

AD-994 610

NAVAL RESEARCH LAB. WASHINGTON DC
REAL-TIME AND APPLICATIONS SOFTWARE FOR THE NEIL BROWN/W.H.O.I.—ETC(U)
DEC 80 L J ROSENBLUM

F/6 9/2

UNCLASSIFIED

[OP]
AD-994 610

NL

END
DATE FILMED
12-8H
DTIC

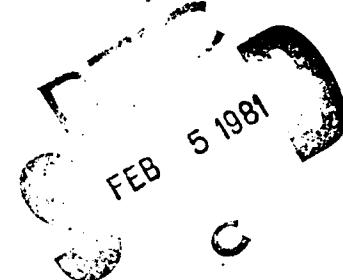
LEVEL 12

NRL Report 8444

Real-Time and Applications Software for the Neil Brown/W.H.O.I. CTD Microprofiler

LAWRENCE J. ROSENBLUM

Oceanographic Computer Applications Group



December 30, 1980



NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited

DDG FILE COPY

81 2 01 50

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
NRL Report 8444	AD-4C94620	
4. TITLE (and Subtitle)	5. TYPE OF REPORT & PERIOD COVERED	
REAL-TIME AND APPLICATIONS SOFTWARE FOR THE NEIL BROWN/W.H.O.I. CTD MICROPROFILER	Final report	
7. AUTHOR(s)	6. PERFORMING ORG. REPORT NUMBER	
Lawrence J. Rosenblum	8. CONTRACT OR GRANT NUMBER(s)	
9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS	
Naval Research Laboratory Washington, DC 20375	J. O. 68056-6013	
11. CONTROLLING OFFICE NAME AND ADDRESS	12. REPORT DATE December 30, 1980	
	13. NUMBER OF PAGES 46	
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)	15. SECURITY CLASS. (of this report) UNCLASSIFIED	
16. DISTRIBUTION STATEMENT (of this Report)	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE	
Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Oceanography	Real-time processing	
Ocean microstructure measurements	Minicomputer applications	
CTD microprofiler data acquisition		
CTD microprofiler real-time data display		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
The Neil Brown/W.H.O.I. CTD Microprofiler has been extensively used by a variety of NRL researchers in at-sea experiments. This paper reports on software for both real-time data acquisition and display and for processing collected data.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S-N-2102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE When Data Entered

CONTENTS

THE CTD MICROPROFILER.....	1
Introduction.....	1
Interfacing.....	1
REAL-TIME COLLECTION AND DISPLAY SOFTWARE.....	3
The Real-Time Program Set	3
System Requirements	3
Data File Structure	4
Data Acquisition and Display	5
Real-Time Programs—FORTRAN Source Listings	9
USER'S GUIDE TO REAL-TIME CTD PROGRAMS	24
Preparation.....	24
Program Start-Up.....	24
Special Functions	26
DATA PROCESSING PROGRAMS	27
Parameters of Interest.....	27
User's Guide to Program PLTBK.....	28
Examples of Program PLTBK Graphics	28
Data Processing Programs—FORTRAN Source Listings.....	32
REFERENCES.....	39
APPENDIX—Scientific Subroutines—FORTRAN Source Listings.....	40

