensors in the array.
samp_freq. The sampling frequency for the traces.
min_freq. The minimum frequency in the band for which ORCA broadband should compute a normal mode model (must be positive).
max_freq. The maximum frequency in the band for which ORCA broadband should compute a normal mode model (must be less than samp_freq).
num_fft. The number of points in the FFT to be performed on the frequency domain data, and which corresponds to the number of points in the traces (power of 2). Care should be taken to make this large enough to prevent wrap-around artifacts in the traces.
source_depth. The depth of the source.
source_range. The range of the source.
source_bearing. The bearing of the source.
tdata. A 2D array containing the shot traces for the sensors.

Wave

The Wave structure holds an acoustic wavelet, and contains the following tags:

num. The number of elements in the wavelet.
samp_freq. The sampling frequency for the wavelet points.
wdata. A vector containing the elements of the wavelet.

Wssp

The Wssp structure holds a water column sound speed profile, and contains the following tags:

depth. A vector containing the depths for the profile.
speed. A vector containing the sound speeds for the profile at the corresponding depths.

Appendix B: Program Modules

Simulate TD Shot Data

This component is implemented by the Sim_shot_data.pro module. Its main component functions and procedures are as follows:

Sim_shot_data. Sets up the GUI for specification of the files and input conditions.
Set_sim_shot_data. Initializes default values for the fields of the GUI.
Check_sim_shot_data. Checks for existence of files and consistency of the data entered by the user.
Sim_shot_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Run_sim_shot_data. Performs a run using the input data specified by the user, i.e., calling ORCA broadband, processing the resulting FFT file to convolve the impulse response with a wavelet, generating traces for each sensor of the array and adding noise to the traces.
Read_aray. Reads in the data for the array.
Read_wave. Reads in the wavelet to convolve with the impulse responses.
Read_orca_fft. Reads in the ORCA "FFT file" – the frequency domain impulse responses for each sensor of the array.
Gen_array_geom_file. Uses the data in the Aray file to generate an array geometry file for use by ORCA.
Gen_bb_opt_file. Uses the data input by the user to generate an options file for ORCA to perform a run using the broadband option.

In addition, an executable ORCA program is required which will input the specified data and run conditions and output the FFT file.

The Sim_shot_data component uses the Aray, Wave, and Shot data structures. A description of these structures and their tags is given in the previous section.

Simulate Ship FD Hpdt Data

This component is implemented by the Sim_ship_data.pro module. Its component functions and procedures are as follows:

Sim_ship_data. Sets up the GUI for specification of the files and input conditions.
Set_sim_ship_data. Initializes default values for the fields of the GUI.
Check_sim_ship_data. Checks for existence of files and consistency of the data entered by the user.
Sim_ship_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Run_sim_ship_data. Performs a run using the input data specified by the user, i.e., calling ORCA or RAM, adding noise, performing cross-spectral matrix estimation if specified, and writing the data to an Hpdt file.
Read_aray. Reads in the data for the array.
Gen_sim_orca_sv. Sets up files for an ORCA run and then calls ORCA to simulate a signal vector at an array for a particular source, array, and frequency.
Gen_array_geom_file. Uses the data in the Aray file to generate an array geometry file for use by ORCA.
Gen_cw_opt_file. Generates an options file for the ORCA run in cw mode.
Gen_ram_sv. Sets up files for a RAM run and then calls RAM to simulate a signal vector at an array for a particular source, array, and frequency.
Gen_ramgeo_file. Generates an input data file for RAM.
Read_ram_grid. Reads in a grid of field values computed by RAM.
Gen_wn_matrix. Generates a white noise matrix.
Gen_sn_matrix. Generates a spherical noise matrix.
Gen_cn_matrix. Generates a cylindrical noise matrix.
Chol_matrix. Performs a Cholesky decomposition on the input matrix.
Gen_nv. Generates an estimated noise vector, based on the Cholesky decomposition.

In addition, executable ORCA and RAM programs are required which will input the specified data and run conditions and output the complex fields to be used as signal vectors.

The Sim_ship_data component uses the Aray and Hpdt data structures. A description of these structures and their tags is given in a later section.

Convert TD Shot Data to TD Ship Data

This component is implemented by the Shot_to_ship_data.pro module. Its component functions and procedures are as follows:

Shot_to_ship_data. Sets up the GUI for specification of the files and input conditions.
Set_shot_to_ship_data. Initializes default values for the fields of the GUI.
Check_shot_to_ship_data. Checks for existence of files and consistency of the data entered by the user.
Shot_to_ship_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Run_shot_to_ship_data. Performs a run using the input data specified by the user, i.e., reading in the shot data, generating ship data and writing to a Ship file.

Convert P16 Data to Shot or Ship Data

This component is implemented by the P16_to_ss_data.pro module. Its component functions and procedures are as follows:

P16_to_ss_data. Sets up the GUI for specification of the files and input conditions.
Set_ p16_to_ss_data. Initializes default values for the fields of the GUI.
Check_ p16_to_ss_data. Checks for existence of files and consistency of the data entered by the user.
P16_to_ss_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Run_ p16_to_ss_data. Performs a run using the input data specified by the user, i.e., reading in the P16 data, converting to ship data and writing to a Shot or Ship file.

Preprocess TD Shot Data to FD Hpdt Data

This component is implemented by the Prep_shot_data.pro module. Its component functions and procedures are as follows:

Prep_shot_data. Sets up the GUI for specification of the files and input conditions.
Set_prep_shot_data. Initializes default values for the fields of the GUI.
Check_prep_shot_data. Checks for existence of files and consistency of the data entered by the user.
Prep_shot_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.

Run_prep_shot_data. Performs a run using the input data specified by the user, i.e., reading in and FFTing the shot data, and writing the cross-spectral matrices to an Hpdt file.

Preprocess TD Ship Data to FD Hpdt Data

This component is implemented by the Prep_ship_data.pro module. Its component functions and procedures are as follows:

Prep_ship_data. Sets up the GUI for specification of the files and input conditions.
Set_prep_ship_data. Initializes default values for the fields of the GUI.
Check_prep_ship_data. Checks for existence of files and consistency of the data entered by the user.
Prep_ship_data_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Run_prep_ship_data. Performs a run using the input data specified by the user, i.e., reading in, overlapping, windowing and FFTing the ship data, performing cross-spectral matrix estimation, and writing the cross-spectral matrices to an Hpdt file.

Generate nD Ambiguity Function

These components are implemented by the modules Mfi_1d_amb_fun.pro, Mfi_2d_amb_fun.pro, and Mfi_3d_amb_fun.pro. The component functions and procedures for the 2D case (the others are analogous) are as follows:

Amb_2d. Sets up the GUI for specification of the files and input conditions.
Set_amb_2d. Initializes default values for the fields of the GUI.
Check_amb_2d_parms. Checks for existence of files and consistency of the data entered by the user.
Amb_2d_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Run_amb_2d. Performs a run using the input data specified by the user, i.e., sets up the values of the parameters and computes the matches at these values.
Read_aray. Reads in the data for the array.
Read_wssp. Reads in the sound speed profile to use for the run.
Read_servers. Reads in the default values for the IP addresses and ports for the remote orcaservers.
Ip_addr_array_to_string. Converts a string vector of IP addresses to their numeric 2D array equivalents.
Ip_addr_string_to_array. Converts a numeric 2D array of IP addresses to their string vector equivalents.

Gen_mfi_orca_sv. Sets up and performs an ORCA run to generate a signal (replica) vector based on the run conditions.
Gen_array_geom_file. Uses the data in the Aray file to generate an array geometry file for use by ORCA.
Gen_cw_opt_file. Uses the data input by the user to generate an options file for ORCA to perform a run using the continuous wave option.
Gen_svp_file. Generates an svp file for ORCA, based on the parameter values.
Bartlett. Computes the output of the Bartlett power processor, for either vector or matrix data.

