Thermosalinograph Data Processing Guide

Procedures, techniques and analytical tools currently (year 2009) used to process thermosalinograph data for the Hawaii Ocean Time-Series (HOT) / Physical Oceanography group (PO)

Updated by Paul Lethaby and Christin Shacat (compiled from materials made by Lal Ratnapala, Don Wright, Fernando Santiago-Mandujano, Sharon DeCarlo, and Daniel S. Fitzgerald)

Document located on PC Kaimana in *D:\Mark_share$\Archived Staff Folders\DanF\tsgprocguide.doc*
Document located on UNIX
Processing programs are located on UNIX systems in: */home/malino5/matlab5/tsg*

1 Overview

A continuous record of near-surface salinity and temperature data at the location of the ship is collected during every typical HOT cruise using a system of conductivity and temperature sensors installed into the uncontaminated seawater supply of the research vessel. The data are processed with a series of Matlab programs, and the thermosalinograph record is compared with corresponding CTD data and with salinity samples collected from the seawater supply during the cruise to recalibrate the thermosalinograph conductivity sensor and recalculate final salinities.

Thermosalinograph (TSG) data processing involves four main steps: obtaining the raw data (Section 2), creating the computer files needed by the processing software (Section 3), preliminary analysis of the data (Section 4), and final analysis (Section 5). The procedures in this manual should be followed to ensure correct and consistent processing of the HOT TSG data.

1.1 Thermosalinograph System Description

The details of a thermosalinograph system will vary from ship to ship, but each system is essentially composed of a thermosalinograph sensor, which contains both a temperature sensor and a conductivity sensor, in conjunction with a remote temperature sensor located at the entrance of the seawater intake pipe. Salinities are calculated based on the temperature and conductivity data at the thermosalinograph sensor (as well as the internal pressure of the intake pump), and near-surface seawater temperatures are reported from the remote temperature sensor. A sampling area is situated near the thermosalinograph, and bottle samples of the uncontaminated seawater are collected at 4-hour intervals. The bottle sample salinities are later measured and compared against the thermosalinograph data from the same time, and are used to recalibrate the conductivity portion of the thermosalinograph sensor. When working with thermosalinograph data, all times are reported in GMT (+10 hrs. from HST). See Fernando and/or Paul if you have questions regarding GMT.

The thermosalinograph systems aboard the UH research vessels are maintained by OTG. Contact Paul or Fernando with questions about the thermosalinograph system.

2 Obtaining the Raw Thermosalinograph Data

To obtain the raw thermosalinograph data files, first, create a working directory, then transfer the raw thermosalinograph files into the new directory.

UNIX terminal prompts (what is supposed to be typed into the terminal) are indicated by the `%` symbol and text in `courier` font. Command responses (standard output = stdout) will be in `courier` but without the `%` sybmol. Matlab prompts (what is supposed to be typed into a terminal where Matlab has been started) are indicated by the >> symbol and are *italicized*. Matlab stdout is also *italicized* without the >> symbols and user responses are **bold faced**. Parenthetical comments are in normal text. The symbols ### always denote the current HOT cruise.

2.1 Create the Working and sub-Directories

An appropriately named working directory needs to be created to hold the thermosalinograph files. HOT thermosalinograph data are stored on the UNIX network in a subdirectory within each HOT cruise directory (e.g., HOT-137 thermosalinograph data are located in */home/malino5/hot/137/thermosal*).

To create the thermosal working directory, open a terminal in UNIX and execute the following commands.

```
% cd /home/malino5/hot/###
% mkdir thermosal
% cd thermosal
```

The previous commands change to the appropriate cruise directory, create the thermosal directory and change into the new directory.

The command `pwd` displays the current working directory, and `ls` will list the contents of the directory, if any.

A new Thermosalinograph was installed on the R/V Kilo Moana during HOT-200 in February 2008. The SBE 45, also referred to as the Micro-TSG, sampling at 1 Hz, is currently down-sampled to a 10-second format (the sampling rate of the standard SBE 21) in order to compare the data of the two systems. Eventually, the SBE 21 will be discontinued and the processing scripts will be adjusted to use the 1-sec data as input.

In order to process both systems in the mean time, a sbe21 and a sbe45 directory have to be created. Depending which system is being processed at one time, the processing related files need to be moved from e.g., the sbe21 directory to the thermosal directory. Once the SBE 21 processing is complete, all processing related files need to be moved into the sbe21 directory.

Here is a list of the 208 SBE 21 processing related files. As a safety measure, a copy of the CTD header and keepsalt file are saved in this directory as well.

aloha_time.dat	fig4b_zoom.ps
bot_ctdindex.m	fig5.fig
condcalib.mat	fig5.ps
cr_dat.mat	fig6.fig
cr_day.mat	fig6.ps
ctd4dbdwn.mat	flagtsg.dat
ctdbot.out	hot208_speed.dat
datacmpcond.out	hot208ctd.hdr%
fig1.ps	hot208nav_final.jpg
fig1b_thsl_ctd_salinity.fig	hot208nav_final.ps
fig1b_thsl_ctd_salinity.ps	hot208salt.dat
fig2a.ps	hot208thsl.dat
fig2b.ps	hot208thsl.mat
fig2c.ps	hot208thsl_final.jpg
fig3a_thsl_ctd_temperature.fig	hot208thsl_final.ps
fig3a_thsl_ctd_temperature.ps	jpp.fig
fig3b.fig	jpp.ps
fig3b.ps	keepsalt.dat%
fig4a.fig	procsail.rep
fig4a.ps	salinity.dat
fig4b.fig	shipdata.mat
fig4b.ps	thctd.mat
fig4b_zoom.fig	thsb.mat

Files to remain in the thermosal folder that do NOT need to copied into the sbe folders for HOT-208 were:

208tsgrpt_draft.doc	jpp_detail_21.ps
cmp_sal_flo.fig	jpp_detail_45.ps
cmp_sal_flo.ps	keepsalt.dat
cmp_sal_flo_detail_1.ps	km0902_cgrav.gz
cmp_sal_flo_detail_2.ps	km0902_fluoro_wfix
cmp_sal_flo_detail_3.ps	km0902_sh
cmp_thsl.m	km0902_thsl_wfix
fig_1_dthsl.ps	km0902_uthsl
fig_2_dthsl.ps	km_uthsl_wfix

flo.ps	read_utsg_data.m
flo_drop_detail.fig	hot208ctd.hdr
flo_drop_detail.ps	hot_procsail_u.m
thsl.cal	

Notice that the SBE 45 pre-processing scripts and the TSG comparison script need to be copied from a previous HOT thermosal folder: hot_procsail_u.m, read_utsg_data.m, cmp_thsl.m

2.2 Transfer the Raw Data Files

There are two methods to obtain the raw thermosalinograph data files: transferring via ftp, and copying from DVD-ROM.

2.2.1 File Transfer via FTP

Thermosalinograph data from most HOT cruises are posted at an ftp site (ftp.soest.hawaii.edu) that is maintained by OTG personnel. This data is usually available beginning one week after a cruise.

Use the following steps to ftp the data into the appropriate working directory:

1) Go to the thermosal directory (*/home/malino5/hot/###/thermosal*) that you just created. This is the location where you want the thermosalinograph data to be. For example, for HOT-152 thermosal data type:
   `% cd /home/malino5/hot/152/thermosal`
2) `% ftp ftp.soest.hawaii.edu`
   This connects you to the ftp server, but you need to
3) Login. Use username = **anonymous**
   your password should be your regular login name (including the @soest.hawaii.edu) or your initials @soest for example: **jb@soest**. The server allows only 15-30 seconds to login; therefore a shorter password is recommended.
4) Upon successful login, you receive the following prompt:
   ftp>
   Check the files in this directory:
   ftp> ls
5) Change directories:
   ftp> cd otg
6) Change to the cruise directory, e.g., for HOT-121 data:
   ftp> cd kok2020_HOT121
   Each cruise is given a unique number by OTG, based on the ship name and year. The OTG cruise number is found in the HOT cruise binder, usually on the first page of the cruise log. If the number is not there then check the thermosalinograph log at the back of the binder.

