Copy No.________

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Reverberation Inversion Using SWAMI

C. Calnan
xwave

xwave
36 Solutions Drive
Halifax, NS B3S 1N2

Project Manager: J. Theriault, 902-426-3100 ext 376

Contract Number: W7707-03-2310

Contract Scientific Authority: J. Theriault, 902-426-3100 ext 376

xwave Contract Number: 1004920 REV IN&A

Defence R&D Canada – Atlantic

Contract Report
DRDC Atlantic CR 2004-113
December 2004

This page intentionally left blank.

Reverberation Inversion Using SWAMI

C. Calnan
xwave

xwave
36 Solutions Drive
Halifax, NS
B3S 1N2

Project Manager: J. Theriault 902-426-3100 Ext 376

Contract Number: W7707-03-2310

Contract Scientific Authority: J. Theriault 902-426-3100 ext 376

xwave Contract Number: 1004920 REV IN&A

Defence R&D Canada - Atlantic

Contract Report

DRDC Atlantic CR 2004-113

December 2004

Author

C. Calnan

C. Calnan

Approved by

J. Theriault
Contract Scientific Authority

Approved for release by

K. Foster
Chair, Document Review Panel (DRP)

Terms of release: The scientific or technical validity of this Contract Report is entirely the responsibility of the contractor and the contents do not necessarily have the approval or endorsement of Defence R&D Canada.

Terms of release: The information contained herein is proprietary to Her Majesty and is provided to the recipient on the understanding that it will be used for information and evaluation purposes only. Any commercial use including use for manufacture is prohibited. Release to third parties of this publication or information contained herein is prohibited without the prior written consent of Defence R&D Canada.

Abstract

The SWAMI suite of programs in use at DRDC Atlantic enables a user to produce modelled reverberation time series data based on a series of input parameters that describe the acoustic parameters of an area of a seabed. The modelled data can be compared to recorded reverberation data and a quantitative measurement can be made that essentially tests the goodness of fit of the modelled data to the measured values.

A contract was issued to design a program that would iteratively vary the values of the geoacoustic parameters input to the SWAMI programs and then run them, each time comparing the model results to a set of measured data. In this manner the program would arrive at a set of values that would produce model results that were as close to the measured data as SWAMI could provide.

Résumé

La suite de programmes de l'initiative de modélisation de sonar actif en eau peu profonde (SWAMI) utilisée par R & D pour la défense Canada - Atlantique (RDDC – Atlantique) permet à l'utilisateur de produire des données chronologiques de modélisation de réverbération basées sur une série de paramètres d'entrée décrivant les paramètres acoustiques d'une zone de fond marin. Les données de modélisation peuvent être comparées à des données de réverbération enregistrées, et il est possible de prendre une mesure quantitative essentiellement dans le but d'établir une correspondance entre les données de modélisation et les valeurs mesurées.

Un contrat a été attribué pour la conception d'un programme qui permettrait de faire varier de façon itérative les valeurs des entrées de paramètres géoacoutisques dans les programmes SWAMI, puis de les exécuter, tout en permettant chaque fois une comparaison des résultats de modélisation à un ensemble de données mesurées. Le programme permettrait ainsi d'arriver à un ensemble de valeurs susceptibles de produire des résultats de modélisation aussi près des données mesurées que possible au moyen de la suite de programmes SWAMI.

This page intentionally left blank.

Executive Summary

Introduction

The SWAMI suite of programs in use at DRDC Atlantic enables a user to produce modelled reverberation time series data based on a series of input parameters that describe the acoustic parameters of an area of a seabed. The modelled data can be compared to recorded reverberation data and a quantitative measurement can be made that essentially tests the goodness of fit of the modelled data to the measured values.

A contract was issued to design a program that would iteratively vary the values of the geoacoustic parameters input to the SWAMI programs and then run them, each time comparing the model results to a set of measured data. In this manner the program would arrive at a set of values that would produce model results that were as close to the measured data as SWAMI could provide.

Results

This document reports on the software developed to undertake the inversion process.

Significance

This effort lends itself to two particular applications. The first is Rapid Environmental Assessment (REA) using “through-the-sensor” techniques. The methodology extracts geoacoustic parameters from reverberation time series, one of the many goals of REA. The second application is in the development of Tactical Decision Aids (TDA). Real-time sonar performance estimation using in-situ measurements would provide the sonar operator with the capability to better employ the sensor.

Future plans

In parallel to this effort, the performance of the system has been published using synthetic data. An effort to invert reverberation data measured by DRDC Atlantic’s Towed Integrated Active-Passive Sonar (TIAPS) is underway. The intention is to produce estimates of geoacoustic parameters and compare with independent measurements. A further extension is to use a SWAMI component to predict transmission loss to compare with independently measured transmission loss in the area.

Calnan, C. 2004. *Reverberation Inversion Using SWAMI* (inversion de réverbération
au moyen de la suite de programmes SWAMI) DRDC Atlantic CR 2004-113.
Defence R&D Canada – Atlantic.

Sommaire

Introduction

La suite de programmes de l'initiative de modélisation de sonar actif en eau peu profonde (SWAMI) utilisée par R & D pour la défense Canada - Atlantique (RDDC Atlantique) permet à l'utilisateur de produire des données chronologiques de modélisation de réverbération basées sur une série de paramètres d'entrée décrivant les paramètres acoustiques d'une zone de fond marin. Les données de modélisation peuvent être comparées à des données de réverbération enregistrées, et il est possible de prendre une mesure quantitative essentiellement dans le but d'établir une correspondance entre les données de modélisation et les valeurs mesurées.

Un contrat a été attribué pour la conception d'un programme qui permettrait de faire varier de façon itérative la valeur des paramètres géoacoutisques saisis dans les programmes SWAMI, puis de les exécuter, tout en permettant chaque fois une comparaison des résultats de modélisation à un ensemble de données mesurées. Le programme permettrait ainsi d'arriver à un ensemble de valeurs susceptibles de produire des résultats de modélisation aussi près des données mesurées que possible au moyen de la suite de programmes SWAMI.

Résultats

Le présent document porte sur le logiciel développé dans le but d'effectuer le processus d'inversion.

Portée

Le projet mène à deux applications particulières. La première est l'analyse environnementale rapide (REA) à l'aide de techniques par capteurs. La méthodologie permet d'extraire les paramètres géoacoustiques des données chronologiques de réverbération, un des nombreux objectifs de la REA. La seconde application réside dans le développement d'aides à la prise de décisions tactiques (TDA). L'estimation en temps réel du rendement d'un sonar à l'aide de mesures prises sur place permettra à l'opérateur d'utiliser plus efficacement le sonar.

Recherches futures

Parallèlement au projet décrit aux présentes, le rendement du système fondé sur des données synthétiques a été publié. Une démarche d'inversion des données de réverbération mesurées par le sonar actif-passif intégré remorqué (TIAPS) de RDDC Atlantique est en cours. L'objectif est de produire des estimations des paramètres géoacoustiques et de les comparer à des mesures indépendantes. Une application supplémentaire vise l'utilisation d'un composant de la suite de programmes SWAMI pour la prédiction des pertes de transmission en vue d'une comparaison à des mesures indépendantes de pertes de transmission dans la zone.

Calnan, C. 2004. *Reverberation Inversion Using SWAMI* (inversion de réverbération au moyen de la suite de programmes SWAMI) RDDC Atlantique CR 2004-113. R & D pour la défense Canada – Atlantique.

Table of Contents

List of Tables

List of File Listings

1. Introduction

By means of “through the sensor probing,” array beam reverberation time series data may be recorded by a towed array. As well, given a series of postulated environmental parameters, it is possible to model reverberation and produce a time series that may be compared to measured data. Assuming sufficient model fidelity and environmental parameters sufficiently close to the “real” parameters, the modelled time series will be close to the measured data.

The closer the model environmental parameters values are to the actual values, the closer the modelled time series will be to the measured data.

To obtain the “best” values for the physical parameters an initial series of values is selected. The reverberation time series resulting from these values is then compared to the measured data and a “goodness of fit” value is obtained. In an iterative series of steps the test values are continuously varied to produce modelled output that is continuously closer to the measured data, as defined by an improving value for the goodness of fit metric.

In an N×2D model a geographic location is selected for which a sound speed profile is known. The model uses a constant sound speed profile over the entire study area. A number of straight lines of the same length, N, may be drawn radially from this point. These lines are not necessarily separated by equal angles. A selected number of evenly spaced points may then be chosen on each radial, values may be assigned for the half-space bottom parameters of compressional sound speed, compressional attenuation, seabed density, scattering strength, and bottom depth. These environmental values change with location, resulting in a varying bottom half-space.

Performing the inversion on the reverberation data ultimately results in values for the desired environmental parameters. These values can be used to model the environment for any number of purposes. For one example, transmission losses can be calculated.

The SWAMI reverberation model can produce any number of desired theoretical towed array beam reverberation time series based on an input environment that uses the above-mentioned parameters. This model was chosen for the inversion.

Reverberation data and sound speed profiles were recorded during sea trials Q246 and Q277 in the Emerald Basin/Bank area. These data may be used as the basis for comparison with SWAMI-produced modelled data.

This report documents the development, testing, and usage of the SWAMI-based reverberation inversion program **BREVER**. As well, all required inputs, the code that comprises **BREVER**, and instructions on the program’s use are presented.

BREVER was written in IDL and is compliant with the standards for the DRDC Atlantic IDL Library, as specified in *DRDC IDL Library Modernization Project* [Calnan, 2002].

In all cases in this project, all IDL program files contain only one main program or function, and the name of the program or function is the same as that of the file containing it. The only exception is that all filenames are in lower case letters only, whereas program and function names are given in capital letters. As two examples: the program **BREVER** resides in the file **brever.pro**, and the function PPR_SETUP resides in the file **ppr_setup.pro**.

Besides common English typographic conventions, the following conventions are used in this document:

- **bold text** is used for filenames (e.g. **test.pro** or **/local/files/test.pro**)
- ***bold italics text*** is used for directories (e.g. ***/usr/tmp***)
- *italics* text is used for computer names (e.g. *Grebe*)
- **Bold Arial text** is used to indicate program names (e.g. **BREVER**)
- Arial text is used to indicate function and subroutine names (e.g. PPR_SETUP)
- *italic Arial text* is used for computer program, system, or environment variable names (e.g. *IDL_PATH*)
- `Courier text` is used for text to be typed on the keyboard, code or text file lines, etc. (e.g. enter "`idl`")

2. Inversion Method

2.1 Choosing an Inversion Methodology

Once it was decided to obtain the best-fit geoacoustical parameters by iteratively running SWAMI with varying parameter values, the next step was to decide on a way in which to perform this task.

The technique that was decided upon was the Adaptive Simulated Simplex Annealing (ASSA) method.

There were several reasons for choosing this method:

- both the Scientific Authority and this author have had experience with the technique,
- IDL code to perform many of the ASSA-related tasks already exists at DRDC Atlantic,
- the fact that the parameters that were being solved for were very loosely related, if at all, meant that the "energy space" resulting from varying the values would be very likely highly complex, and so well suited to ASSA,
- depending on the user's choices, the number of dimensions to be solved for could become quite high, something that ASSA handles very well, and
- ASSA has already been successfully used to solve similar multi-dimensional problems in several projects at DREA/DRDC Atlantic.