Perform MFI Search/Optimization

This component is implemented by the Mfi_search_opt.pro module. Its component functions and procedures are as follows:

Mfiop. Sets up the GUI for specification of the files and input conditions.
Set_mfiop. Initializes default values for the fields of the GUI.
Check_mfiop_parms. Checks for existence of files and consistency of the data entered by the user.
Mfiop_event. Handles the events generated by the GUI and sets the values in the data structures based on the input by the user.
Run_mfiop. Performs an MFI run using the input data specified by the user, i.e., performs the initial search, optimizes the objective function for multiple starting estimates and performs the grouping and peak analysis.
Read_aray. Reads in the data for the array.
Read_wssp. Reads in the sound speed profile to use for the run.
Read_servers. Reads in the default values for the IP addresses and ports for the remote orcaservers.
Ip_addr_array_to_string. Converts a string vector of IP addresses to their numeric 2D array equivalents.
Ip_addr_string_to_array. Converts a numeric 2D array of IP addresses to their string vector equivalents.
Gen_mfop. Returns an Mfop structure based on the input parameters, which are the component structures of the Mfop structure.
Mfi_func. Evaluates the objective function to be minimized in MFI. This is a composite function consisting of the sum of the Bartlett processor, a penalty function if any parameter exceeds its bounds, and a regularization term.
Mfi_dfunc. Estimates the gradient of the objective function using central differences.
Plot_opt. Produces a plot of the evolving function value and the values of the parameters.
Gen_mfi_orca_sv. Sets up and performs an ORCA run to generate a signal (replica) vector based on the run conditions.

Gen_array_geom_file. Uses the data in the Aray file to generate an array geometry file for use by ORCA.
Gen_cw_opt_file. Uses the data input by the user to generate an options file for ORCA to perform a run using the continuous wave option.
Gen_svp_file. Generates an svp file for ORCA, based on the parameter values.
Gen_group. Groups multiple optima corresponding to the same peak.
Analyze_group. Performs statistical analysis in the multiple parameter estimates for each group.
Analyze_peak. Performs post-processing on the best optimum to estimate peak width, sensitivity, etc., for each parameter.
Bartlett. Computes the output of the Bartlett power processor, for either vector or matrix data.

Support of Gulf of Mexico Hydrate Research Consortium: Activities to Support Establishment of Seafloor Monitoring Station

TECHNICAL PROGRESS REPORT

Reporting Period Start Date: June 1, 2005

Reporting Period End Date: September 30, 2005

Principal Investigator (Author): Bob A. Hardage

Date Issued: May 15, 2006

DOE Cooperative Agreement No. DE-FC26-02NT41628

Seismo-acoustic characterization of seafloor properties
and processes at the hydrate-monitoring station

Submitting Organization:
Bureau of Economic Geology
John A. and Katherine G. Jackson School of Geosciences
The University of Texas at Austin
University Station, Box X
Austin, TX 78713-8924

Abstract

We have developed software that implements a new theory to create higher-resolution P-P images of near-seafloor geology from 4C OBC seismic data. This report summarizes this software development and illustrates the quality of the P-P images that can be generated.

List of Figures

Introduction

Work subcontracted to the Bureau of Economic Geology (Bureau) has been expanded to allow the Exploration Geophysics Laboratory (EGL) at the Bureau to process and interpret all horizontal-array and vertical-array seismic data acquired at the seafloor observatory constructed across Block MC118. Work during this quarter has focused on developing software that will generate optimal-resolution P-P images of data acquired with multicomponent seismic sensors positioned on the seafloor.

Executive Summary

Researchers at the EGL have been subcontracted to create deliverables for Work Task 6, *Seismo-acoustic characterization of seafloor properties and processes at the hydrate-monitoring station*. Investigations have shown that standard P-P imaging of data acquired using 4C seafloor sensors does not produce the resolution of near-seafloor geology that is desired for this project. EGL scientists have developed a new concept for processing deep-water 4C OBC data that yields a significant improvement in the spatial resolution of P-P images. The fundamental theory is that, in deep water, the large elevation difference between a sea-surface source and a seafloor sensor allows P-P data to be processed in much the same way as standard VSP data.

Experimental

Experimental activity during this period focused on developing and testing software that creates high-resolution P-P images of near-seafloor geology from deep-water 4C OBC seismic data.

Results and Discussion

In deep-water multicomponent seismic data acquisition, there is a large elevation difference between source stations (an air gun at the sea surface) and receiver stations on the seafloor. Conventional processing of deep-water 4-C seismic data involves a wave-equation datuming step that transforms the data to a domain in which sources and receivers are on the same depth plane. This step effectively removes the water layer and allows the data to be processed as if the source were on the seafloor. This adjustment of source-receiver geometry allows deep-water multicomponent data to be processed using software already developed for shallow-water environments, where marine multicomponent data-acquisition technology was originally developed and applied. An example of a good-quality, deep-water P-P image of near-seafloor geology made using this wave-equation datuming approach is shown as Figure 1a. This image shows local geology associated with a fluid-gas expulsion chimney that extends to the seafloor.

If a person wishes to study near-seafloor strata in greater detail, a new approach to P-P imaging of deep-water multicomponent seismic data is not to eliminate the large elevation difference between sources and receivers, as is done in wave-equation datuming, but to take advantage of that elevation difference. The objective is to process deep-water multicomponent data in a way similar to how vertical seismic profile (VSP) data are processed, because VSP data acquisition also involves large elevation differences between sources and receivers (Fig. 2). Users of VSP technology know that VSP data provide high-resolution images of geology near downhole receiver stations. That same logic leads to the conclusion that deep-water multicomponent seismic data processed using VSP-style techniques should yield higher resolution images of geology near deep-seafloor receivers.

The P-P processing illustrated here can be done using either 2-C or 4-C seafloor sensors. The fundamental requirement is to acquire data with a sensor having a hydrophone and a vertical geophone. The seafloor hydrophone response (**P**) and the seafloor vertical-geophone response (**Z**) are combined to create downgoing (**D**) and upgoing (**U**) P-P wavefields as

$\mathbf{D = P + Z/cos(\Phi)}$ and

$\mathbf{U = P - Z/cos(\Phi)}$,

in which Φ defines the incident angle at which the downgoing compressional wave arrives at the seafloor. Once this wavefield separation is done, deep-water multicomponent seismic data are defined in terms of downgoing and upgoing wavefields just as VSP data are. Having access to downgoing (**D**) and upgoing (**U**) wavefields means that subseafloor reflectivity can be determined by taking the ratio **U/D**. This reflectivity wavefield is then segregated into stacking corridors, and data inside these corridors are summed to create image traces, just as VSP data have been processed for the past 20+ years. Figure 1b shows a P-P image made with this technique, using the same deep-water data displayed in Figure 1a. The improvement in resolution is obvious.

Applying this VSP-style imaging technique to deep-water multicomponent seismic data is proving to be invaluable for gas hydrate studies, geomechanical evaluations of deep-water seafloors, and other applications where it is critical to image near-seafloor geology with optimal resolution. However, every seismic data-processing technique has constraints and pitfalls. Two principal constraints of the technology described here are:

1. There has to be a significant difference between elevations of sources and receivers. The technique is not appropriate for multicomponent seismic data acquired in shallow water.

2. The improvement in image resolution over that of production processing of marine multicomponent seismic data diminishes as the image space extends farther (deeper) from the receivers. At significant subseafloor depths, production-style, wave-equation-datuming-based, P-P imaging (Fig. 1a) is equivalent or superior to the VSP-style imaging described here.

Conclusions

Important software needed for ongoing research at the hydrate-monitoring station has been developed and tested using data similar to those expected to be acquired across Block MC 118. Test results indicate that the software that has been developed is robust and creates higher-resolution P-P images of near-seafloor geology than what can be provided by commercial contractors.

References

None.