7) Type the following to transfer the data in binary instead of ASCII format:
   ftp> bin
8) Type the following to turn off interactive mode
   ftp> prompt
9) Type the following to get the cruise data (**mget** means: get multiple files):
   ftp> mget kok2020* or type
   ftp> mget *.Z (Z is a capital letter and means the file is compressed)
10) Type the following to end the session
    ftp> bye
11) View the presence of new thermosal files in your directory with the **ls** command. There will typically be eight different files, all beginning with the OTG cruise number (kok$$$$ or km$$$$), followed by one of the following extensions: _fluoro, _fluoro_wfix, _nav, _sh, _thsl, _thsl_wfix, _wind2T5, or _windT5.
12) Make note of the OTG cruise number (kok$$$$ or km$$$$).
13) Contact OTG and inform them that you completed transferring the files so that they can remove the files from their server to save space.
14) Remember that there are some programs that read files that are initially compressed (end in a **.Z**.). These files need to be uncompressed or the programs will not run. Uncompress all your files received from ftp and rename them to all lower case letters. For example:
    ```
    % gunzip KM0724_SH.GZ
    % mv  KM0724_SH km0724_sh
    ```

2.2.2 Directions to Transfer Files from CD-ROM

Occasionally the raw thermosalinograph data files are only available on DVD-ROM format. Usually the DVDs are readable by both the UNIX machines and by the PCs. Simply load the DVD onto a UNIX machine, and then copy the thermosalinograph files into the appropriate working directory (see Section 2.1). If the CD is loaded onto a PC, then use the ftp program "**WS_FTP95**" to transfer the files into the working directory. See Fernando if you are unfamiliar with the use of this ftp program. Remember to uncompress all the thermosalinograph data files after they have been transferred into the working directory (see Step 14 above).

Also get a file called THSL.CAL from OTG. That file includes information on the serial numbers and calibration dates of the thermosalinographs used on the different ships.

3 Create input files for initial processing

The following files need to be created/updated in order to perform the preliminary analysis of the thermosalinograph data:

hot###ctd.hdr
keepsalt.dat

UNIX terminal prompts (what is supposed to be typed into the terminal) are indicated by the `%` symbol and text in `courier` font. Command responses (standard output = stdout) will be in `courier` but without the `%` sybmol. Matlab prompts (what is supposed to be typed into a terminal where Matlab has been started) are indicated by the >> symbol and are *italicized*. Matlab stdout is also *italicized* without the >> symbols and user responses are **bold faced**. Parenthetical comments are in normal text. The symbols **###** always denote the current HOT cruise (e.g., HOT-198). The symbols **$$$$** always denote the current HOT cruise number that is used by OTG (e.g., km0724).

3.1 Create the file hot###ctd.hdr

The file **hot###ctd.hdr** is used to correlate CTD near surface temperature and conductivity data with the thermosalinograph record. The file contains the date and start time for every CTD cast conducted during the cruise while the thermosalinograph was recording. The following steps detail a procedure to create this necessary file.

1) Copy the **hot###ctd.hdr** file from the previous cruise into the current cruise directory with the correct cruise number. For example, type the following text into a UNIX terminal:
   ```
   % cp ../../121/thermosal/hot121ctd.hdr  hot122ctd.hdr
   ```

   This would copy the **hot121ctd.hdr** file from */121/thermosal* to */122/thermosal/* directory and rename the file to **hot122ctd.hdr**.

2) Open **hot###ctd.hdr** in a text editor in order to update the file. (Depending upon which terminal you are using the command to open the text editor will be different, i.e., "**ed**" or "**te**". Check with Paul or Fernando to determine which text editor you could potentially use.) The following UNIX command is one example of how to open the file (in this case, for HOT-122):
   ```
   % ed hot122ctd.hdr
   ```

3) The **hot###ctd.hdr** file has 7 columns. The number of rows is equal to the total number of casts. The following is a typical sample of the file contents:

   ```
   1   1   01 15 2001 23 35
   6   1   01 16 2001 03 51
   2   1   01 16 2001 13 21
   2   2   01 16 2001 18 07
   2   3   01 16 2001 21 01
   2   4   01 17 2001 00 00
   2   5   01 17 2001 02 58
   2   6   01 17 2001 06 01
   2   7   01 17 2001 09 02
   ```

2 8 01 17 2001 12 06
2 9 01 17 2001 15 00
2 10 01 17 2001 18 02
....
First two columns refer to **Station** and **cast #**.
Next three columns refer to the day the rosette went into water: mo/date/yr
Next two columns refer to the time (GMT) the rosette went into water: hr/min

4) Edit this file with the data from the CTD console logs located in the current cruise binder. Occasionally, the date will advance during the period between the "start-log" time and the "in-water" time, in which case the later date should be entered into the **hot###ctd.hdr** file. For example, if the "start-log" time for a particular cast is 23:53 on 12/18/2000, and the "in-water" time is 00:01, then the correct date to be entered in the **hot###ctd.hdr** file would be 12/19/2000.
5) Occasionally, the "in-water" time is not recorded in the console log. In such a case, "in-water" time may be estimated by adding a lapse time (~ 5-10 minutes) to the "start-log" time.
6) It is imperative that the information in the **hot###ctd.hdr** file be correct. Any error in this header file would not allow to the Matlab program pltbot_p.m to run, or it may produce awkward looking Figures 3a or 3b, or the error may be undetectable. Therefore, make certain that times, dates, cast # and station # are matched correctly.

3.2 Create the file keepsalt.dat

1) Copy the **keepsalt.dat** file from the previous cruise into the current cruise directory with the correct cruise number. For example, type the following UNIX command:
```
% cp ../../121/thermosal/keepsalt.dat
```

2) Open the **keepsalt.dat** file in a text editor to update the contents. For example, type the following command at the UNIX prompt:
```
% ed keepsalt.dat
```

3) This file has 9 columns. The number of rows is equal to the total number of thermosalinograph bottles. The following is a typical sample of the file contents:

1	1	2000	12	18	20	24	40	34.6796
2	2	2000	12	19	00	27	50	34.6343
3	3	2000	12	19	04	03	00	34.7406
4	4	2000	12	19	08	04	40	35.0024
5	5	2000	12	19	12	01	01	34.8971
6	6	2000	12	19	16	30	45	34.8993
7	7	2000	12	19	20	57	52	34.9174

8 8 2000 12 20 00 38 50 34.8906
9 9 2000 12 20 04 11 40 34.9113
10 10 2000 12 20 08 03 05 34.9406
....
First two columns refer to the thermosalinograph bottle number.
Next three columns refer to the date the thermosalinograph bottle taken (yr/mo/day)
Next three columns refer to the time (GMT) the thermosalinograph bottle sample was collected (hr/min/sec).

4) Edit only columns one to eight of this file with the thermosalinograph bottle data from current cruise binder. Be very careful to enter the data correctly. (It is useful to keep a xerox copy of the thermosalinograph bottle data sheet in your current cruise folder).

4 Preliminary Analysis of the Thermosalinograph Data

The thermosalinograph data from every cruise need to be screened for errors and for outlying data points prior to the final processing and recalibration of the conductivity sensor. The following Matlab programs assist in the preliminary processing and must be executed in the order listed below to ensure the thermosalinograph data are processed correctly.