2.2 BREVER's Method of Operation

The general ASSA analysis form for **BREVER** was derived from the FORTRAN program **ASSA**, described in Calnan [2003]. That program was based on FORTRAN code by Chris Gillard and IDL programs by Stan Dosso and Anna-Liesa Lapinski.

2.2.1 General

The reverberation inversion program **BREVER** must be given a number of fixed parameters with respect to the area over which seabed parameters are to be determined. The items that must be specified ahead of time are:

- the number of radials
- the bearings of the radials
- the length of the radials (all are the same length)
- the location of the radials' centre
- estimates of water depth along the radials

A user must also obtain other input related to the measured data, including:

- transmitter/pulse parameters
- receiver parameters
- a sound speed profile for the area at the time of the data measurements

Once these parameters have been settled on, the program can determine the values of other parameters. These target parameters include:

- the number of points per radial
- the seabed density
- compressional sound speed
- compressional attenuation
- scattering strength
- water depth

For the first target parameter, points/radial, the user provides the program with a minimum and maximum number of points to produce solutions for. The program steps through these values in increments of 1 and at each step solves for all the other target parameters, producing a minimum energy for each number of points/radial. At the end of its run, the program tells the user which number of points/radial produced the best overall minimum energy.

For the next four parameters the user provides the program with a minimum and maximum value. For the last parameter, depth, the user provides a measure of the uncertainty of the depth profile values. The depth's uncertainty is combined with the depths provided to get a minimum and maximum value for each point. During the program's run for each number of points/radial, values for each parameter are randomly produced from within the range provided. As the program proceeds towards a solution for a given points/radial value, each radial point's target values settle to a value that minimizes the difference between the slopes of the measured data and the model's output.

The number of parameters that **BREVER** solves for simultaneously (in ASSA terminology the number of dimensions, *NDIM*) depends on the number of radials, the number of points/radial, and the number of parameters to be solved for at each radial point. The relation between these items is:

$$NDIM = NPPRP \times NURP$$

where: *NPPRP* - is the number of target parameters per radial point, as selected from seabed density, compressional sound speed, compressional attenuation, scattering strength, and water depth

NURP - is the number of unique radial points

Since the radials have a common origin the target parameters only have to be solved once for that point. Remembering this, *NURP* is found via:

$$NURP = [\ NRAD \times (PPR - 1)\ ] + 1$$

where: *NRAD* - is the number of radials
PPR - is the number of points/radial
-1 - removes the centre point, common to all radials
+1 - adds the common centre point, but only once

An interesting feature of **BREVER** is that since it can vary the number of points/radial, the number of dimensions for which solutions are being sought changes with each points/radial loop. This is unlike the majority of ASSA-based programs currently in use at DRDC Atlantic, which assign or calculate the number of dimensions at the start of the problem and use that as a fixed parameter throughout the run of the program.

BREVER has a feature that allows a user to skip solving for certain parameters at the radial points. If the minimum and maximum values of a parameter's range have the same value the program will use that value throughout its run and not try to solve for it.

The opting-out of the search for a parameter's values is slightly different for water depth. Among the inputs to **BREVER** are water depth profiles along each radial, which are linearly interpolated to the locations of the radial points. Instead of specifying a minimum and maximum range from which to choose test depths, the program's input specifies an uncertainty in the depth's values. If this uncertainty is zero then the interpolated depths are used as is and depths are not varied or solved for. However if a non-zero uncertainty is specified then depths are solved for at each radial point with a minimum and maximum created by respectively subtracting and adding the uncertainty from the interpolated depth.

There are a couple of limitations to the fixed value scheme:

- with the exception of depth, the same fixed value is used for all the radial points
- either all the occurrences of a target parameter are solved for or none of the are; i.e. it is not possible to, for example, fix the scattering strengths on radials 1 and 2 and solve for them on radials 3 and 4

Every time a new set of target parameter values is arrived at a SWAMI run is made to create a modelled reverberation time series. The "slope" of the time series is calculated by subtracting adjacent time series via:

$$MOS_i = MOTS_i - MOTS_{i+1}$$
$$MES_i = METS_i - METS_{i+1}$$

where *MOS* - is the array of model slope values
MOTS - is the array of model time series values
MES - is the array of measured slope values
METS - is the array of measured time series values

Once the arrays of slope data have been calculated an "energy" value for the model is found. This is calculated by way of:

$$ENERGY = \Sigma|MOS_i - MES_i|$$

By calculating *ENERGY* in this manner the calibration of the reverberation calculation is irrelevant. It is only the slope of the curves that matters, and that is independent of the calibration.

2.2.2 Specific

This section gives a more specific description of how **BREVER** works and is, more or less, a "plain English" stepwise explanation of the program's operation.

- Input files are read in and the consistency of some of the data is verified.
- The number of target parameters/radial point is determined by comparing target minimum and maximum values or, for depth, checking the stated uncertainty.
- The measured data to be used as a basis for comparison are read in and converted to value-to-value differences, a.k.a. slopes.
- If no target parameters are to be searched for, the input values are used to perform a SWAMI run and the *ENERGY* resulting from comparing the model output to the measured data is calculated and written out. **BREVER** then ends.

- Otherwise a loop is run that steps through the numbers of user-selected points/radial.
    - At the top of each loop the time is checked, and if it has been more than 5 seconds since the last printout, the current status of the loop counter is printed out and the printout clock is reset.
    - The distance between radial points is calculated, the depths at the radial points are interpolated, and necessary arrays for test data, resulting energy, etc. are created.
    - An array of test target values is filled with an initial series of estimates, SWAMI is run with these values, and *ENERGY* is calculated for the model output. This is done for each simplex model space, of which there are *NDIM*+1 members.
    - A simplex simulated annealing loop is run for the number of temperature steps specified in the main input file. For each loop the temperature is decreased. ("Temperature" is, roughly, a measure of the width of the space from which random values are obtained; smaller temperatures result in a smaller range of values and usually occurs as the "best" values are approached.)

        - A perturbation loop is run until either an input specified number of loops have been run or a user-specified convergence value is reached. The value is reached when either the difference between the minimum and maximum *ENERGY* values over all the simplex model space or the minimum *ENERGY* itself is less than the value. The min/max *ENERGY* difference is used to escape cases where the minimum is actually reached (as seen by convergence of all the *ENERGY* values) but is not a very small value. If the convergence value is reached then a flag is set that allows an escape from the temperature loop.

- If the convergence value was not reached the temperature value is decreased by a user specified amount.

- Depending on whether the temperature loop was exited due to the number of steps being exceeded or the convergence value being reached, an appropriate message is written out.
- The parameters that gave rise to the minimum *ENERGY* are sequestered and the rest of the simplex model space is filled once again with random values.

- A downhill simplex quenching loop is started. The loop exits when either a specified number of loops were run or the specified downhill convergence value is reached. The value is reached in the same way and for the same reasons as in the perturbation loop.
  - This loop finds the minimum *ENERGY*, which is assumed to be a local near-minimum and "drops" to the minimum *ENERGY* near this point. It drops by varying the simplex model space parameters, running SWAMI, and obtaining a new *ENERGY* until either loop escape clause is reached.
- Depending on whether the downhill simplex loop was exited due to the number of steps being exceeded or convergence value being reached, an appropriate message is written out.
- The minimum energy and the target parameters that gave rise to it are printed out to both the screen and the output file.
- The minimum *ENERGY* and the average minimum *ENERGY* per radial point for the current number of points/radial are stored in an array.

- End of the loop over the specified points/radial.
- The stored *ENERGY* values are examined and both the overall minimum *ENERGY* and the overall minimum *ENERGY* per radial point are printed out to both the screen and the output file, along with their respective number of radial points.

3. Required Input

BREVER requires four “levels” of input files in order to run:

1. The single input file required by the program. Section 3.1 presents a sample of this file and describes its contents item by item. This file is referred to as the **BREVER** primary input file.
2. Several input files listed within the **BREVER** primary input file. Section 3.2 describes these files, which are referred to as the **BREVER** secondary input files.
3. The four input files initially read by the four SWAMI programs used by **BREVER**. Section 3.3 describes these files, the SWAMI primary input files.
4. Files named in the SWAMI primary input files. These are described in Section 3.4 and are referred to as the SWAMI secondary input files.

BREVER can solve the parameters for more than one radial at a time. However if there are more than one radial the user must be careful about how several parameters are entered. There are three locations where radial-related information is stored:

- **BREVER** primary input file data line 14 (Section 3.1), which lists one file for each radial
- **BREVER** primary input file data line 19 (Section 3.1), which has information for each radial’s radial description file (Section 3.4.4)
- the radials file (Section 3.4.3), which contains the bearings of the radials

The user must be careful to always refer to the radials in the same order in each of these locations. If the ordering is not maintained then the results will be incorrect.

3.1 BREVER Primary Input File

BREVER opens and reads its primary input file at the start of its run, storing the contents or derivations thereof in common area variables. It validates and verifies some of the contents (e.g. checks that named files exist, verifies that numerical values are valid), and if there are any problems writes fairly specific error messages. Once the entire file has been read in it is closed and not touched again.

3.1.1 Sample File

The following listing of a sample **BREVER** primary input file has some lines that are numbered. These lines are the actual data lines of the file and a description of each data line is given in Section 3.1.2.

The rules associated with this input file are:

- The file may contain blank lines and comment lines in order to assist the user and enhance readability. Comment lines are those with "#" as the first non-blank character.
- In all data lines (i.e. lines with data) the character "#" denotes the start of an in-line comment; everything from that character to the end of the line will be stripped away before the line is processed.
- Blank spaces are used to separate multiple items on a line.
- Because of the previous rule, neither filenames nor directory names may contain spaces, even if the operating system running **BREVER** allows it.
- Data lines must appear in the file in the order shown.