Abbreviations and Acronyms

4-C: four-component
EGL: Exploration Geophysics Laboratory
MC: Mississippi Canyon
OBC: ocean-bottom cable
P-P: standard P-wave seismic data
P-SV: converted-shear mode (P-wave to SV-shear wave conversion)
VSP: vertical seismic profile

(a)

(b)

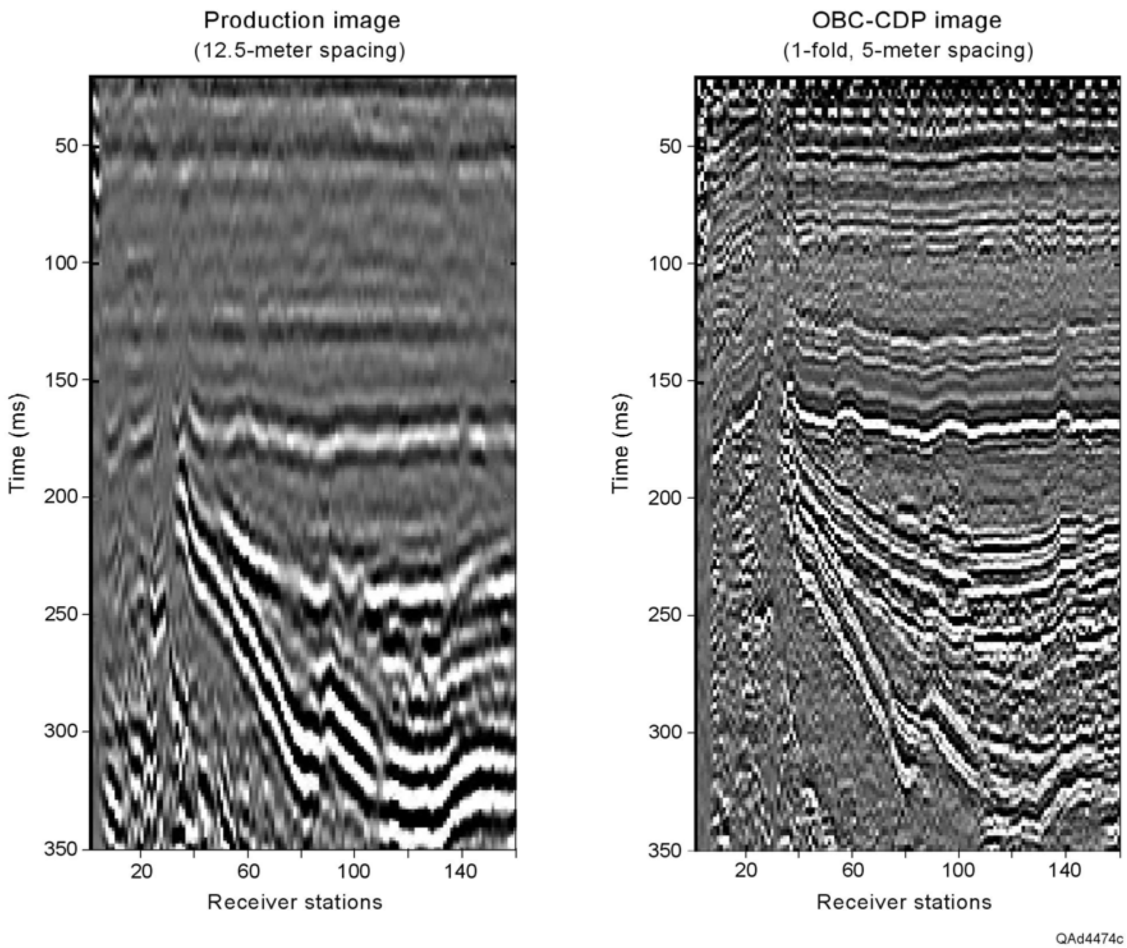


Figure 1. (a) Standard production processing of deep-water 4-C OBC seismic data along a profile that traverses a seafloor gas-expulsion chimney. (b) Improved resolution of near-seafloor geology using VSP-style concepts for processing deep-water OBC data. Both images are flattened to the seafloor.

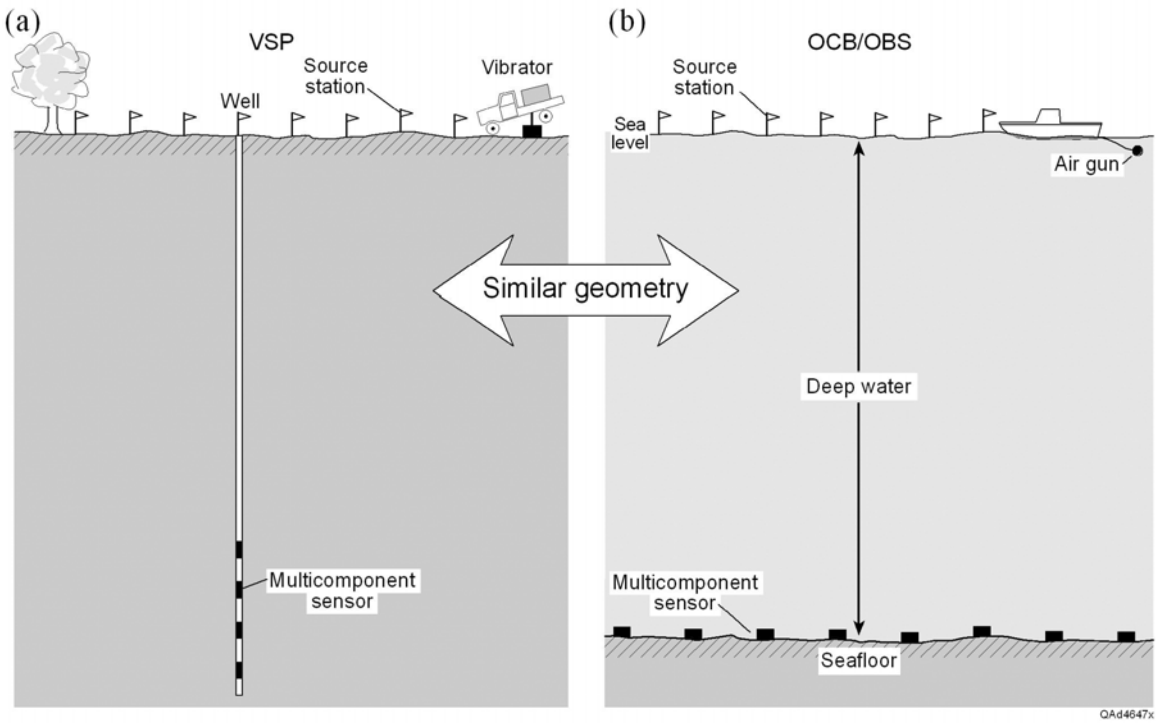


Figure 2. Illustration of similar source-receiver geometries used for acquiring (a) VSP data and (b) deep-water OBC/OBS seismic data.

Coupling of continuous geochemical and sea floor acoustic measurements
6 month progress report
Technical report: June 1-Sept 30, 2005

Gulf of Mexico Gas Hydrate Research Consortium

From Florida State University

Principal Investigator:
Jeffrey Chanton
Department of Oceanography
Florida State University
Tallahassee, Florida, 32306-4320
Tel:850-644-7493
Fax: 850-644-2581
jchanton@mailer.fsu.edu,

Associated Collaborators:
Laura Lapham, Department of Marine Sciences, UNC-Chapel Hill 27599
And
Paul Higley, Specialty Devices Inc. 1104 Summit Ave #104, Plano Texas 75074

ABSTRACT
Nutrients, methane and sulfate concentrations and methane stable carbon isotopes were measured in 10 cores collected at MC 118 to help understand the spatial variability in geochemical indicators at a hydrate seep site and the role biogeochemical processes play in hydrate stability. The questions addressed are: What is the source of methane at MC 118? Are sulfate reduction and methane oxidation important processes at MC118? Do rates of methane oxidation affect hydrate stability? This is the first study to address spatial variability of biogeochemical processes affecting hydrate stability at MC 118 and lays the foundation for the hydrate seafloor observatory where hydrate stability is assessed through long term monitoring of both seismic events and biogeochemistry.