- whichship.m
- awkspeed.m
- hot_procsail_p.m (SBE 21) / hot_procsail_u.m (SBE 45)
- grabtsg_p.m
- pltbot_p.m
- cmpcond_p.m
- jpp.m

In order to process the SBE 45 TSG data, the pre-processing script read_utsg_data.m needs to be edited and run. Change the filenames of the ship TSG data sets to the correct ship ID number.

i.e., change 0902 to current cruise ID (make sure file is unzipped and change to lower case)
LINE 26: fid=fopen('km0902_thsl_wfix','r');
LINE 31: fid=fopen('km0902_uthsl','r');
LINE 35: fid=fopen('km0902_fluoro_wfix','r');

then save the file and run from Matlab prompt in thermosal directory, e.g.,:
/home/malino5/hot/208/thermosal

The program will create the file km_uthsl_wfix which includes the 1-sec SBE 45 data down-sampled to 10-sec with the exact same time and position info as the SBE 21 km0902_thsl_wfix data.

As described further below, a script called hot_procsail_p will be run for the SBE 21 processing. For SBE 45 the script hot_procsail_u (u stands for greek letter micro) will be run in place of hot_procsail_p from the thermosal directory.

UNIX terminal prompts (what is supposed to be typed into the terminal) are indicated by the `%` symbol and text in `courier` font. Command responses (standard output = stdout) will be in `courier` but without the `%` sybmol. Matlab prompts (what is supposed to be typed into a terminal where Matlab has been started) are indicated by the >> symbol and are *italicized*. Matlab stdout is also *italicized* without the >> symbols and user responses are **bold faced**. Parenthetical comments are in normal text. The symbols **###** always denote the current HOT cruise (e.g., HOT-198). The symbols **$$$$** always denote the current HOT cruise number that is used by OTG (e.g., km0724).

4.1 Run the program whichship.m

The file **shipdata.mat** is created by opening Matlab and running the program **whichship.m** (located at */home/malino5/matlab5/tsg/whichship.m*). The resultant **.mat** file contains the header information for the **ctdbot.out** file that is created and used during further processing.

1) Ensure that you are in the appropriate thermosal directory for the cruise you are working on. For example if you are working on the thermosalinograph data from HOT-157, then make sure you are in the directory */home/malino5/hot/157/thermosal*. In UNIX, type `pwd` to display the current working directory.
2) Open Matlab. For most users, type the following at a UNIX prompt to open a new Matlab window in the current directory.
   `% matlab -nojvm &`
3) Run **whichship.m** in the Matlab window.
   >> whichship

4) The program will display the following prompts.

   >> Enter Ship name (m(Wave), k(KOK), km(Kilo Moana), w(Wecoma), r(Revelle), mel(Melville)) : **km**
   >> SHIP = Kilo Moana
   DATA INTERVAL = 10
   PUMP PRESSURE = 6.145
   INTAKE DEPTH = 8

Necessary input files	***Output files***	***Possible problems***
User input on command line	shipdata.mat	

4.2 Run the program awkspeed.m

The file **hot###_speed.dat** is created by opening Matlab and running the program **awkspeed.m** (located at */home/malino5/matlab5/tsg/awkspeed.m*). The resultant file contains the ship speed at ten second intervals as stripped from the ADCP ship data (the file ending with "**.sh**").

Run **awkspeed.m** in the Matlab window. Type the following into the Matlab terminal and then hit the return key.
>> awkspeed

The program will display the following prompts.

>> Enter cruise # (i.e. 100):
(type the number, e.g., if you are working on data from HOT-154, then type **154**)
>> Enter a number for the ship code (1=kok, 2=km, 3=other): (kok is the default)
(enter the code that corresponds to the research vessel used for that cruise: **1** for the *R/V Ka'imikai-O-Kanaloa* (kok), **2** for the *R/V Kilo Moana* (km), or **3** for a ship other than the kok or the km.)
>> Enter km # enclosed between quotes:
(enter the number as assigned by OTG. For example, HOT-154 is named km0325 by OTG; therefore type **'0325'**. Make sure the number is bracketed by single quotation marks.)
>> rm: remove kmsh.sh (yes/no)?
(type **y** to remove the temporary file from the workspace.)

The process outlined above will produce the **hot###_speed.dat** file. Matlab will display the head and tail of this file where you can find the beginning and ending julian days (the second column) of the cruise.

A note on Julian days

Julian days are the number of days from the beginning of the year or from some other set starting point. The Matlab program "**julday.m**" is useful to convert between calendar dates and Julian days (type "help julday" at a Matlab prompt for instructions on how to use the program). While Julian days can be useful, they can sometimes be a source of confusion because some groups consider the first day of the year to be day 1 (OTG), while others consider it to be day 0 (HOT/PO group). When working with Julian days, it is important to know which numbering convention your group is using. Please contact Fernando and/or Sharon if you have questions regarding Julian days.

For plotting purposes pick a day longer than the cruise duration. **Never use half days!** For example, if the numbers displayed are 346 to 351, pick 345 to 351 as the beginning and ending Julian days to make the plots a day longer and to account for the one day difference.

For example, output of head and tail of **hot123_speed.dat** file:

2001 043 19 04 20 1.03
2001 043 19 05 20 0.37
2001 043 19 05 30 0.42
2001 043 19 05 40 1.11
2001 043 19 05 50 1.46

2001 047 07 25 20 7.11
2001 047 07 25 30 7.11
2001 047 07 25 40 7.11
2001 047 07 25 50 7.11
2001 047 07 26 00 7.11

In the above example, the cruise began on the 43rd day of 2001 and ended on the 47th day; so, for plotting purposes, "42" would be used for the beginning Julian day and "47" would be used for the end day.

Necessary input files	***Output files***	***Possible problems***
User input on command line	hot###_speed.dat	

4.3 Run the program hot_procsail_p.m for SBE 21 or hot_procsail_u.m for SBE 45

The Matlab program **hot_procsail_p.m** screens the thermosalinograph data for timing errors and for temperature and salinity data points that lie outside the accepted range of values. (Make sure to uncompress all the zip files (file name ends with a "**.Z**") that were transferred via ftp, see Sect. 2, or the program may not work.) The program outputs three files, **procsail.rep**, **hot###thsl.mat** and **cr_dat.mat**, all of which will be used in later steps of the processing procedures. Figures 1, 2b and 2c area also created as postscript files.

Use the following procedures to run **hot_procsail_p.m**:

> Open a Matlab window in the current cruise's thermosalinograph directory. (The command **pwd** identifies the present working directory.) Then run the program:
> *>> hot_procsail_p*

This program takes a long time to run, and it will request input from the user at several instances. It is important to monitor Matlab while **hot_procsail_p.m** is running until you know when input is needed.

>> /home/mana6/hot/198/thermosal
>> Enter HOT Cruise Number: **198**
typstr = hot
Input file = km0724_thsl_wfix
Loading in the data .....
.....Loading complete
#########
Did not find temperature offset corrections
in hottsgoffset.dat file. Offsets will not be applied
Checking for gross errors
Checking for timing errors
Running the median filter
medfiltc finished ngtcht= 0
medfiltt finished
Please create flagtsg.dat in data directory.
This file should have 3 columns:
1) Starting flag index 2) ending flag index 3) flag
>> Enter Begining julian day for X-axis: **352**
This sets the minimum value of the x-axis, and is the number you recorded earlier after running the Matlab program "awkspeed.m" (Sect. 3.1). This number is usually one day before the start of the cruise (to allow enough space for all of the thermosalinograph data to fit in the plot). Do not worry if you did not write down the number earlier; you can determine the number if you look closely at the graph you just moved – the beginning Julian day will be the first x-axis value displayed. Remember to use only whole (integer) days as the beginning and ending Julian days.
>> Enter Ending julian day for X-axis: **357**
This will be the maximum value of the x-axis, and should be chosen such that all the thermosalinograph data from a given cruise will fit in the plot.

>> Enter year (i.e. 2000): **2007**
Loading speed data

When complete, the program will generate three separate graphs (Fig. 1 or 2a, depending on the processing stage, Fig. 2b and 2c).