Listing 1. Sample BREVER Primary Input File

```
    #####################################################################
    #
    #  Input file for BREVER, the Reverberation Inversion program.
    #
    #####################################################################
    #
    #  File type specification for this file: BREVER file, format 1
1)      FTYPE  ;  BREVER  ;  1
    #
    #  Title (for BREVER-specific output files)
2)      Data Analysis Run 01

    ########################################################
    #
    #  ASSA parameters
    #
    #  T0   Tfact  Ntemp   Npert   NDHS    NAVE
3)     1.0    0.99   30       5     500   20000
    #
    #  NPERTM   PVC   PVC Multiplier
    #   (PVC = Perturbation Value Code = 1 or 2)
4)       5       1        3.0
    #
    #  Convergence values:
    #     SSA      DHS
5)      1.0E-02  1.0E-04

    ########################################################
    #
    #  Inversion target parameter min and max values.  If min and
    #  max are within 1.0E-5 the mean will be used as a constant
    #  and that parameter will not be solved for.
    #
    #  Number of points per radial (>1)
6)       60    81
    #
```

```
     #  Density  (>0. - 10.)
 7)       1.5   2.5
     #
     #  Compressional sound speed  (1000. - 5000.)
 8)       1800. 1800.
     #
     #  Compressional attenuation  (0. - 1.)
        # 0.20  0.20
 9)       0.15  0.25
     #
     #  Scattering Strength  (-80. - <0.)
10)       -35.  -20.
     #
     #  Depth uncertainty (0. means no uncertainty; use interp values)
11)       5.           # metres

     ########################################################
     #
     #  Files needed by the SWAMI programs:
     #
     #  INTEGRATE input filename
12)       integ.input
     #
     #  PMODES input file
13)       pmodes.input
     #
     #  MONOGO input file(s) - 1/radial; each name unique
14)       monogo.input1
14)       monogo.input2
     #
     #  CMBRAD input file
15)       cmbrad.input

     ########################################################
     #
     #  Other required information:
     #
     #  Directory holding the SWAMI executables
16)       /home/models/SWAMI/bin/
     #
     #  Length of the radials (km)
17)       60.
     #
     #  Template ".env" file
     #  (for its non-depth and non-target parameter data)
18)       template.env
     #
     #  ".des" file data, one set of values for each radial
19)       DESINFO
19)        Radial 1
19)         GRADIENT = R1dprof.dat
```

```
19)         Surface Scattering model = ChapmanHarris
19)         Sea State = 2.0
19)         Wind Speed = 15 knots
19)        Radial 2
19)         GRADIENT = R2dprof.dat
19)         Sea State = 2.2
19)         Surface Scattering model = ChapmanHarris
19)         Wind Speed = 15 knots
19)       EODESDATA
     #
     #  Measured data for comparison and Energy calculations
20)       test.dat
     #
     #  Time zero in the measured data file
21)       1998-06-02  00:11:05
     #
     #  Bins to use in the measured data file
22)       1  11  23:27
     #
     #  BREVER output results file
23)       brever.out.t4
     #
     #  Measured data copied out?
24)       N
25)       N
     #
     #  Best fit output files
26)       y bf_rev_t4
27)       y bf_dif_t4

     ########################################################
     #
     #  Note:
     #  - The number of input files to MONOGO and the number of radials
     #    in the DESINFO block are obtained by getting the number of
     #    radials specified in the radials file named in INTEGRATE's
     #    input file.
```

3.1.2 File Description

This subsection describes the **BREVER** input file by data line number.

1) This is the file type specifier line. It indicates that the line is a file type specifier line (“FTYPE”), the program that reads the file (“BREVER”), and gives the input file’s format version number. The line must appear exactly as shown. If the **BREVER** input file format is ever modified then the last item, the digit “1”, will be incremented to indicate format version number 2.

2) This is the title used in output.

3-5) These lines contain data needed for the ASSA procedure:

Line 3 has:

T0 - temperature zero, the initial temperature of the simplex loop

TFACT - the amount the current temperature is multiplied by in each temperature loop. This value must be greater than 0.0 and less than 1.0.

NTEMP - the maximum number of temperature loops to run. The temperature loop can be "escaped" if the energies reach the convergence value *SSACV* before *NTEMP* loops are run.

NPERT - number of perturbations to perform for each temperature.

NDHS - the maximum number of downhill simplex steps to take in the final quenching run. The downhill loop can be "escaped" if the energies reach the convergence value *DHSCV* before *NDHS* loops are run.

NAVE - number of dimensional values to average to determine perturbation size. See Dosso et al. [2001] and Fallat et al. [1999] for details.

Line 4 has:

NPERTM - if a perturbation loop runs more than *NPERT*×*NPERTM* times without finding a new solution, the perturbation size is reduced. This value is used as part of an escape mechanism to get out of a potentially infinite loop.

PVC - a perturbation value code that can be either:

1 - get new perturbation values based on the average of previous values

2 - get new perturbation values based on the maximum of previous values

PVCM - the perturbation value code multiplier; new perturbation values obtained according to the instructions of *PVC* are multiplied by this value before being used.

Line 5 has two items, both of which make use of E_{RATIO}. That value is calculated from the minimum and maximum energies over the simplex model space by:

$$E_{RATIO} = (E_{MAX} - E_{MIN}) / [0.5 \times (E_{MAX} + E_{MIN})]$$

The items on line 5 are:

SSACV - simplex simulated annealing loop convergence value. If, during a temperature loop run, either the minimum energy or E_{RATIO} becomes less than or equal to this value, the temperature loop is "escaped."

DHSCV - downhill simplex loop convergence value. If, during a temperature loop run, either the minimum energy or E_{RATIO}

becomes less than or equal to this value, the downhill simplex loop is “escaped.”

6) This line indicates the minimum and maximum number of points/radial to use in the testing. The program steps from the minimum to the maximum number in steps of 1, and performs a full analysis for each step.

   Rules on the minimum/maximum limits are:
   - the minimum must be 1 or greater
   - the maximum must be at least the minimum

7-10) These lines have the minimum and maximum ASSA values for the remaining four parameters that **BREVER** solves for. However, if the minimum and maximum are within 10^{-5} of each other the mean will be used as a constant and that parameter will not be solved for.

   Rules on their min/max limits are:
   7) must be > 0. and ≤ 10.
   8) must be ≥ 1000 and ≤ 5000.
   9) must be ≥ 0. and ≤ 1.
   10) must be ≥ -80. and < 0.

11) This is the depth uncertainty at the radial points as read from the profile files and interpolated to the radial points. If this is 0.0 then the interpolated depths are used as is, but if this is > 0.0 then radial point depths are solved for in the range of interpolated depth ± this uncertainty. This value must be positive.

12-15) These are the names of the input files to be used by the SWAMI programs **INTEGRATE**, **PMODES**, **MONOGO**, and **CMBRAD**, in that order. Section 3.3 presents sample versions of these files and describes them.

   The following rules must be followed for the files listed in data lines 12 and 15 (the **INTEGRATE** and **CMBRAD** input files):
   - Both input files name a radials file (Section 3.4.3); they must both name the same file.
   - Both input files have a filename root used to create files. This filename root must be the same in both files.

14) There must be one file named for each radial, as specified in the radials files listed in the **INTEGRATE** and **CMBRAD** input files. If there is more than one radial, these files are checked to ensure that each name is unique and that the time data in all the files are the same.

16) This is the directory that holds the four SWAMI executables used by **BREVER**. The directory name must have “/” or “\” at the end, depending on the operating system of the computer **BREVER** is being run on.

17) The radial length must be > 0. This length is checked for consistency against

the maximum time for a calculation (located in the **MONOGO** input file; Section 3.3.3) and the maximum sound speed in the sound speed profile (located in the template **.env** file named in data line 18 of the **BREVER** input file; Section 3.2.1).

18) This template file's contents will be copied to all the **.env** files created for **PMODES** except for the water depth, density, compressional sound speed, and compressional attenuation. Values for these parameters will be provided by **BREVER**.

19) These lines contain data that will be put into the environment description files named in the **MONOGO** input files, of which there is one per radial. This data type must start with a line containing "DESINFO" (**.des** file information) and end with a line containing "EODESINFO" (end of DESINFO). In between there must be one line for each radial in order starting with "1". All radial specification lines must be followed by a line with "GRADIENT =" and the name of a file with the depth profile for that radial. The user may also optionally include "SURFACE SCATTERING MODEL", "SEA STATE", and "WIND SPEED" lines. The lines within a radial's section may be in any order.

Section 3.4.4 has more information on these data lines.

20) The file of measured data in DREA .DAT or DREA .DAT32 format.

21) The analysis zero time in ISO 8601 format. This is the reference time zero as used by **MONOGO**, and data are only read from this time plus T_{MIN} of the **MONOGO** input file (Section 3.3.3).

22) This line lists the beams to be analysed. The number of beams selected must agree with the number of beams specified in the receiver file (i.e. the **.rx** file named in the **INTEGRATE** input file of data line 12) and must be consistent with the number of beams in the input data file named in data line 20. The rules for decoding this line are:

"*"	- selects all beams
"1 12 5"	- selects beams 1, 5, and 12 in that order
"2:5 14:12"	- selects beams 2, 3, 4, 5, 14, 13, and 12
"1 to 23 by 5"	- selects beams 1, 6, 11, 16, and 21
"1:100 x 12 x21:29"	- selects all the beams from 1 to 100 except beams 12, and 21 through 29

Two points two be aware of:

- If a beam is named more than once it will only be used once.
- The first channel of the input file is referred to as beam 1.

24) One of two options:

Y - indicates "yes, print out the measured reverberation data used in analysis as a **SAPLOT** input file"; "Y" must be followed by a

filename root, to which the program appends **.sap**
N - indicates not to print out the data; it must appear alone

25) One of two options:
Y - indicates "yes, print out the measured reverberation differences used in analysis as a **SAPLOT** input file"; "Y" must be followed by a filename root, to which the program appends **.sap**
N - indicates not to print out the data; it must appear alone

24-25) If both start with "Y" the filename roots must be different.

26) One of two options:
Y - indicates "yes, print out the best fit reverberation data produced by the program for a particular points/radial value as a **SAPLOT** input file"; "Y" must be followed by a filename root, to which the program appends **_P.sap**, where **P** is the number of points/radial
N – indicates not to print out the data; it must appear alone

27) One of two options:
Y - indicates "yes, print out the best fit reverberation differences produced by the program for a particular points/radial value as a **SAPLOT** input file"; "Y" must be followed by a filename root, to which the program appends **_P.sap**, where **P** is the number of points/radial
N - indicates not to print out the data; it must appear alone

26-27) If both start with "Y" the filename roots must be different.

3.2 BREVER Secondary Input Files

This section describes three input files named in **BREVER's** primary input file and read by **BREVER**. There are other input files named in the **BREVER** primary input file, but since they are primary input files for SWAMI's programs, they are discussed in Section 3.3.

3.2.1 Template ".env" File

The SWAMI program **PMODES** reads data files of environmental data (known familiarly as "**.env**" files since their names usually end in **.env**) from the study area, and from them creates binary files of mode data for use by **MONOGO**. These **.env** files are specific to each point on each radial and contain several of the parameters that **BREVER** solves for. Because of this **BREVER** continually creates new **.env** files with varying values of these parameters for **PMODES** to process during an inversion run.

However **.env** files also contain other data needed by **PMODES** that **BREVER** has nothing to do with. Because of this **BREVER** requires a template **.env** file. The data that are untouched by **BREVER** are read from the file and stored during **BREVER's** startup. These values are then copied back unchanged to the new files that **BREVER** creates. This also means that the "untouchable" data in this template file are used over the entire study area.

For that reason **BREVER** needs a template **.env** file with the data to be copied unchanged to the **.env** files it creates. This file is named on data line 18 of the **BREVER** primary input file

Following is a line-numbered sample template **.env** file. The items in bold type on lines 2 and 6 are those that are changed by **BREVER**; everything else is copied unchanged.

Listing 2. Sample Template ".env" File Listing

```
 1) Data Analysis 01
 2) 90.   1.0   0.       ! depth (m), density (g/cm3), atten
 3)   0.   1500.         ! depth, sound speed, start of profile
 3) 120.   1500.         ! depth, sound speed
 4) -1.   -1.            ! flag end of profile
 5)  0.    0.            ! shear speed and attenuation
 6) -1.    2.0  1800.  0.2  ! layer thick, dens, comp spd & atten
 7)  0.    0.        ! surface and bottom roughness
 8)  1  1100.        ! number of frequencies, frequency (Hz)
 9)  1    50.        ! number of sources, source depth (m)
10)  1    60.        ! number of targets (m)
11) 11    -1         ! no. of water column layers, max no. of modes
```

By line, this file contains:

1) This is a run title; it is ignored by **BREVER**.

