



**National
Oceanography Centre**

NATURAL ENVIRONMENT RESEARCH COUNCIL

National Oceanography Centre

Cruise Report No. 31

RRS *James Clark Ross* Cruise JR306

10 - 20 JAN 2015

Hydrographic measurements on WOCE line SR1b

Principal Scientist

B A King

Editor

Y Firing

2015

National Oceanography Centre, Southampton
University of Southampton Waterfront Campus
European Way
Southampton
Hants SO14 3ZH
UK

Tel: +44 (0)23 8059 6438

Email: bak@noc.ac.uk

DOCUMENT DATA SHEET

<i>AUTHOR</i> KING, B A et al	<i>PUBLICATION</i> <i>DATE</i> 2015
<i>TITLE</i> RRS <i>James Clark Ross</i> Cruise JR306, 10-20 Jan 2015. Hydrographic measurements on WOCE line SR1b.	
<i>REFERENCE</i> Southampton, UK: National Oceanography Centre, Southampton, 35pp. (National Oceanography Centre Cruise Report, No. 31)	
<i>ABSTRACT</i> <p>Twentieth complete occupation of the Drake Passage section, established during the World Ocean Circulation Experiment as repeat section SR1b, first occupied in 1993 and re-occupied most years since then. Full-depth temperature, salinity, and lowered ADCP velocity profiles were measured at the 30 repeat stations (with the 9 northern stations along 58 W, as from 2009 on). Underway measurements included navigation, VM-ADCP, sea surface temperature and salinity, water depth, and meteorological variables. The objectives were to investigate interannual variability in Antarctic Circumpolar Current (ACC) structure and transport, the positions of the ACC fronts, and Drake Passage temperature and salinity distributions.</p>	
<i>KEYWORDS</i>	
<i>ISSUING ORGANISATION</i> National Oceanography Centre University of Southampton Waterfront Campus European Way Southampton SO14 3ZH UK Tel: +44(0)23 80596116 Email: nol@noc.soton.ac.uk <i>A pdf of this report is available for download at: http://eprints.soton.ac.uk</i>	

Page intentionally left blank

Contents

1 Personnel	7
2 Itinerary and Cruise Track	9
3 Objectives	10
4 Narrative	10
5 CTD Systems Operation	10
5.1 CTD and Sensors	10
6 CTD Data Processing and Calibration	11
6.1 Data Processing	11
6.2 CTD Conductivity and Temperature Calibration	12
6.2.1 Temperature calibration	12
6.2.2 Conductivity calibration	13
6.3 Water Sample Salinity Analysis	13
6.4 Sample recording and merging with CTD data	14
7 Lowered Acoustic Doppler Current Profiler (LADCP)	15
7.1 Instrument setup and performance	15
7.2 Data processing	15
7.2.1 Data processing setup	15
7.2.2 Linking to CTD data	16
7.2.3 Linking to VMADCP data	17
7.2.4 Edits to .m files required at start of a cruise for IX processing	17
7.2.5 IX processing	17
7.2.6 UH processing	18
7.3 Data quality	18
8 Computing and SCS data processing	19
8.1 Configuration of linux workstation 'folaf'	19
8.2 SCS data streams	19
8.3 Access to SCS data	19
8.3.1 The mscs source directory	21
9 Underway temperature and salinity data	21
9.1 Underway salinity calibration	21
9.2 Underway surface thermosalinograph	22
10 Daily processing of SCS streams	23
10.1 Navigation	23
10.2 Surface meteorological sampling (SURFMET)	23
10.3 Bathymetry	23
10.3.1 EA600	24
10.3.2 EM122	24
10.3.3 Merging of EM122 and EA600	25
10.4 Chernikeef EM log	25

11 Vessel Mounted ADCP	25
11.1 Real-time data acquisition and VMDAS files	25
11.1.1 Settings	26
11.2 Post-processing	27
11.2.1 Setting up the directories and using quick_adcp	27
11.2.2 Angle and amplitude calibration	27
11.2.3 Gautoedit	27
11.2.4 Applying the edits	28
11.2.5 Creating the output files	28
11.2.6 Extracting station profiles	29
12 Acknowledgments	29

1 Personnel

Scientific Personnel

Brian King	NOC MPOC
Yvonne Firing	NOC MPOC
Freya Garry	UoS
Damien Desbruyeres	NOC MPOC
Cristian Florindo Lopez	UoS
Michael King	NOC

Technical Personnel

Jeremy Robst	BAS IT
Paul Woodroffe	BAS AME

Doctor

Julie Hunt	BASMU
------------	-------

Ship's Personnel

Jerry Burgan	Master
Tim Page	Chief Officer
Philippa Bowden	2nd officer
Greg Johnston	3rd Officer
Lucy Faulkner	Ex. 3rd Officer
Mike Gloistein	ETO Comms
Neil MacDonald	Chief Engineer
Gert Behrmann	2nd Engineer
Marc Laughlan	3rd Engineer
Jevgenijs Tolks	4th Engineer
Craig Thomas	Deck Engineer
Steve Amner	ETO
Rich Turner	Purser
Dave Peck	Bosun/Sci Ops
Martin Bowen	Bosun
George Dale	Bosuns Mate
Ian Raper	AB
Sheldon Smith	AB
Dick Robinson	AB
Alan Howard	AB
Glyn Henry	Motorman
John Pratt	Chief Cook
Colin Cockram	2nd Cook
Lee Jones	Senior Steward
Nick Greenwood	Steward
Riff Raworth	Steward
Rod Morton	Steward

2 Itinerary and Cruise Track

JR305/306 departed Rothera, Adelaide Island, Antarctica on 10 January 2015, stopping a half day at Vernadsky Base for tide gauge servicing by the NOCL JR305 participants, and then continuing to Elephant Island. Hydrographic stations (listed in Table 5) were occupied starting on 12 January from the southernmost station, near Elephant Island (Figure 2), and progressing northwards. Underway navigation, meteorological, and VMADCP data were acquired starting from departure from Rothera until the end of science, while underway ocean surface data and bathymetry were acquired starting from the Bransfield Strait on 12 January. JR305 work occurred before CTD station 1 and between stations 3 and 4 and stations 24 and 25. The weather was good, with only two intervals of a few hours delay in work over the course of the cruise. All science sampling was completed by the early hours of 18 January and the *James Clarke Ross* arrived off Punta Arenas on 20 January, although docking was not possible until the 22nd. As part of JR305, three Bottom Pressure Recorders (BPRs) were to be recovered and these were to be replaced with three new BPRs deployed in the same locations. In addition to the usual lander recoveries and deployments, an EM-APEX float was deployed to the east of Elephant Island at 60° 54.438'S, 53° 14.178'W in the western end of a deep trough. The BPR at the southern end of the Drake Passage was not recovered and the new BPR was deployed 100m east of that position. At the northern end of the Drake Passage, two BPRs were recovered and replaced with two new BPRs, deployed at the same positions as the recovered landers. Full details can be found in the JR305 cruise report, prepared by NOCL.

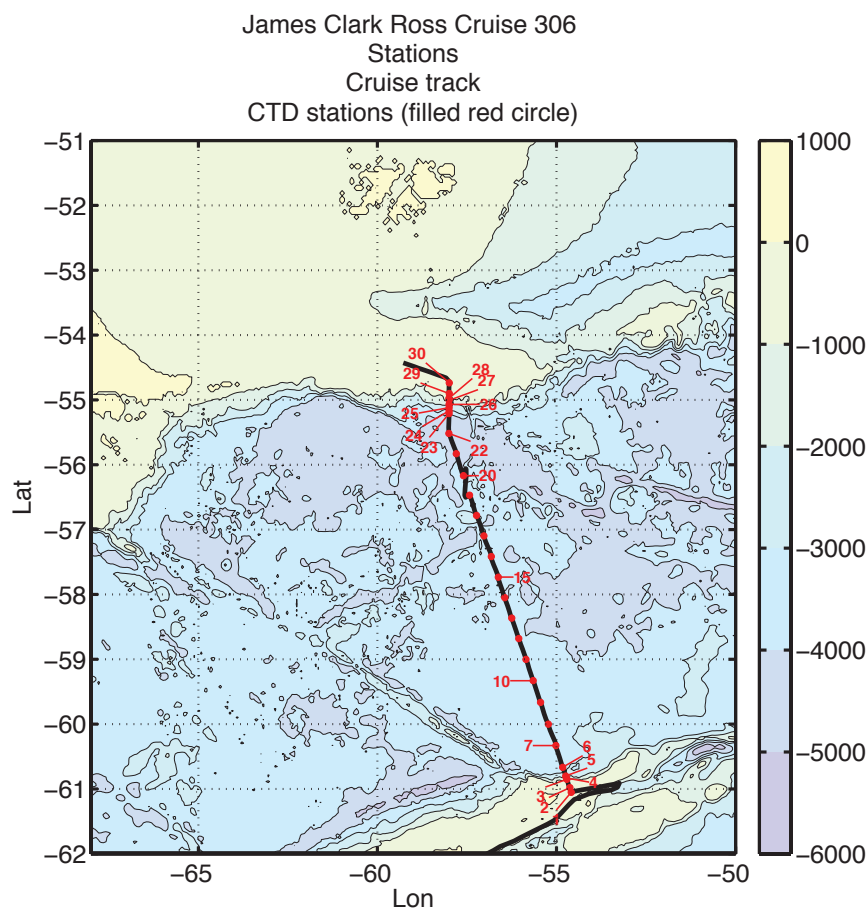


Figure 1: JR305/306 cruise track with JR306 SR1b CTD stations marked by red dots.

3 Objectives

RRS *James Clark Ross* cruise JR306 was a repeat occupation of World Ocean Circulation Experiment (WOCE) line SR1b, performed annually since 1993 for the National Capabilities Drake Passage repeat hydrography programme by scientists at the National Oceanography Centre, Southampton. This long, ongoing time series is designed to monitor Southern Ocean watermasses and Antarctic Circumpolar Current volume and property transport. The principal objective for JR306 was to make repeat hydrographic measurements on SR1b, with a secondary objective of obtaining current measurements across Drake Passage. The objectives were achieved successfully.

4 Narrative

A total of 30 CTD (conductivity-temperature-depth) stations were occupied. Continuous lowered acoustic Doppler current profiler (LADCP) profiles were also obtained on each station. Up to 12 bottle salinity samples per station were analysed on board for CTD conductivity calibration, and a SBE-35 thermometer provided coincident discrete high-accuracy temperature measurements for CTD temperature calibration. Although oxygen data were recorded by the CTD we do not have bottle oxygen samples for calibration. Underway measurements included vessel-mounted ADCP (VMADCP) currents, surface ocean measurements, and surface meteorology. The cruise was part of the National Capabilities Drake Passage repeat hydrography programme along WOCE line SR1b, and also included science from JR305, described in a separate cruise report prepared by NOCL.

BAK, YLF

5 CTD Systems Operation

5.1 CTD and Sensors

One stainless steel CTD system was prepared with a 24-way carousel. A spare frame was on board but it was apparently distorted out of shape and was not prepared for deployment. It may or may not have been useable as a backup. Since there were no mishaps with the primary frame, this was not tested.

The sensor configuration was as follows, unchanged throughout the cruise.