When the program is run for the first time for a particular TSG system an external temperature offset is not determined yet. Also, the flagtsg.dat file is not created at this point. The offset as well as the flags are being determined at a later stage of processing (see below).

Necessary input files	***Output files***	***Possible problems***
km$$$$_thsl_wfix (or km_uthsl_wfix), shipdata.mat, hottsgoffset.dat (if offsets already determined) keepsalt.dat, hot###ctd.hdr, hot###speed.dat	procsail.rep, cr_dat.mat, fig1.ps (or if flagtsg.dat exists, fig2a.ps), fig2b.ps, fig2c.ps, hot###thsl.mat	Ran whichship or awkspeed? Input files all lower case and unzipped? Correct date changes in CTD header file?

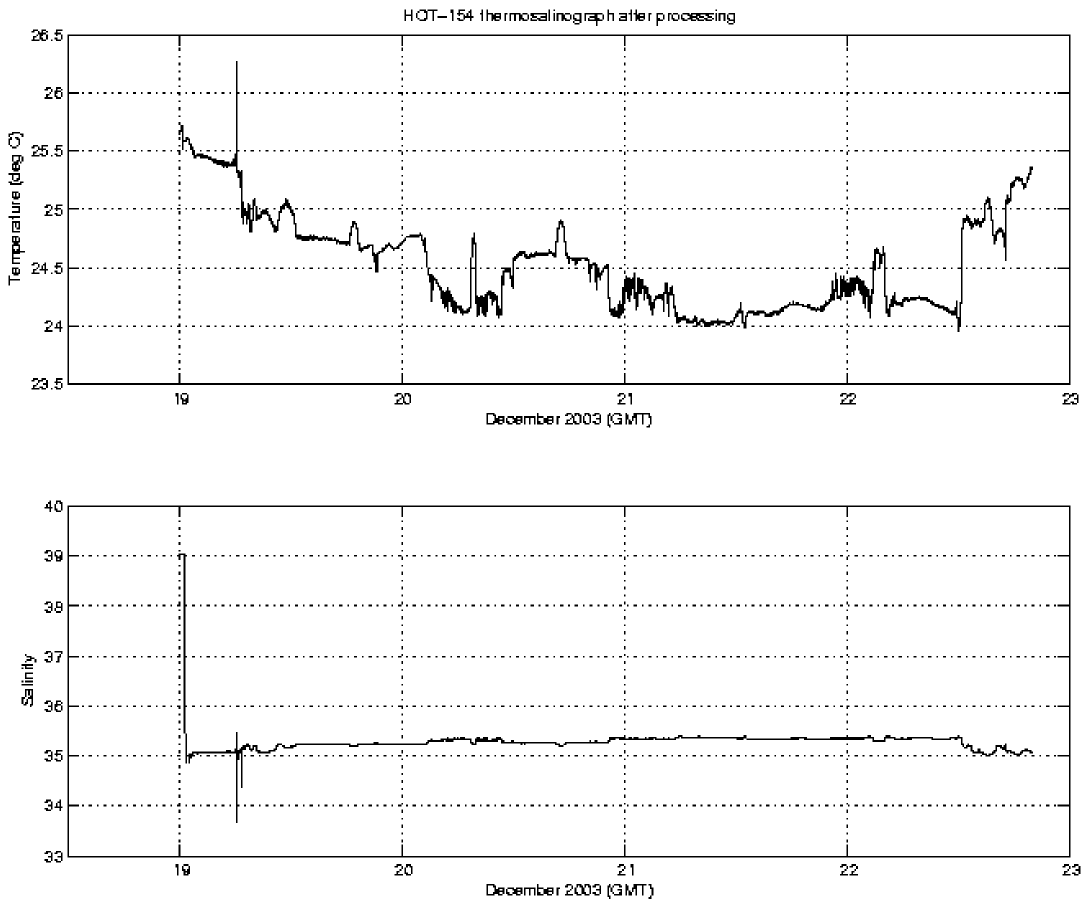

Figure 1. Example of the first plot (Fig1.ps) generated by hot_procsail_p.m.

4.4 Run the program grabtsg_p.m

The Matlab program **grabtsg_p.m** creates the file **hot###salt.dat** that is used to correlate the measured salinities of the thermosalinograph bottle samples with the thermosalinograph record. The file contains the serial number of each sample bottle and the date and time (in GMT, to the nearest second) when each sample was collected. It also creates the files **bot_ctdindex.m**, **ctd4dbdwn.mat** and **ctdbot.out** used in further processing.

>> Enter the cruise type: h: HOT a:AC
>> Enter the cruise Type: h
>> Enter the cruise : 198
>> Enter 99 to print out a copy:
Creating bot_ctdindex.m in TSG data area
To exclude a bottle modify bot_ctdindex.m and rerun cmpcond.m

Necessary input files	***Output files***	***Possible problems***
hot###ctd.hdr, keepsalt.dat, shipdata.mat	bot_ctdindex.m, ctd4dbdwn.mat, ctdbot.out, salinity.dat, hot###salt.dat	

4.5 Run the program pltbot_p.m

The Matlab program **pltbot_p.m** will create two graphical comparisons of the thermosalinograph salinity bottle data with the thermosalinograph and then output the files **cr_day.mat**, **thsb.mat** and **thctd.mat** that will be used in a later step in the processing. The program uses as an input the file **hot###salt.dat** that you already created with the program **grabtsg_p.m**. The program is helpful to determine 'suspicious' bottle samples, which will be considered outliers and removed from the statistical comparison. Figures 3a and 4a are also created. Figure 3a shows a comparison of the external temperature data with the CTD data from the according depth. Figure 4a displays the differences between the salinity of the individual thermosalinograph samples and the salinity from the continuous thermosalinograph record at the time each bottle sample was collected. **Print copies of the figures for the report and label the prints with the corresponding figure names.**

Use the following steps to run **pltbot_p.m**
1) Open a Matlab window in the current thermosalinograph directory (e.g., */home/malino5/hot/155/thermosal*).
2) *>> pltbot_p*
   /home/mana6/hot/198/thermosal
   >> Enter HOT Cruise Number: **198**
   typstr = hot
   xl = 352.5000 357.0000

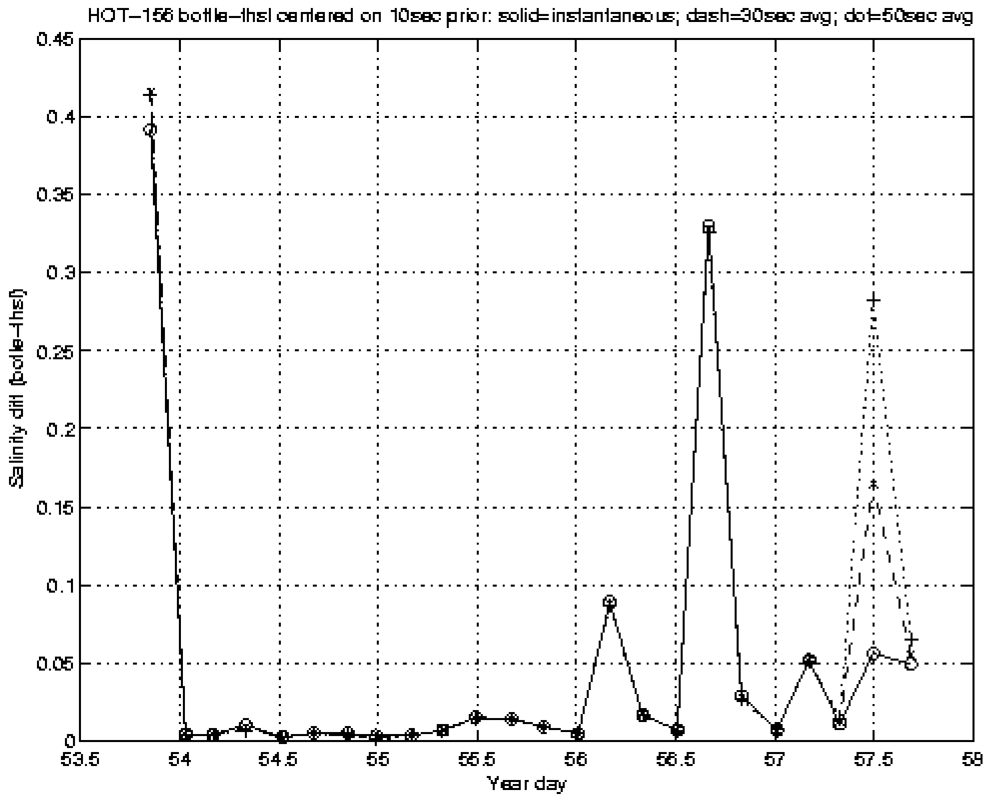

Figure 2. Example of the bottom panel of Figure 4a generated by pltbot_p.m.