METHODS
Shipboard
In May 2005, ten gravity cores (<4.5m in length) were collected off the Research Vessel (R/V) Pelican (operated by Louisiana Universities Marine Consortium) at MC 118. Cores sites were chosen based on geophysical chirp lines collected in May, 2005, by C&C Company. Ideal sites were areas containing wipe-out zones, as chosen by geophysicists participating on the cruise. Core barrels were sliced lengthwise and

sampled at ~50 cm intervals. Sediment plugs (6mL) were collected at each specified interval, capped into a glass serum vial, and frozen for future analysis of methane concentrations, stable carbon isotopes and porosity. Remaining sediment from each interval was then collected into 50mL centrifuge tubes and frozen for later analysis of pore-water sulfate, chloride, and nutrient (nitrate, phosphate, and ammonium) concentrations.

Analytical

Methane concentrations for core pore waters were measured on a Shimadzu Mini II Gas Chromatograph (GC). Ten-mL of methane-free deionized water was added to each sample vial to displace 10-mL needed for the GC injection. Samples were analyzed by head space extraction. Integrated areas were compared to standards (101.6ppm CH_4). These samples, along with the DIC (dissolved inorganic carbon) samples, were then analyzed for carbon isotopic signatures using a gas chromatograph-isotope ratio mass spectrometer (GC-IRMS) with a Hewlett-Packard 5890 GC equipped with a 2M Poropaq Q column set at 30 C and a Finnigan Mat 252 IRMS. Low concentration gases were analyzed by pre-concentrating gas aliquots of ~10-30mL in-line with a liquid nitrogen/ethanol slush at approximately -130 C and a Poro-plot Q column and then introduced into the GC-IRMS system. Microliter volumes of high concentration gases were directly injected onto the column to be introduced to the mass spectrometer. Standards were run.

Sulfate and chloride was measured by diluting 100uL of sample to 10mL with eluent and injecting 1mL into a Dionex Ion Chromatograph. Sample concentrations were determined after running standard curves. Sediment patties were freeze-dried and ground for determination of total organic carbon concentrations and carbon isotopic composition. Approximately 30mg of sample were introduced by flash combustion into a Carlo Erba Elemental Analyzer and resultant carbon dioxide swept into the same IRMS system as described for dissolved gases. Nutrients (ammonium, nitrate, and phosphate) were measured on a Lachet QuickChem 8000 equipped with an auto analyzer using colormetric techniques.

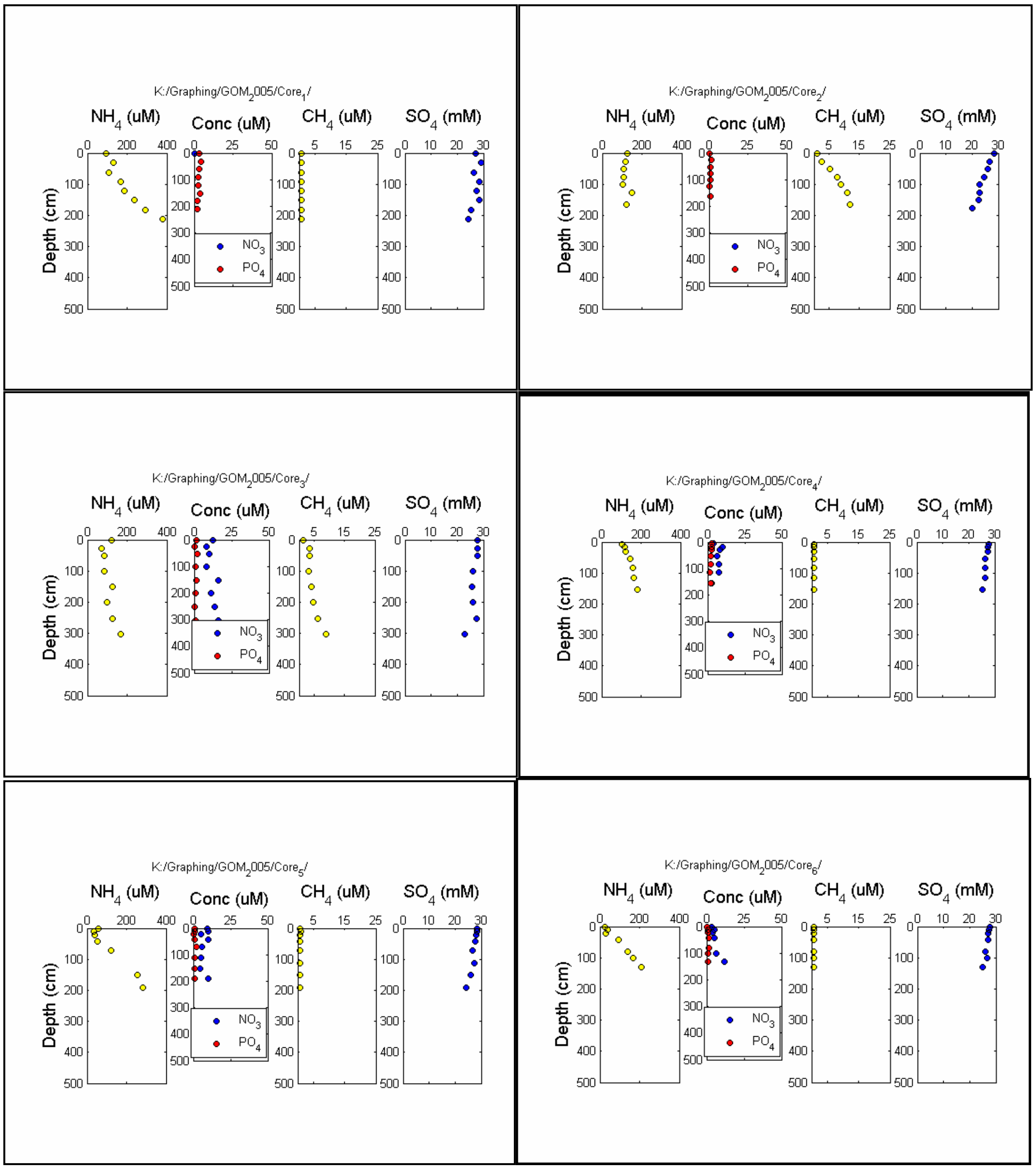
RESULTS

Non-seep cores: Cores 1-8 and 10 did not show any evidence of a seep site (Figure 1). Methane concentrations were low throughout the cores. Sulfate concentrations were consistent with overlying seawater values (~28mM), with only a slight decrease deeper in the cores to ~20mM. Ammonium concentrations increased from ~50uM at the seawater interface to ~400uM in half of the cores (1, 5, 6, 9, and 10). Phosphate and nitrate do not show any particular trends. Chloride concentrations for cores 1-6 were consistently seawater values down core.

Hot spot core: Core 9 also showed evidence of having hit a hot spot (Figure 2). Sulfate concentrations were 28mM at the sediment water interface and decreased to near zero around 50cmbsf. Methane concentrations were very low at the sediment water interface, began to increase at the depth of no sulfate and remained around 3.5mM at 75cmbsf for the remainder of the core, to 450cmbsf. However, these concentrations are still considered minimums due to gas expansion during core retrieval. Carbon isotopic composition of the methane showed a subsurface minimum of -70‰ at 50cmbsf. Below this depth, the isotopic signature becomes more enriched in 13C to a value of -49.73 ± 1.11‰ (n=10), indicating a thermogenic source gas from deep below. Above the minimum value, the isotopic composition becomes enriched in 13C to -60.66 ± 0.20‰. Ammonium concentrations were low at the sediment water interface and remained low until around 300cmbsf where concentrations increased to ~400uM at the bottom of the core. Nitrate and phosphate were low and did not show any trends.

Additional data analysis can be found in the next report, for the period Oct 1, 2005-March 31, 2006.