Sea-Bird 9plus underwater unit, s/n 0541

Sea-Bird 3P temperature sensor, s/n 5043, (primary)

Sea-Bird 4C conductivity sensor, s/n 3491, (primary)

Digiquartz temperature compensated pressure sensor, s/n 0541

Sea-Bird 3P temperature sensor, s/n 2307, (secondary)

Sea-Bird 4C conductivity sensor, s/n 4090, (secondary)

Sea-Bird 5T submersible pump, s/n 3415, (primary)

Sea-Bird 5T submersible pump, s/n 2371, (secondary)

Sea-Bird 32 Carousel 24 position pylon

Sea-Bird 11plus deck unit, s/n 11P-15759-0458

Sea-Bird 43 dissolved oxygen sensor, s/n 0620

PA200 altimeter, s/n 10127.244739

Fluorometer, s/n 12-8513-01

Transmissometer C-Star, s/n CST1497DR

PAR sensor, s/n 7235

Swivel, s/n 196111

SBE35 thermometer, s/n 0620

TRDI WorkHorse 300kHz LADCP, s/n 14897 (downward-looking)

The instrumentation on the underwater package operated normally on the 30 stations conducted during the cruise. Data were acquired with SeaSave V 7.22.3. Immediately after a station was completed, a

batch script was run to prepare a sound velocity profile and a CTD listing for transmission to the UK Met Office. This batch script also copied the raw data onto a network drive.

SBE processing software was then run in three steps to export as text file (.cnv), apply time alignment of oxygen data (5s for sbeox0Mm/Kg and sbeox0V; but NOT to perform hysteresis correction, which is done later in mexec processing), and cell thermal mass for conductivity ($\alpha = 0.03$, $\tau = 7.0000$ on both primary and secondary). The resulting files with suffix `_align_ctm` were copied manually to the `legdata ctd` directory.

SBE35 data were downloaded after most stations and stored in individual station files, except for stations 4, 5, 6 and 7, which were stored in a single file. Mexec processing retrieved each station from an appended SBE35 file, based on bottle times. It is therefore important to keep the SBE35 clock correct to within a few minutes.

6 CTD Data Processing and Calibration

6.1 Data Processing

The CTD data processing followed the methods used on previous SR1b and other NOC MPOC cruises, using the mexec software suite. The initial SeaBird data conversion, align, and cell thermal mass corrections were performed using SBE Data Processing, Version 7.22.2 software. The network data drive, `legdata`, was linked to `ctd/ASCII_FILES/jcrfs_ctd` and `ctd_linkscript` was used to copy files to `fola` and set up additional symbolic links to filenames following `mstar` convention. At the beginning of processing empty sample files `sam_jr306_nnn.nc` for all casts `nnn` were generated using `msam_01`, as described in the comments at the beginning of `msam_01b`.

For each cast the following m-files were then run, using wrapper script `ctd_all_part1`: `mctd_01`, `mctd_02a`, `mctd_02b`, `mctd_03`, `mdcs_01`, `mdcs_02`.

The processes completed by these scripts include

read ASCII `cnv` data from `ctd/ASCII_FILES/ctd_jr306_001_ctm.cnv`

convert variable names from SBE names to mexec names using `data/templates/ctd_jr306_renamelist.csv`

copy raw file to 24hz file

make oxygen hysteresis adjustment on 24hz file

average to 1hz

calculate derived variables `psal`, `potemp`

extract information from bottom of cast identified by maximum pressure.

Subsequently `mdcs_03g` was run to inspect the profiles and hand-select cast start and end times. The way oxygen time lag is handled in the SBE align algorithm, and the weak dependence of oxygen calculation on salinity, means that when air is ingested into the conductivity cell at the end of the cast, the oxygen becomes biased a few seconds earlier than the `psal`. Care should therefore be taken to select a cast end time for which all the important variables are free from bias. The `mdcs_03g` program was modified so that the pressure record is coloured red at times when the CTD pumps status indicated that the pumps were off. This highlighted one instance (station 15) where the pumps went off when the package was hauled too close to the surface after soaking. On that station it was necessary to delete approximately the first 27 metres of the downcast, until the pumps restarted and the sensors stabilised.

The start, bottom and end data cycles are stored in files with names like `dcs_jr306_001.nc`. After selecting the limits for start and end, `ctd_all_part2` was then run, executing `mctd_04`, `mfir_01`, `mfir_02`, `mwin_01`, `mwin_03`, `mwin_04`. The processes completed by these scripts include

Extract down and upcasts using scan numbers stored in `dcs_jr306_001`, and average into 2 dbar files (2db and 2up)

Read the `data/ctd/ASCII_FILES/ctd_jr306_001.bl` file and extract scan numbers corresponding to bottle firing events.

Add time from CTD file, merging on scan number

Add CTD upcast data (P,T1,T2,S1,S2, etc) corresponding to bottle firing events
Paste these data into the master sample file data/ctd/sam_jr306_001.nc
Load winch telemetry data from winch SCS file
Add winch wireout data to the fir_jr306_001 file
Paste winch wireout data into the master sample file

Processed data could then be examined using `mctd_checkplots` to view sensor and up-down cast differences as well as compare nearby profiles, with particular attention paid to any drift in deep temperature or salinity (expected to be relatively stable) over time. The 24-Hz data were checked for spikes in either of the temperature or conductivity sensors using `mctd_rawshow` and, if necessary, edited using `mctd_rawedit`.

A variety of extra steps is available after other processing has been carried out; these steps can be run in any order.

After LADCP processing has been completed there is a best estimate of water depth available from the LDEO IX processing. This is found in cast processing log files by searching for “bottom found at”; the list of stations and depths is placed in `station_depths/station_depths.txt`. `populate_station_depths` can be run to convert `station_depths_jr306.txt` to `station_depths_jr306.mat`, which is the file required by `mdep_01`. `populate_station_depths.m` also allows missing depths to be set or bad depths to be overwritten using `switch/case` for each cruise.

After navigation data processing has been completed the file `data/nav/seapos/bst_jr306_01` will be available. `mdecs_04` will generate files `dc_s_jr306_001_pos.nc` which include position at start, bottom and end of profiles. `mdecs_05` will then paste the position at the bottom of the cast into the header of all relevant files in `data/ctd`.

`mdep_01` and `mdecs_05` can be run multiple times. The headers will not be updated if the updating values are identical to the values already in the file.

When a conductivity calibration is available, it is applied to the 24hz files using `mctd_condcal`, as described below. A subset of scripts should now be rerun, specifically `mctd_02b`, `mctd_condcal` with `senscal = 1`, `mctd_condcal` with `senscal = 2`, `mctd_03`, `mctd_04`, `mfir_03`, `mfir_04`, `msam_updateall`. This collection of calls can usefully be put in a script like `smallscript.m`

Selection of data cycle start and end points is preserved by `smallscript`, as well as edits to the raw file made using `mctd_rawedit`. Water depth and position data will also be preserved and do not need to be re-entered after conductivity calibration.

Final processed and calibrated CTD data and bottle salinity data can be exported to CCHDO format using `mcchdo_02` and `mcchdo_01` for uploading to the CCHDO repository. Template files may need to be revised to control the list of variables exported in CCHDO csv exchange format. Temperature and salinity sections from JR306 are shown in Figure 2.

6.2 CTD Conductivity and Temperature Calibration

6.2.1 Temperature calibration

SBE35 temperature data can be logged when a Niskin bottle is fired. If the SBE35 is set to 8 samples, it requires approximately 13 seconds to make a measurement, calculated as $8 * 1.1$ seconds plus an overhead. Data are stored internally and must be downloaded at the CTD deck unit as a separate process from the CTD data transfer. The SBE35 data are then transferred as a collection of ASCII files. On jr306 these were found in `data/ctd/ASCII_FILES/SBE35`.

`msbe35_01` reads the data from a single station. The script works in `data/ctd/SBE35/`. The script requires a list of all available SBE35 data files, so on jr306 the list was called `data/ctd/SBE35/lsbe` and included

- `../ASCII_FILES/SBE35/RefThermJR306_003.asc`
- `../ASCII_FILES/SBE35/RefThermJR306_004_005_006_007.asc`
- `../ASCII_FILES/SBE35/RefThermJR306_008.asc`

msbe35_01 operates one station at a time. It reads all ascii files in the lsbe list file and extracts data cycles for the station, based on the start and end times with a 15-minute extra window in case the SBE35 clock timestamps are not perfectly accurate.

msbe35_02 pastes the SBE35 data from one station into the master sample file. msam_update_all could be used to paste the data into the appended sample file sam_jr306_all.

ctd_evaluate_temp compares the SBE35 temperature measurements taken each time a bottle is fired with the coincident measurements by the two CTD temperature sensors. The CTD and the SBE35 were found to be in acceptable agreement. CTD temp1 and temp2 agreed very closely with no apparent bias. There were a number of occasions when the SBE35 read slightly warmer than the CTD. This was attributed to the measurement being made approximately 1 metre shallower than the CTD. The median SBE35 minus CTD value suggested the SBE35 was reading slightly cooler than the CTD, by an amount smaller than 0.001°C. We concluded that there was no reason to question or adjust the CTD temperature data for either sensor.

6.2.2 Conductivity calibration

After the bottle salinity data start to appear in sam_jr306_all.nc, ctd_evaluate_sensors.m can be used to examine the difference between the CTD cond1 and cond2 sensors, and the residuals with bottle salinity. The script performs comparisons in both salinity and conductivity space. Suggested adjustments of conductivity ratio or converted to equivalent salinity offsets to judge their importance. Plots are generated to reveal biases between sensors, and either pressure- or station-dependence of bottle minus sensor differences.

When a possible calibration adjustment has been identified for either or both sensors, ctd_evaluate_sensors.m can be edited to see the effect of that calibration without changing the data in any files. A new switch/case for the cruise name should be introduced, following previous examples. If sensors have been changed, subgroups of stations corresponding to sensor configurations can be declared, using, eg, k11, k12, k13 for sets of stations with the first, second, third, primary sensor. Indexes k21, k22, k23 would correspond to successive secondary sensors.

Calibrations in ctd_evaluate_sensors.m can be tried until a satisfactory set of residuals is obtained for both sensors. This calibration adjustment can then be transferred to a switch/case in cond_apply_cal.m. A further switch is required for each sensor.

The wrapper smallscript.m can then be run on a set of stations (see section on CTD processing). This will produce calibrated CTD profiles in all the derived files (24hz, 1hz, psal, 2db, 2up) and paste the adjusted CTD data into the sam bottle files. With the exception of the ctd_jr306_nnn_raw.nc file, which still contains raw data, adjusted conductivity and associated adjusted salinity will occur in all derived files from the 24hz file onwards.

To check that the adjustment was well chosen and has been applied correctly, ctd_evaluate_sensors.m should now be edited to have the trail adjustment removed, so there is no adjustment in that script. The set of residuals now correspond to the data in the sam_jr306_all.nc file, and should have zero mean and no obvious vertical shape. On jr306, cond1 was adjusted with a simple scaling equivalent to -0.0007 salinity. cond2 had some vertical shape equivalent to salinity offset of 0.002 at the surface, interpolated linearly to 0.0005 at 3500 dbars or below. A further uniform offset of -0.0007 was then applied. cond1 and cond2 adjustments were fixed without any station dependence.

6.3 Water Sample Salinity Analysis

Salinity sample analysis was performed on the BAS Guildline 8400B Salinometer, Serial No. 65763, in the Bio Lab. The salinometer water bath temperature was set to 24°C and the laboratory temperature maintained between 20.5°C and 22.9°C.