3) Take notes on which bottles lie off the average. In this example, the suspicious bottles are nos. 1, 15, 18, 21, 23, and 24. Normally there will be between 3 to 5 suspicious thermosalinograph bottles for a typical HOT cruise. The times of the suspicious bottles should be rechecked; the data could be okay, just an error with the times.

Necessary input files	***Output files***	***Possible problems***
hot###salt.dat, hot###ctd.hdr, hot###thsl.mat, shipdata.mat, ctd4dbdwn.mat,	fig1b_thsl_ctd_salinity.fig and *.ps, fig3a_thsl_ctd_temperature.fig and *.ps, fig4a.fig and *.ps, thctd.mat, cr_day.mat, thsb.mat	

4.6 Run pltbot_p.m again.

Follow the instructions above to run **pltbot_p.m** a second time.

This time, however, type control-C (^C) to terminate the program after the first plot appears. This plot overlays the salinity of each bottle sample onto the continuous thermosalinograph record at the time each bottle was collected, and it can be useful to determine if any of the suspicious thermosalinograph bottles (Sect. 4.5, step 3) coincide with glitches.

Use the following steps to run **pltbot_p.m** a second time and to examine the first plot.

1) In the Matlab window, type >> *pltbot_p*; and then enter the cruise number when prompted.

2) After the first plot appears (entitled "HOT-### Thermosalinograph O=bottle X=thsl when bottle taken") and Matlab tells you to "*press any key to continue*", simultaneously depress both the "ctrl" and the "c" keys. This will terminate the program.

3) Use the zoom in/out icons of the Matlab figure window to focus in on the suspicious thermosalinograph bottles that were identified in Section 4.5. If a bottle coincides with a spike in salinity, label the bottle number on the zoomed portion of the graph (see below), and then print the graph. Use the zoom in/out icons to investigate all the suspicious thermosalinograph bottle samples, and print out a zoomed and labeled figure for any and every bottle that coincides with a salinity glitch.

   To attach a text label to a Matlab graph:

   a) In the Matlab command window, type the following
      >> *gtext('text that you want to appear on the graph')*
      For example, if you wish to enter a label for the sixteenth thermosalinograph bottle sample then type the following to attach the text "thsl bottle #16" to the figure.
      >> *gtext('thsl bottle #16')*
   b) Matlab will snap to the figure, and a set of crosshairs will appear. Use the mouse to position the crosshairs at the spot in the figure where you wish the label to appear. In this example the words "thsl bottle #16" would appear at the position that was clicked.
   c) NOTE: Labels added with the gtext command do not disappear when the figure is updated or replotted, and so they must be deleted manually. To remove a label, click on the text you wish to delete and wait for the text box to appear (four small black squares that frame the text); then press "delete" or "backspace".

4.7 Run the program cmpcond_p.m

The Matlab program **cmpcond_p.m** is used to compare the thermosalinograph data with the CTD and bottle sample data, and to recalibrate the thermosalinograph conductivity sensor based on those results. This program will be used multiple times during the processing. At this step in the processing procedures, **cmpcond_p.m** is used to identify outlying CTD casts based on temperature differences from the thermosalinograph. Figures 3b, 4b, 5 and 6 are also created.

Use the following steps to run **cmpcond_p.m** to identify the CTD outliers.

1) Open a Matlab window in the current thermosalinograph directory and type:
   >> cmpcond_p
   /home/mana6/hot/198/thermosal
   >> Enter HOT Cruise Number: **198**
   Reading bottle/CTD indices from bot_ctdindex.m
   intmean = -0.1781
   extmean = -0.1696
   Figure 3b saved as .fig and .ps

2) The program will prompt the user to depress a key to continue. At this point in the processing, the user can either continue on displaying the rest of the graphs (type **y** when prompted to overwrite the file **datacmpcond.out**), or terminate the program (simultaneously press ctrl and C).
   Hit the space bar to continue
   linreg(x,y); y = mx + c;
   m = -0.000113359 , c = 0.0373559, r^2= 0.0977849
   linreg(x,y); y = mx + c;
   m = -0.000113359 , c = 0.0373559, r^2= 0.0977849
   Figure 4b saved as .fig and .ps
   Hit the space bar to continue
   linreg(x,y); y = mx + c;
   m = 3.1128e-05 , c = -0.014174, r^2= 0.0632549
   Hit the space bar to continue.
   Hit the space bar to continue
   linreg(x,y); y = mx + c;
   m = 1.68053e-17 , c = -5.87995e-15, r^2= 2.05261e-26
   linreg(x,y); y = mx + c;
   m = -0.000146177 , c = 0.0521229, r^2= 0.15328
   Figure 5 saved as .fig and .ps
   Hit the space bar to continue
   linreg(x,y); y = mx + c;
   m = -0.0010865 , c = 0.387426, r^2= 0.15206
   Figure 6 saved as .fig and .ps
   Hit the space bar to continue

3) Examine the plot for "suspicious" CTD casts that lie off the average and record the number of any casts that stand out from the rest. Check the CTD times that were entered in **hot###ctd.hdr** (Sect. 3.1).

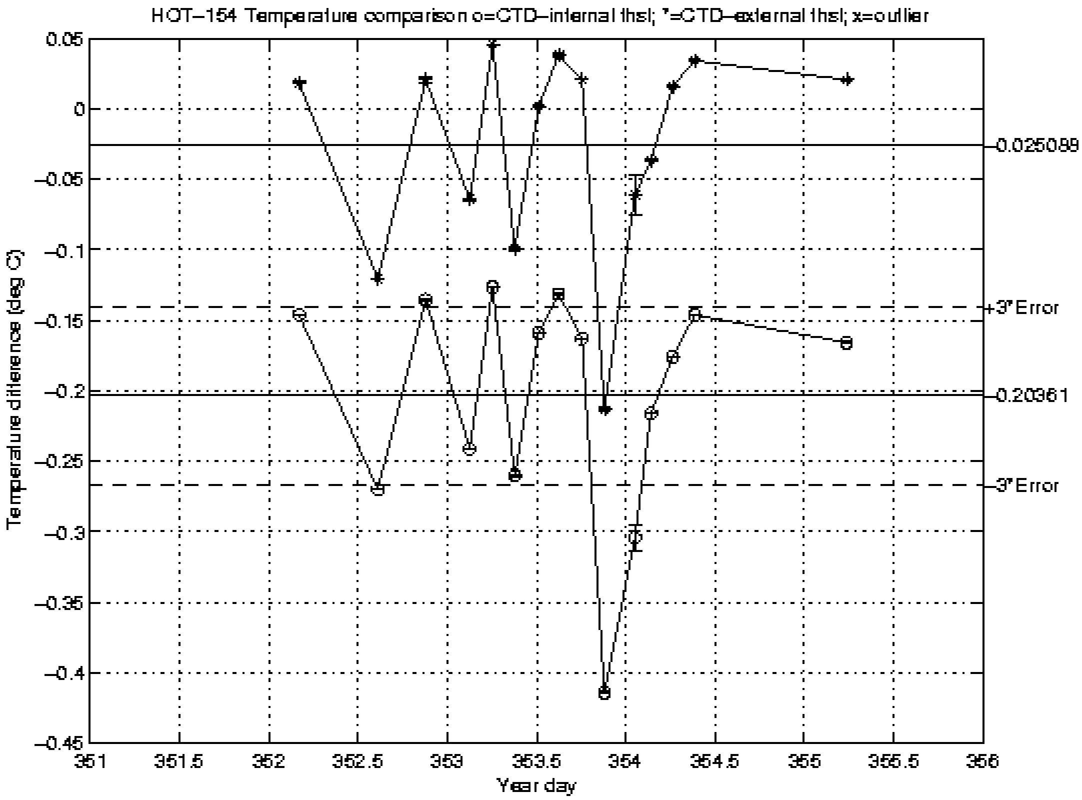

Figure 3. Example of first plot (Fig. 3b.ps) generated by cmpcond_p.m.