2) The three items on this line are:
   - water depth (m), this is calculated by **BREVER** from depth profile and radial point position data.
   - water density (g/cm^3)
   - an attenuation flag. If the flag's value is between 0 and 1, **PMODES** uses the value as the attenuation coefficient for the water column (in dB/Hz-km). If the flag's value equals -1, THORP volume attenuation is used, and if the flag's value equals -2, URICK Volume Attenuation is used (a modified version of THORP).

3) There are a number of lines that define the sound speed profile. The first value on each line is the depth in metres and the second is the sound speed in m/s.

There are some rules associated with the sound speed profile data, all of which are tested for:

- The first line of sound speed profile data must have a depth of zero.
- The depth must increase from line to line in the profile.
- The maximum depth (i.e. the depth on the last line of sound speed profile data) must be at least equal to the deepest value in the depth profiles (Section 3.2.3) plus the depth uncertainty given in data line 11 of the **BREVER** primary input file.

4) A line with two negative values denotes the end of the sound speed profile data.

5) The two items are nominally sheer speed and attenuation. The line's contents are read but ignored and the line itself has been carried over from older code.

6) This line has four items, the last three of which are set by **BREVER**:
   - bottom layer thickness, which must be –1.0 for the inversion to work properly
   - bottom density (g/cm^3)
   - compressional sound speed (m/s)
   - compressional attenuation

7) The two items in this line are the RMS roughness of, respectively, the surface and the bottom.

8) The two items on this line are the number of sound frequencies used to produce the measured data and the frequencies themselves (Hz). This inversion is only designed for single frequency analyses.

9) The two items on this line are the number of sound sources used to produce the measured data and the source depths (m). This inversion is only designed for a single source.

10) The two items on this line are the number of targets used to produce the measured data and the target depths (m). This inversion is only designed for a single target.

11) The two items on this line are the number of computational layers and the maximum number of modes. Default values are used for either of these two parameters that have a negative value in this file.

3.2.2 Measured Data File

Data line 20 of the **BREVER** primary input file specifies the name of the file of measured data that is used as a basis for the program's determination of the geoacoustical parameters of the study area. This file must be presented to **BREVER** in a ready-to-use format, meaning that all filtering, windowing, smoothing, etc. must have already been performed. This file must be in either DREA .DAT or DREA .DAT32 format.

Data are read from the measured data file according to the following conditions:

- When this file is read in only the channels specified in data line 22 of the **BREVER** primary input file are read.
- The data's origin is taken to be the time listed in data line 21 of the **BREVER** primary input file. All data in the file that occur before that point are ignored completely.
- The first record read is the one that occurs are the data line 21 time plus the number of seconds stated on line 7 of the **MONOGO** primary input file (Section 3.3.3).
- The SWAMI analysis time step size is obtained from line 9 of the **MONOGO** primary input file (Section 3.3.3). Accordingly, data are only read from the measured data file at this frequency. In the normal course of events the measured data file contains data with a smaller line-to-line time step than the time interval of the analysis time step. When this happens the measured data file's data are subsampled. For example, if the measured file has data at 2000 Hz (0.0005 s between records), and the analysis time step is 0.5 second, only 1 of every 1000 records will be read from the measured data file.
- The maximum analysis time as stated in line 8 of the **MONOGO** primary input file (Section 3.3.3) is added to the data's origin (data line 21 of the **BREVER** primary input file). Once the routine reading the measured data reaches a recorded data after this time, no more measured data are read.

The required measured data are read as part of **BREVER's** startup procedure. If the user desires, the data actually used during the inversion may be put into a **SAPLOT** input file, thus allowing that program to plot the (usually) subsampled and clipped data. As well, the value-to-value differences, a.k.a. slopes, may be put into a **SAPLOT** input file. These options are available on lines 24 and 25 of the **BREVER** primary input file and are discussed in Sections 5.1.1 and 5.1.2.

3.2.3 Depth Profile File(s)

The first item on the second line of the **.env** files read by **PMODES** (Section 3.2.1) is the water depth at that file's radial point. These depths are calculated by **BREVER**, which linearly interpolates a depth for the radial point based on the values in depth profile data files.

There must be one depth profile file for each radial, and they are listed as data line 19 in the **BREVER** primary input file. The data that comprises data line 19 is placed by **BREVER** into radial description (**.des**) files, as partially described in Section 3.4.4. The depth profile can probably be most easily obtained by locating the position of the centre of the radials on a bathymetric chart, drawing radials from that point at the appropriate bearings, and then reading the depths along the radials for a distance that extends somewhat past a distance representing the length of the radials.

Following is a sample depth profile file.

Listing 3. Sample Depth Profile File

```
  0.0  101.
 11.4   91.
 12.0   88.
 26.3   91.
 34.8   88.
 42.7   91.
 51.4   95.
 60.2   93.
```

On every line of the file are two numbers: the first value is the distance from the radials’ centre point in km, and the second value is the water depth at that point in m.

Depth profile files have a number of constraints:

- This file may only consist of lines with data, blank and comment lines are not allowed.
- There must be two values on each line.
- The distance of the first line must be zero.
- If there is more than one radial and so more than one depth profile file, the line one depth must be the same in all files since they all describe the same physical location.
- Distances must increase from line to line.
- The distance associated with the last line in the file must be greater than the length of the radials specified in data line 17 of the **BREVER** primary input file.

Since **BREVER** linearly interpolates the data in these files to produce depths at the locations of radial points, the selection of data for these files is important. Specifically, if a user selects depths at locations where the depths are changing rather than at the middle of areas of constant depth, the interpolated values are much more likely to be accurate.

3.3 SWAMI Primary Input Files

This section is divided into four parts. Each part presents a sample of the primary input files for the four SWAMI programs used by **BREVER** in order of usage.

The input files to the SWAMI programs have no provision for ad hoc comment lines or blank lines. However the character “ ! ” can be used to indicate the start of an in-line comment. As such, that character can’t be used as part of the data.

3.3.1 INTEGRATE Primary Input File

Files used as input to the SWAMI program **INTEGRATE** are named in data line 12 of the **BREVER** primary input file.

Following is a line-numbered sample **INTEGRATE** primary input file.

Listing 4. Sample INTEGRATE Primary Input File

```
 1) Data Analysis 01 - omni transmit/receive
 2) LR                    ! options, long output, use radials file
 3) da01.tx               !  Transmitter Input file.
 4) da01.rx               !  Receiver Input file
 5) One_radial            !  Input Radials File
 6) da01                  !  File name base for output files
 7) -90.0                 !  minimum vertical angle
 8)  90.0                 !  maximum vertical angle
 9)   0.5                 !  vertical angle increment
10)   0.5                 !  azimuthal angle increment
```

By line, this file contains:

1) This is a run title; it is ignored by **BREVER**.

2) This line contains the processing options. Although **INTEGRATE** has 13 options to choose from only "`LR`" should be selected. The first option causes more information to be written to the **.log** file, which can be useful in case there are problems with the program, and the second indicates that information on radials will be entered explicitly.

3) This is the name of a file with information on the transmitter. See Section 3.4.1 for information on this file.

4) This is the name of a file with information on the receiver. See Section 3.4.2 for information on this file

5) This is the name of the file with the radials' information. See Section 3.4.3 for information on this file. This filename must be the same as that listed in line 3 of **CMBRAD's** primary input file.

6) This is a base for filenames produced by this program. This character string must be the same as that listed in line 4 of **CMBRAD's** primary input file.

7) This is the minimum vertical angle for calculations in degrees, where 0º is horizontal.

8) This is the maximum vertical angle for calculations in degrees, where 0º is horizontal.

9) This is the size of an angular step vertically from the minimum to the maximum vertical angle.

10) This is the azimuthal step size used to split the full 360º of bearing among the various radials. The larger this value is, the coarser the gradation between radials.

3.3.2 PMODES Primary Input File

Files used as input to the SWAMI program **PMODES** are named in data line 13 of the main **BREVER** input file.

Following is a line-numbered sample **PMODES** primary input file. If a user creates a file exactly as described below, **PMODES** will be satisfied.

Listing 5. Sample PMODES Primary Input File

```
1)  L                 ! Long output
2)   1
3)  file.env
4)  file.bin
```

Line 1 contains the **PMODES** processing options that are to be used. There are more than just “L”, but that is the only one that is really needed to perform reverberation inversion. If line 1 of the file contains the options “T” or “O”, **BREVER** will print an error message and stop. There is nothing inherently wrong with these options, it’s just that these options involve inserting extra lines in the file, and the **BREVER** IDL function that reads this file (READ_PMODES_IN) does not yet have code to handle the extra lines properly, and so will simply reject the file. At present the safest case is to only use option “L”, which provides for extra output in the **.log** file. It is because READ_PMODES_IN may eventually be expanded to take all the **PMODES** options into account that **BREVER** even bothers to use a **PMODES** input file instead of just creating one with just the “L” option.

Before **BREVER** runs **PMODES** it creates a new primary input file for that program. The first line of the new file is a copy of line 1 of the user-provided file. That line is followed by a line with the number of unique radial points in the study area. (See Section 2.2.1’s description of *NURP* for this item.)

Line 2 is then followed by the names of **PMODES** secondary input and output files. All filenames are generated by **BREVER** and so are beyond control of the user.

3.3.3 MONOGO Primary Input File(s)

Files used as input to the SWAMI program **MONOGO** are named in data line 14 of the main **BREVER** input file. There must be one **MONOGO** input file for each radial, and each file must have a unique name since that program can only process one radial at a time.

All **MONOGO** input files must exist in their correct form before **BREVER** can run since **BREVER** does not modify their contents. However **BREVER** does read the file to get the line 2 data since it creates those files and must know what to call them. **BREVER** also verifies that the data in lines 7-9 are the same in all **MONOGO** primary input files.

Following is a line-numbered sample **MONOGO** primary input file.

Listing 6. Sample MONOGO Primary Input File(s)

```
1) Data Analysis 01
2) L                            ! options, long output
3) da01-R1.des                  ! Input Environmental Description
4) da01_0692.sys                ! Input System File
5) da01_0692_ngv.rev            ! Output No-Group Reverb file
6) da01_0692_gv.rev             ! Output Reverb file inc GV
7)   1.0                        ! minimum time
8)  80.0                        ! maximum time
9)   0.5                        ! time increment
```

By line this file contains:

1) This is a run title; it is ignored by **BREVER**.

2) This line contains the processing options. Although **MONOGO** has eight options to choose from only "L" should be selected. That option causes more information to be written to the **.log** file, which would be useful in case there are problems with the program.

3) This is the name of a file that describes the input environment and is created by **BREVER**. Its name should be unique in each **MONOGO** input file, although this is not required. In the sample file above the **da01** is taken from the filename root in the **INTEGRATE** and **CMBRAD** primary input files and the **R1** was chosen to indicate radial 1. This file is partly described in Section 4.2.3 and more completely described in the SWAMI documentation.