For each CTD cast, one water sample was drawn per Niskin bottle for salinity analysis, from bottles that were fired, up to 12 per station. The distribution of bottle sample locations is shown in Figure 2. Samples were taken in 200ml glass sample bottles, which were rinsed three times and sealed with a new clean dry

disposable plastic stopper, after drying the neck of the bottle. The crate was then transferred to the bench in the Prep Lab. Caps were rinsed in freshwater and dried before being added to secure the stoppers. Samples were stored in the Bio Lab for a minimum of 24 hours before analysis to allow equilibration to the laboratory temperature. A logsheet was maintained of when crates were moved into the Bio Lab to keep track of when they would be ready to analyse. Sample drawing for calibration of the underway salinity data is described in Section 9.1.

All watchkeepers carried out the analysis following standard procedure. A sample of IAPSO Standard Seawater was run before and after each set of 24 samples for salinometer calibration. The Standard Seawater batch used was P156, with a K15 value of 0.99981, giving a 2xK15 value of 1.99968. The standardisation dial was set to 548 after the first standardisation, and not altered during the cruise. The majority of values given by the standards were in the range of 1.99971 to 1.99983, with two lower-valued outliers (ignored). The standby (SBY) and zero numbers varied little, from 6038 to 6040 and +0.00001 to +0.00003 respectively.

Eight 24-bottle crates (6 provided by NMF and 2 provided by BAS) were used for collecting salinity samples. 273 CTD cast Niskin bottle salinities and 34 underway samples were analysed.

6.4 Sample recording and merging with CTD data

Bottle sample data (conductivity and salinity) are recorded on a laptop provided by NMF using the data-logging program `autosal_2009` into an excel file to which sample numbers are manually entered in the format `sssnn` (where `sss` indicates the station number and `nn` the bottle number), giving a 5-digit number. A hard copy is also kept of bottle number, the three conductivity ratios and the average ratio. Standards are given the sample number value of `999nnn`, where `nnn` is a sequential standard number starting with 001. Standards are recorded in the Autosal logsheet but then transcribed to a standards log in order to assess Autosal drift and choose an offset (see below) for each crate. The excel file is edited to produce, for instance, Table 1. This file is then saved as a comma-separated csv file, with the name `sal_jr306_nnn.csv`, which was then copied across to `foia` and saved in `CTD/BOTTLE_SAL/`. The csv files are modified by the following shell commands on `foia`:

```
mac2unix -n sal_jr306_001.csv sal_jr306_001.csv_linux
```

```
echo " " >> sal_jr306_001.csv_linux
```

All `csv_linux` files are then concatenated to `sal_jr306_01.csv`.

Table 1: This table illustrates salinity sample data from a typical station, here using salinity standard number 14, and salinity sample bottles 164 and 165, drawn from station 012, Niskin bottles 01 and 02, respectively. The `sampnum` column is added to the spreadsheet generated by the `autosal` software as column 10, and sample numbers are entered manually from the logsheets. (The date and time columns are omitted from this example, but not from the real files.)

Bottle number	Sample 1	Sample 2	Sample 3	Average	offset	Salinity	sampnum
CTD012_9014	1.999584	1.999638	1.99959	1.999604	-0.000013	34.992	999014
CTD012_164	1.976222	1.9762	1.976221	1.976214	-0.000013	34.5321	01201
CTD012_165	1.974472	1.974491	1.9745	1.974488	-0.000013	34.4982	01202

Offsets to adjust for standardization of the `autosal` are chosen by station or crate and entered into `msal_01` using `switch/case` to set the offsets for a particular cruise. An array `g_adj` consists of pairs of rows defining the station number and sample number ranges (e.g. 100 and 599 for stations 1 to 5, 1700 and 1799 for station 17, etc.), bath temperature, and the offsets chosen for each station number range. The offsets used on this cruise are given in Table 2.

We then run `msal_01` to read in the concatenated bottle samples and extract the sample data for each station based on the sample number, generating `sal_jr306_nnn.nc`. The file `msal_02` pastes this information to `sal_jr306_nnn.nc`.

Table 2: Autosal offsets by CTD station or TSG crate. An offset of -5 means the Guildline Autosal reading should be adjusted by -0.00005.

Stations	Bath temperature	Offset
1-5	24	-5
6-10	24	-10
11-12	24	-13
13-16	24	-8
17	24	-5
18	24	-7
19	24	-10
20	24	-12
21	24	-13
22	24	-10
23-24	24	-9
25-27	24	-11
28-30	24	-7
TSG 1	24	-10
TSG 2	24	-9

7 Lowered Acoustic Doppler Current Profiler (LADCP)

7.1 Instrument setup and performance

The 300-kHz Workhorse LADCP was installed in a downward-looking configuration on the CTD rosette for JR306. The configuration (Table 3) was 16 x 10m bins, with data collected in beam co-ordinates and rotated to earth co-ordinates during processing. The LADCP was connected to a charger and by a serial cable to a BAS AME-supplied laptop in the Chem Lab for programming prior to each station and data download after each station, using BBTalk. Pre-deployment tests were performed approximately once daily. Data downloaded after each station were copied to the network legdata drive in ladcp/JR306, with names of the form JR306_NNNm.000, along with deployment logs, JR306_NNNm.txt.

7.2 Data processing

Data for each station were processed on fola using two different software packages. The first software package was developed at the University of Hawaii (UH) and computes shear, then using integrated instrument-relative velocity and ship navigation to compute the depth-averaged velocity reference. The second package was developed at the Lamont-Doherty Earth Observatory (LDEO), and uses an inverse method to calculate velocity profiles, optionally including LADCP bottom tracking and/or VMADCP upper ocean velocities as constraints. LDEO (IX) processing was performed using only ship navigation (ladcp/ix/DL_GPS), navigation and bottom tracking (ladcp/ix/DL_BT), and navigation, bottom tracking, and VMADCP (ladcp/ix/DL_SADCP).

The LADCP profiles from the LDEO processing, including shear method (relative to the depth mean), inverse including navigation data, and inverse including navigation data and bottom track are shown in Figures 3 to 5. VMADCP profiles are also shown for comparison.

7.2.1 Data processing setup

A single raw data directory on fola is used for both IX and UH processing. Each processing suite can access the raw files by making links to them. Raw data are copied onto fola in cruise/data/ladcp/raw/, where there is a link to the ladcp directory on legdata: rawdata → ../jcrfs/current/ladcp/JR306/. This link may need to be edited to get the detailed filename correct on each cruise. There is a unix script

Table 3: LADCP deployment commands, contained in JR306_ladcp.cmd.

```

CR1
RN JR306
WM15
TC2
LP1
TB 00:00:02.80
TP 00:00.00
TE 00:00:01.30
LN25
LS0800
LF0
LW1
LV400
SM1
SA011
SB0
SW5500
SIO
EZ0011101
EX00100
CF11101
CK
CS

```

exec/lad_syncscript which uses rsync to make sure the raw directory is up to date, using cd cruise/data/ladcp/raw, rsync -av rawdata/*. Two linkscripts are available, lad_linkscript_ix and lad_linkscript_uh. Each linkscript starts by calling lad_syncscript, and then makes links from the raw data directory for that data processing suite to the raw files in cruise/data/ladcp/raw/.

lad_linkscript_ix: IX processing uses raw filenames like 001DL000.000 in directory 001/, so the script links ladcp/ix/data/raw/rawdata → ../../../../ladcp/raw and ladcp/ix/data/raw/001/001DL000.000 → ../rawdata/JR306_001m.000. lad_linkscript_ix will need to be edited for each cruise depending on the names of the raw files downloaded from legdata, eg JR302M_???.000 became JR306_???.000.

lad_linkscript_uh: UH processing uses raw filenames like j001_02.000, so the script links

ladcp/uh/pro/jr1501/ladcp/proc/Rlad/rawdata → ladcp/raw and

ladcp/uh/pro/jr1501/ladcp/proc/Rlad/j001_02.000 → rawdata/JR306_001m.000. lad_linkscript_uh will need to be edited for each cruise depending on the names of the raw files downloaded from legdata, e.g. JR302M_???.000 became JR306_???.000.

7.2.2 Linking to CTD data

Each processing suite also expects CTD filenames to have a certain form. 1-Hz CTD data can be exported from mexec into ascii using the mexec command list_ctd_1hz(nnn) for station number nnn. This exports the variables time, press, temp, psal, latitude, longitude. Note that lat and lon are required for the IX processing, and become the source of navigation data for that suite. If list_ctd_1hz has been run from Matlab, then ladctd_linkscript_ix, ladctd_linkscript_uh will make those ascii files available to the IX and UH processing in the correct locations and with the correct file names. In IX, links to CTD data are made in ladcp/ix/data/CTD/1Hz/ctd → ctd and ladcp/ix/data/CTD/1Hz/ctd_jr306_028_1hz.txt → ctd/ctd_jr306_028_1hz.txt. ladctd_linkscript_ix should not require modification for each cruise, but ladctd_linkscript_uh will require modification to include the correct cruise directory name in the ladcp/uh directory tree.

7.2.3 Linking to VMADCP data

A link was made to the VMADCP directory. The codas/mexec processing produced single profiles of VMADCP data acquired during each station. `ladcp/ix/data/SADCP` → `vmadcp/jr306_os75`. A typical filename of data in the `vmadcp` directory was `ladcp/ix/data/SADCP/os75_jr306_ctd_028.mat`.

7.2.4 Edits to .m files required at start of a cruise for IX processing

In addition to setting up directories, links and linkscripts, the `set_cast_params*.m` files must be edited with cruise details. These scripts reside in `data/ladcp/ix/data/`.

Edits required (lines from `jr306_set_cast_params.m` shown):

```
f.sadcp = sprintf('SADCP/os75_jr306_ctd_%03d.mat',stn);
```

```
f.ctd = sprintf('CTD/1Hz/ctd_jr306_%03d_1hz.txt',stn);
```

```
p.cruise_id = 'jr306';
```

```
p.whoami = 'B. King';
```

The different versions of `set_cast_params*.m` are called by different versions of `process_cast*.m` to run IX processing of the LADCP data with different constraints:

- `process_cast_v5.m` calls `set_cast_params_v5.m` to include only navigation data
- `process_cast_v4.m` calls `set_cast_params_v4.m` to include navigation and bottom tracking data
- `process_cast_v3.m` calls `set_cast_params_v3.m` to include navigation, bottom tracking, and VMADCP data

`set_cast_params.m` (and thus `process_cast.m`) are currently set to use v5 (navigation only).