In this example the suspicious CTD casts would be 2, 4, 6, 11, and 12. Ideally there will be fewer outliers for your cruise.

Necessary input files	***Output files***	***Possible problems***
hot###salt.dat, hot###ctd.hdr, hot###thsl.mat, shipdata.mat, cr_day.mat, thsb.mat, ctd4dbdwn.mat, thctd.mat, bot_ctdindex.m	fig3b.ps and *.fig, fig4b.ps and *.fig, fig5.ps and *.fig, fig6.ps and *.fig, datacmpcond.out, condcalib.mat	

After running cmpcond_p.m the external temperature offset can be determined from the standard output on the screen, i.e., from the example above: *extmean = -0.1696* and add this value to the hottsgoffsets.dat file located in /home/malino5/hot/ Also, the file bot_ctdindex.m can now be filled in with outliers in CTD and bottle numbers. Remember to delete the outliers in the list of samples to be kept:

% CTD data to use & exclusions

```
ic=[3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18];
ixc=[1 2 19];

% bottle data to use & exclusions
ir=[1 2 5 6 7 8 10 11 12 13 14 15 16 17 18 19 20 21];
ix=[3 4 9 22 23 24];
```

4.8 Run the program jpp.m and determine flags

The Matlab program **jpp.m** produces a plot that is used to determine the thermosalinograph data "flags". Each thermosalinograph temperature and conductivity datum is given a numerical flag that indicates the reliability of the data point. The flags most often used are "2" (for good data), "3" (for suspicious data), and "4" (for bad data); and less often used is "1" (for uncalibrated data).

To run **jpp.m** and generate the plot, type **jpp** at a Matlab prompt, and then enter the number of the HOT cruise (e.g., 157). Matlab will generate a plot that overlays the following six thermosalinograph data sets over time: salinity, conductivity, internal temperature, remote temperature, change in latitude, and change in longitude.

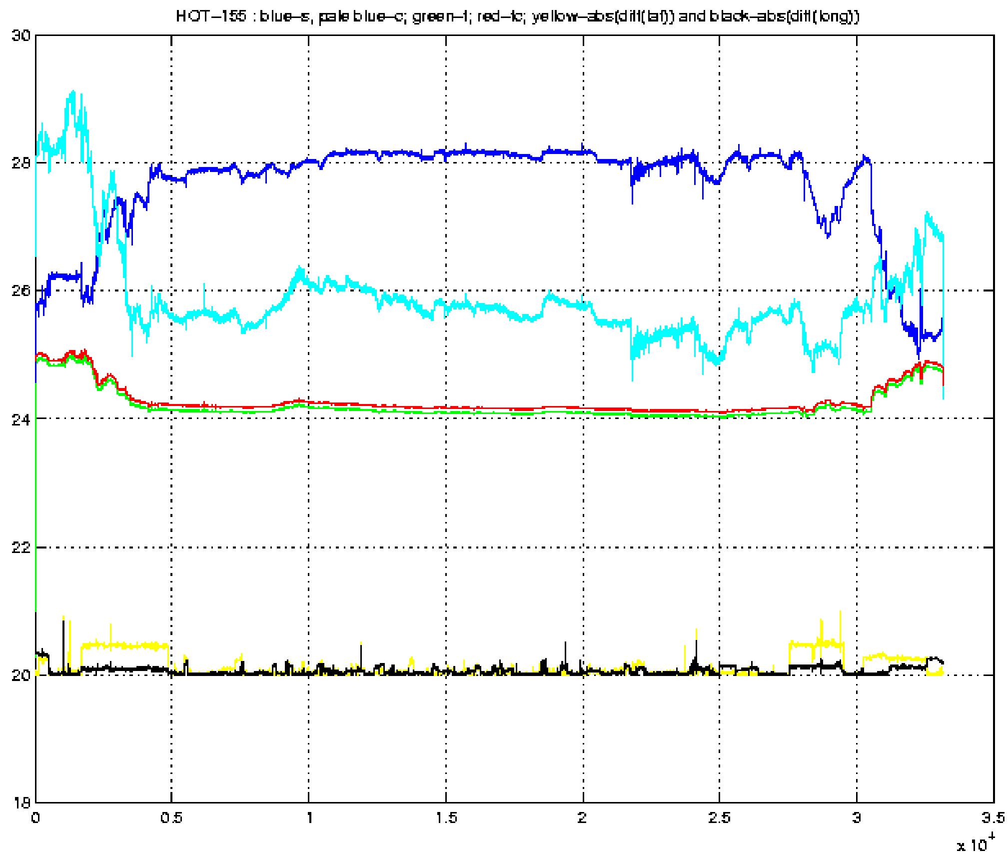

Figure 4. Example of the plot (jpp.ps) generated by jpp.m.

The plot needs to be examined closely to determine flags for both the conductivity and the temperature data. By default, any data point that is not specifically assigned a flag will be assumed to be good data (flag "2"). Use the zoom in/out tools on the graph window to inspect for suspicious data points and record the corresponding beginning and ending x-axis values of all that are found. The following guidelines can be helpful in determining flags.
Temperature data are considered suspicious (flag "3') when there is a spike in either the internal or remote temperature set without a corresponding spike in the other set. Zoom in on the temperature data traces and record the x-axis values that correspond to the start and end of any suspicious points.

Conductivity data are considered suspicious when there is a spike in conductivity without a corresponding spike in temperature. This tends to be the most common type of error that occurs with HOT thermosalinograph data and is often caused by bubbles entering the ship's seawater supply, usually as the result of the ship transiting through rough seas.

Spikes in conductivity without a corresponding change in temperature also often occur as "bottle glitches" on cruises aboard the *R/V Ka'imikai-O-Kanaloa* (*KOK*). This type of glitch is caused by the activation of a pump that drains the thermosalinograph sampling area aboard the *KOK*, though following proper sampling procedures (i.e., waiting 30 seconds after the pump deactivates before collecting the sample) should allow the conductivity spike to be easily flagged without interfering with the data processing.

Data should only be flagged as bad (4) when there is documented evidence in the logs to do so. For example, it may have been recorded in the logs that data logging was halted or that the seawater intake pumps were disabled at various times during a cruise, then data points that were collected during those times would be flagged as bad (4).

Data are flagged as uncalibrated (1) when the data have been corrected for some sort of offset or drift. For example, the remote temperature data from the *R/V Kilo Moana* tend to be warmer than the CTD data collected from the same depth at the same time because of a problem with the remote temperature system, so the average offset between the two is calculated and then the remote temperature data are adjusted to agree with the CTD temperature data. **Offsets can be applied to the temperature data by first entering the cruise number and offset in the file hottsgoffset.dat, located in the following directory /home/malino5/hot/; and then running the Matlab program "hot_procsail_p, grabtsg_p.m, pltbot_p.m and cmpcond_p.m.**

Record the x-axis values for all the suspect data points, and be sure to record whether the points are in the conductivity data or in the temperature data. This information will all be used in the next step of the thermosalinograph processing.