4) This is the name of an input system file that is created by **INTEGRATE**. The name of this file is predetermined and is taken from a couple of places. While

creating the **MONOGO** primary input file the user must create the filename according to the rules used by **INTEGRATE** to name it. For our example:

da01 - **INTEGRATE** got this string from its primary input file
_ - put there by **INTEGRATE**
0692 - calculated by **INTEGRATE** from the information in its radials file, the name of which it got from its primary input file. To get this item **INTEGRATE** took the radial's bearing angle, multiplied it by 10., and found the nearest integer. In this case the bearing would have been 69.2º in the radials file. (Actually it could have been any angle from 69.15001º through 69.24999º.)
.sys - put there by **INTEGRATE**

5) This will be the name of a reverberation file created by **MONOGO** that lacks group data (**n**o **g**roup re**v**erb = "ngv"). The naming scheme used in this example must be followed to prevent confusion. As can be surmised, the first part of the filename is obtained from the same location as the line 4 file, and the latter part is used to indicate the type of data contained in the file.

6) This will be the name of a reverberation file created by **MONOGO** that has group data. The naming scheme used in this example must be followed to prevent confusion and arises from the same origin as the line 5 filename.

7) This is the minimum time in seconds for which **MONOGO** will produce time series reverberation data. This value must be the same in all **MONOGO** primary input files used for a **BREVER** run.

8) This is the maximum time in seconds for which **MONOGO** will produce time series reverberation data. This value must be the same in all **MONOGO** primary input files used for a **BREVER** run.

9) This is the size of the time step for which reverberation data will be calculated. It must be a multiple of the time step size of the measured data (Section 3.2.2) being used as a basis for comparison. This value must be the same in all **MONOGO** primary input files used for a **BREVER** run.

3.3.4 CMBRAD Primary Input File

Files used as input to the SWAMI program **CMBRAD** are named in data line 15 of the main **BREVER** input file.

Following is a line-numbered sample **CMBRAD** primary input file.

Listing 7. Sample CMBRAD Primary Input File

```
1) DA 01 - VLA transmit/HLA receive
2) LRG      ! options, long output, radials file, group velocity files
3) One_radial    ! The radials file
4) da01          ! Filename root
```

By line this file contains:

1) This is a run title; it is ignored by **BREVER**.

2) This line contains the processing options. Although **CMBRAD** has seven options to choose from these three should always be selected and the others should be omitted, although they will not affect processing. The effects of these listed options are:
   - `L` - makes **CMBRAD** produce "long" output with extra information in the **.log** file
   - `R` - makes the program use a radials file
   - `G` - makes the program produce group velocity output

3) This is the name of the file with the radials' information. See Section 3.4.3 for information on this file. This filename must be the same as that listed in line 5 of **INTEGRATE's** primary input file.

4) This is a base for filenames produced by this program. This character string must be the same as that listed in line 6 of **INTEGRATE's** primary input file.

3.4 SWAMI Secondary Input Files

This section describes the SWAMI secondary input files that a user must have in place before **BREVER** can be run. These files are all listed in SWAMI primary input files.

These files have no provision for ad hoc comment lines or blank lines. However the character "`!`" can be used to indicate the start of an in-line comment. As such, that character can't be used as part of the data.

3.4.1 Transmitter Characteristics File

This file is named on line 2 of the **INTEGRATE** primary input file and presents the characteristics of a single frequency transmitter. It is completely ignored by **BREVER** but is required by **INTEGRATE**.

Following is a line-numbered sample transmitter characteristics file.

Listing 8. Sample Transmitter Characteristics File

```
 1) TRANSMITTER                  ! flag transmitter
 2) vla transmitter              ! Title
 3) This example file describes a vla transmitter.
 4) TV1                          ! Beam Pattern file name root
 5) 1100    50                   ! frequency (Hz), bandwidth
 6)    1   223                   ! pulse length(s), source level(dB)
 7) HFM   0.0                    ! pulse type, parameter
 8) Vertical Line Array          ! transmitter type
 9) SQUARE                       ! weighting
10)  1.0  -20                    ! weighting parameter, floor (dB)
11)  2      0.8064               ! number of elements, spacing (m)
12)  0      0.0                  ! tilt (deg),  Steering Angle (deg)
```

By line this file contains:

1) This must consist of the string “TRANSMITTER” in that exact manner since it is used as a file type identifier.

2) A job title; the exact contents are irrelevant to **BREVER’s** operation.

3) A job description; the exact contents are irrelevant to **BREVER’s** operation.

4) A filename root for a number of **INTEGRATE** output files. In order to avoid potential problems it should be unique.

5) The frequency and bandwidth of the transmitter’s output, both in Hz.

6) The transmitter’s pulse length (s) and source level (dB).

7) The pulse type and a pulse parameter.

8) The transmitter type is put here, along with keywords describing further transmitter information. The keywords that are searched for are:
   BINARY - the next line will be the last line in the file and it must contain the name of a binary file that contains beam pattern data
   ASCII - the next line will be the last line in the file and it must contain the name of an ASCII file that contains beam pattern data
   OMNI - this is the last line in the file, **INTEGRATE** will produce beam pattern data itself.

   If none of these words are present the next four lines must be in the file.

9) This string describes the weighting method to use. This string may contain:
   FILE:xxx - **xxx** is the name of a file from which weights are read

```
HANNING
HAMMING
RCOSINE
TAYLOR
TRIANGULAR
SQUARE
```

All but the “`FILE:`” option cause **INTEGRATE** to create the appropriate set of weights for the method named.

10) Two values will be read, but the first, a weighting parameter, is only used if the previous line has “`RCOSINE`”. The second parameter is the minimum sensitivity in dB.

11) This line contains the number of elements in the transmitter array and the distance between them in m. The assumption is that the transmitters are evenly spaced.

12) This line has two items:
    - the array tilt in degrees. For a vertical line array, this is measured from the vertical in the direction of the array heading. For horizontal line array, it is measured down from the horizontal.
    - a steering angle in degrees.

3.4.2 Receiver Characteristics File

This file is named on line 3 of the **INTEGRATE** primary input file and presents the characteristics of a receiver. It is completely ignored by **BREVER** but is required by **INTEGRATE**.

Following is a line-numbered sample receiver characteristics file.

Listing 9. Sample Receiver Characteristics File

```
 1) RECEIVER                   ! flag receiver
 2) HLA Receiver               ! Title
 3) 32-element 1/2 wavelength spacing HLA 1100 Hz    ! description
 4) RHL                        ! Beam Pattern file name root
 5)  1                         ! Number of Beams
 6) -2.0    3.0                ! DT, sysloss (dB)
 7) MULTIPLE                   ! flag for multiple beams
 8) Horizontal Line Array      ! receiver type
 9) HAMMING                    ! weighting
10) 1.0   -30                  ! weighting param, floor(dB)
11) 32     0.385               ! no. of elements, spacing(m)
```

```
12) 0.0                           ! Tilt (deg)
13) 0.0                           ! Noise Equiv. bandwidth (Hz)
14) Broadside beam                ! beam name
15)  0.0   -300                   ! steering angle, self noise
```

By line this file contains:

1) This must consist of the string “RECEIVER” in that exact manner since it is used as a file type identifier.

2) A job title; the exact contents are irrelevant to **BREVER’s** operation.

3) A job description; the exact contents are irrelevant to **BREVER’s** operation.

4) A filename root for a number of **INTEGRATE** output files. In order to avoid potential problems it should be unique.

5) The number of beams. Lines 8 through the end are repeated this number of times with exceptions noted below.

6) Two items that must be here since they are read, but neither is used.

7) This line must contain the string “MULTIPLE” if and only if the data represent a multi-beam case. Otherwise the line may contain anything else.

8) The receiver type for the current beam is put here, along with keywords describing further receiver information. The keywords that are searched for are:
   `BINARY` - this line is followed by two lines, the first must contain the name of a binary file that contains beam pattern data and the second has a label for that file; lines 9-13 must not be in the file for the current beam
   `ASCII` - this line is followed by two lines, the first must contain the name of an ASCII file that contains beam pattern data and the second has a label for that file; lines 9-13 must not be in the file for the current beam
   `OMNI` - this line must be followed by a line containing a self-noise value in dB/1 Hz; lines 9-15 must not be in the file for the current beam
   `VERTICAL` - indicates a vertical array, lines 9-15 follow
   `HORIZONTAL` - indicates a horizontal array, lines 9-15 follow
   `HORIZONTAL` and `DIPOLE` - indicates a horizontal dipole array, lines 9-15 follow

9) This string describes the weighting method to use. This string may contain:
   `FILE:xxx` - **xxx** is the name of a file from which weights are read
   `HANNING`
   `HAMMING`
   `RCOSINE`

```
TAYLOR
TRIANGULAR
SQUARE
```

All but the "`FILE:`" option cause **INTEGRATE** to create the appropriate set of weights for the method named.

10) Two values will be read, but the first, a weighting parameter, is only used if the previous line has "`RCOSINE`". The second parameter is the minimum sensitivity in dB.

11) This line contains the number of elements in the receiver array and the distance between them in m. The assumption is that the receivers are evenly spaced.

12) This line contains the angle of tilt from the horizontal.

13) This line has the receivers' bandwidth (Hz).

14) This string is the beam's label. However if the string "`OFF`" is in the line the program treats the beam as if it is off and will do no processing for it.

15) This line has the steering angle (degrees) and self-noise (dB/1Hz)

3.4.3 Radials File

This file is named on line 4 of the **INTEGRATE** primary input file and line 3 of the **CMBRAD** primary input file, and both names must be identical. This file presents the information pertaining to all the radials being used and must be constructed by the user before **BREVER** can be run.

BREVER reads this file as well to verify a number of parameters, but this is only done as part of its setup routine.

The data in the file is read via the use of key text strings. That is, certain strings must occur on a line for the line to be processed by the programs that use the file. The sample radials file only contains the lines with the appropriate key strings, which are case sensitive and must appear in the order shown. As well, all the data to be read must appear before the key string.

Following is a line-numbered sample radials file.

Listing 10. Sample Radials File

```
1) 43.2863 62.7748 50.0 341.0 10.0  50.0 (Transm: L/L, dep, hdg, spd)
2) 2                           (Number of Radials)
```

```
3) 1   510769  4848749  69.2 (RadialNo EndPtEast EndPtN Bear)
4) 2   510769  4848749 122.0 (RadialNo EndPtEast EndPtN Bear)
```

By line this file contains:

1) The first key string is "(Transm", which indicates a line of transmitter related information. The data on this line are:
   - transmitter's latitude, positive for degrees N (decimal degrees)
   - transmitter's longitude, positive for degrees W (decimal degrees)
   - transmitter's depth (m)
   - transmitter's heading (ºT)
   - transmitter's speed (kts)
   - receiver's depth (m)

2) The second key string is "`Number of Radials`", and the value preceding the string is that value, which must be 1 or more.

3) The third key string is "`(RadialNo`", which indicates radial related data. The data on this line are:
   - the radial number, which may help the user keep track of how many radials have been entered
   - end position of the radial as a UTM northing value
   - end position of the radial as a UTM easting value
   - radial's heading (ºT)

   Both **INTEGRATE** and **CMBRAD** read all the items on this line, but neither program uses the first three values. Only the radial's headings are used. Therefore a user may enter any numerical value at all in the first three locations.

   The value on the "`Number of Radials`" line dictates how many "`(RadialNo`" lines are in the file.

3.4.4 Radial Description File(s)

These files are named in the **MONOGO** primary input files and are created by **BREVER** under those names. These contain, among other items, information read from data line 19 of the **BREVER** primary input file (Section 3.1). Since the files contain environment description data and their names usually end with "**.des**", they are referred to as **.des** files.