7.2.5 IX processing

Quick look notes for LADCP processing in IX

1. Make sure the raw file is on `legdata`:

```
cd /cruise/data/ladcp/raw  
ls ld rawdata/* (see files on legdata)  
ls ld * (see files on fola)
```

The latest station should have a name like `rawdata/JR306_002m.000`.
2. Run unix script `lad_linkscript_ix`. This will find any files in `legdata` that aren't on `fola`, copy them across to the correct place on `fola` using `rsync`, and create links in the IX processing tree so that IX is aware of the raw files: `ladcp/ix/data/raw/002/002DL000.000` → `../rawdata/JR306_002m.000` and so on. This script can be run from anywhere, and will echo to the screen the identity of any files copied and any links made. Check again using step 1 if you wish.
3. Run unix script `ladctd_linkscript_ix`. If the CTD 1-Hz ASCII file is available, this will make a link in the IX processing tree so that IX is aware of the CTD data:
`ladcp/ix/data/CTD/1Hz/ctd_jr306_nnn_1hz.txt` → `ctd/ctd_jr306_nnn_1hz.txt`. IX can be run without CTD data for first-look checking, but it will produce an error in calculating the magnetic variation, because position is unknown and is found in the CTD 1-Hz file. If you know the CTD file is not available, skip this step.
4. If you intend to run `process_cast_v3` to use VMADCP data, check that `ladcp/ix/data/SADCP/os75_jr306_ctd_028.mat` is available.
5. Run the IX process:

```
cd ladcp/ix/data or cd /cruise/data/ix (an abbreviation: ix → ladcp/ix/data)  
matlab &  
>> ixpath  
>> process_cast(nnn), where nnn is the station number.
```
6. Repeat steps 3, 4 and 5 as CTD and VMADCP data become available.

7. Note that on jr306 there are three versions of process_cast available.
 process_cast_v5 calls set_cast_params_v5, and processes data into DL_GPS/. Navigation is used but no BT or VMADCP constraint.
 process_cast_v4 calls set_cast_params_4, and processes data into DL_BT. Navigation and BT are used but no VMADCP constraint.
 process_cast_v3 calls set_cast_params_v3, and processes data into DL_BT_GPS. Navigation, BT, and VMADCP are all used.

7.2.6 UH processing

Quick look notes for LADCP processing in UH

The initial stages of processing allow the user to examine the quality of the data and to calculate relative velocity profiles in the absence of CTD data.

1. Run unix script lad_linkscript_uh.
2. After navigating to the directory /cruise/data/ladcp/uh, source LADall sets up the paths required for the processing.
3. cd proc; perl S scan.prl NNN_02 to scan the raw data and create a station specific directory in the proc/casts directory. Data printed to screen should be checked to ensure the details of the cast (i.e. depth, downcast/upcast times) agree approximately with the CTD log sheet.
4. matlab; m_setup; putpos(NNN,02) gets position of the cast by accessing the SCS datastreams. magvarsm(NNN,02) obtains the required magnetic correction to the compass on the LADCP. Quit matlab.
5. perl S load.prl NNN_02 loads the raw data, using magvar.tab for the magnetic correction to load data. It is very important that this step is only carried out once. If it needs to be repeated the database files (/proc/casts/dNNN_02/scdb) must be deleted first.
6. perl S domerge.prl c0 NNN_02 to merge the velocity shear profiles from individual pings into full upcast and downcast profiles. The option c0 refers to the fact that CTD data has not yet been included.
7. cd Rnav; matlab; make_smscs makes a smoothed navigation file for the cast. Quit matlab. A modified version of make_sm will be needed on techsas ships.
8. matlab; plist = NNN.02; do_abs; calculates the relative velocity profiles. Check that these plots look sensible, i.e. reasonable agreement between downcast and upcast and that the vertical velocity changes sign between downcast and upcast (it may be necessary to rescale some of the plots).
 Once the CTD data has been processed this can be incorporated into the LADCP processing to make more accurate estimates of depth and sound velocity and to obtain a final absolute velocity profile.
9. The inclusion of CTD data requires an ASCII file containing 1-Hz CTD data for the station. If CTD processing has been completed, ladctd_linkscript_uh will make it available in proc/Rctd.
10. cd proc/Pctd; ctd_in(NNN,02) will read the 1-Hz CTD data in. plist=NNN.02; fd aligns the LADCP and CTD data sets in time. Quit matlab.
11. cd proc; perl S add_ctd.prl NNN_02 adds the CTD data to the *.blk LADCP files in scdb/.
12. perl S domerge.prl -c1 NNN_02 merges the single pings into corrected shear profiles. The -c1 option now states that we have included CTD data.
13. matlab; plist=NNN.02; do_abs; calculates the velocities again with the merged pings.

7.3 Data quality

The LDEO inverse velocity solutions generally agree with shear method shear solutions and VMADCP currents, but for more than half of JR306 stations (10-15, 18-23, 26-29) either the barotropic reference or the low-wavenumber shear or both differed noticeably. The incorporation of bottom tracking in the inverse solutions improved agreement with the upper ocean VMADCP velocities only some of the time. We advise care in interpreting the LADCP data.

8 Computing and SCS data processing

8.1 Configuration of linux workstation 'fola'

The NOC MPOC OCP group brought a linux workstation (fola), which was the primary platform for data analysis during the cruise. The jcr cruise data directory was made available by mounting on fola. That directory includes SCS data streams, data from other sources such as CTD, LADCP, VMADCP, and the legwork directory. The network data directory was mounted on fola so that /mnt/data/cruise/jcr was the parent directory of the individual cruise data directories identified by date. Cruise jr306 was current → 20150109.

A link was made on fola in /local/users/pstar/cruise/data so that data/jcrfs → /mnt/data/cruise/jcr. This enabled access to the legdata directory as well as the legwork part of it. Since the current/ cruise directory sits below /mnt/data/cruise/jcr, useful directories could be accessed using the following example links:

data/ctd/ASCII_FILES/jcrfs_ctd → ../../jcrfs/current/ctd/JR306

data/legwork → jcrfs/current/work

data/vmadcp/jcrfs_adcp75 → ../jcrfs/current/adcp

data/scs_raw_ship → jcrfs/current/scs/Compress

If the mount point of the legdata directory changes in future cruises, it should be necessary only to change the data/jcrfs link to point to the correct location on the network. All primary data were acquired and backed up on legdata. Underway and time series data were logged into SCS files. CTD, LADCP and VMADCP data were archived in instrument-specific directories, in binary files generated by the data acquisition software of those instruments.

Workstation fola was backed up twice per day. A complete dump of cruise data and software was copied using rsync from fola to one of four Transcend portable hard drives. Two out of four available disks were in use at a time, and these were periodically rotated during the cruise. These drives were used to carry data back to NOC at the end of the cruise, including a final identical backup of fola on all of the four drives.

8.2 SCS data streams

The SCS data streams (ashtech [nav/ash], ea600 [sim], anemometer [met/surfmet], oceanlogger [ocl], emlog-vhw [chf], gyro [nav/gyros], seatex-gll [nav/seaapos], em122 [em122], seatex-hdt [nav/seahead]) were processed on fola during the cruise. Most were processed in 24-hour segments, using m_jr306_daily_processing.m, with cleaning and appending as required. Winch data were processed by CTD station as part of standard CTD processing (ctd_all_part2.m).

8.3 Access to SCS data

SCS data are stored in *.ACO files, which are plain text. However, non-numeric characters make these slow to parse in Matlab. An ad hoc but fairly robust and quick-running system has evolved to make these data more easily readable for mexec processing.

1. Parse the ACO files with linux sed, to remove non-numerical characters. Input SCS files are in data/scs_raw. On the ship and during the cruise, data/scs_raw → scs_raw_ship, data/scs_raw_ship → jcrfs/current/scs/Compress. At the end of the cruise, the contents of scs_raw_ship are copied to a local directory, data/sca_raw_postcruise. Then the link is changed so data/scs_raw → scs_raw_postcruise, so that raw SCS files can be accessed after the cruise.

The sed parsing allows continually-filling SCS files to be continually edited into data/scs_sed/. Parsing is started and stopped with linux commands sedexec_startall and sedexec_stopall. If there is a problem with any aspect of sed processing, the safest action is to delete all of scs_sed/*ACO and stop and restart sedexec processing.

The sedexec scripts live in data/exec/jcr. sedexec_startall needs a list of ACO files to parse. This can be generated by a Matlab script, data/exec/jcr/make_list_for_sedexec_startall.m. This will

make list_jr306, linked to by data/exec/jcr/list.

SCS files note 1 The SCS data consist of .ACO files and .TPL files. The TPL files contain lists of variables. When a cruise begins, the data/scs_raw/*.TPL should be copied to data/scs_sed. oceanlogger.TPL needs to be manually edited, as oceanlogger.ACO has time information for which the number of fields changes when parsed through sed. The original oceanlogger.TPL begins thus:

```
126,oceanlogger-sampletime,YYY DDD HH:MM:SS
```

```
127,oceanlogger-airtemp1,celsius
```

```
128,oceanlogger-humidity1,
```

```
129,oceanlogger-par1,umol/S.m2
```

so the first line must be replaced with 3 lines thus:

```
126,oceanlogger-sampletimeyyyy,YYYY
```

```
126,oceanlogger-sampletimedddd,DDD
```

```
126,oceanlogger-sampletimehhmmss,HH:MM:SS
```

```
127,oceanlogger-airtemp1,celsius
```

```
128,oceanlogger-humidity1,
```

```
129,oceanlogger-par1,umol/S.m2
```

SCS files note 2 The default SCS variable names include the name of the data stream for each variable. e.g. oceanlogger-humidity1. The daily processing of SCS files removes the extra part of the variable name. The new names need to be in lookup template .csv files in data/templates. For example the first few lines of the oceanlogger template, data/templates/scs_jr306_renamelist_oceanlogger.csv, are

```
oceanlogger_sampletimeyyyy,sampletimeyyyy,YYYY
oceanlogger_sampletimedddd,sampletimedddd,DDD
oceanlogger_sampletimehhmmss,sampletimehhmmss,HH:MM:SS
oceanlogger_airtemp1,airtemp1,celsius
oceanlogger_humidity1,humidity1,
```

The set of required template files can be made in Matlab with

```
ms_generate_varname_translation_all.m, which makes successive calls to
```

```
ms_generate_varname_translation(stream). If this script fails, templates can be copied from a previous cruise. The template contents will be unchanged unless the variables in the SCS file have changed, e.g. a new sensor in the oceanlogger stream.
```

2. Convert ACO text files to Matlab.

Even the clean and numeric ACO files are too slow to routinely read into Matlab to access data 'on demand'. Therefore a process is run in which Matlab versions of the SCS data are accumulated into data/scs_mat/. In a Matlab window, the command `ms_update_aco_to_mat('ea600')` will read data from data/scs_sed/ea600.ACO and save the data in data/scs_mat/ea600.mat. This process also notes the number of bytes of the ACO file converted. Next time the process is run, those bytes are skipped so that only new data are converted.

The command `update_allmat` will update all the SCS streams with multiple calls to `ms_update_aco_to_mat`. `update_allmat` will need to be edited to add new streams if new SCS streams are required, otherwise it does not require any modification at the start of a cruise. Some Matlab scripts (eg `mday_00.m`) that read data 'up to the present' call the update script before loading data. Sometimes it is necessary to run the update script manually from the command prompt, to ensure all ACO data are available to following scripts.

If the scs_sed files have got into a mess and been regenerated or scs_sed parsing restarted, the safest course of action is to delete the contents of scs_mat and run `update_allmat`. This takes a few minutes at the start of a cruise and rather longer, maybe up to half an hour, if 30 or 40 days data are waiting to be parsed. This is necessary because the scs_mat file keeps a count of the number of bytes of the ACO file parsed. If the ACO file has been altered before that byte count,

the ms_update_aco_to_mat process may fail to copy the data correctly, but may not give an error.