Necessary input files	***Output files***	***Possible problems***
hot###thsl.mat	jpp.ps	

4.9 Create the file flagtsg.dat

The file **flagtsg.dat** will list all the thermosalinograph data points that are flagged as being something other than good (i.e., suspicious, bad, or uncalibrated).

The file contains three columns of data, which will look something like this:

1	**55**	**33**
99	**111**	**23**
254	**314**	**23**
316	**420**	**44**

These values are, from left to right, the beginning point of the flag, the ending point of the flag, and then the flag (a two-digit number). The first number of the flag is the temperature flag, the second number is the salinity flag.

Use a text editor to update the file with the data points and flags for the current cruise. For instance, if for a given cruise there was a suspicious temperature glitch between points 337 and 359, but the conductivity data at the same time was good, then the following line would be entered into flagtsg.dat: **337 359 32**
If there was section of data between 420 and 426 with suspicious conductivity data and good temperature data, the following line would be entered: **420 426 23**
And so on, until every non-good point identified in Section 4.8 above is represented in the **flagtsg.dat** file. Flagging is a difficult process that requires some experience, but it becomes easier with practice. Consult Fernando and/or Sharon for flagging advice.

After all the flags are recorded into **flagtsg.dat**, the thermosalinograph data are ready for final analysis.

5 Final Analysis of the Thermosalinograph Data

After completing the preliminary analysis and making the file **flagtsg.dat**, the following processing programs must be executed in the order given below. If any changes are made such as editing the flags, or changing the CTD and bottle outliers, then these four programs must be consecutively run to ensure that the changes have been incorporated into the final analysis.

- hot_procsail_p.m / hot_procsail_u.m
- cmpcond_p.m
- finalize_p.m
- tsnoise_tsg.m

UNIX terminal prompts (what is supposed to be typed into the terminal) are indicated by the `%` symbol and text in `courier` font. Command responses (standard output = stdout) will be in `courier` but without the `%` sybmol. Matlab prompts (what is supposed to be typed into a terminal where Matlab has been started) are indicated by the >> symbol and are *italicized*. Matlab stdout is also *italicized* without the >> symbols and user responses are **bold faced**. Parenthetical comments are in normal text. The symbols **###** always denote the current HOT cruise (e.g., HOT-198). The symbols **$$$$** always denote the current HOT cruise number that is used by OTG (e.g., km0724).

5.1 Run hot_procsail_p.m / hot_procsail_u.m

The **hot_procsail_p.m** program will replot the thermosalinograph salinity and temperature data after taking into consideration any applied temperature offset (in **hottsgoffset.dat**, see Sect 4.8) and accounting for data that have been flagged as good, suspicious, bad, or uncalibrated as indicated in the **flagtsg.dat** file (Sect. 4.9).

For final analysis, run **hot_procsail_p.m** in Matlab (see Sect. 4.3 for instructions).
The program will generate three separate plots similar to those from Section 4.3 earlier, however this time the plots will not include bad or suspicious data.

5.1.1 Analyze the plots to determine if additional flagging is necessary

Before printing the three plots, examine the first graph (with the title “after quality control analysis”) for any extra spikes or glitches or sections of suspect data that were not flagged earlier (Sect. 4.8 and Sect. 4.9). Take note of the areas that need further flagging, then run **jpp.m** again and update **flagtsg.dat** with the additional flags, then rerun **hot_procsail_p.m** to incorporate the changes. Repeat this step as often as necessary.

5.1.2 Print the three separate plots

After all the data flags and outliers have been updated; print a copy of each of the three plots generated by **hot_procsail_p.m**. These will be Figures 2a, 2b, and 2c of the thermosalinograph processing report.

5.1.3 Print the file procsail.rep

The **hot_procsail_p.m** program will create the file **procsail.rep** in the current thermosalinograph directory. This file lists all the data flags you determined and other quality control information needed for the processing report.

Print **procsail.rep** with **lp** command. For example at UNIX prompt type:

```
% lp procsail.rep
```

The printout will be Appendix B of the thermosalinograph processing report.

5.2 Run “cmpcond_p.m”

After running **hot_procsail_p.m** with flags, you must run the program **cmpcond_p.m** again (Sect 4.7) to recalibrate the thermosalinograph sensor based on the CTD and bottle data (excluding outliers). This will generate a series of plots and create other files needed for final analysis and for writing the processing report.

5.2.1 Print the requested plots

The program analyzes a series of the comparisons (conductivity, temperature, etc.) and plots them. Print the requested plots; these will be Figures 3b, 4b, 5, and 6 of the processing report. Be sure to adjust the y-axis of each graph so that the data are visible and nicely centered (Either manually change the axis settings in the m-file and rerun **compcond_p.m**, or edit the axis properties temporarily by using the menu at the top of the plot.)

5.2.2 Print the output file datacmpcond.out

The program **cmpcond_p.m** outputs the file **datacmpcond.out** to the current thermosalinograph directory. This file lists several statistical comparisons between the quality controlled thermosalinograph data and the CTD and bottle data, which all be needed for the final processing report.

Print the output file using the lp command. For example, at the UNIX prompt type:

```
% lp datacmpcond.out
```

This printout will be Appendix A of the final processing report.

5.3 Run finalize_p.m

This program will create an output file of the corrected conductivity and temperature data (**hot###thsl.dat**), and it will also generate a color plot of the final salinity and near-surface temperature data.

To run **finalize_p.m**, go to a Matlab prompt and type **finalize_p**, and then enter the cruise number when requested.

Print copies of the final plots (hot###thsl_final.ps = Fig. 7a, hot###nav_final.ps = Fig. 7b) in color.

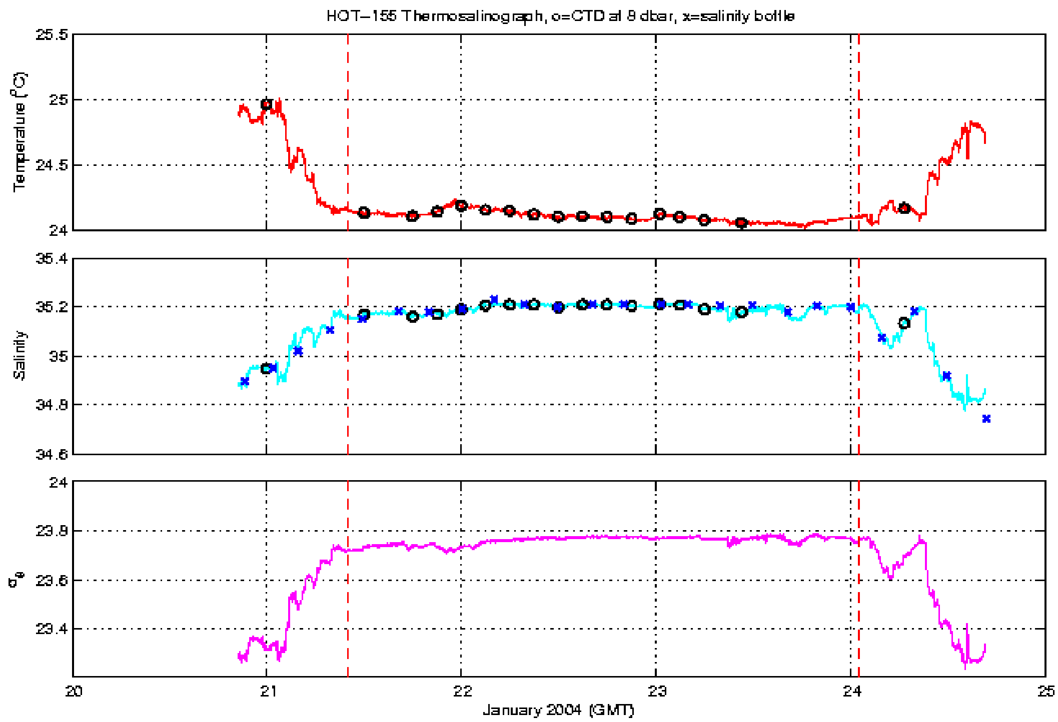

Figure 5. Example of the first plot generated by finalize_p.m (Fig. 7a).