Figure 1: Concentration profiles from cores 1-8, 10



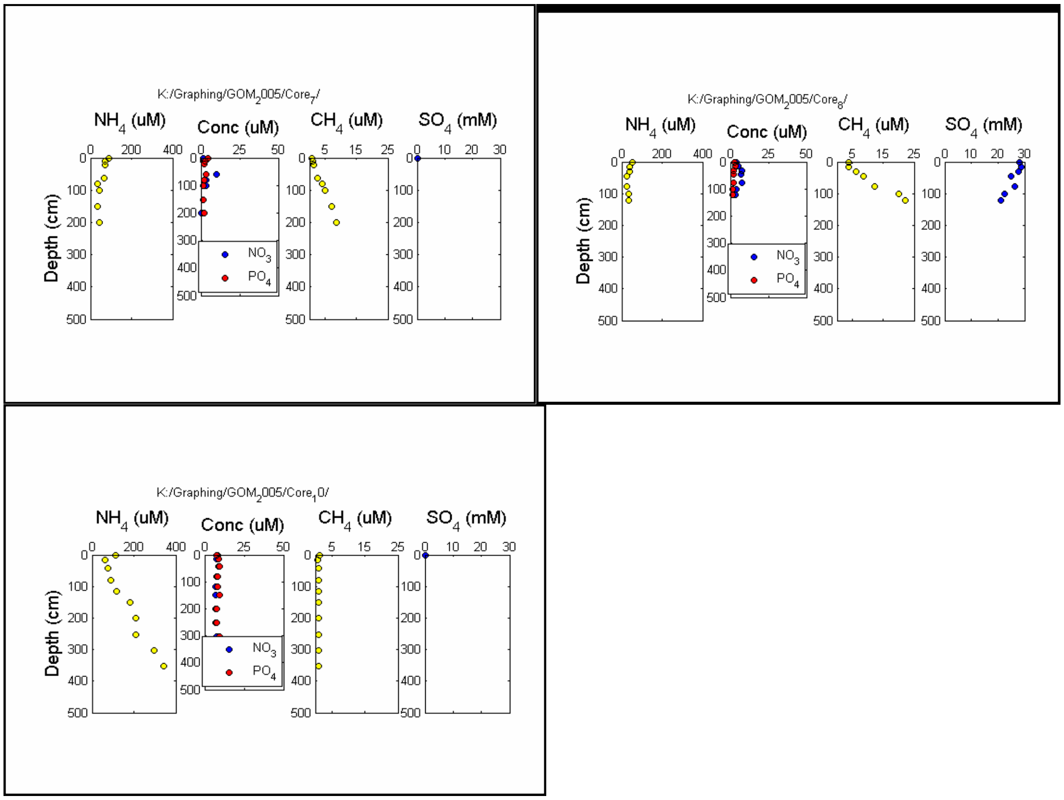
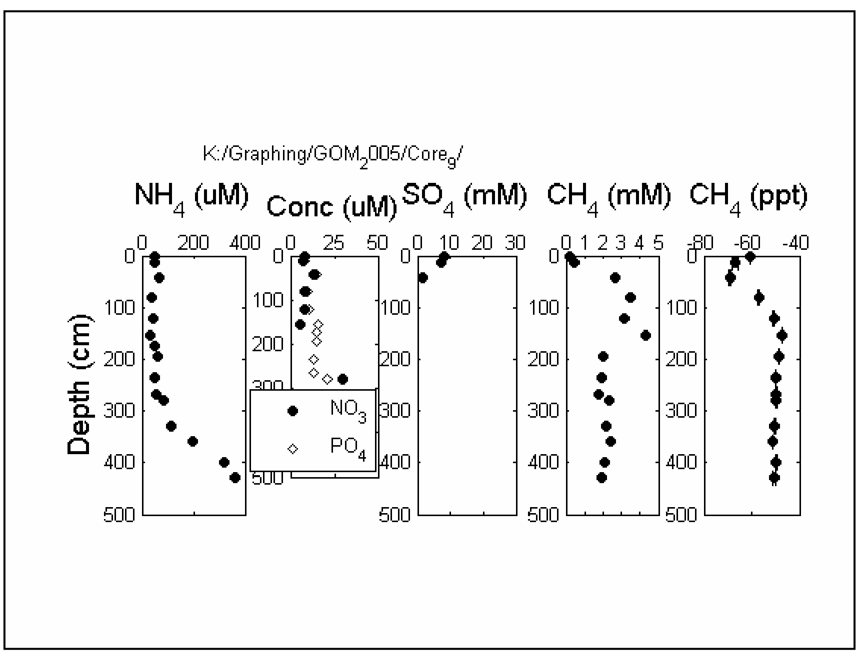


Figure 2: Core 9 concentration and isotope profiles.



Laboratory Analysis and Comparison of Cores from MC 118 and MC 798

R.E. Rogers[1], J.L. Dearman[1]

[1]Swalm School of Chemical Engineering, P.O. Box 9595, Mississippi State, MS 39762

May 10, 2006

ABSTRACT

Sediments in zones of the Gulf of Mexico where hydrates occur may exhibit hydrate-forming properties that differ from non-hydrate-bearing sediments. If so, an analysis could not only be valuable in establishing gas-hydrate formation mechanisms in ocean sediments, but such data could provide guides in locating gas-hydrate deposits worthy of production. To evaluate this possibility, sediments from two locations in the Mississippi Canyon, MC-118 and MC 798, were analyzed for propensity to form gas hydrates. Hydrate formation rates and crystal initiation times were measured in the laboratory as a function of depth below seafloor and as a function of lateral displacement. Trends in these parameters resulted.

INTRODUCTION

The preceding quarterly report, 1st quarter report 2005, detailed the hydrate-forming propensity of sediments from the Dufresne cruise in the Mississippi Canyon. (MD02-2570 cores were taken in 631 m water depths at the West Mississippi site of the Mississippi Canyon.) Those sediments were from a core that reached almost 30 m below seafloor, a depth not often available for gas-hydrate studies. Hydrate formations were determined from replicated laboratory tests to determine any trend with depth. Strong correlations with depth resulted: Formation rates peaked at about 20 meters depth and induction times decreased greatly to a minimum value at about 15 meters depth and remained at that low value through 30 meters. The top 5 meters of the Dufresne sediments exhibited a somewhat erratic hydrate-formation pattern within the 5 meters.

The results from the Dufresne sediments' analyses raise intriguing questions regarding parameters affecting hydrate formation rates and hydrate initiation times in ocean sediments. The current report begins to address some of those parameters in other Mississippi Canyon sediment studies.

In the absence of cores as deep as the 30 m from the Dufresne cruise, investigations were made of cores in the upper 6 m of various locations in the Mississippi Canyon. Specifically, tests on sediments from MC 118 and MC 798 are reported this quarterly in an initial attempt to determine hydrate-formation variability near-surface as a function of lateral displacement.

It is helpful to remember approximate depths of the hydrate zone as compared to the depth of cores being analyzed. Typically, the hydrate zone depth would be about 200 mbsf in the Mississippi Canyon, whereas the Dufresne cores extended to 30 mbsf and available push cores reach to about 6 mbsf. In fact, most of the limited hydrate-formation data reported in the literature for GOM are restricted to the first few meters below seafloor.

EXPERIMENTAL PROCEDURE

A few important features of the laboratory procedure for analyzing cores are listed below.

1. Twenty grams of mud samples are evenly dispersed through 60 grams of coarse Ottawa sand that has been cleaned. The reasons for doing this are the following: (a) to give maximum access of natural gas to the sediment samples, thus exposing maximum surface areas of the sediment particles to the reacting gas, (b) to provide larger porosities in which hydrates are allowed to form and expand, (c) to be able to utilize the small mud sample sizes.
2. The samples are tested for hydrate formation in Teflon containers. Teflon provides a hydrophobic surface that does not affect hydrate formation.
3. For testing, the samples are placed in annular spaces of concentric Teflon cylinders. Both cylinders have many gas-access holes spaced over their entire areas. This allows maximum contact of gas with sediments and prevents mass transfer from limiting hydrate kinetics.
4. Only original seawater removed with the cores is present in the tests.
5. Teflon containers with samples are placed in stainless steel Parr reactors, sealed, pressurized with natural gas and submerged in constant temperature baths maintained at $0.5^{o}C$.
6. Pressures and temperatures are recorded continuously during the tests.