This section only discusses the data line 19 contents since everything else in the **.des** files is controlled from other sources.

Data lines 19 must contain one item in particular and may contain any of three optional ones. The type of data on each line is indicated by a key text string. None of key text strings are case sensitive but the three optional ones all have one, and only one, space character between two non-blank strings.

The compulsory key string is "GRADIENT", and to run **BREVER** properly it must appear in a line with the format "GRADIENT = dpfile", where **dpfile** is the name of the depth profile file for that radial (Section 3.2.2).

Files may not be named **FLAT** or **SLOPE** in any case combination (i.e. **Flat**, **flat**, **flAt**, etc. are all forbidden) since **MONOGO** uses those strings as instructions incompatible with **BREVER's** operation.

The three optional key text strings are:

"SURFACE SCATTERING" - Without this key string the default Lambert surface scattering model is used. Its used may by ensured by following the key string with "= LAMBERT".

Alternatively, the Chapman-Harris model may be specified by following the key string with "= CHAP".

It is not clear what **MONOGO** will do if any other entry is used.

"SEA STATE" - This must be followed by "= ss", where "ss" is a sea state value. **MONOGO** converts this value to a wind speed.

"WIND SPEED" - This must be followed by "= ws", where "ws" is a wind speed in knots.

If both or multiple copies of "SEA STATE" and "WIND SPEED" are included in the data lines for a specific radial, the last one will be used.

4. BREVER Output

BREVER will always produce two forms of output and may produce up to four others, depending on a user's requests. Of the output that is produced, one stream is to the screen (and so can be redirected to a file) and all others are to named output files.

This section describes these outputs.

4.1 Screen Output

BREVER prints three type of information to the screen:

- error messages
- status messages
- results

Error messages are written if **BREVER** discovers a problem with any of the input during the startup check. All errors are reported as soon as they are found. Depending on the error, **BREVER** may continue, if only to continue checking the input. Regardless, if any error is found, as soon as the input check is finished (or sooner, depending on the severity of the error) the program exits.

It may happen that all the inputs check out but a SWAMI program encounters an error condition. If this happens **BREVER** will pick up on it, write an error message pointing to the appropriate program's **.log** file, and exit. All error messages were designed to be as informative as possible.

Status messages are simply messages written to indicate that the program is still running as expected. The first ones reiterate the name of the **BREVER** primary input file, and after the input data have been checked out okay, give that information. Following that, messages are written from within processing loops. The simplex initialization program RI_SIMPLEX_INIT writes out the number of each dimension as it is being filled, and the temperature and downhill simplex loops both write out the current and total loop count every time a loop is started more than 5 seconds after the previous one. The timer is there to prevent loops that finish very quickly from blurring past on the screen. Status messages also inform the user when the loops exit and the reason for leaving them.

Results are written to the screen every time they have been found. These data are also written to the main output file, but are put to the screen in case a user wants to make note of them. If a **BREVER** run is being made with more than one value for the points/radial, the results from one run will immediately be followed by the status messages from the next.

4.2 File Output

4.2.1 Main Output File

The main output file, named in data line 23 of the **BREVER** primary input file, is always created. A sample is presented in the following listing.

Listing 11. Sample Main Output File

```
 Data Analysis Run 03

 ASSA Parameters:
  Initial temperature:          1.00000
  Temperature scaling factor:   0.990000
  Number of temp. reductions:   30
  Pert/temp multiplier:         5
  No. downhill simplex steps:   500
  No. of averaging data sets:   20000
  Perturbations/temperature:    5
  Perturbation value code:       1 - factor * mean of prev. 20000 pert.
values
  Pert. value code multiplier:  3.00000
  SSA convergence value:        0.01000000
  Downhill simplex conv. value: 1.00000e-04

 Target Parameters:
  Points/radial:                fixed at 6
  Density:                      will vary from 1.50000 through 2.50000
  Compressional sound speed:    will vary from 1700.00 through 1900.00
  Compressional attenuation:    will vary from 0.150000 through 0.250000
  Scattering strength:          will vary from -35.0000 through -20.0000
  Depth uncertainty:            5.00000 m on interpolated depths

 Radial information:
  Length:                      60.2000
  Number of radials:           1
  Bearings:                       71.0000

 Source parameters:
  Latitude:                    43.2863
  Longitude:                   62.7748
  Depth:                       50.0000
  Heading:                     341.000
  Speed:                       10.0000

 Receiver Parameters:
  Latitude:                    43.2863
  Longitude:                   62.7748
  Depth:                       50.0000
  Heading:                     341.000

 SWAMI program input filenames:
  INTEGRATE:                   integ.input
  PMODES:                      pmodes.input
  MONOGO:                      monogo.input1
  CMBRAD:                      cmbrad.input

 Measured data filename:       test_c2_beam17.dat
 No. of measured data values:  158

 ***************************************************************************

 6 points/radial
 12.0400 km between radial points

 The SSA convergence value was not reached before the temperature loop
    ran to its limit of 30 runs.
 The last convergence ratio was 0.19111301
    and E min/max were 23.076070 / 27.952149

 Reached the downhill convergence value with a ratio of 9.5207348e-05
    and E min/max of 21.005782 / 21.007782
 356 DHS runs were needed

 Total number of SWAMI runs is         1472
 Minimum E              =        21.005782
 Minimum E/radial point =        3.5009637
 Best parameter estimates are:

                          Compressional  Compressional  Scattering
 Radial  Point   Density   Sound Speed    Attenuation    Strength    Depth
 -------------------------------------------------------------------------
     Centre       1.8528     1791.523       0.18387      -26.3835    95.40

    1      2      1.9552     1800.084       0.21195      -26.6694    90.26
    1      3      1.9760     1822.601       0.19264      -26.6616    91.42
    1      4      2.0206     1799.080       0.20336      -27.1464    87.65
    1      5      2.0136     1789.349       0.20516      -27.0981   103.72
    1      6      2.0376     1788.587       0.19900      -26.6129    97.85
```

As can be seen above, the first thing written to the main output file is a reiteration of the input data. Following the input data is a line of asterisks. This is followed by the results for a specific number of points/radial.

These results consist of the energy statistics and the values that **BREVER** settled on for the various parameters that it was solving for.

The above main output file was produced for a 1 radial run, however if the **BREVER** primary input file had specified 3 radials rather than one, the end of the file would have looked like the following sample listing.

Listing 12. Sample Multi-radial Main Output File End Results

```
                         Compressional  Compressional  Scattering
Radial  Point   Density   Sound Speed    Attenuation    Strength     Depth
--------------------------------------------------------------------------
    Centre       1.8528     1791.523       0.18387      -26.3835     95.40

   1      2      1.9552     1800.084       0.21195      -26.6694     90.26
   1      3      1.9760     1822.601       0.19264      -26.6616     91.42
   1      4      2.0206     1799.080       0.20336      -27.1464     87.65
   1      5      2.0136     1789.349       0.20516      -27.0981    103.72
   1      6      2.0376     1788.587       0.19900      -26.6129     97.85

   2      2      1.9552     1800.084       0.21195      -26.6694     90.26
   2      3      1.9760     1822.601       0.19264      -26.6616     91.42
   2      4      2.0206     1799.080       0.20336      -27.1464     87.65
   2      5      2.0136     1789.349       0.20516      -27.0981    103.72
   2      6      2.0376     1788.587       0.19900      -26.6129     97.85

   3      2      1.9552     1800.084       0.21195      -26.6694     90.26
   3      3      1.9760     1822.601       0.19264      -26.6616     91.42
   3      4      2.0206     1799.080       0.20336      -27.1464     87.65
   3      5      2.0136     1789.349       0.20516      -27.0981    103.72
   3      6      2.0376     1788.587       0.19900      -26.6129     97.85
```

All the results from the various radials are presented, but the centre point, common to all the radials, is only written once.

If **BREVER** had been instructed to run a case with more than one value for the number of points/radial, then the part of the output from the line of asterisks to the end of the file would have been repeated once for each points/radial value.

After all the points/radial solutions were written, they would be followed by text telling the user which number of points/radial had the lowest overall energy, and which had the lowest average energy per radial point.

4.2.2 Measured Data SAPLOT File(s)

Data line 24 of the **BREVER** primary input file allows the user to instruct **BREVER** to write to a **SAPLOT** format ASCII file the measured time series reverberation data used as a basis for determining the goodness of fit of modelled data. The data in this file have been extracted from the program's measured data input file, specified on data line 20, in the manner described in Section 3.2.2.

Data line 25 of the **BREVER** primary input file allows the user to instruct **BREVER** to write to a **SAPLOT** format ASCII file the differences in reverberation between consecutive lines in the measured data being used for model comparison. These are the data that are actually used for testing the fit of the model. These data are intended to stand in for the slopes along the “curve” of the time series. These differences can be converted to slopes by dividing the differences by the size of the time step between points. However, since the time step size is the same for both measured and modelled data simple differences are functionally equivalent to the actual slopes.

Section 3.1 discusses data lines 24 and 25 of the **BREVER** primary input file

4.2.3 Best Fit Modelled Data SAPLOT File(s)

Time series of both modelled data and consecutive differences in the modelled data can be written to ASCII files if the user desires by way of data lines 26 and 27 of the **BREVER** primary input file. The data in these files were produced by running the SWAMI programs using, as input, the best fit results presented in **BREVER’s** main output file.

If these files are being produced, they are made for each points/radial value analysed by **BREVER**. Using **SAPLOT** to plot these data allows a user to compare the best fit results with the measured data.

Section 3.1 discusses data lines 26 and 27 of the **BREVER** primary input file

5. Performing a Reverberation Inversion

5.1 Program Locations

The required programs are in several locations on *Grebe*.

- the programs that comprise **BREVER** are located in the directory ***/home/calnan/projects/RevInv/code***
- the SWAMI program executables are located in the directory ***/home/calnan/projects/RevInv/dmos/bin***
- **SAPLOT** is located in ***/home/calnan/projects/idllib/SAPLOT***

As well, directory ***/home/calnan/projects/HPA/code*** has two IDL programs that make it simple to examine the measured data file:

- **READ_HDR** (main file **read_hdr.pro**) reads the header of a DREA .DAT or DREA .DAT32 file and writes it to the screen. To save this data in a file a user can either start IDL and redirect its output to a file, or cut and paste the text written to the screen.
- **VIEW_DAT** (main file **view_dat.pro**) produces screen, and optionally PostScript file, plots of user selected portions of a DREA .DAT or DREA .DAT32 file.

It may be noted that some IDL files of the same name exist in more than one directory. This is particularly true of directories ***/home/calnan/projects/RevInv/code*** and ***/home/calnan/projects/HPA/code***. A user must not assume that the files are identical. The HPA code was written first and then adopted for use with **BREVER**. Some of the programs have had their capabilities enhanced and so can not be blithely exchanged back and forth. It was hoped that the HPA analysis routines that call the improved routines could be modified so as to make use of the extra capabilities, but there was insufficient time for this. As well, the task was outside the scope of the project.

5.2 Running the Programs

It is a simple task for a user with an account on *Grebe* to run **BREVER** on that computer. Data line 16 of the **BREVER** primary input file must have the directory of the SWAMI programs and the directory that has the **BREVER** routines must be added to the user's IDL path. This can be done from within IDL via the command:

```
   !PATH = !PATH + ';/home/calnan/projects/RevInv/code'
```

A similar tack must be taken in order to use **SAPLOT** or the HPA routines. However, modifying the IDL path should not be done for both **BREVER** and the

HPA routines in the same IDL session since the wrong routines will be accessed at some point.