8.3.1 The mscs source directory

The mscs source directory cruise/sw/mexec_v2/source/mscs contains the scripts for accessing the SCS data either directly or via scs_mat. Some of the scripts, e.g. scs_to_mstar2.m are called by wrapper scripts such as mday_00.m. Some of the scripts can provide quick access to SCS data that have not yet been processed into mexec files. Examples include: msgaps, mslast, mslistit, msload, mslookd (usually run as mslookd f), msnames, msposinfo, msposinfo_noup (quick version for finding ship position at earlier time so not necessary to update scs_mat/ first), msvars. See help mscs for further information. The techsas equivalent library has help mtechsas which has extra documentation and help.

9 Underway temperature and salinity data

9.1 Underway salinity calibration

A total of 34 underway samples were analysed for oceanlogger salinity calibration. Samples were drawn from the underway supply in the Prep Lab, nominally every 4 hours, and the time noted in a logsheet to the nearest minute.

Samples were analysed following the procedure described for CTD salinity samples. Crates of these samples were identified as "jr306 tsg001_" and so on for successive crates. In the xls spreadsheet, the column-10 sampnum column was added. The edited xls files were named tsg_jr306_001.csv. Standards were given sampnums following the usual convention: Sampnum = 999011 for standard 011, Sampnum for a sample is the time of sampling in the form ddhhmmss, where ss is 00 because seconds are not recorded. Thus CTD samples have sampnums in the range 101 to 99924, standards in the range 999001 to 999999, and underway samples in the range 001000000 to 366235900. Each type of analysis (CTD, standard, underway) can be identified based on the sample number.

All underway salinity sample files are handled in CTD/BOTTLE_SAL/. The csv files are converted using mac2unix in a shell window, and a line feed added:

```
mac2unix n tsg_jr306_001.csv tsg_jr306_001.csv_linux
```

echo " " ; tsg_jr306_001.csv_linux The tsg files are then appended into the master salinity sample file, sal_jr306_01.csv. Underway samples are identified by sampnum, as described above. mtsg_01.m loads underway samples and applies the standardization offset. A switch/case for the cruise declares the bath temperature and Guildline offset for each crate. mtsg_01 prompts for a crate number to be extracted and analysed: this creates for example data/ocl/tsg_jr306_001.nc. At the end of mtsg_01, all available tsg_jr306_???.nc files are appended into data/ocl/tsg_jr306_all.nc, which is now ready for examining the oceanlogger salinity calibration.

The list of oceanlogger variables is as follows:

mexec directory	mexec short name	mexec directory abbreviation	mexec file root	variables
ocl	oceanlogger	M_OCL	ocl	tstemp
				conductivity
				salinity
				sound_velocity
				chlorophyll
				sampletemp
				flowrate
				sstemp
				trans
				sstemp2

Note the alternative temperatures: sstemp and sstemp2 for seawater intake, sampletemp for fluorometer sample temperature, ttemp for the SBE housing temperature for calculation of salinity.

9.2 Underway surface thermosalinograph

As noted above, the calibrated TSG record is in ocl/. First note that no calibration or data cleaning has been applied to the fluorometer or transmissometer data, except that it has sometimes been set to absent where the salinity is absent. Data were set to absent at any time that the pumps were known to be off, either close to port or where the flowrate indicated there was little or fluctuating supply.

Data were read in as part of the daily processing. Relatively simple cleaning and calibration has been applied to the appended dataset, following the path worked out on jc069. The full processing sequence was thus

1. `mtsg_medav_clean_cal.m`
calls `mtsg_cleanup.m`
input: `ocl_tsg_jr306_01`
output: `ocl_tsg_jr306_01_medav_clean`
creates 1-minute median bin average of data, then removes known bad data in `mtsg_cleanup`. `mtsg_cleanup` has times of pumps off hardwired in for each cruise. In addition it loads times of bad data from file `ocl/bad_time_lims.mat`. This file is added to by `mtsg_findbad`. Between the times identified as bad, all variables are set to NaN. This is because this procedure was originally conceived as a means of discarding data when the pumps were off, not merely when salinity had a problem. Note that in fact this step only cleans out bad data; no calibration is performed until step 5 (the name is retained for backwards compatibility).
2. `mtsg_findbad.m`
allows graphical identification of bad data using an interface similar to `mdcs_03g`. See help `mtsg_findbad`. Note the use of 'n' to store the start and end of bad data and move on to the next segment of bad data is CRITICAL. This script should be run at least once, if only to initialise the `bad_time_lims` file.
3. After `mtsg_findbad`, you can loop through steps (1) and (2) as often as required, until a sufficiently clean salinity file is obtained. Limits of bad times are accumulated by successive uses of `mtsg_findbad`, so it is a good idea to back up the `bad_time_lims.mat` file. You **must** run `mtsg_medav_clean_cal` one final time after the run of `mtsg_findbad`.
4. `mtsg_bottle_compare_jr306.m`
input: `ocl_jr306_01_medav_clean` or
`ocl_jr306_01_medav_clean_cal`
output: `ocl_jr306_01_botcompare.nc`
merges the clean 1-minute data onto bottle samples. The script is set up to make it easy to switch between uncalibrated (variable `salin`) or calibrated (variable `salin_cal`) data (by setting the case switch at the start of the .m file). Individual bottle residuals are plotted, as well as a smoothed time series of the residuals, which can then be used as a slowly-varying adjustment to the TSG salinity. Figure 6 shows the resulting comparison for the entire set of surface bottles (34 samples, taken nominally every four hours). The choices of `cal` and `uncal` make it easy to check after step 5 below that the final calibrated data have zero-mean residuals.
5. `mtsg_apply_salcal.m`
calls `mtsg_salcal.m`
input: `ocl_jr306_01_medav_clean`
and `ocl_jr306_01_botcompare.nc`
output: `ocl_jr306_01_medav_clean_cal`
This step applies a salinity calibration to the clean file and makes a `clean_cal` file. The subroutine `mtsg_salcal.m` can be used to create any calibration procedure required, selected by `switch/case` on the cruise number. On jr306 a simple adjustment of +0.005 was applied to the whole dataset. On jr302 a time-varying adjustment was created by making a smoothed version of the sample

residuals contained in `ocl_jr306_01_medav_clean_botcompare.nc`. On jr302, the bottle residuals were passed through successive filters to remove outliers, until an acceptable smoothed adjustment was reached. This was then interpolated on time onto the `ocl_jr302_01_medav_clean` time series. Step 4 can be repeated after this step, to check that the residuals are now acceptable. Figure 6 shows residuals with calibrated data; the residuals have median 0.0004 and iqr 0.0047.

BAK

10 Daily processing of SCS streams

As part of the regular watchkeeping logging every 4 hours, we checked that the SCS streams were updating with reasonable values. Data were transferred from the onboard logging system (SCS) to fola daily using `mday_00_get_all.m`. This script also appends daily data files for streams other than bathymetry (see below).

10.1 Navigation

The list of navigation streams processed from SCS was given in Section 8.2. After daily processing of SCS streams, navigation processing steps can be completed on the appended `xxx_jr306_01.nc` files. The steps are combined into a single wrapper `mbest_all.m`. This script performs tasks such as subsampling to 30-second intervals, adding ship speed and course over ground, and merging on ship heading. Ship headings are properly vector averaged when subsampled. A best navigation file, `data/nav/seapos/bst_jr306_01.nc`, is the final output.

10.2 Surface meteorological sampling (SURFMET)

RRS *James Clark Ross* is equipped with a variety of meteorological sensors to measure air temperature and humidity, atmospheric pressure, total irradiance, photosynthetically active radiation, wind speed and wind direction throughout the cruise. The radiation and pressure variables were logged in `data/met/surflight/`. The remaining data were logged in `/met/surfmet`. The raw data files have extensions of the form `_jr306_dNNN_raw.nc`, where NNN represents the day number.

Once the best navigation file is available, `mtruew_01.m` is run to combine ship navigation data with relative data from the anemometer to produce absolute and relative winds averaged to 1-minute resolution. As with best navigation, all headings are properly vector averaged, and the final product is the file `data/met/surfmet/met_jr306_trueav.nc`.

10.3 Bathymetry

The raw data files have extensions of the form `_jr306_dNNN.nc` where NNN is the number of the Julian day. `msim_01.m` was run to remove data outside a tolerated range and apply a 5-minute median despiking, outputting the file `sim_jr306_dNNN_smooth.nc`. `msim_plot.m` copied the smoothed sim data to the file `sim_jr306_dNNN_edited.nc` and called `mplxyed` to allow a manual removal of the remaining spikes in sim data; `mem122_plot.m` was run to do the same for the em122 data (note that each script must be used to edit the correct data type; sim must be edited in `msim_plot` and em122 in `mem122_plot`). Although both swath and EA600 data quality was generally acceptable there was a period of several hours where the echosounder and swath were interfering with each other, as well as times when the echosounder range was not properly adjusted or the echosounder detected slightly below the bottom.

Two bathymetry streams are available via SCS: the ea600 single-beam echo sounder, and the centre beam of the em122 multibeam (swath) system. Data are read in by the daily processing (`mday_00_get_all.m`); for day number 014 (for instance), ea600 and em122 data are read into `sim/sim_jr306_d014.nc` and `em122/em122_jr306_d014.nc`, respectively. For streams other than bathymetry, daily processing appends each day onto a growing appended file, but the bathymetry records require manual editing before they can be appended, so there is no append action for sim or em122 in the daily processing.

The bathymetry streams inevitably contain bad data where the automatic digital determination of depth has failed to correctly infer depth from the analog signals. As the original analog records are not generally available to the person doing data editing/cleanup, some judgement is required to decide what are good or bad data. In order to aid this process, there is a display of both of the two data streams, along with an estimate of bathymetry from satellite gravity data. Thus the ea600 data are edited by reference to em122 data, and vice versa. There is a symmetric set of scripts for cleaning the two data streams.

The first step is to do some automated cleanup of both the ea600 and em122 streams, with msim_01 and mem122_01. Both must be run before proceeding with manual graphical editing.

10.3.1 EA600

For ea600, the process is as follows:

- msim_01
reads raw data for the day (from e.g. sim_jr306_d014_raw.nc);
picks data in the depth range 5 to 10000, to discard zeros;
takes the median depth over 300-second bins, to discard noise (example output file: sim_jr306_d014_smooth.nc)
makes duplicate file in sim_jr306_d014_edited.nc
- msim_02
merges on swath centre beam data for the day (if available) to be used in cleaning. Note that mem122_01.m must be run before this step.
- msim_plot
calls mplxyed for manual removal of bad data. Note that the files are set up so this file is used to edit bad data in the ea600 file. The equivalent file in the em122 sequence should be used to edit bad data in the em122 stream.
msim_plot displays an extra window with the full ea600 data, to guide the use of mplxyed. A 'quick and dirty' edit was made to this plotting code to add satellite bathymetry and em122 centre beam depths. Positions for the satellite bathymetry are loaded on the fly from seapos, culled to approximately 5-minute intervals, and depths retrieved using pos_to_ssdep. Em122 data are loaded on the fly from the SCS em122 file, and displayed with depths less than 50 m discarded. This script would benefit from an overhaul to make its approach more consistent with other mexec scripts. The 'spare' window is useful for Matlab zooming, as well as for comparing all available estimates of seabed depth.
EA600 data quality: when the ea600 is noisy, it is often very difficult to infer just from the digitised records whether the instrument was finding the bottom or simply returning random noise close to the last known good depth. Accordingly the mplxyed editing was done in a fairly harsh manner, preferring to discard data which seemed noisy and for which there was no good evidence that the data were correct. The dataset therefore has many gaps.
- After cleaning in mplxyed, daily files were appended into a single file, sim_jr306_01.nc. At this stage, the em122 depths used for data quality checking are discarded.
- Next, navigation was added, in sim_jr306_01_nav.nc, and a Carter area correction applied, using mcalc, in sim_jr306_01_nav_cordep.nc. This last file is therefore the master EA600 file, containing corrected bottom depths in 5 minute bins.