Request for Information	Serial Number
Comments	23
Comments	24
Comments	25
Comments	26
Comments	27
Comments	28
Comments	29
Comments	30
Comments	31
Comments	32
Comments	33
Comments	34
Comments	35
Comments	36
Comments	37
Comments	38
Comments	39
Comments	40
Comments	41
Comments	42
Comments	43
Comments	44
Comments	45
Comments	46
Comments	47
Comments	48
Comments	49
Comments	50
Comments	51
Comments	52
Comments	53
Comments	54
Comments	55
Comments	56
Comments	57
Comments	58
Comments	59
Comments	60
Comments	61
Comments	62
Comments	63
Comments	64
Comments	65
Comments	66
Comments	67
Comments	68
Comments	69
Comments	70
Comments	71
Comments	72
Comments	73
Comments	74
Comments	75
Comments	76
Comments	77
Comments	78
Comments	79
Comments	80
Comments	81
Comments	82
Comments	83
Comments	84
Comments	85
Comments	86
Comments	87
Comments	88
Comments	89
Comments	90
Comments	91
Comments	92
Comments	93
Comments	94
Comments	95
Comments	96
Comments	97
Comments	98
Comments	99
Comments	100
Comments	101
Comments	102
Comments	103
Comments	104
Comments	105
Comments	106
Comments	107
Comments	108
Comments	109
Comments	110
Comments	111
Comments	112
Comments	113
Comments	114
Comments	115
Comments	116
Comments	117
Comments	118
Comments	119
Comments	120
Comments	121
Comments	122
Comments	123
Comments	124
Comments	125
Comments	126
Comments	127
Comments	128
Comments	129
Comments	130
Comments	131
Comments	132
Comments	133
Comments	134
Comments	135
Comments	136
Comments	137
Comments	138
Comments	139
Comments	140
Comments	141
Comments	142
Comments	143
Comments	144
Comments	145
Comments	146
Comments	147
Comments	148
Comments	149
Comments	150
Comments	151
Comments	152
Comments	153
Comments	154
Comments	155
Comments	156
Comments	157
Comments	158
Comments	159
Comments	160
Comments	161
Comments	162
Comments	163
Comments	164
Comments	165
Comments	166
Comments	167
Comments	168
Comments	169
Comments	170
Comments	171
Comments	172
Comments	173
Comments	174
Comments	175
Comments	176
Comments	177
Comments	178
Comments	179
Comments	180
Comments	181
Comments	182
Comments	183
Comments	184
Comments	185
Comments	186
Comments	187
Comments	188
Comments	189
Comments	190
Comments	191
Comments	192
Comments	193
Comments	194
Comments	195
Comments	196
Comments	197
Comments	198
Comments	199
Comments	200
Comments	201
Comments	202
Comments	203
Comments	204
Comments	205
Comments	206
Comments	207
Comments	208
Comments	209
Comments	210
Comments	211
Comments	212
Comments	213
Comments	214
Comments	215
Comments	216
Comments	217
Comments	218
Comments	219
Comments	220
Comments	221
Comments	222
Comments	223
Comments	224
Comments	225
Comments	226
Comments	227
Comments	228
Comments	229
Comments	230
Comments	231
Comments	232
Comments	233
Comments	234
Comments	235
Comments	236
Comments	237
Comments	238
Comments	239
Comments	240
Comments	241
Comments	242
Comments	243
Comments	244
Comments	245
Comments	246
Comments	247
Comments	248
Comments	249
Comments	250
Comments	251
Comments	252
Comments	253
Comments	254
Comments	255
Comments	256
Comments	257
Comments	258
Comments	259
Comments	260
Comments	261
Comments	262
Comments	263
Comments	264
Comments	265
Comments	266
Comments	267
Comments	268
Comments	269
Comments	270
Comments	271
Comments	272
Comments	273
Comments	274
Comments	275
Comments	276
Comments	277
Comments	278
Comments	279
Comments	280
Comments	281
Comments	282
Comments	283
Comments	284
Comments	285
Comments	286
Comments	287
Comments	288
Comments	289
Comments	290
Comments	291
Comments	292
Comments	293
Comments	294
Comments	295
Comments	296
Comments	297
Comments	298
Comments	299
Comments	300
Comments	301
Comments	302
Comments	303
Comments	304
Comments	305
Comments	306
Comments	307
Comments	308
Comments	309
Comments	310
Comments	311
Comments	312
Comments	313
Comments	314
Comments	315
Comments	316
Comments	317
Comments	318
Comments	319
Comments	320
Comments	321
Comments	322
Comments	323
Comments	324
Comments	325
Comments	326
Comments	327
Comments	328
Comments	329
Comments	330
Comments	331
Comments	332
Comments	333
Comments	334
Comments	335
Comments	336
Comments	337
Comments	338
Comments	339
Comments	340
Comments	341
Comments	342
Comments	343
Comments	344
Comments	345
Comments	346
Comments	347
Comments	348
Comments	349
Comments	350
Comments	351
Comments	352
Comments	353
Comments	354
Comments	355
Comments	356
Comments	357
Comments	358
Comments	359
Comments	360
Comments	361
Comments	362
Comments	363
Comments	364
Comments	365
Comments	366
Comments	367
Comments	368
Comments	369
Comments	370
Comments	371
Comments	372
Comments	373
Comments	374
Comments	375
Comments	376
Comments	377
Comments	378
Comments	379
Comments	380
Comments	381
Comments	382
Comments	383
Comments	384
Comments	385
Comments	386
Comments	387
Comments	388
Comments	389
Comments	390
Comments	391
Comments	392
Comments	393
Comments	394
Comments	395
Comments	396
Comments	397
Comments	398
Comments	399
Comments	400
Comments	401
Comments	402
Comments	403
Comments	404
Comments	405
Comments	406
Comments	407
Comments	408
Comments	409
Comments	410
Comments	411
Comments	412
Comments	413
Comments	414
Comments	415
Comments	416
Comments	417
Comments	418
Comments	419
Comments	420
Comments	421
Comments	422
Comments	423
Comments	424
Comments	425
Comments	426
Comments	427
Comments	428
Comments	429
Comments	430
Comments	431
Comments	432
Comments	433
Comments	434
Comments	435
Comments	436
Comments	437
Comments	438
Comments	439
Comments	440
Comments	441
Comments	442
Comments	443
Comments	444
Comments	445
Comments	446
Comments	447
Comments	448
Comments	449
Comments	450
Comments	451
Comments	452
Comments	453
Comments	454
Comments	455
Comments	456
Comments	457
Comments	458
Comments	459
Comments	460
Comments	461
Comments	462
Comments	463
Comments	464
Comments	465
Comments	466
Comments	467
Comments	468
Comments	469
Comments	470
Comments	471
Comments	472
Comments	473
Comments	474
Comments	475
Comments	476
Comments	477
Comments	478
Comments	479
Comments	480
Comments	481
Comments	482
Comments	483
Comments	484
Comments	485
Comments	486
Comments	487
Comments	488
Comments	489
Comments	490
Comments	491
Comments	492
Comments	493
Comments	494
Comments	495
Comments	496
Comments	497
Comments	498
Comments	499
Comments	500
Comments	501
Comments	502
Comments	503
Comments	504
Comments	505
Comments	506
Comments	507
Comments	508
Comments	509
Comments	510
Comments	511
Comments	512
Comments	513
Comments	514
Comments	515
Comments	516
Comments	517
Comments	518
Comments	519
Comments	520
Comments	521
Comments	522
Comments	523
Comments	524
Comments	525
Comments	526
Comments	527
Comments	528
Comments	529
Comments	530
Comments	531
Comments	532
Comments	533
Comments	534
Comments	535
Comments	536
Comments	537
Comments	538
Comments	539
Comments	540
Comments	541
Comments	542
Comments	543
Comments	544
Comments	545
Comments	546
Comments	547
Comments	548
Comments	549
Comments	550
Comments	551
Comments	552
Comments	553
Comments	554
Comments	555
Comments	556
Comments	557
Comments	558
Comments	559
Comments	560
Comments	561
Comments	562
Comments	563
Comments	564
Comments	565
Comments	566
Comments	567
Comments	568
Comments	569
Comments	570
Comments	571
Comments	572
Comments	573
Comments	574
Comments	575
Comments	576
Comments	577
Comments	578
Comments	579
Comments	580
Comments	581
Comments	582
Comments	583
Comments	584
Comments	585
Comments	586
Comments	587
Comments	588
Comments	589
Comments	590
Comments	591
Comments	592
Comments	593
Comments	594
Comments	595
Comments	596
Comments	597
Comments	598
Comments	599
Comments	600
Comments	601
Comments	602
Comments	603
Comments	604
Comments	605
Comments	606
Comments	607
Comments	608
Comments	609
Comments	610
Comments	611
Comments	612
Comments	613
Comments	614
Comments	615
Comments	616
Comments	617
Comments	618
Comments	619
Comments	620
Comments	621
Comments	622</

REAL-TIME AND APPLICATIONS SOFTWARE FOR THE NEIL BROWN/W.H.O.I. CTD MICROPROFILER

THE CTD MICROPROFILER

Introduction

The W.H.O.I./Brown conductivity, temperature, and depth (CTD) profiler is a device to make fine-scale measurements of these physical parameters in the ocean. The CTD was first used by the Naval Research Laboratory (NRL) in July 1977 and has been extensively used in field work in physical oceanography, chemical oceanography, acoustics, and special projects ever since. In August 1979 one of the NRL CTD devices was modified to include a fluorometer. All field work utilizing the CTDs has used on-board computer systems of the Oceanographic Computer Applications Group. This report documents software written by the author for the real-time acquisition, storage, and display of CTD data as well as application programs for processing collected data.