Should a user desire, the necessary files may be copied to a different directory on *Grebe* or to another computer altogether. As long as the SWAMI programs' directory is correct in the **BREVER** primary input file and the IDL path contains the directory in which the files reside there should be no problem in running the routines.

5.3 Running BREVER

To run **BREVER** a user must first start IDL and make sure that the directory containing its code is in the path. (Entering "`print, !path`" displays the directories that are currently in the IDL path.)

By entering "`brever`" at the IDL prompt a user starts the program. At this point the program will open a file selection GUI that requires the user to select a **BREVER** primary input file. Once the file has been selected the program runs to completion without any further input required from the user unless errors are encountered.

Alternatively a user can enter "`brever, 'fname'`" at the IDL prompt, where **fname** is the name of a **BREVER** primary input file. Once again, after this has been done the program runs to completion without any further input required from the user unless errors are encountered.

This second option allows a user to write an IDL routine that will perform multiple **BREVER** runs. For example, a routine named **runall.pro** can be written to contain:

```
pro runall

brever, 'infile1'
brever, 'infile2'
brever, 'infile3'
brever, 'infile4'

end
```

Then, by entering "`runall`" at the IDL prompt the four cases may be run one after the other. If a user does this, care must be taken to specify different output names for the main **BREVER** output file, at least. Otherwise a run's results will overwrite those from a previous run.

6. Suggestions for Further Work

This section has some suggestions for further work that may be performed relating to **BREVER**. The first four suggestions involve expanding the **BREVER** code, but the last one involves potential changes to the SWAMI code.

Suggestion 1: Expand READ_PMODES_IN

The function READ_PMODES_IN can't handle the input **PMODES** options "T" and "O". Currently if either of these options is present the function reports the fact and exits. It would not be a major task to modify the function to handle the options properly.

Suggestion 2: Allow random choice of points/radial

Currently **BREVER** reads a minimum and maximum number of points/radial (**BREVER** primary input file data line 6) and performs its analysis by stepping from the former to the latter one at a time. That is, if the minimum is 5 and the maximum is 8, then **BREVER** will run analyses for 5, 6, 7, and 8 points/radial. A method should be implemented to allow "randomly" selected numbers of points. A user should also be able to repeat certain points/radial values in order to be able to more easily compare results from different runs using the same data. This process could be used to provide error estimates for the calculated results.

Suggestion 3: Allow the fixing of specific parameters

Currently **BREVER** solves for either all or none of the values for a specific parameter. For example a user can only tell the program to solve for the scattering strength at all of the radial points or at none of them. As well, if the option was for all points, then the same limits are used for all the points. An investigation could be made to determine if it would be useful to fix parameters' values at certain radial points and solve them at others. If this would be useful, then a scheme would have to be devised to permit it. The fact that the number of radial points can vary during a **BREVER** run might make this option tricky to implement.

Suggestion 4: Allow multiple fixed values

If a parameter is not to be solved for, then the analysis proceeds by providing all the instances of that parameter with the same value. It might be useful to be able to fix parameters' values individually. The fact that the number of radial points can vary during a **BREVER** run might make this option tricky to implement.

Suggestion 5: MONOGO modification

MONOGO operates by reading an environment description file and using it until a new one is presented to it. This can result in discontinuities in the results if consecutive environments have markedly different parameters. The program could be modified so that it interpolates between environments, thus producing a smoother transition between adjacent areas.

Appendix A BREVER Program Files

This appendix lists the program files that comprise **BREVER**. The files are listed in two tables:

- Table 1 contains the major **BREVER** routines and their purpose; these files are listed in their usual order of appearance.
- Table 2 contains utility functions and their purposes; these files are listed roughly in order of appearance. The majority of these functions were originally written for a project involving HPA data analysis although a few are new to this project. Several of the HPA-originating functions have been modified, usually to expand their capabilities but occasionally to make them more efficient.

Table 1. Major BREVER routines

IDL File	Purpose
brever.pro	Main program
brevercom.pro	"Include" file that is a common area, and so contains the names of most of the variables used in **BREVER**. This is used by most major functions.
read_brever_in.pro	Called by **BREVER** to read its main input file and some other input files.
read_integ_in.pro	Called by READ_BREVER_IN to read the input file for the SWAMI program **INTEGRATE**.
read_radial.pro	Called by READ_BREVER_IN to read a SWAMI radials file.
read_cmbrad_in.pro	Called by READ_BREVER_IN to read the input file for the SWAMI program **CMBRAD**.
read_desinfo.pro	Called by READ_BREVER_IN to read a SWAMI radial description (**.des**) file.
ver_monogo_in.pro	Called by **BREVER** to verify the contents of the input file for the SWAMI program **MONOGO**.
read_pmodes_in.pro	Called by **BREVER** to verify the contents of the primary input file for the SWAMI program **PMODES**.
read_temp_env.pro	Called by **BREVER** to read the template **.env** file. This is used as a basis for the **.env** files read by the SWAMI program **PMODES**.
read_meas_data.pro	Called by **BREVER** to read the measured data file, which is either a DREA .DAT or DREA .DAT32 format file.
read_depth_profs.pro	Called by **BREVER** to read the depth profile data from the files listed in the main **BREVER** input file.

IDL File	Purpose
read_stderr.pro	Called by **BREVER** to read the standard error file produced by the SWAMI programs **INTEGRATE** and by COST to read the files produced by the SWAMI programs **PMODES**, **MONOGO**, and **CMBRAD**.
write_out_hdr.pro	Called by **BREVER** to write out the header of the output file. The header consists mainly of the input file's data.
ppr_setup.pro	Called by **BREVER** to perform setup functions related to the selection of a new points/radial value.
cost.pro	Called by **BREVER**, RI_SIMPLEX_INIT, RI_SIMPLEX, and RI_AMOEBA after a series of test target parameter values have been chosen in order to run SWAMI and calculate *ENERGY*.
pmodes_setup.pro	Called by COST to prepare files for a run of the SWAMI program **PMODES**.
ri_simplex_init.pro	Called by **BREVER** to fill the simplex model space with test target parameter values and call COST for each set of values. This is a version of the general purpose function SIMPLEX_INIT that has been modified for reverberation inversion, hence the "RI_" prepended to the file and function's names.
ri_simplex.pro	Called by **BREVER** to perform the simplex operation over the entire simplex model space. See the notes on **ri_simplex_init.pro** for an explanation of the "**ri_**".
ri_amoeba.pro	Called by RI_SIMPLEX to recalculate dimensional parameter values to obtain smaller *ENERGY* values. See the notes on **ri_simplex_init.pro** for an explanation of the "**ri_**"

Table 2. Utility Functions used by BREVER

IDL Utility Function File	Purpose
has_data.pro	Called by READ_BREVER_IN and READ_DESINFO to see if a line read form an input file has data.
get_header.pro	Related to reading in .DAT and .DAT32 format files: called by READ_BREVER_IN and READ_DAT_DAT32 to read the header of the measured data file.
ba2str.pro	Related to reading in .DAT and .DAT32 format files: called by other functions to convert a byte array to a string so it can be printed out.

IDL Utility Function File	Purpose
make_dat_hdr.pro	Related to reading in .DAT and .DAT32 format files: called by other functions to create a structure to hold the file's header data.
read_head.pro	Related to reading in .DAT and .DAT32 format files: called by other functions to read in the file's header.
chk_hbs.pro	Related to reading in .DAT and .DAT32 format files: called by other functions to check if the header data need to be byte-swapped.
validate_hdr.pro	Related to reading in .DAT and .DAT32 format files: called by other functions to validate the header's data.
timestamp2jd.pro	Called by READ_BREVER_IN to convert an ISO 8601 format timestamp to a Julian date.
decode_cs.pro	Called by READ_BREVER_IN to decode the list of channels to process as specified in the input file.
int_check.pro	Called by a number of routines to see if a character string contains an integer.
reality_check.pro	Called by a number of routines to see if a character string contains an real number.
read_dat_dat32.pro	Called by READ_MEAS_DATA to read in the measured data.
get_chans.pro	Called by READ_DAT_DAT32 to see which channels the user wants to read in; not used by **BREVER** due to options choice.
datts2jd.pro	Called by READ_MEAS_DATA to convert the .DAT or .DAT32 date/time data to a Julian date.

Appendix B SWAMI Components

BREVER requires four SWAMI programs in order to run: **INTEGRATE**, **PMODES**, **MONOGO**, and **CMBRAD**. These programs must exist as files named, respectively, **INTEGRATE**, **PMODES**, **MONOGO**, and **CMBRAD**, and they must all be located in the same directory on the computer. If these conditions are not met **BREVER** will notice this error, report on it, and exit.

This section provides information on these four programs, including brief notes on what they do, the FORTRAN files used to produce the executables, their inputs, and their outputs.

B.1 SWAMI Programs and Program Files

Table 3 contains the names of the SWAMI executables (complete programs in themselves) needed by **BREVER** and their purposes. This table also contains the names of the individual FORTRAN files that are required to produce the various SWAMI executables, but without comments on each. It may be noted that some FORTRAN files are common to more than one SWAMI program.

Table 3. SWAMI programs used by BREVER

SWAMI Program	Purpose
INTEGRATE	Performs beam pattern integration in order to generate a system file and beam patterns. This only has to be run once for each **BREVER** run. The files that comprise this program are: **bpat_io.f** **bpat_libs.f** **input.f** **INTEGRATEBP.f** **integrate.f** **intvars.include** **open_file.f** **params.include** **util.f**
PMODES	Computes **MONOGO** input mode files for a set of environments. This must be run for each new set of target parameter values. The files that comprise this program are:

SWAMI Program	Purpose
	input.f **monvar.include** **open_file.f** **orcmod.f** **params.include** **parse.include** **PMODES.f** **prplos.f** **swami_io.f** **util.f**
MONOGO	Generates reverberation predictions for one radial. This must be run once for each radial of each new set of target parameter values. The files that comprise this program are: **input.f** **MONOGO.f** **monvar.include** **ogomonRD.f** **open_file.f** **params.include** **parse.include** **swami_cmp.f** **swami_io.f** **util.f**
CMBRAD	For each new set of target parameter values this combines predicted reverberations for all radials, thus producing combined reverberation files. When there is only one radial this must still be run since its output is in a different format than the input files. This must be run for each new set of target parameter values. The files that comprise this program are: **CMBRAD.f** **input.f** **monvar.include** **open_file.f** **params.include** **parse.include** **swami_cmp.f** **swami_io.f** **util.f**

B.2 Running SWAMI

In order to run **BREVER** a user must produce the input files needed by the SWAMI programs, as described in Sections 3.3 and 3.4. Once that has been done, **BREVER** takes care of running the SWAMI programs without any further input from the user. However, it was deemed useful to include a section that summarizes what the SWAMI programs actually do. This might assist in understanding the overall program and could come in handy in the debugging process should **BREVER** not run properly.