10.3.2 EM122

Em122 centre beam depths were logged in SCS and downloaded in daily files to em122/. Two scripts are used to reduce and edit EM122 data, based on the scripts for EA600:

- mem120_01
reads raw data (from e.g. em122_jr306_d014_raw.nc),
runs mdatpik to require depth \geq 20 m,
runs mavmed to take the median over 300-s bins, producing em122_jr306_d014_smooth.nc, and

copies the file to an 'edited' version (em122_jr306_d014_edited.nc) ready for manual editing.

- mem120_02
merges on ea600 for display in mplyed for QC.
- mem120_plot
plots data for editing: if all the relevant files exist, displays the Matlab window with EM120, EA600 and satellite bathymetry, then enters mplyed on the edited file. This step can be repeated as often as required. The Matlab plot window displays the data as found in the most recently updated 'edited' file.

10.3.3 Merging of EM122 and EA600

Since each data stream has some good measurements where the other's are absent, the best available estimate of cruise bathymetry requires the two streams to be merged. This task was not undertaken at sea.

10.4 Chernikeef EM log

Chernikeef EM log data were recorded and downloaded daily. Besides appending the files, no further processing was carried out and the calibration of the EM log was not checked.

11 Vessel Mounted ADCP

A vessel-mounted 75-kHz Teledyne RD Instruments (RDI) OceanSurveyor Acoustic Doppler Current Profiler (ADCP) was run throughout the cruise to measure horizontal velocity from 30 m to approximately 400-700 m. The depth of the transducer is 5 m. The phased-array transducer should be insensitive to sound speed (temperature) changes (but see discussion below). The data may be particularly affected by bubbles under a certain range of ship heading relative to the seas, which appears to have dramatically reduced data quality at a few of the CTD stations.

11.1 Real-time data acquisition and VMDAS files

The data from the instrument were acquired using the RDI VMDAS software package version 1.42, installed on a PC in the main laboratory. The software allows data acquisition in a number of configurable formats and performs preliminary screening and transformation of the data from beam to Earth coordinates. A perl script to subsample ship navigation data, NavigationRepeater, must be run on the same machine throughout data collection. VMDAS was run using a selection of preset configuration files (selected under "edit data options") which allowed the instrument to be promptly switched between bottom tracking and water tracking; set for different depth ranges; and self-triggered or triggered by the K-sync unit (see below) so as not to interfere with other acoustic measurements. More detail on settings is given below. Data collection was typically stopped and restarted, generating a new file segment number, daily during the cruise, to facilitate incremental processing. Files were automatically placed on the network legdata/adcp directory and rsynced to fola daily using vmadcp_linkscript.

The files produced have names of the form JR306 os75nnn_mmmm.ext, where nnn is the file sequence number, mmmm is the number of the segment file within the sequence and ext is the extension. A new segment file is automatically started when one of the file types (see below) reaches a size of 10 Mb. VmDas automatically increments the file segment number every time data collection is stopped and restarted.

The list of files produced is given below:

- .ENR files are the binary raw data files (beam coordinates).
- .ENS files are binary ADCP data after being screened for RSSI and correlation and with navigation data included.

- .ENX files are ADCP single ping data and navigation data after having been bin-mapped, transformed to Earth coordinates and screened for error velocity and false targets.
- .STA files are binary files of short-term average ADCP data (120 s, user-specified in VmDas).
- .LTA files are binary files of long-term average ADCP data (600 s, user-specified in VmDas).
- .N1R files are ASCII text files of raw NMEA navigation data from the NMEA1 stream.
- .NMS files are binary files of navigation data after screening.
- .VMO files are ASCII text files specifying the option settings used for the data collection.
- .LOG files are ASCII text files logging all output and error messages.

The 'R', 'S' and 'L' tabs on the VmDas menu bar allow the user to swap between graphical output from the .ENR, .STA and .LTA files. When in 'R' mode, the default upper left hand display in VmDas is the raw velocity parallel to each beam, while 'S' (short-term average) and 'L' (long-term average) modes display currents averaged over 2 or 10 minutes, in Earth coordinates. Various display options are available but data were generally monitored in either raw or short-term average form, and checked in more detail following daily processing. During the every-4-hours regular watchkeeping tasks the VMADCP ensemble number was checked to ensure the expected amount of data recording. As described in the caption of Table 4, which lists VMADCP file sequence start and end times, the ensemble number reset itself at various points, for unknown reasons (i.e., not wrapping over at any obvious or consistent number). We did not check if this led to any missed pings; 5-minute ensemble averages did not appear to be affected.

Table 4: VMADCP file sequences with start and end times (Julian day and time) and brief configuration notes. Starting with 040, K-sync triggering (indicated by K-s in the configuration column) produced a set of 3 pings, 3 seconds apart, then a 6-second gap to allow the EM122 and EA600 to ping (such that the VMADCP produced 3 pings per 12 seconds). Ensembles refers to number of single-ping ensembles. ¹The ensemble counter reset, for unknown reasons, during sequence 040, segment 003, sequence 041, segment 003, sequence 042, segment 002, sequence 043, segment 001, and sequence 045, segmen 001. The ensemble totals listed for these sequences are the minimum totals and estimated totals based on duration.

sequence	segment range	ensembles	start time	end time	configuration
037	000-004	41791	010 12:25:19	012 00:05:02	BT, 500 m
038	000-002	16375	012 00:08:05	012 11:13:50	WT, 800 m, K-s
039	000	1254	012 11:16:57	012 15:26:45	BT, 800 m, K-s
040	000-003	23546-24660 ¹	012 15:27:30	013 12:57:49	WT, 800 m, K-s
041	000-003	20566-23990 ¹	013 12:59:34	014 11:20:01	WT, 800 m, K-s
042	000-003	23537-25760 ¹	014 11:22:08	015 11:07:07	WT, 800 m, K-s
043	000-003	20529-23790 ¹	015 11:08:05	016 11:01:31	WT, 800 m, K-s
044	000-003	24028	016 11:01:47	017 11:04:24	WT, 800 m, K-s
045	000-003	17340-20121 ¹	017 11:04:38	018 06:45:02	WT, 800 m, K-s

11.1.1 Settings

The VMADCP was run in single-ping mode, with a 16-m pulse, 50 16-m bins, and an 8-m blanking distance. At the beginning of the cruise, in shallow water (<500 m generally), the instrument was run in bottom track mode to obtain phase and amplitude calibrations. Once we reached deeper water in the Bransfield Strait, on 12 January, the instrument was switched to watertrack mode, and watertrack calibration data were obtained during CTD station-keeping. For most of the cruise the VMADCP was set to be triggered by the K-sync system in order not to interfere with the EM122 or EA600. As the EM122 trigger frequency depends on water depths, the VMADCP ended up pinging 3 times every 11.9 seconds, with the pings grouped together separated by 3 seconds.

11.2 Post-processing

The final onboard processing of the data was done using the (deprecated) Matlab version of the CODAS (Common Ocean Data Access System) suite of software provided by the University of Hawaii. This suite of shell, perl, python, and Matlab programs allows manual inspection and editing of bad profiles and provides best estimates of the required rotation of the data, either from water profiling or bottom tracking. CODAS was run on fola, after rsyncing the raw data into /vmadcp/raw_os75, linked to by vmadcp/jr306_os75/rawdata. This processing was performed daily, following daily navigation data processing.

11.2.1 Setting up the directories and using quick_adcp

Once loaded into the rawdata directory, the following steps were followed:

1. vmadcp_linkscript was typed in the command window. This creates a new directory, rawdataNNN (NNN denoting the file sequence), and makes links to the relevant data in this new location.
2. The command adcptree.py jr306NNNnbenx –datatype enx was typed at the command window. This command sets up a directory tree for the CODAS dataset and an extensive collection of configuration files, text files and m files.
3. The control files q_py.cnt, q_pyrot.cnt, and q_pyedit.cnt were copied to each directory and q_py.cnt edited to refer to the relevant NNN.
4. From jr306NNNnbenx/ quick_adcp.py –cntfile q_py.cnt was run to load the data and perform routine processing and automatic editing, including estimates of bottom track and water track calibrations.
5. Calibration values in cal/watertrack/adcpcal.out and cal/btrack/btcaluv.out were examined and q_pyrot.cnt adjusted to use the chosen amplitude and phase calibrations (see below).

11.2.2 Angle and amplitude calibration

The quick_adcp.py script estimates amplitude and phase corrections for each set of data. It is necessary to specify the rotation angle and along-track amplitude scale factor in q_pyrot.cnt to obtain accurate velocities. The best calibration estimates are obtained when the velocity data is collected using the seabed as a reference. However, bottom track calibration estimates are only obtainable when the water depth is within the ADCP profiling range. Bottom tracking was performed over the Antarctic Peninsula shelf at the beginning of the cruise; we also examined bottom tracking data from the Antarctic shelf and the South American shelf and Burdwood Bank from the immediately preceding cruise (JR307/308). The bottom tracking values show no significant additional rotation. Water track amplitude values obtained while the ship was on-station for CTD casts ranged from 0.994 to 1.007. Bottom track amplitude values varied from 1.013 in the north to 1.019 in the south; we set an amplitude scale of 1.016 for the whole cruise, associated with an uncertainty of less than 2 cm/s in alongtrack velocity.

The final calibrations were applied to each file sequence using quick_adcp.py –cntfile q_pyrot.cnt in jr306NNNnbenx/ in the terminal window. This rotates the data by the phase and amplitude specified by the user in the control file q_pyrot.cnt. A recalculated calibration (after taking the first calibration into account) is printed to the *.out file(s). The data were then checked in Gautoedit to ensure that any vertical striping associated with on/off station differences had been removed by application of the calibration. Any alterations that needed to be made to the files, for example due to bad profiles or bad bins, were made using Gautoedit.

11.2.3 Gautoedit

The gautoedit package within CODAS allows the user to review closely the data collected by VM-DAS and flag any data that are deemed to be bad. These flags can then be passed forward and, using

the `q_pyedit.cnt` control file, the data removed. For each sequence we opened `gautoedit` in Matlab in `jr306NNNnbenx/edit` using `m_setup`; `codaspaths`; `gautoedit`. Typically we worked from the start of the data in 0.2-day steps, as this allowed us to see the individual 5-minute ensembles. Upon pressing Show Now, two plots are displayed according to the default plot selections. One contains four subplots: the first displays the absolute east-west (U) velocity component, the second shows the absolute north-south (V) component, the third shows the percent good parameter and the fourth shows the ship speed (in m/s) and an editing parameter called jitter. The second figure contains subplots of the ship's track and mean absolute velocity vectors at the reference layer. We noted that this plot sometimes was not generated until Show Now was pressed again. Routine editing for each section included:

- looking for bad profiles (i.e. those in which the u and/or v had a systematic offset over all depth levels). These were flagged using the `del bad times` command and choosing the select time range option.
- looking at the jitter parameter in the bottom subplot. A high level of jitter either indicates noise in the navigation and/or rapidly changing velocities (for instance during coming onto station maneuvers).