The CTD consists of a shipboard data terminal deck unit and an underwater unit which provides continuous sampling of the variables as it is lowered through the water. Conductivity is measured with a miniature four-electrode ceramic cell. Temperature is sensed by electronically combining the outputs of a high-speed thermistor and a platinum resistance thermometer. Pressure is sensed by a strain-gauge pressure transducer. Data from the underwater unit is transmitted in real time to the shipboard data terminal through a single electrical conductor in the core of the steel cable used to support the instrument in the water. The data are in Teletype format using frequency-shift key modulation of 5- and 10-kHz ac signals superimposed on the dc power supply to the underwater unit through the same cable. The deck unit decodes the signals, provides digital data in both parallel and serial format for computer processing, and displays conductivity, temperature, and pressure. For greater detail on the CTD see references 1 to 5.

The NRL Oceanographic Computer Applications Group uses Hewlett-Packard (HP) 1000-series computers and a variety of peripherals depending on individual requirements. For the September/October 1979 cruise aboard the USNS *Lynch* two HP21MX-E computers were used, one for taking real-time data from various physical oceanographic data collection devices and ship's sensors and the other for recording CTD and XBT data. The CTD system produced real-time plots on a HP2647 terminal and used an HP2631G line printer for producing hard copy of these plots. Minor program modifications are necessary for other peripherals. Figure 1, taken aboard the USNS *Hayes* in 1977, shows a typical shipboard computer system.

Interfacing

The CTD offers both serial and parallel data ports for use in interfacing. In the CTD data flow each byte (8 bits) is preceded by one bit containing a logic "zero" and terminated by two bits, each of which contains a logic "one." When a digitized channel requires two bytes to convey magnitude, the first byte is the least significant byte. The original interface, developed by the Oceanographic Computer Applications Group in 1977, performed a sample-and-hold operation on an entire scan of data holding the three 16-bit words containing pressure, temperature, and conductivity. These three words are then sent to the computer on demand using three computer microcircuit interface boards, one for each parameter. A driver developed by the Oceanographic Computer Applications Group for doing sampling of successive I/O channels on HP1000-series computers was employed to control the I/O.



Figure 1 — Computer system aboard USNS *Hayes* in 1977

The original approach had some disadvantages. While synchronization problems have been found to be rare, the "sync" and utility frames were unavailable. Also, the requirement for three successive channels was inconvenient and occasionally necessitated the use of an I/O extender to provide additional computer-interface channels. The extender requires additional rack space and creates the potential for hardware failure aboard ship. Accordingly, the CTD system required revision to work with a CTD microprofiler modified to include fluorometer measurements. The data were still taken at the deck unit parallel output port; however, the interrupt strobes were "or"ed together and data was sent to the computer in word serial mode with no indication of which word (or byte) of data was being sent. Thus, the software had to determine where the sync frame was in order to start data acquisition and to assure that the data flow remained in sync. The data were sent to the computer through an I/O board using the same driver as before. This method brings in all the collected data bytes, but greater programming care was required since the internal timing of the CTD within a scan had to be considered in the programming to assure that sufficient time was available between words to perform all CTD operations without data loss.

The fluorometer CTD system has the following data format:

- Byte 1: Frame sync. Alternate between 11110000 and 00001111
- Byte 2: Pressure (LSBs)

- Byte 3: Pressure (MSBs)
- Byte 4: Temperature (LSBs)
- Byte 5: Temperature (MSBs)
- Byte 6: Conductivity (LSBs)
- Byte 7: Conductivity (MSBs)
- Byte 8: Utility. Contains pressure and temperature signs.
- Byte 9: Fluorometer "Signal." All "ones" correspond to full scale input.
- Byte 10: Fluorometer "Range." All "ones" correspond to full scale input.

As currently interfaced, pressure, temperature, and conductivity are packed into words by the interface unit, with the remaining parameters placed in the lower half of full words.

This report deals with the CTD software for the fluorometer CTD. However, both program sets are quite similar and differences are likely to be transparent to a user.

REAL-TIME COLLECTION AND DISPLAY SOFTWARE

The Real-Time Program Set

The real-time data collection and display programs for the fluorometer CTD unit are described in this chapter.

There are nine real-time programs used. These are:

- CTDRT — Obtains data and schedules other programs.
- CTDIN — Initializes parameters prior to data collection and writes an initialization record on magnetic tape.
- CTDMT — Writes data records on magnetic tape.
- CTDP1 — Produces real-time plots of salinity and temperature vs pressure.
- CTDP2 — Produces real-time plots of salinity, temperature, and pressure vs time.
- CTDCN — Changes certain control functions while data are being taken.
- CTDTR — Writes termination file.
- CTDMS — Writes message to terminal for CTDRT.
- CTDFL — Sets fluorometer On/Off switch.

System Requirements

The programs are designed to operate under RTE-II, RTE-III, or RTE-IV (Real-Time Executive operating systems) on HP1000-series computers with 64K words of memory. A background system common area of 2000₈ is required. The collection program CTDRT is loaded as a foreground program using reverse (background) common. The remaining programs are loaded as background programs using background common.

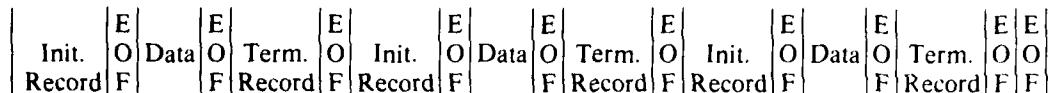
The programs listed in this section utilize the HP2647 terminal and HP2631G printer. The HP2631G is also used to produce hard copy of terminal plots. The programs require minor revisions if other peripherals are used since the means for producing hard copy of plots is device dependent and also the screen size of the HP2647 requires a minor scaling modification of the plot.

Data File Structure

One CTD lowering (or towing) results in a data tape file with the following format:

| INITIALIZATION RECORD | EOF | DATA RECORDS | EOF | TERM RECORD | EOF | EOF |

Prior to the next event, the program F2EOF is run to position the tape between the double end-of-file marks and the next initialization record is written over the last EOF. Hence a tape with several CTD events will have a double EOF only after the last event. A sequence of three events would appear as



The initialization record is a 47-word header record containing identification parameters for the event:

Words 1-6:	12 Character Identifying ASCII String
Word 7:	Station Number
Word 8:	Year
Word 9:	Month
Word 10:	Day
Word 11:	Lowering Number
Word 12:	Instrument Number
Words 13-24:	Conversion Constants: C1, ..., C6
Words 25-30:	Pressure Axis Constants
Words 31-36:	Salinity Axis Constants
Words 37-42:	Temperature Axis Constants
Words 43-44:	Latitude
Words 45-46:	Longitude
Word 47:	Zero

The initialization record is followed by an EOF and then the CTD data. Each data record contains 456 words (approximately 2 seconds of data) consisting of 8 words of header information followed by 448 data words:

Word 1:	Record Number
Word 2:	Time (Hours = 24 × Days + Hours)
Word 3:	Time (Seconds = 60 × Minute + Seconds)
Word 4:	Time (Tens of milliseconds)
Word 5:	Special Segment ID Marker
Word 6:	Fluorometer on/off switch
Word 7:	0
Word 8:	0
Words 9-456:	Digital Data

The last data record is followed by an EOF and an 8-word termination record:

Word 1:	-1
Word 2:	Time Hours
Word 3:	Time Seconds
Word 4:	Termination Method Indicator
Words 5-8:	0

Finally, a double EOF is written on the tape.