RESULTS AND DISCUSSION

As stated in the preceding quarterly report, formation rates and induction times determined in laboratory measurements of Gulf of Mexico (GOM) sediments (Dufresne MD02-2570 cores) show distinct trends to 30-m depths. In the current report, shallower cores from different locations in the Mississippi Canyon are compared.

Generally, when induction times decrease, hydrates begin forming more quickly. This can be important when gases percolate through sediments and have a limited residence time. After the hydrates are initiated, higher formation rates are indicative, of course, of favorable hydrates occurring in the sediments.

<u>Core 11, MC-118</u>

Presented in Fig. 1 are the formation rates and induction times of laboratory hydrates developing in the sediments and indigenous waters of MC118, Core 11. The patterns of formation rate and induction time in Fig. 1 are somewhat atypical patterns in that induction times increase at about 200 cm depth.

In interpreting the data, the atypical behavior could mean that the bottom of the sulfate zone has not been reached at 297.5 cm—the deepest sample from Core 11. If so, this would mean a relatively low methane flux through the sediments. It could also suggest turbulence in the top sediments.

In the laboratory, anionic bioagents increase the formation rates of hydrates and decrease the induction times of hydrates, especially in the presence of smectite clays. Since different microbial communities exist above and below the sulfate zone, each community would be associated with a unique bioproduct. These bioproducts may have significantly different effects on gas-hydrate formation.

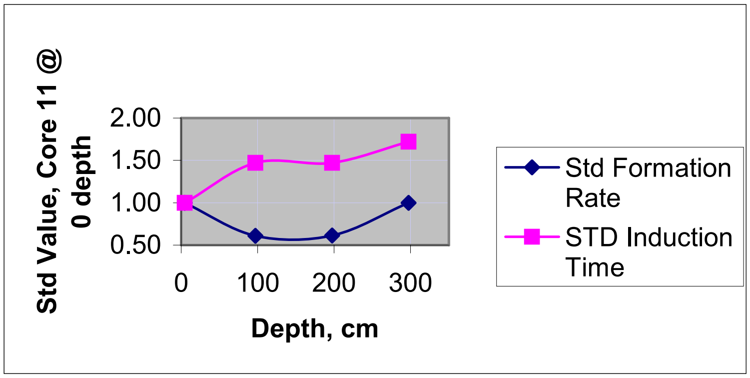


Fig. 1. Hydrate formation in Core 11 from MC-118

Core 04, MC118

In Fig. 2, analysis of sediment from MC798, Core 04, gives a more typical pattern.

The bottom of the sulfate zone could possibly be within a few centimeters of the surface; this would explain patterns of the formation rates and induction times. If so, a relatively high methane flux might pass through the sediments at the core location.

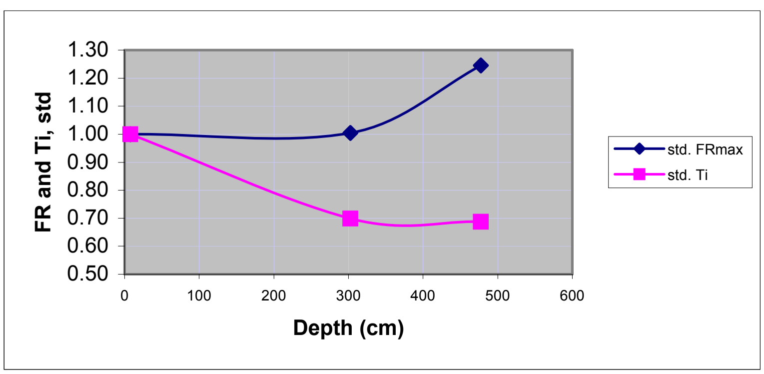


Fig. 2. Hydrate formation in Core 04 from MC-798

Hypothesized Mechanism Affecting Near-Surface Hydrate Formation

Sulfate from the overhead seawater permeates the near-surface sediments. Anaerobic oxidation of methane occurs. Archaea work in clusters with sulfate-reducing bacteria. The bacteria reduce sulfate to H_2S, and archaea concurrently oxidize methane to form CO_2. The CO_2 precipitates as carbonates at the point of anaerobic methane oxidation. As the upward methane flux increases through the sediments, the sulfate zone ascends to near the sediment-sea interface. For example, the bottom of the sulfate zone has been reported to be only a few centimeters deep near gas hydrate outcrops and methane gas vents. Carbonate nodules solidify and become an indicator of the current bottom of the sulfate zone or a previous sulfate boundary.

Bioproducts from the microbes in the sulfate zone differ from those below the sulfate zone, and the catalytic effects of those bioproducts on hydrate formation differ greatly. (The catalytic effect of bioproducts probably depends on whether there are distinct hydrophobic and hydrophilic components in the same molecular structure. The hydrophobic moieties collect the methane and the nearby hydrophilic moieties collect and structure water, thus setting up the nuclei for hydrate initiation. Anionic bioagents and anionic synthetic agents have proven to be hydrate catalysts.) The sediment particles in the sulfate zone may be covered with polymeric bioproducts that do not have distinguishing hydrophilic and hydrophobic components, contrasting to those below the sulfate zone. A nonionic polymeric coating of near-surface mineral particles could slow or prevent local hydrate formation.

If this hypothesis is used to interpret the hydrate formation and induction curves generated in the laboratory from MC-118 sediments, then from generated curves one might determine the location of the sulfate zone and have a good indicator of the magnitude of the methane flux at that location.

CONCLUSIONS

Variations in ease of gas-hydrate formation occur as one moves laterally at near-surface depths of the seafloor in Mississippi Canyon. These variations do not follow a distinct pattern as witnessed in the 30-m deep Dufresne core sediments. However, even the Dufresne sediments gave near-surface variations in hydrate formation. The near-surface variations are possibly caused by multiple parameters. It is hypothesized that extent and depth of the sulfate zone could be a primary cause of the variations. Extenuating parameters will be investigated in upcoming work.

CONCLUSIONS

This report covers the accomplishments of the sixth six-month period funding of Cooperative agreement Project #DE-FC26-02NT41628, between the Department of Energy and the Center for Marine Resources and Environmental Technology, University of Mississippi. The efforts of the Hydrates Research Consortium are reviewed and plans for the final phases of the project presented. Conclusions of various projects are happening and every effort is being made to coordinate site surveys and sensor emplacements in a sequence that allows all participants maximum access to and benefit from the cruises being scheduled for summer and fall, 2006. While plans remain incomplete, several cruises have been budgeted for and will occur if vessels are available.

Project summaries of the subcontractors' efforts appear in their reports contained within this document. All FY03 subcontractors have completed their technical reporting although financial reports are not yet complete. The CMRET is working with the sponsored programs officials at several institutions to resolve these delays. The VLA and the SFP are complete and have been proven. The "bubble counter" is complete but awaits testing in deep water, something that the Consortium is arranging. The "SphereIR" is essentially complete though it has never been field tested. Both it and the acoustic device have depleted funds prior to completion of their sensor packages. Laboratory studies of gas hydrates have expanded to new areas and new depths revealing more of the complex "habitat" hosting gas hydrates.

Software development and innovative processing techniques are on schedule as are the pore-fluid experiments. The initial components of the station, a pore-fluid sampling probe and a thermistor geophysical probe, were emplaced on the sea floor in May of 2005. The data-logger and sample-collecting box of these components contain the first date produced at the MS/SFO. Their retrieval is a priority for the upcoming field season. Additional components will be added during subsequent visits to the station site with completion of the station anticipated in 2007.