11.2.4 Applying the edits

Once the `a*.asc` files have been created, the edits are applied using the following command at the terminal prompt from within `jr306NNNnbenx/`:
`quick_adcp.py cntfile q_pyedit.cnt` The `q_pyedit.cnt` file has to have the correct `instname` command line (i.e. OS75).

11.2.5 Creating the output files

Once the editing and rotations were completed, the final velocities were collated into mstar files (*.nc) using the following commands, run from `jr306NNNnbenx/`:

`m_setup`, `mcod_01`, `mcod_02` (type the file number and instrument number when prompted to specify the input file). The first command sets up the Mstar suite of programs and the relevant paths. The other two commands load in the final data for the file sequence and save it as two mstar files. The first command produces a file of `os75_jr306NNNnnx.nc` that includes the following variables:

- time - (in seconds since [2010 1 1 0 0 0])
- lon - (0 to 360)
- lat - (-90 to 90)
- depth - (of bin)
- uabs - (absolute u velocity in cm/s)
- vabs - (absolute v velocity in cm/s)
- uship - (u velocity of ship over ground)
- vship - (v velocity of ship over ground)
- decday - (decimal day of year)

The second file is of the form `os75_jr306NNNnnx_spd.nc` and includes, (in addition to the above variables):

- speed - (scalar water speed, cm/s)
- shipspd - (scalar ship speed over ground, m/s).

The individual `os75_jr306NNNnnx_spd.nc` files are then appended together into a single output file for the cruise using `mcod_mapend`. This command relies on an input file containing the paths of all the individual files to be merged. This was created in `jr306_os75/` and was named `nc_files`. The final output file was `os75_jr306nnx_01.nc` which contain appended on-station and underway data.

11.2.6 Extracting station profiles

mcod_03 uses CTD station times from the dcs files to extract VMADCP data corresponding to CTD stations and reduce them to a single profile, in files such as os75_jr306nnx_ctd_006_ave.nc. mcod_stn_out then exports this profile to os75_jr306_ctd_006.mat, which is loaded to use as “SADCP” by LDEO IX processing. A useful script, mcod_03_all, shows how to run these two scripts over a large number of stations. mcod_addrows was used, after mcod_01, to pad a file with fewer than 50 rows so it could be appended with other files; this script would need to be edited for a different cruise. The VMADCP station profiles are plotted in Figures 3 to 5 along with IX LADCP profiles.

12 Acknowledgments

We gratefully acknowledge the assistance of the Master, officers, crew, technical support personnel, and volunteers in making the cruise and data collection highly successful.

Table 5: List of CTD stations. day is year day (001 for 1 Jan) in 2015. Water depth is corrected water depth from the EA600; alt min is the minimum altimeter reading; N_{dep} is the number of bottle depths; N_{sal} the number of bottle salts reported.

stn	start day	start time	bot. day	bot. time	end day	end time	latitude	longitude	water depth	CTD depth	alt min	wire out	max press	N_{dep}	N_{sal}
001	013	0722	013	0731	013	0746	61 03.04 S	54 35.37 W	359	350	9	347	353	6	6
002	013	0830	013	0843	013	0902	60 58.86 S	54 37.77 W	581	572	8	569	579	8	8
003	013	1004	013	1023	013	1050	60 51.00 S	54 42.53 W	962	949	10	947	960	16	10
004	013	1404	013	1435	013	1513	60 49.99 S	54 43.27 W	1754	1751	7	1746	1775	10	10
005	013	1549	013	1632	013	1725	60 48.02 S	54 44.52 W	2520	2506	8	2500	2545	12	12
006	013	1831	013	1924	013	2029	60 40.04 S	54 49.55 W	3107	3098	8	3107	3151	11	11
007	013	2242	013	2341	014	0050	60 19.93 S	55 01.04 W	3431	3422	9	3414	3483	24	10
008	014	0313	014	0414	014	0521	60 00.04 S	55 13.95 W	3499	3491	8	3482	3554	8	8
009	014	0739	014	0841	014	0952	59 39.97 S	55 26.61 W	3675	3665	8	3657	3733	8	8
010	014	1220	014	1324	014	1436	59 19.94 S	55 39.04 W	3749	3740	9	3792	3809	8	8
011	014	1700	014	1805	014	1914	59 00.07 S	55 50.98 W	3775	3748	27	3837	3817	8	8
012	014	2138	014	2240	014	2354	58 41.00 S	56 03.23 W	3757	3748	8	3765	3817	10	10
013	015	0223	015	0331	015	0454	58 21.90 S	56 14.88 W	3892	3884	8	3882	3957	10	10
014	015	0726	015	0833	015	0952	58 03.00 S	56 26.83 W	3963	3954	8	3944	4029	10	10
015	015	1225	015	1325	015	1430	57 44.00 S	56 37.67 W	3582	3557	13	3550	3620	10	10
016	015	1658	015	1759	015	1910	57 25.07 S	56 49.15 W	3675	3654	17	3650	3721	10	10
017	015	2139	015	2244	016	0002	57 05.95 S	57 01.31 W	3897	3894	7	3896	3967	12	12
018	016	0216	016	0311	016	0415	56 46.93 S	57 13.98 W	3086	3076	9	3068	3128	10	10
019	016	0625	016	0731	016	0854	56 28.12 S	57 25.63 W	3901	3889	11	3878	3961	10	10
020	016	1724	016	1823	016	1928	56 10.13 S	57 35.63 W	3358	3345	12	3337	3403	10	10
021	016	2156	016	2318	017	0101	55 49.89 S	57 48.11 W	4755	4745	10	4759	4843	12	12
022	017	0322	017	0437	017	0603	55 31.04 S	58 00.10 W	4224	4213	10	4258	4295	12	12
023	017	0812	017	0919	017	1039	55 12.77 S	58 00.01 W	3782	3772	7	3774	3841	10	10
024	017	1139	017	1232	017	1331	55 10.19 S	57 59.99 W	3050	3039	10	3042	3089	8	8
025	017	1943	017	2030	017	2124	55 07.23 S	58 00.02 W	2789	2777	8	2774	2821	8	8
026	017	2209	017	2248	017	2338	55 04.20 S	58 00.00 W	2306	2297	9	2294	2331	8	8
027	018	0032	018	0102	018	0138	55 00.41 S	57 59.84 W	1661	1652	8	1648	1674	8	8
028	018	0215	018	0232	018	0255	54 58.66 S	58 00.22 W	982	971	10	968	983	6	6
029	018	0358	018	0409	018	0426	54 54.24 S	57 59.60 W	589	579	9	577	586	6	6
030	018	0602	018	0608	018	0615	54 44.25 S	57 59.57 W	206	197	9	194	199	4	4

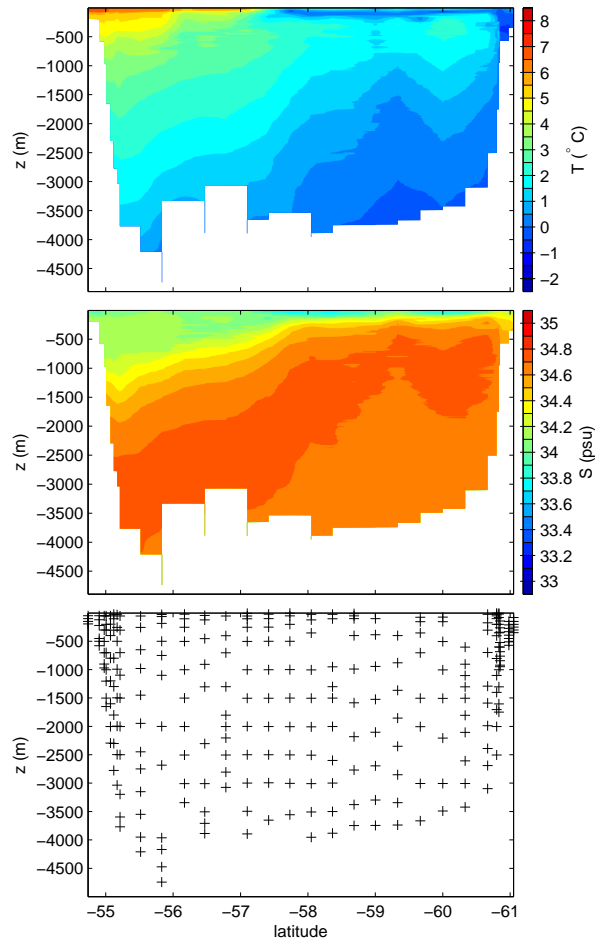


Figure 2: Temperature (top) and salinity (middle) sections from JR306, with salinity sample bottle locations (bottom).

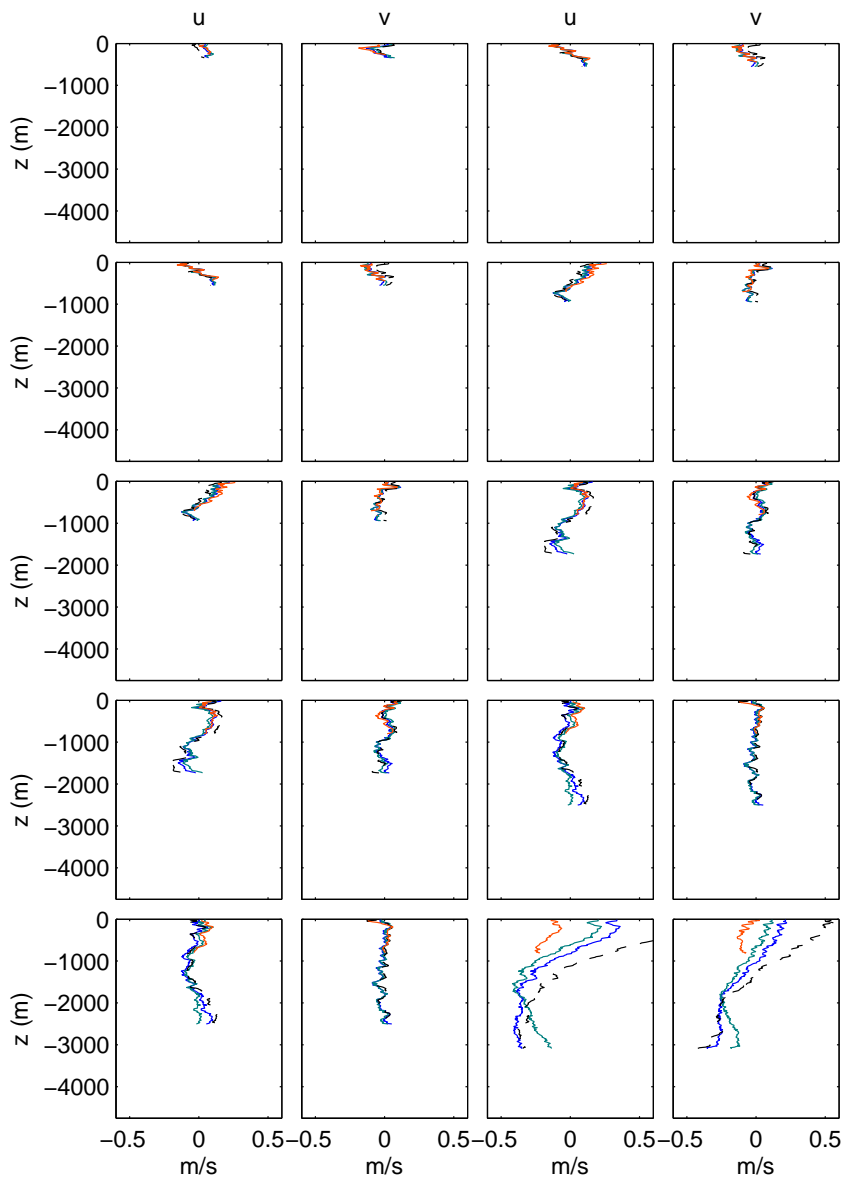


Figure 3: Comparison between LADCP solutions (shear method: black dashed line, shows velocity relative to the depth mean; inverse with navigation data: blue line; inverse with navigation and bottom tracking: teal line) and VMADCP velocity (red), for stations 1 through 10.