Data Acquisition and Display

The data-acquisition programs are started by running the program CTDRT (a full description of questions and answers is given in the next chapter). CTDRT then schedules programs CTDMS for execution. CTDMS asks the operator whether the operation will be a CTD lowering or tow and if temperature or potential temperature is to be a plotting parameter. For a lowering, temperature (or potential temperature) and salinity are both plotted against pressure, while in the case of a tow, temperature (or potential temperature), salinity, and pressure are plotted as a function of time. Note that pressure in decibars is very nearly identical to depth in meters.

Now that the plot type has been defined, program CTDRT schedules programs CTDIN for execution to obtain identifying parameters and plot scale parameters. These parameters are written to magnetic tape in a 47-word header record for the CTD cast. The calibration constants are used to convert the CTD measurement into pressure, temperature, and salinity. If M_1 , M_2 , M_3 are the real values of the measurements obtained from the CTD unit for conductivity, pressure and temperature, then

$$\begin{aligned} \text{Pressure} &= C1 * (M_2 - C2) \\ \text{Temperature} &= C3 + (C4 * M_3) \\ \text{Conductivity} &= C5 * M_1 * (1 + C6 * \text{Temperature}) \end{aligned}$$

$C1$ is the only constant with which the user need be concerned; it depends upon the pressure sensor used within the CTD. The axes parameters determine the parameter limits for the real-time CTD display. Points outside the user-defined axes limits will not appear on the real-time plot, so the user should exercise care in assuring that these limits span the data without being so wide that the display loses meaning. Note, however, that these limits affect only the real-time display and not the collection of the data or the process of writing the data records onto magnetic tape. Thus, if the data is outside the user-defined limits, the ability to view the data in real-time is lost but the collection process will not be affected.

Once the initialization parameters have been obtained, the program CTDIN displays the values and asks if the values are correct. If a mistake has been made while inputting the parameters, the user answers NO and the questions are repeated. If they are correct, CTDIN writes the initialization record and an end-of-file (EOF) onto magnetic tape and terminates, which allows the scheduling program CTDRT to resume.

Depending upon whether a lowering or towing real-time display has been requested, CTDRT next schedules either program CTDP1 or CTDP2. In either case the appropriate axes are drawn on a plotting terminal, with the axes parameters obtained by CTDIN. When these axes have been drawn, CTDP1 (or CTDP2) suspends execution with the point of suspension being retained. This enables plotting to occur without the screen being erased each time a pair of points are to be drawn. Program CTDRT now has program CTDMS print the message "PAUSE—TYPE: 'GO, CTDRT' FOR DATA ACQUISITION" on the system console and then suspends. This completes the initialization phase. The operator then types "GO, CTDRT" on the system console to initiate data collection.

Figures 2 and 3 were made aboard the USNS *Lynch* in October 1977 near Bermuda and show typical plots for the lowering mode. For Fig. 2 the CTD was lowered near the surface while in Fig. 3 the device "yo-yoed" near the bottom. Figures 4 and 5 were made during the return run from Bermuda to Newport, Rhode Island, and show the real-time display for the towing mode. Here pressure, temperature, and salinity are displayed as a function of time.