Necessary input files	***Output files***	***Possible problems***
hot###ctd.hdr, hot###salt.dat, hot###_speed.dat, hot###thsl.mat, shipdata.mat, cr_day.mat, ctd4dbdwn.mat, condcalib.mat	aloha_time.dat, hot###thsl.dat, hot###nav_final.ps (Fig. 7b), hot###thsl_final.ps (Fig. 7a)	

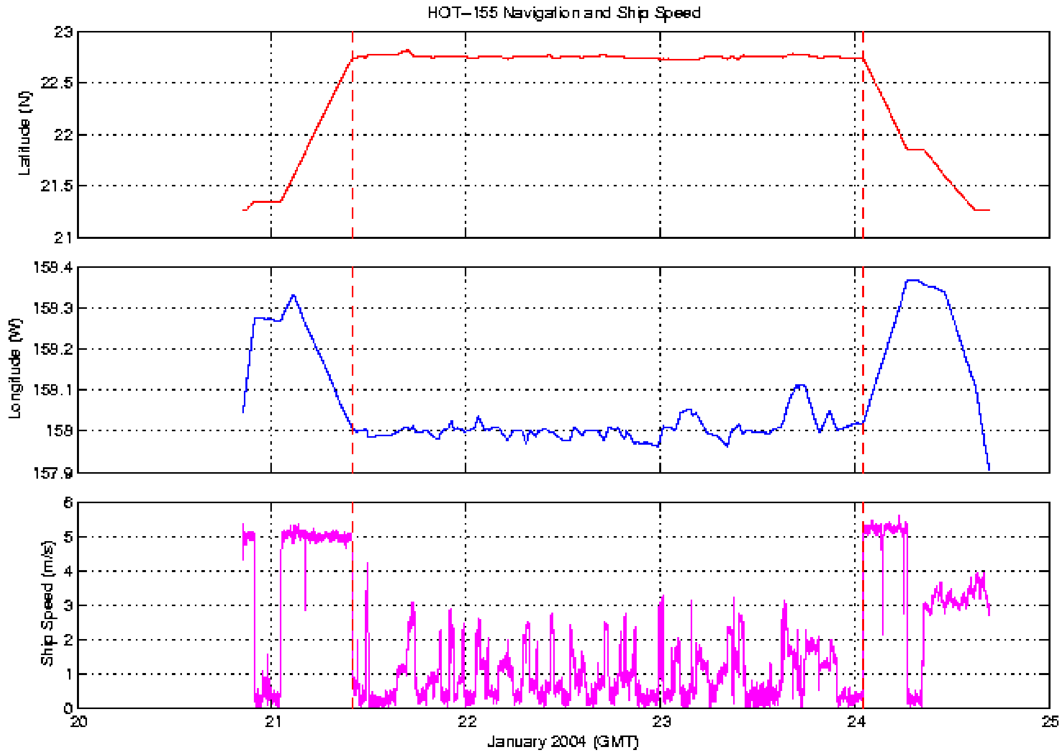

Figure 6. Example of the second plot generated by finalize_p.m (Fig. 7b).

5.4 Run tsnoise_tsg.m

The m-file **tsnoise_tsg.m** is used to estimate salinity and temperature noise of the thermosalinograph data. The noise is the standard deviation of the difference between actual data points and the 17-minute running mean.

To run **tsnoise_tsg.m**, at a Matlab prompt type **tsnoise_tsg**, then enter the cruise number when requested. The program will also request the number of data sections that you will want to use to estimate the noise (three sections is usually good).

Next you will need to click on the graph at the beginning and ends of the sections you want to use to estimate the noise. Use sections that are relatively flat and constant. The resultant noise estimates will be displayed in the Matlab window; write these down because they will be needed for the processing report. Rerun **tsnoise_tsg.m** repeatedly until the salinity and temperature noise values are consistent.

Necessary input files	***Output files***	***Possible problems***

Necessary input files	***Output files***	***Possible problems***
hot###thsl.dat		

>> Enter Cruise Number: **198**
How many sections? **3**
Select the start and end of the section
Select the start and end of the section
Select the start and end of the section
Cruise: 198 External temperature noise (St. Dev): 0.004302 deg
How many sections? **3**
Select the start and end of the section
Select the start and end of the section
Select the start and end of the section
Cruise: 198 Salinity noise (St. Dev): 0.00099551 psu
>> tsnoise_tsg
Enter Cruise Number: **198**
How many sections? **3**
Select the start and end of the section
Select the start and end of the section
Select the start and end of the section
Cruise: 198 External temperature noise (St. Dev): 0.004129
How many sections? **3**
Select the start and end of the section
Select the start and end of the section
Select the start and end of the section
Cruise: 198 Salinity noise (St. Dev): 0.00096825 psu
>> tsnoise_tsg
Enter Cruise Number: **198**
How many sections? **3**
Select the start and end of the section
Select the start and end of the section
Select the start and end of the section
Cruise: 198 External temperature noise (St. Dev): 0.0041313
How many sections? **3**
Select the start and end of the section
Select the start and end of the section
Select the start and end of the section
Cruise: 198 Salinity noise (St. Dev): 0.00087497 psu

Take the mean of the three temperature and salinity noise stddev
>> mean([0.004302, 0.004129, 0.0041313])
0.00418743

>> mean([0.00099551, 0.00096825, 0.00087497])
0.000946243

Make as note of the two means
HOT 198 noise
T = 0.00418743
S = 0.000946243

Comparing the two TSG systems

Once both TSG systems are fully processed, i.e., all CTD and bottle outliers are excluded, flags are determined and all processing related files put in the appropriate folders, the script cmp_thsl.m can be edited and run to create comparison figures.

5.5 Write the draft thermosalinograph processing report

Now the data processing procedures are complete, and you have enough information to write the draft version thermosalinograph processing report. Usually it is easiest to copy the report from the previous cruise into the current directory, then rename it and update it with the numbers from the current cruise. At the time this processing manual was written, reports are written with the Star Office software on the UNIX machines (file name ends with "**.sdw**"). When you are finished with the report, turn it in to Fernando for review.

5.6 Finalize the thermosalinograph data processing report

Once the processing report has been returned to you with edits, then you need to update the final report. If any outliers or data flags are changed, then all the programs listed in Section 5 need to be rerun consecutively to incorporate the changes into the final analysis.

After finalizing the report with any changes and reprinting the necessary graphs, make a copy of the report (include the extra color graphs with this copy) and turn it in to Roger for his review. Place the original report and graphs in the file cabinet with the completed HOT CTD and QC reports.

You will also need to convert a copy of the processing report into a text only format, and then paste it into the file **history.###** (where ### is the HOT cruise number) located in the current cruise directory.

5.7 Compress the raw files

The only thing left to do is to clean up the workspace and to compress the raw files to conserve disk space. Remove any temporary files that are no longer needed from the thermosalinograph directory. If the thermosalinograph data from a cruise required special processing (such as applying offsets) then create a read-me file that details everything that differed from normal processing procedures.

Now compress the raw files (the ones that were transferred via ftp from the OTG server) so as to save space on the server. In UNIX use the **compress** command (type **man compress** for help). For example if you wanted to compress the raw files from HOT-154 you could use the following command: **compress km0325***

In UNIX, type **ls** to list the contents of the directory. All the raw thermosalinograph data files should now have the extension "**.Z**".

Now get ready for the next cruise.