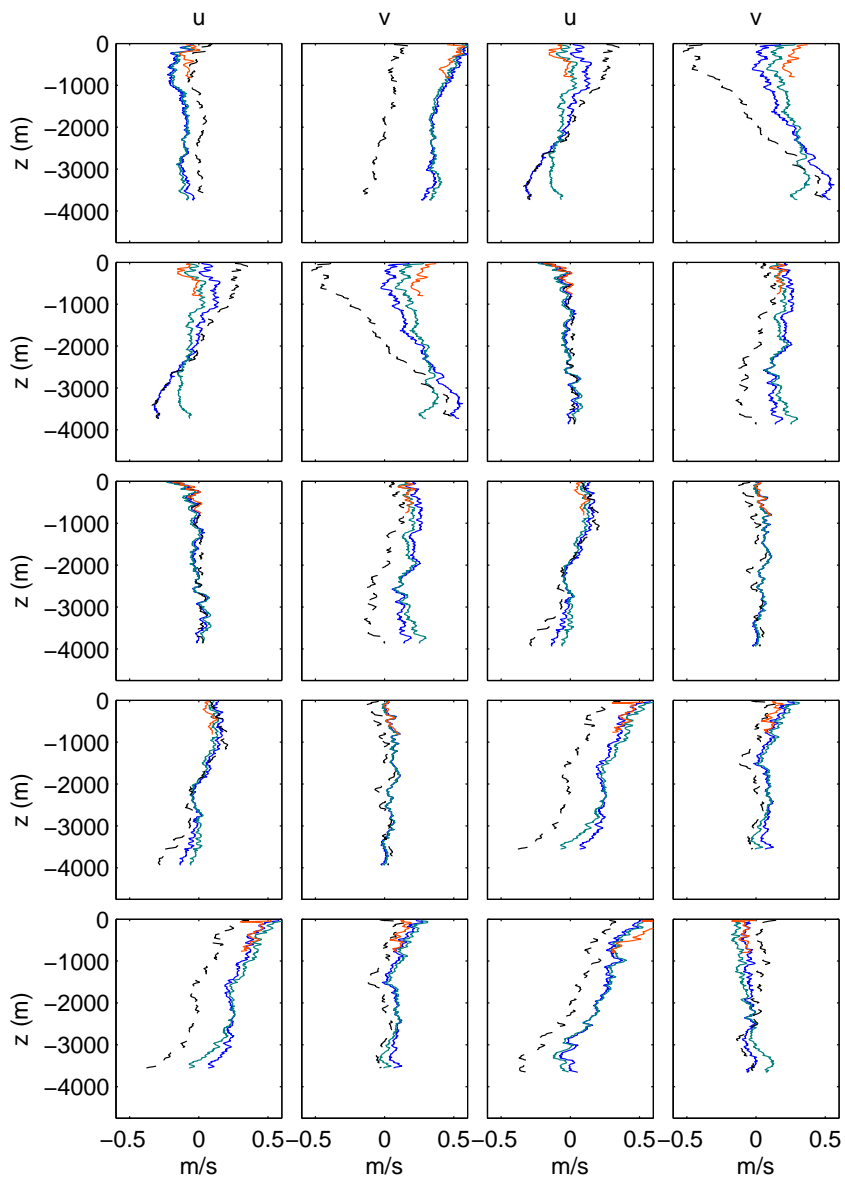


Figure 4: Comparison between LADCP solutions (shear method: black dashed line, shows velocity relative to the depth mean; inverse with navigation data: blue line; inverse with navigation and bottom tracking: teal line) and VMADCP velocity (red), for stations 11 through 20.

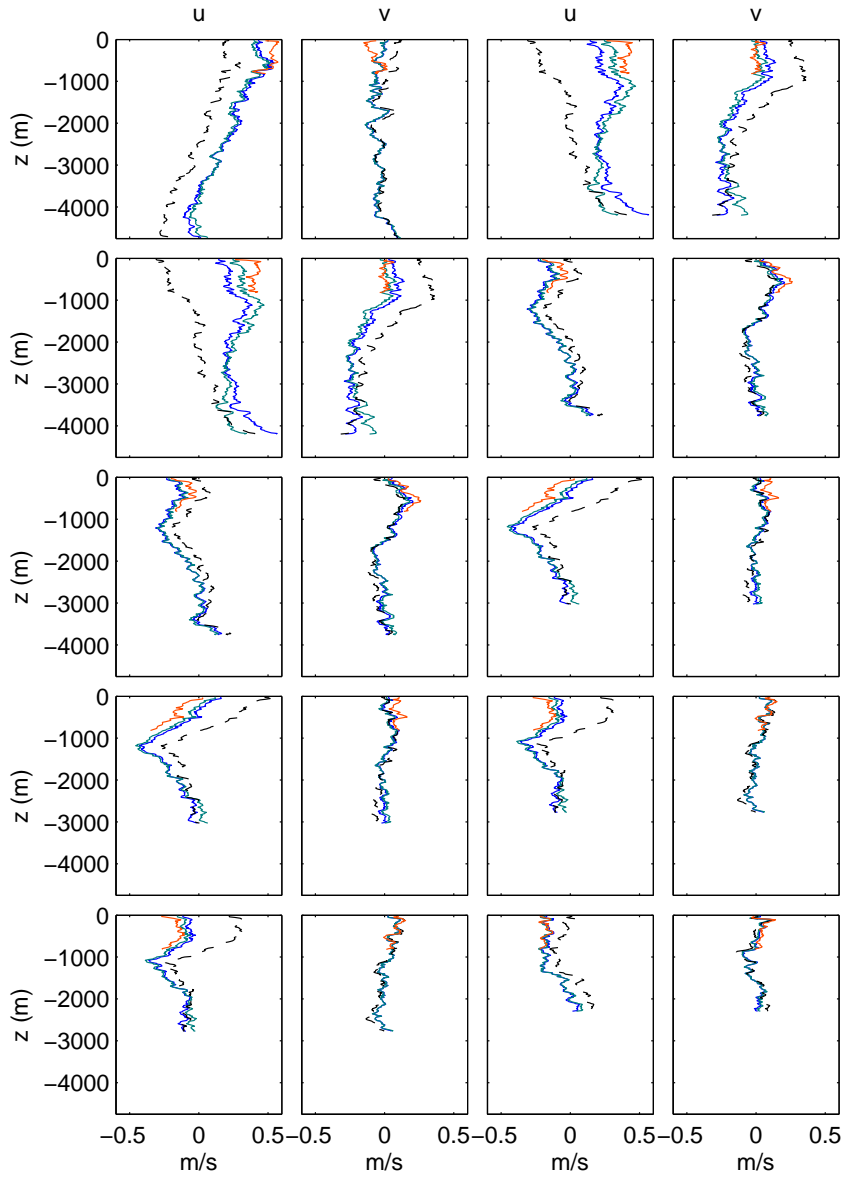


Figure 5: Comparison between LADCP solutions (shear method: black dashed line, shows velocity relative to the depth mean; inverse with navigation data: blue line; inverse with navigation and bottom tracking: teal line) and VMADCP velocity (red), for stations 21 through 30.

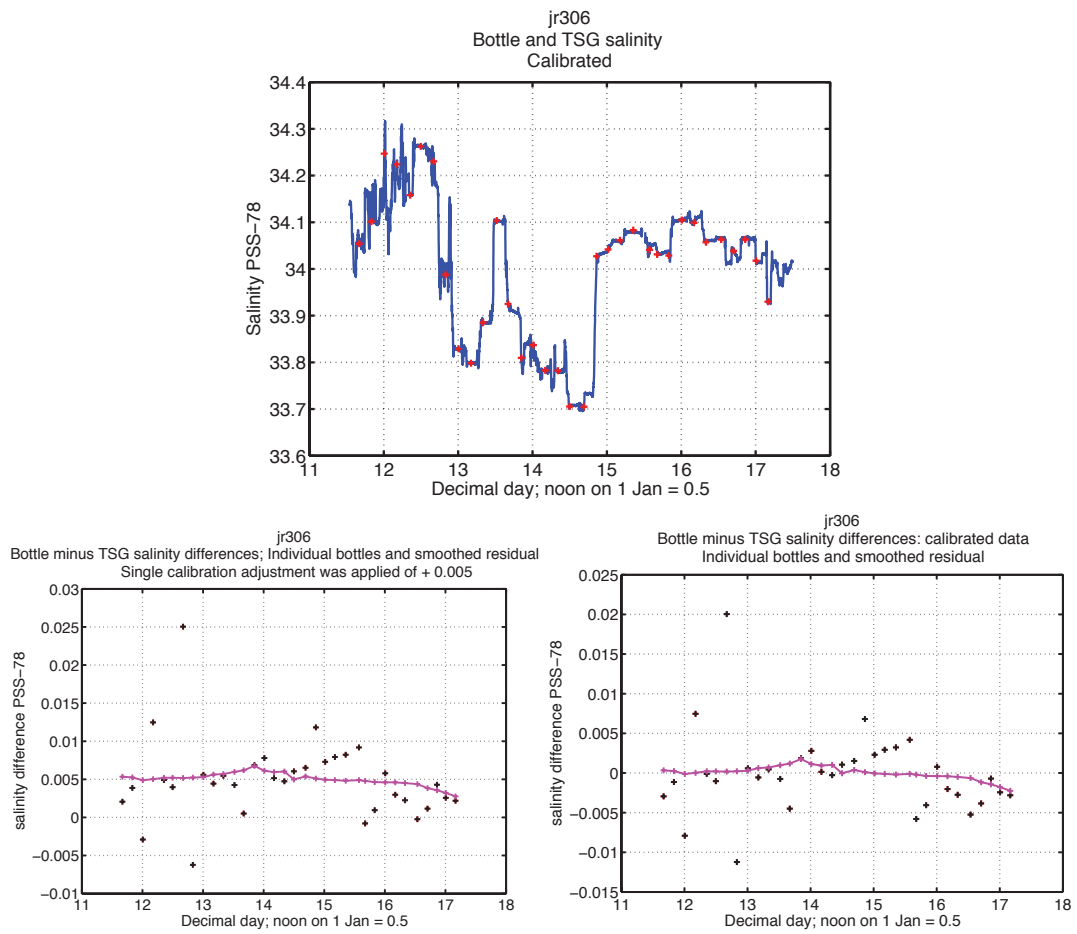


Figure 6: Top: underway salinity time series (blue) with bottle salinities (red); bottom left: uncalibrated salinity differences; bottom right: calibrated salinity differences after application of constant offset (see text).