L. J. ROSENBLUM

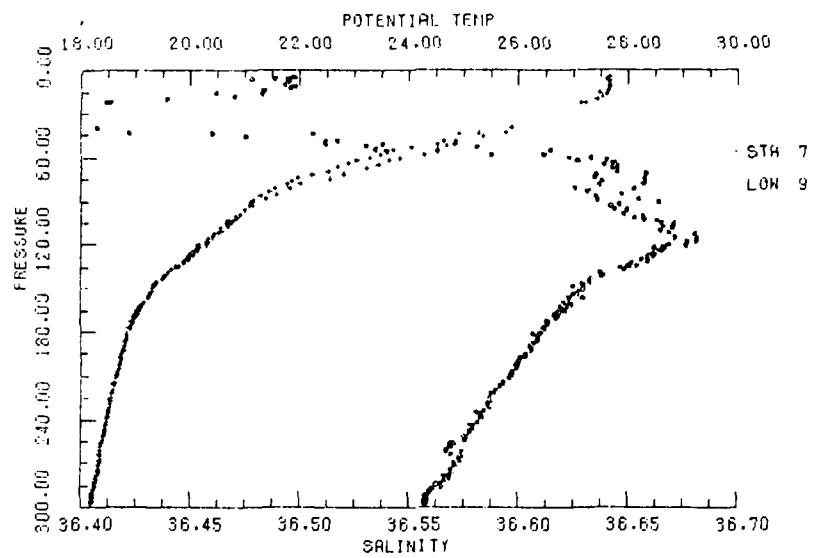


Figure 2 — Real-time CTD plot for near-surface lowering

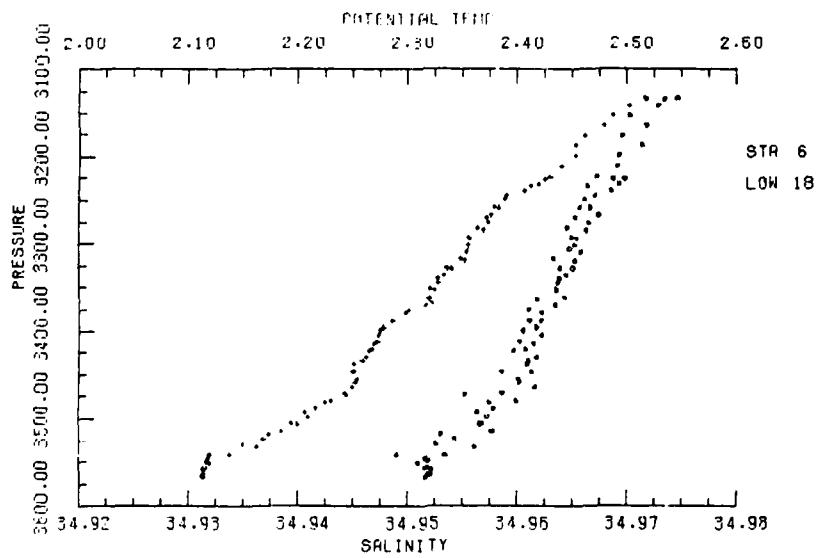


Figure 3 — Real-time CTD - "yo-yo" in vicinity of bottom

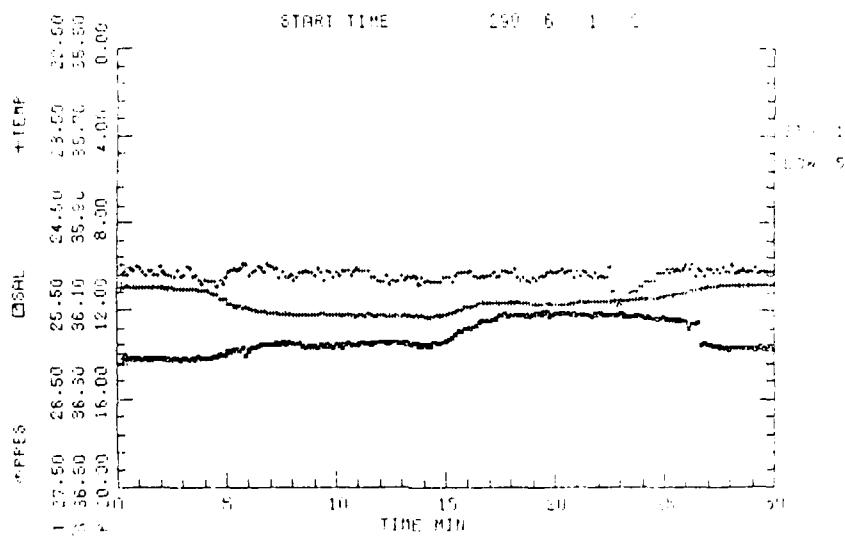


Figure 4 — Real time CTD plot—"tow" mode

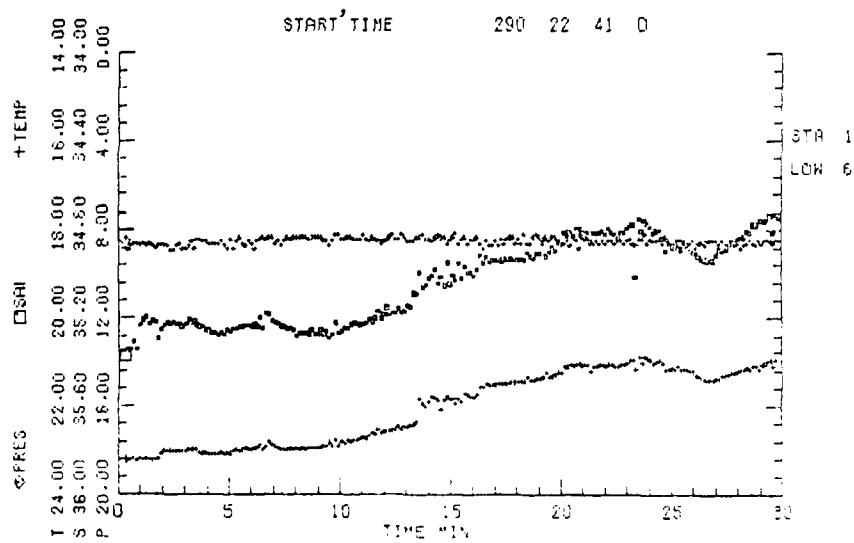


Figure 5 — Real time CTD plot—"tow" mode

L. J. ROSENBLUM

The basic programming technique used for data acquisition and processing is double buffering. The data are acquired in (approximately) 2-second buffers and the processing of the previous buffer must be accomplished within the 2 seconds allotted for obtaining the new data. Since the CTD scan rate is 31.25 hertz, one scan takes 32 milliseconds. As previously discussed, the CTD interface sends the data words in a serial format. The CTD interface, in one scan, sends the computer:

Word 1: Sync Frame
Word 2: Pressure
Word 3: Temperature
Word 4: Conductivity
Word 5: Utility
Word 6: Fluorometer "Signal"
Word 7: Fluorometer "Range"

where those words which contain only one byte of information have the data in the LSBs.

Returning to the data processing flow, program CTDRT compares words against the CTD sync frame values until the sync frame is recognized. It next determines which of the two buffers is to be filled. Once this is done, program CTDRT schedules program CTDMT and requests 448 words of data from the CTD interface. Thus, while the foreground program CTDRT obtains the next 2 seconds of data, the background programs (CTDMT and either CTDP1 or CTDP2) processes the previous 2 seconds of data. CTDMT keeps track of the buffer times and fills in the appropriate entries in the header—the first 8 words of the buffer as given above. The record is then dumped onto magnetic tape, and every fourth buffer the plotting program CTDP1 (or CTDP2) is scheduled to plot a point and print out on the line printer and system console the values of temperature, pressure, and salinity. These values are printed even if a point is not plotted due to one of the values being off the plot scale. When program CTDP1 (or CTDP2) is scheduled for execution, it takes the first three data points in the buffer and uses the constants C_1, C_2, \dots, C_6 to obtain pressure, temperature, and conductivity. Salinity is computed using an algorithm of Fofonoff. These points are then scaled for plotting and, if within the user defined plot axes limits, they are plotted on the real-time plot display. The values are printed out as mentioned above. If the CTD is being operated in the tow mode each plot will contain 1/2 hour of data. At the end of the half-hour program CTDP2 will automatically copy the plot onto a hard-copy unit (the 2631G terminal, a Tektronix hardcopier, etc., depending upon the system in use). The axes are then redrawn and reannotated with the same value as previously used and the plot continues.

The remaining programs are used for various control functions. Program CTDFL sets the fluorometer on/off switch (Word 6 of the buffer). It should be noted that this switch is for convenience in locating fluorometer data, but is not used to alter the data; hence, operator failure to properly set the switch will not affect the data tape. The program CTDCN is used for several purposes. It can be used to put a marker in word 5 of the data record in order to identify special segments of data for easy future location. Alternately, program CTD is used to put the CTD programs in "idle" mode and to terminate data collection. Occasionally, in the course of a CTD lowering, events such as cable snags will necessitate holding the CTD at a fixed depth for a certain period, and it has been found useful to have the capability of suspending data collection while such problems are being corrected. Accordingly, one mode of operation of program CTDCN will put all programs into a suspend state in such a way that the operator can resume data collection by typing a command on the system console. CTDCN is also used to terminate data collection. Full details of how to use program CTDCN are given the next section.