An Appendix is included for informational/historical purposes. Revision to it since the previous progress report has been minimal but it provides a useful reference when reviewing current projects.

REFERENCES

Relevant references appear following contributions by the individual subcontractors.

ACRONYMS

1D	one-dimensional
2D	two-dimensional
3D	three-dimensional
4-C	four-component
ALA (=VLA)	acoustic line array
Ambg	ambiguity function (file)
Aray	structure that holds the data for the array
ASCII	American Standard Code for Information Interchange
AUV	autonomous underwater vehicle
BA	bottom accelerometer
BBLA	benthic boundary layer array
BCOMFI	Barrodale Computing Matched Field Inversion
BCS	Barrodale Computing Services, Ltd.
BEG	Bureau of Economic Geology (University of Texas)
BHA (=BLA)	borehole array
BLA (=BHA)	borehole line array
BS	Battery System
CDP	continuous Doppler profile
CH4 (=CH_4)	methane
cmbsf	centimeters below sea floor
CMRET	Center for Marine Resources and Environmental Technology
CO2 (=CO_2)	carbon dioxide
CSA	chimney sampler array
CTD	conductivity, temperature, depth (sensors)
DATS	Data Acquisition and Telemetry System
dB	signal to noise ratio
DIC	Dissolved Inorganic Carbon
DOC	Department of Commerce
DOE	Department of Energy
DOI	Department of the Interior
DOS	Disk Operating System
DRS	Data Recovery System
DS	Docking Station
EGL	Exploration Geophysics Laboratory
FD	frequency domain
FFT	fast fourier transform
FORTRAN	formula translating system
FY	Fiscal Year
GC	gas chromatograph
GLA	geophysical line array
GOM	Gulf of Mexico

GOM-HRC	Gulf of Mexico-Hydrates Research Consortium
GUI	graphical user interface
HLA	horizontal line array
Hpdt	structure that holds simulated and real FD data input for MFI
HRC	Hydrates Research Consortium
H_2S	hydrogen sulfide
HSZ	Hydrate Stability Zone
ID	identification
IDL	Interactive Data Language
IDLVM	Interactive Data Language Virtual Machine
IDP	Integrated Data Power Unit
IP	Internet Protocol
IRMS	isotope ratio mass spectrometer
JIP	Joint Industries Program
LAN	local area network
mbsf	meters below sea floor
MC	Mississippi Canyon
MD	Marion Dufresne
ME	microbial experiments
MFI	matched-field inversion
Mfop	matched field inversion output (file)
MFP	matched field processing
MMRI	Mississippi Mineral Resources Institute
MMS	Minerals Management Service
MPC	multi-purpose cable
MS/SFO	monitoring station/sea-floor observatory
M/V	Merchant Vessel
NETL	National Energy Technology Laboratory
NIUST	National Institute for Undersea Science and Technology
NOAA	National Oceanographic and Atmospheric Administration
NURP	National Undersea Research Program
OBC	ocean-bottom cable
OBS	ocean-bottom seismometer
OLA (=OVA)	Oceanographic Line Array
ORCA	a range-independent acoustic model (Microsoft) code
OVA (=OLA)	Oceanographic Vertical Line Array
PCB	pressure-compensated battery
PCA (=PFA)	pore-fluid array
PFA (=PCA)	pore-fluid array
PFP	pore-fluid probe
PE	parabolic equation
P-P	standard P-wave seismic data
P-SV	converted-shear mode (P-wave to SV-shear wave conversion)

P-wave	compressional wave
RAM	Range-dependent parabolic equation program Acoustic Model
RDI	RD Instruments
ROV	remotely operated vehicle
RSI	Research Systems, Inc.
R/V	Research Vessel
SDI	Specialty Devices, Inc.
SFO	Sea Floor Observatory
SFP	Sea Floor Probe
Shot	structure that holds the traces for a single shot
SNR	Signal to noise ratio
SSD	Station Service Device
svp	sound velocity profile
S-wave	shear wave
TD	time domain
TP	thermistor probe
UNC	University of North Carolina at Chapel Hill
US	United States
USBL	ultra-short base-line (locating system)
USGS	United States Geological Survey
VLA	vertical line array
VSP	vertical seismic profile
Wave	structure that holds an acoustic wavelet
Wssp	water sound speed profile (file)

APPENDIX

GULF OF MEXICO HYDRATE RESEARCH CONSORTIUM: ESTABLISHMENT OF A SEA FLOOR MONITORING STATION, AN UPDATE

Introduction

Since the Gulf of Mexico Gas Hydrates Research Consortium (GOM-HRC) was organized in 1999, considerable progress has been made toward establishing a monitoring station or sea-floor observatory (MS/SFO) to monitor and investigate the hydrocarbon system within the hydrate stability zone of the northern Gulf of Mexico. The intention has been to equip the MS/SFO with a variety of sensors designed to determine a steady-state description of physical, chemical, thermal and, most recently, microbiological conditions in its local environment as well as to detect temporal changes of those conditions.

In the original design, the heart of the MS/SFO was a network of five vertical line arrays (VLAs), each consisting of 16 channels of hydrophones spaced over the lower 200m of the water column. Each VLA would be suspended from glass floats and would have been anchored to the sea floor. Since water currents would cause the VLAs to deviate from vertical, each would also include inclinometers and compasses for determining the location of each hydrophone within the water column.

The intention was to use standard surveying techniques to determine the configuration of sub-bottom strata and to monitor that configuration by applying Matched Field Processing (MFP) to the acoustic energy received by the VLAs. The source of the energy could be either the intentional firing of conventional seismic devices or the opportunistic noise of passing ships.

In either case, MFP would require knowledge of the source location. In the former, the location would be measured directly. In the latter, it would be estimated relative to the known location of the VLAs by triangulation. The net of five VLAs would provide 20 independent estimations that would be analyzed statistically to minimize error in the final determination.

Significant disagreement between the MFP results and the sub-bottom configuration determined previously would indicate that a change had occurred within the sea floor. A new survey could then be carried out to determine the structural nature of the change and the output of other sensors examined to determine chemical and thermal changes.

This original strategy came under question during 2003, however, due to a number of external factors that surfaced. Discussions arose among some Consortium members as to whether or not the design of the MS/SFO could be modified to accommodate, and perhaps even to capitalize on, those factors. There was agreement to explore a number of modifications but not to alter the original intention or basic mission of the MS/SFO. This update documents that exploration and other developments.

Modifications

Modifications to the design of the monitoring station/sea floor observatory are described below and are illustrated in Figure A1.

CHANGE 1: ARRAY TYPE

One external factor affecting the establishment of the station is the development of an ocean acoustics technique by which the sound of waves at the sea surface can be used to image the sea floor. The method requires that at least two horizontal line arrays (HLAs) be deployed on the sea floor perpendicular to each other. Each HLA should be as long as the water is deep and contain as many hydrophones as is feasible. If each hydrophone comprises a separate data channel, the cross of HLAs will also be capable of triangulating on ship noise. One VLA would still be required to separate the up-going and down-going wave-fields, but the sound of waves could be utilized as an energy source by redeploying the other four VLAs as two HLAs. This would allow the sound of wind-driven waves to be used without forfeiting the use of either intentional seismic sources or ship noise.

A second external factor is the opportunity to deploy an array of sensors in a borehole that will be drilled by the Department of Energy/Joint Industry Program (DOE/JIP) Consortium. The borehole array (BHA) will consist of hydrophones, three-component accelerometers and temperature sensors that would remain in the hole after the drill stem is recovered, letting the hole collapse and making the installation permanent. It would provide long-term monitoring from within the hydrate stability zone. If located at a site appropriate to the other requirements of the monitoring station, it would comprise a valuable addition to the MS/SFO.