The SWAMI programs are all written so they read their input interactively. However this can be circumvented by creating the SWAMI primary input files, which contain the required data, and running the programs using these data files redirected into the programs' system input. This may be done by entering, for example:

```
PMODES < pmode.input
```

The primary input files contain the names or parts thereof of other input files, output files, and various numerical and text input parameters. All input files that are mentioned below and not noted as having been created by a specific program must already exist before **BREVER** can be run. Otherwise the SWAMI programs will fail when the time comes to access these files.

On a program-by-program basis input and output files for each program are:

INTEGRATE Input

- ASCII transmitter information file that contains a filename root (Section 3.4.1).
- ASCII receiver information file that contains a filename root (Section 3.4.2).
- Radial information file with the location of the centre of the radials, the number of radials, and their bearings (Section 3.4.3).
- The primary input file also contains a text string that is used as the base for output filenames, e.g. "`da01`".

Output

- One ASCII system file is created per radial. The files' names are based on the filename base and the bearing of the radial. For example, a radial with a bearing of 87.5º and a filename base of "`da01`" would result in a filename of **da01_0875.sys**. The system files are used by **MONOGO**.
- Transmitter files based on the filename root in the transmitter input file will be created. For example, if

		the root is “TOM” then files named and 1 radial at 90.0º is listed in the radial information file, then the files **TOMB.dat**, **TOM_ROM_001_0900B.dat**, **TOM_ROM_001_0900V.dat**, **TOM.dat**, **TOM_ROM_001_0900.dat**, and **TOMV.dat** will be created. The only reference to these was found in the scripts that use **PLOTXMGR** to produce graphics. If this is the case then **BREVER** does not require or use these files. Some of these files were needed by **MONOGO**, but **BREVER** doesn’t need them.
		- Receiver files based on the filename root in the transmitter input file will be created. For example, if the root is “ROM” then files named and 1 radial at 90.0º is listed in the radial information file, then the files **ROM_001B.dat**, **ROM_001.dat**, and **ROM_001V.dat** will be created. The only reference to these was found in the scripts that use **PLOTXMGR** to produce graphics. If this is the case then **BREVER** does not require or use these files.
PMODES	Input	- The primary input file specifies how many environmental files (Section 3.2.1) are to be processed, and specifies their names. They contain three of the four target parameters **BREVER** is to solve for: density, compressional speed, and compressional attenuation. This means that **PMODES** must be run for each change in any of these parameters during the inversion process. Environmental files are ASCII and must be created before **PMODES** can be run.
	Output	- One binary mode file is always created for each input environment file and the names of these files are specified in the primary input file. The output files are used as input to **MONOGO**.
		- If the primary input file specifies the “T” (transmission loss) processing option, then one ASCII file is created for each input environment file. The names of these files are specified in the primary input file. These files were used in SWAMI testing and are not used in reverberation inversion.
MONOGO	Input	- Environmental description file (Section 3.4.4 for a partial description; SWAMI documentation for a full description) that names binary mode files created by **PMODES**.

- System file as created by **INTEGRATE**.

Output

- One ASCII no-group velocity reverberation file for each radial. If, for example, the system file is named **da01_0900.sys**, then this file will be named **da01_0900_ngv.rev**. The no-group velocity reverberation file is not used in reverberation inversion.

- One ASCII reverberation file including group velocity for each radial. If, for example, the system file is named **da01_0900.sys**, then this file will be named **da01_0900_gv.rev**. These files are used by **CMBRAD**.

CMBRAD Input

- The radial information file used earlier by **INTEGRATE** (Section 3.4.3).

- The primary input file also contains a text string that is used as the base for output filenames. This string must be the same as the root used by **INTEGRATE** to create the files used by **MONOGO**.

- The "**_gv.rev**" reverberation file created by **MONOGO**.

Output

- A combined ASCII reverberation file including group velocity. If, for example, the filename base is "`da01`" and the **MONOGO**-produced reverberation file is named **da01_0900_gv.rev**, then this file will be named **da01_gv.rev**. It is basically a reformatted version of the input reverberation file. If **MONOGO** was used to produce more than one reverberation file (with each having a different bearing for each radial) **CMBRAD** produces results that are calculated by combining the results of all radials.

BREVER runs the SWAMI programs by means of the IDL `SPAWN` command. That command runs the program passed to it and causes the program issuing the command to wait until the spawned program runs to completion before continuing.

It was mentioned above that **BREVER** runs SWAMI programs by redirecting a primary input file to the SWAMI program, thus allowing the primary input file to be used as interactive input.

This is true but incomplete. The complete command associated with running a SWAMI program from within **BREVER** for **PMODES** is:

```
PMODES < pmodes.input > PMODES.log 2> PMODES.err
```

The name of the input file is one of the parameters passed to **BREVER**; in this example "pmodes.input" was used. The names of the **.log** and **.err** file are produced by **BREVER** as displayed, however.

The **.log** file stores any informational messages written by the program and the **.err** file stores the program's STOP message, printed when the program ends operations.

The SWAMI programs have all been modified so that a successful runs ends with the program printing the STOP message "OKAY". However if an error is encountered during a program's run the information associated with this error is written to the **.log** file and the STOP message "ERROR" is printed. **BREVER** has been written so that after each run of a SWAMI program, that program's **.err** file is examined. If the file contains "OKAY" then **BREVER** continues, but if it contains "ERROR" a message warning the user about this is written to the screen and **BREVER** stops.

The reason for making the program come to a complete stop in the case of an error is simple: there should be no errors during a SWAMI run. The fact that there is one means that the afflicted SWAMI program can't operate as it should, in which case there is no use in running **BREVER**. The usual cause of an error is that there is something wrong in one of the programs' input files (**BREVER** only checked the files for consistency, not for validity).

The next most common cause of an error is that an array in a program is not large enough to contain the data the program wishes to put in it. This is normally corrected by increasing the size of a program parameter in one or more of the SWAMI "include" files and remaking the executable. It must be remembered that if the source code for the different SWAMI programs are segregated and if a file that is used for more than one program is edited, then all versions of the file should be modified in the same way.

If executables are remade the user must remember that the executable must be given the correct name and put into the same directory as its predecessor.

Bibliography

Calnan, C. (2002). *DRDC Atlantic IDL Library Modernization Project* (DRDC Atlantic CR 2002-187, November 2002)

Calnan, C. (2003). *Adaptive Simplex Simulated Annealing Function Minimization Program* (DRDC Atlantic CR 2003-033, publication pending)

Dosso, Stan E., Wilmut, Michael J., and Lapinski, Anna-Liesa S. (2001). "An Adaptive-Hybrid Algorithm for Geoacoustic Inversion," *IEEE Journal of Oceanic Engineering*, **26** (3), July 2001.

Fallat, Mark R., and Dosso, Stan E. (1999). "Geoacoustic Inversion via Local, Global, and Hybrid Algorithms," *J. Acoust. Soc. Am.*, **105** (6), June 1999.

Initialisms and Acronyms

ASCII	American Standard Code for Information Interchange
DRDC	Defence Research and Development Canada
HPA	Horizontal Projector Array

Distribution List

Internal Distribution

2	DRDC Atlantic LIBRARY FILE COPIES
4	DRDC Atlantic LIBRARY (SPARES)
3	J. Theriault
1	S. Pecknold
1	M. Boutin
1	D. Ellis
1	P. Hines
1	M. Lefrancois
1	B. Roger

External Distribution

1	AUTHOR
1	LCdr W. Renauld, SSO METOC, MARLANT CFB Halifax
1	D. McGaughey, Dept Electrical and Computer Engineering Royal Military College of Canada Box 17000, Station Forces Kingston, Ontario K7K 7B4
1	R. Marsden, Dept Electrical and Computer Engineering Royal Military College of Canada Box 17000, Station Forces Kingston, Ontario K7K 7B4
1	M. Prior, NATO Undersea Research Center
1	S. Taylor Defence Science and Technology Organization
1	C. Burt Defence Science and Technology Laboratory
1	NDHQ/ CRAD/ DRDKIM 3

Total 23 copies

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (the name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g. Establishment sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Xwave
36 Solutions Drive, Halifax, NS B3S 1N2

2. SECURITY CLASSIFICATION (overall security classification of the document including special warning terms if applicable).

UNCLASSIFIED

3. TITLE (the complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S,C,R or U) in parentheses after the title).

Reverberation Inversion Using SWAMI

4. AUTHORS (Last name, first name, middle initial. If military, show rank, e.g. Doe, Maj. John E.)

C. Calnan

5. DATE OF PUBLICATION (month and year of publication of document)

December 2004

6a. NO. OF PAGES (total containing information Include Annexes, Appendices, etc).

61 (approx.)

6b. NO. OF REFS (total cited in document)

4

7. DESCRIPTIVE NOTES (the category of the document, e.g. technical report, technical note or memorandum. If appropriate, enter the type of report, e.g. interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered).

CONTRACT REPORT

8. SPONSORING ACTIVITY (the name of the department project office or laboratory sponsoring the research and development. Include address).

Defence Research & Development Canada – Atlantic
PO Box 1012
Dartmouth, NS, Canada B2Y 3Z7

9a. PROJECT OR GRANT NO. (if appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant).

9b. CONTRACT NO. (if appropriate, the applicable number under which the document was written).

W7707-03-2310

10a ORIGINATOR'S DOCUMENT NUMBER (the official document number by which the document is identified by the originating activity. This number must be unique to this document.)

10b OTHER DOCUMENT NOs. (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

DRDC Atlantic CR 2004-113

11. DOCUMENT AVAILABILITY (any limitations on further dissemination of the document, other than those imposed by security classification)

(X) Unlimited distribution
(   ) Defence departments and defence contractors; further distribution only as approved
(   ) Defence departments and Canadian defence contractors; further distribution only as approved
(   ) Government departments and agencies; further distribution only as approved
(   ) Defence departments; further distribution only as approved
(   ) Other (please specify):

12. DOCUMENT ANNOUNCEMENT (any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected).

13. ABSTRACT (a brief and factual summary of the document. It may also appear elsewhere in the body of the document itself. It is highly desirable that the abstract of classified documents be unclassified. Each paragraph of the abstract shall begin with an indication of the security classification of the information in the paragraph (unless the document itself is unclassified) represented as (S), (C), (R), or (U). It is not necessary to include here abstracts in both official languages unless the text is bilingual).

The SWAMI suite of programs in use at DRDC Atlantic enables a user to produce modelled reverberation time series data based on a series of input parameters that describe the acoustic parameters of an area of a seabed. The modelled data can be compared to recorded reverberation data and a quantitative measurement can be made that essentially tests the goodness of fit of the modelled data to the measured values.

A contract was issued to design a program that would iteratively vary the values of the geoacoustic parameters input to the SWAMI programs and then run them, each time comparing the model results to a set of measured data. In this manner the program would arrive at a set of values that would produce model results that were as close to the measured data as SWAMI could provide.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS (technically meaningful terms or short phrases that characterize a document and could be helpful in cataloguing the document. They should be selected so that no security classification is required. Identifiers, such as equipment model designation, trade name, military project code name, geographic location may also be included. If possible keywords should be selected from a published thesaurus. e.g. Thesaurus of Engineering and Scientific Terms (TEST) and that thesaurus-identified. If it not possible to select indexing terms which are Unclassified, the classification of each should be indicated as with the title).

Underwater Acoustics
Data Analysis
Reverberation
Acoustic Model
Geoacoustic Inversion
Active Sonar

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

DEFENCE DÉFENSE

www.drdc-rddc.gc.ca