When program CTDCN is executed in either its termination or its abort mode, it assures that the real-time programs are properly terminated, writes an EOF onto the magnetic tape, and then schedules program CTDTR which writes an 8-word termination file and two EOFs. This completes the data collection process for one event. For further collection using the same magnetic tape, the tape is positioned between the EOF's and the process repeated.

Real-Time Programs—FORTRAN Source Listings

```

0001  FTN4.L
0002  PROGRAM CTDRT(10,15)
0003  C
0004  C      REAL TIME PROGRAMS FOR COLLECTION OF CTD DATA WITH FLOROMETER
0005  C      CAPABILITY. DOUBLE BUFFERED DATA TAKEN EVERY TWO SECONDS.
0006  C
0007  C      REAL TIME PROGRAMS USED ARE:
0008  C      CTDRT - SCHEDULES OTHER PROGRAMS AND CONTROL PARAMETERS
0009  C      CTDIN- INITIALIZES PARAMETERS AND WRITES INIT FILE ON MT
0010  C      CTDMT - WRITES DATA BUFFERS TO MT
0011  C      CTDPI- REAL TIME PLOT OF SAL, TEMP VS. PRESSURE
0012  C      CTDP2- REAL TIME PLOT OF SAL, TEMP, PRESS VS. TIME
0013  C      CTDTR- WRITES TERMINATION FILE
0014  C      CTDCN- USES COMMON PARAM ISTOP TO CONTROL FUNCTIONS
0015  C      CTDMIS- USED TO WRITE MESSAGES FOR CTDRT
0016  C      CTDFL- SETS FLOROMETER ON/OFF SWITCH
0017  C      OFF LINE PROGRAMS USED ARE:
0018  C      PLTBK - USED FOR POST PROCESSING OF DATA
0019  C      F2EOF - POSITIONS MT BETWEEN DBL EOF FOR NEW RT DATA
0020  C      MTPOS - FINDS STA & LOW NUMBERS AND POSITIONS MT AT DATA
0021  C
0022  C      DATA FILE FORMAT
0023  C      WORD 1      RECORD NUMBER
0024  C      WORD 2      TIME(HOURS)
0025  C      WORD 3      TIME(SECONDS)
0026  C      WORD 4      TIME(10XMSEC)
0027  C      WORD 5      ID MARKER FOR SPECIAL SEGMENTS - SEE CNTL
0028  C      WORD 6      FLOROMETER ON/OFF 0=OFF 1=ON
0029  C      WORDS 7-8    0
0030  C      WORDS 9-456 DIGITAL DATA
0031  C
0032  C      SEE CTDIN FOR 47 WORD INITIALIZATION FILE AND CTDTR FOR 8 WORD
0033  C      TERMINATION FILE
0034  C
0035  C      SCIENTIFIC SUBROUTINES USED ARE TAKEN FROM WOODS HOLE CTD
0036  C      PROGRAM AND ARE LISTED WITH THE USER PROGRAM
0037  C
0038  C      PROGRAMMER: LARRY ROSENBLUM
0039  C      VERSION: SEPT 20, 1979
0040  C
0041  DIMENSION NAME(3),NAME1(3),NAME2(3),NAME3(3),ITIME(5)
0042  DIMENSION NAME4(3),ISYNC(7),NAME5(3),IPRAM(5)
0043  COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0044  COMMON IBUF1(456),IBUF2(456)
0045  COMMON C1,C2,C3,C4,C5,C6,INSTR
0046  COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0047  COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0048  DATA NAME/2HCT,2HDM,2HT /
0049  DATA NAME1/2HCT,2HDI,2HN /
0050  DATA NAME2/2HCT,2HDT,2HR /
0051  DATA NAME3/2HCT,2HDP,2H1 /
0052  DATA NAME4/2HCT,2HDP,2H2 /

```

```

0053      DATA NAME5/2HCT,2HDM,2HS /
0054      LUCTD=10
0055      MSK1=1774000
0056      MSK2= 3778
0057 C      CHOOSE TYPE OF PLOT WANTED
0058      CALL EXEC(9,NAME5,1)
0059      CALL RMPAR(IPRAM)
0060      IANS=IPRAM(2)
0061      IANS1=IPRAM(3)
0062      ISTOP=-1
0063      IREC=1
0064      IFLOR=0
0065 C      SCHEDULE INIT AND PLT1 OR PLT2
0066      CALL EXEC(9,NAME1,IANS)
0067      IF(IANS.EQ.1)CALL EXEC(10,NAME3,IANS1)
0068      IF(IANS.FLT.2)CALL EXEC(10,NAME4,IANS1)
0069 C      WAIT UNTIL AXES HAVE BEEN DRAWN & EXPT READY TO COLLECT DATA
0070      CALL EXEC(9,NAME5,2)
0071      PAUSE
0072 C      GET INTO SYNC WITH THE SYNC FRAME
0073 2      CALL EXEC(1,LUCTD,ISYNC,1)
0074      ITST=IAND(ISYNC(1),MSK1)
0075      IF(ITST.NE.170000B)GO TO 2
0076      CALL EXEC(1,LUCTD,ISYNC,6)
0077 C      DETERMINE WHICH BUFFER IS TO BE FILLED
0078 12     KPAR=IREC-2*(IREC/2)
0079     IF(KPAR.EQ.0)GO TO 40
0080 C      ODD RECORD NUMBER
0081     IBUF1(1)=IREC
0082     IBUF1(6)=IFLOR
0083 C      SCHEDULE MAGTP
0084     IF(IREC.NE.1)CALL EXEC(10,NAME,IANS,IANS1)
0085 C      AQUIRE DATA
0086 60     CALL EXEC(1,LUCTD,IBUF1(9),448)
0087     IREC=IREC+1
0088     IF (ISTOP.GE.0) GO TO 99
0089     IF(IREC.LT.4)GO TO 12
0090     ITST1=IAND(MSK1,IBUF2(9))
0091     IF(ITST1.EQ.170000B)GO TO 12
0092     IF(ITST1.EQ.74000B)GO TO 12
0093     ITST1=IAND(MSK2,IBUF2(9))
0094     IF(ITST1.EQ.3600B)GO TO 12
0095     IF(ITST1.EQ.17B)GO TO 12
0096     GO TO 2
0097 C      EVEN RECORD NUMBER
0098 40     IBUF2(1)=IREC
0099     IBUF2(6)=IFLOR
0100 C      SCHEDULE MAGTP
0101     CALL EXEC(10,NAME,IANS,IANS1)
0102 C      AQUIRE DATA
0103     CALL EXEC(1,LUCTD,IBUF2(9),448)
0104     IREC=IREC+1
0105     IF(ISTOP.GE.0) GO TO 99
0106     IF(IREC.LT.4)GO TO 12