If both these array modifications were to be incorporated, the seismo-acoustic components of the SFO would comprise three mutually perpendicular axes of a Cartesian coordinate system. One VLA would be the vertical axis in the water column and the horizontal axes would consist of the other four VLAs deployed horizontally. The BHA would comprise the sub-bottom portion of the vertical axis.

A second VLA has been constructed to accommodate geochemical sensors: off-the-shelf thermistors, CTDs, fluorometers and transmissometers. This array will provide the capability of studying hydrate-related hydrocarbon fluids in the water column. It will be possible to deploy this array either in an autonomous mode or as a component of the MS/SFO.

The original design of the MS/SFO calls for each of the VLAs to be equipped with a sea-floor data-logger. The five data loggers were to be connected to a central integrated data/power (IDP) module that would collect data from, and supply power to, the individual loggers. The change to using HLAs would not affect this arrangement.

The BHA has been funded separately by DOE/JIP and it would not represent a cost increase to the SFO. The only cost increase would be associated with increasing the length of the four VLAs so they could be re-deployed as two HLAs with lengths equivalent to the water depth. This could be a factor in whether or not the BHA becomes an integral part of the SFO.

Since the Consortium's break from the JIP plan, it appears likely that the placement of a BHA will not happen in the near future. For this reason and because the BHA concept adds so much to the overall station capability, the idea of emplacing shorter arrays via the Sea Floor Probe has been revived. Ten meter arrays, both geochemical and geophysical have been added to the plan for the station. Although these arrays are temporary, they will provide much valuable data at a fraction of the cost of a borehole array. These are the arrays that were emplaced on the sea-floor in May, 2005. Data from these arrays will be retrieved at the earliest opportunity.

CHANGE 2: DATA RECOVERY

External factors have also impacted the way SFO data will be recovered. For some time it was thought that a commercial service would be available in 2004 which would allow the IDP to stream data onto an optic-fiber link for near-to-real time transmission to shore. It was learned in the autumn of 2003, however, that the service would not become available until 2006 or later.

The use of a remotely operated vehicle (ROV) to download data directly from the SFO's data loggers was found to be prohibitively expensive due to the depth of water and the weight of the battery packs that would need to be exchanged. Therefore, until such a link becomes available, the IDP module will stream data onto an optic-fiber data recovery system (DRS) which will be connected via optic fiber to an access connector. Whenever downloading is required, a system of buoys will bring the DRS access connector to the surface so that the data can be downloaded onto computer in a boat. The system has been used successfully before and involves far less expense than repeated use of a deep-water ROV. The system has been dubbed the “Big M” and is illustrated in Fig.A1.

CHANGE 3: POSITIVE SYNCHRONIZATION OF TEST SIGNALS

The DRS will serve yet another need. While surveying to determine the configuration of sub-bottom strata in the vicinity of the SFO, the towed sea-floor sled will be used to generate shear waves for recording by the SFO's arrays. During the course of that survey, an access connector will be brought to the surface and connected to a radio telemetry buoy that will synchronize the firing and receiving of signals.

CHANGE 4: ELECTRICAL POWER FOR THE SFO

The Gulf of Mexico Hydrates Research Consortium funds the development of microbial batteries but it will be some time before they can provide electrical power to the MS/SFO. In the meantime, the IDP module will supply electricity to the MS/SFO by exchanging the pressure compensated battery (PCB) component about once a year. This will involve unplugging the depleted PCB from the IDP and plugging in a fresh one. The emplacement and exchange of PCBs will be accomplished by a station service device (SSD) especially designed for the task.

A docking station will be incorporated into the IDP module to facilitate changing the PCB. The SSD will carry the recharged PCB unit to the sea floor and return with the depleted unit. In addition, the SSD will be capable of recovering pore-fluid samples at *in situ* pressures. Perhaps most significantly, the SSD will be the means by which all station systems are connected to the IDP for data recovery and electrical power.

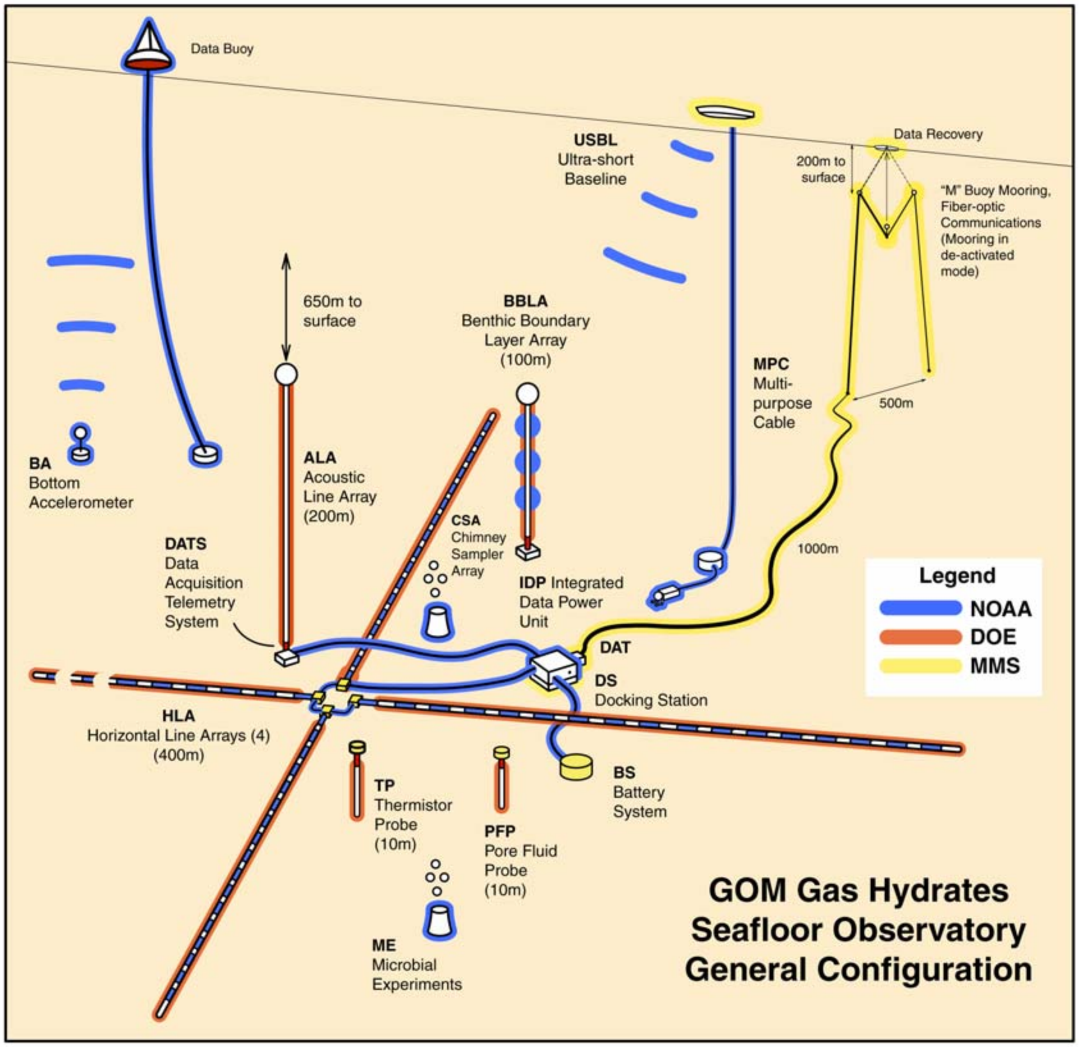


Figure A1. Diagram of the monitoring station/sea floor observatory.

Conclusion

Modifications discussed herein are not intended to change the basic concepts, overall plans and mission for the MS/SFO. Instead, they are expected to enhance the accomplishment of that mission.

Funding has been requested for the supply of components and construction of the new systems in order to adapt to the changing circumstances, as well as for the continuation of the all-important, on-going, studies and systems development projects. Data are anticipated in 2006 and tests of the station's operation and software systems should follow. A fully operational station is anticipated in 2007.