```

```

0107      ITST1=IAND(MSK1,IBUF1(9))
0108      IF(ITST1.EQ.170000B)GO TO 12
0109      IF(ITST1.EQ.007400B)GO TO 12
0110      ITST1=IAND(MSK2,IBUF1(9))
0111      IF(ITST1.EQ.360B)GO TO 12
0112      IF(ITST1.EQ.17B)GO TO 12
0113      GO TO 2
0114 C      END OR IDLE
0115 99      IF(ISTOP.NE.2) GO TO 98
0116      CALL EXEC(9,NAME5,3)
0117      PAUSE
0118      ISTOP=-1
0119      GO TO 2
0120 98      CALL EXEC(9,NAME2,IANS)
0121      END
0122      ENDS$
```

```

0001 FTN4.L
0002      PROGRAM CTDIN
0003 C
0004 C      PROG IS CALLED (WITH WAIT) BY CTDRT TO INITIALIZE PARAMS.
0005 C      PARAMETER NAMES AND INITIALIZATION FILE WERE MADE TO
0006 C      CORRESPOND TO THE WOODS HOLE PROGRAM.  SOME PARAMETERS
0007 C      ARE NOT USED IN PROGRAM.  47 WORD FILE IS WRITTEN TO MT
0008 C      ON COMPLETION.
0009 C      INSTR=INSTRUMENT NUMBER
0010 C      C1-C6 ARE CONSTS USED TO CONVERT MEAS. TO PRES, TEMP, & COND
0011 C      AXES PARAMS ARE AX LENGTH, INIT. VALUE, & INCR/INCH.  AXES
0012 C      LENGTH IS 6" FOR X AXIS AND 5" FOR Y AXIS.
0013 C
0014 C      FILE FORMAT:
0015 C      WORDS 1-6      12 CHARACTER INPUT I. D. STRING
0016 C      WORD 7      STATION NUMBER
0017 C      WORDS 8-10     DATE: YEAR, MONTH, DAY
0018 C      WORDS 11-12    LOWERING #, INSTRUMENT #
0019 C      WORDS 13-24    CALIBRATION CONSTS (REAL VALUED)
0020 C      WORDS 25-30    PRESSURE AXIS CONSTS (REAL VALUED)
0021 C      WORDS 31-36    TEMPERATURE AXIS CONSTS (REAL VALUED)
0022 C      WORDS 37-42    SALINITY AXIS CONSTS (REAL VALUED)
0023 C      WORDS 43-46    LATITUDE & LONGITUDE (REAL VALUED)
0024 C      WORD 47      0
0025 C
0026 C      PROGRAMMER: LARRY ROSENBLUM
0027 C
0028      DIMENSION IBUF5(47),IPRAM(5)
0029      COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0030      COMMON IBUF1(456),IBUF2(456)
0031      COMMON C1,C2,C3,C4,C5,C6,INSTR
0032      COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0033      COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0034      EQUIVALENCE (XC1,IBUF5(13)),(XC2,IBUF5(15))
```

```

0035      EQUIVALENCE (XC3,IBUF5(17)),(XC4,IBUF5(19)),(XC5,IBUF5(21))
0036      EQUIVALENCE (XC6,IBUF5(23)),(XPAXL,IBUF5(25)),(XPIV,IBUF5(27))
0037      EQUIVALENCE (XPINCR,IBUF5(29)),(XTAXL,IBUF5(31)),(XTIV,IBUF5(33))
0038      EQUIVALENCE (XTINCR,IBUF5(35)),(XSAXL,IBUF5(37)),(XSIV,IBUF5(39))
0039      EQUIVALENCE (XSINCR,IBUF5(41)),(XLAT,IBUF5(43)),(XLON,IBUF5(45))
0040      CALL RMPAR(IPRAM)
0041      C      DEFAULT VALUES
0042      5      INSTR=1
0043      C1=0.1
0044      C2=0.8
0045      C3=0.0
0046      C4=.0005
0047      C5=0.0010
0048      C6=0.0
0049      PAXL=5.
0050      PIV=0.
0051      PINCR=1000.
0052      TAXL=6.
0053      IF(IPRAM(1).EQ.2)TAXL=5.
0054      TIV=1.
0055      TINCR=3.0
0056      IF(IPRAM(1).EQ.2)TINCR=4.0
0057      SAXL=6.
0058      IF(IPRAM(1).EQ.2)SAXL=5.
0059      SIV=32.
0060      IF(IPRAM(1).EQ.2)SIV=33.
0061      SINCR=1.0
0062      IBLNK=20040B
0063      DO 37 I=1,6
0064      ISHIP(I)=IBLNUK
0065      37      CONTINUE
0066      WRITE(1,10)
0067      10      FORMAT("PROGRAM CTDIN RUNNING")
0068      C      READ IN PARAMETER VALUES
0069      WRITE(1,12)
0070      12      FORMAT("ENTER@ SHIP + CRUISE # (12 CHARACTERS")*
0071      READ(1,13)(ISHIP(I),I=1,6)
0072      13      FORMAT(6A2)
0073      WRITE(1,14)
0074      14      FORMAT("ENTER DATE(YR,MO,DAY")*
0075      READ(1,*)IYR,IMO,IDAY
0076      WRITE(1,16)
0077      16      FORMAT("ENTER STATION NUMBER AND LOWERING NUMBER")
0078      READ(1,*)IST,LOW
0079      WRITE(1,17)
0080      17      FORMAT("ENTER LATITUDE: DEG, MIN & FRACT. OF MIN XXXX.,XX.XXX")
0081      READ(1,*)XLAT1,XLAT2
0082      WRITE(1,19)
0083      19      FORMAT("ENTER LONGITUDE:DEG, MIN & FRACT. OF MIN XXXX.,XX.XXX")
0084      READ(1,*)XLON1,XLON2
0085      XLAT=60.0*XLAT1+XLAT2
0086      XLON=60.0*XLON1+XLON2
0087      WRITE(1,18)
0088      18      FORMAT("USE DEFAULT VALUES, 1=YES 2=NO")
0089      READ(1,*)KANS
0090      IF(KANS.EQ.1)GO TO 99
0091      WRITE(1,20)
0092      20      FORMAT("DEFAULT VALUES GIVEN IN PARENTHS")
0093      WRITE(1,22)

```

```

0094  22  FORMAT("INSTRUMENT # (I)")
0095  READ(1,*)INSTR
0096  WRITE(1,26)
0097  26  FORMAT("CAL CONST PRESSURE C1,C2 (0.1,0.0)")
0098  READ(1,*)C1,C2
0099  WRITE(1,28)
0100  28  FORMAT("CAL CONST TEMP C3,C4 (0.0,0.0005)")
0101  READ(1,*)C3,C4
0102  WRITE(1,30)
0103  30  FORMAT("CAL CONST COND C5,C6 (0.001,0.0)")
0104  READ(1,*)C5,C6
0105  WRITE(1,31)
0106  31  FORMAT("FOR EACH AXIS THE AXIS LENGTH IS GIVEN IN PARENTHESIS")
0107  WRITE(1,33)
0108  33  FORMAT("ENTER THE INITIAL VALUE AND THE INCREMENT/INCH")
0109  WRITE(1,32)PAXL
0110  32  FORMAT("PRESSURE(AX LNTH=",I2,")ENTER: INIT VALUE, INCR INCH")
0111  READ(1,*)PIV,PINCR
0112  WRITE(1,34)TAXL
0113  34  FORMAT("TEMP(AX LNTH=",I2,")ENTER INIT VALUE, INCR/INCH")
0114  READ(1,*)TIV,TINCR
0115  WRITE(1,36)SAXL
0116  36  FORMAT("SAL(AX LNTH=",I2,")ENTER INIT VALUE, INCR.INCH")
0117  READ(1,*)SIV,SINCR
0118 C      WRITE OUT VALUES
0119  99  DO 98 I=1,6,5
0120  WRITE(I,60)(ISHIP(J),J=1,6)
0121  60  FORMAT(1X,6A2)
0122  WRITE(I,62) IYR,IMO,IDAY,IST,LOW
0123  62  FORMAT(1X, "DATE ",3I4," STA ",I4," LOW ",I4)
0124  WRITE(I,64)INSTR
0125  64  FORMAT(1X,"INSTR ",I2)
0126  WRITE(I,66)C1,C2,C3,C4,C5,C6
0127  66  FORMAT(1X,"CAL CONSTS ",6F9.4)
0128  WRITE(I,68)PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0129  68  FORMAT(1X,"AXES CONSTS ",9F6.2)
0130  WRITE(I,70)XLAT1,XLAT2,XLON1,XLON2
0131  70  FORMAT("LAT:",F4.0,"DEG",F6.2,"MIN",4X,"LON:",F4.0,"DEG",
0132  1   F6.2,"MIN")
0133  98  CONTINUE
0134  WRITE(1,105)
0135  105 FORMAT("ARE INPUT PARAMS O.K.: 1=YES, 2=NO, 3=YES - NO MT FILE")
0136  READ(1,*)IANS
0137  IF(IANS.EQ.2)GO TO 5
0138 C      WRITE FILE OUT TO TAPE
0139  IF(IANS.EQ.3) GO TO 199
0140  DO 110 I=1,6
0141  IBUF5(I)=ISHIP(I)
0142  110 CONTINUE
0143  IBUF5(?)=IST
0144  IBUF5(8)=IYR
0145  IBUF5(9)=IMO
0146  IBUF5(10)=IDAY
0147  IBUF5(11)=LOW
0148  IBUF5(12)=INSTR
0149  IBUF5(47)=0
0150  XC1=C1
0151  XC2=C2
0152  XC3=C3

```

L. J. ROSENBLUM

```

0153      XC4=C4
0154      XC5=C5
0155      XC6=C6
0156      XPAXL=PAXL
0157      XPIV=PIV
0158      XPINCR=PINCR
0159      XTAXL=TAXL
0160      XTIV=TIV
0161      XTINCR=TINCR
0162      XSAXL=SAXL
0163      XSIV=SIV
0164      XSINCR=SINCR
0165  C      WRITE RECORD AND EOF
0166      CALL EXEC(2,8,IBUF5,47)
0167      CALL EXEC(3,8+100B)
0168  199  END
0169  ENDS

```

```

0001  FTN4.L
0002  PROGRAM CTDML(3,90)
0003  C
0004  C      PROGRAM TO WRITE OUT DOUBLE BUFFERED DATA COLLECTED BY
0005  C      CTDRT TO MAG. TAPE. ALSO SCHEDULES RT PLOTTING PROG AND
0006  C      GETS TIME.
0007  C
0008  C      PROGRAMMER: LARRY ROSENBLUM
0009  C      VERSION SEPT 22 1979
0010  C
0011  C      NOTE: 1ST RECORD MAY HAVE BAD TIME
0012  C
0013  DIMENSION ITIME(5),IPRAM(5),NAME3(3),NAME4(3)
0014  COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0015  COMMON IBUF1(456),IBUF2(456)
0016  COMMON C1,C2,C3,C4,C5,C6,INSTR
0017  COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0018  COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0019  DATA NAME3/2HCT,2HDP,2H1 /
0020  DATA NAME4/2HCT,2HDP,2H2 /
0021  5      CALL RMPAR(IPRAM)
0022  IANS =IPRAM(1)
0023  IANS1=IPRAM(2)
0024  KPAR=IREC-2*(IREC/2)
0025  C      GET CURRENT TIME
0026  CALL EXEC(11,ITIME)
0027  IHR=24*ITIME(5)+ITIME(4)
0028  ISEC=60*ITIME(3)+ITIME(2)
0029  IMSC=ITIME(1)
0030  C      SCHED. PLOTTING PROG EVERY 4TH TIME
0031  TEST=IREC-4*(IREC/4)

```

```

0032      IF(TEST.NE.3)GO TO 15
0033      IF(IANS.EQ.1)CALL EXEC(10,NAME3,IANS1)
0034      IF(IANS.EQ.2)CALL EXEC(10,NAME4,IANS1)
0035  15   IF (KPAR.EQ.0) GO TO 10
0036      IBUF2(2)=IHR1
0037      IBUF2(3)=ISEC1
0038      IBUF2(4)=IMSC1
0039      CALL EXEC(2,8,IBUF2,456)
0040      GO TO 20
0041  10   IBUF1(2)=IHR1
0042      IBUF1(3)=ISEC1
0043      IBUF1(4)=IMSC1
0044      CALL EXEC(2,8,IBUF1,456)
0045  C     SAVE OLD TIME
0046  20   IHR1=IHR
0047      ISEC1=ISEC
0048      IMSC1=IMSC
0049      IF(ISTOP.EQ.1.OR.ISTOP.EQ.3)GO TO 99
0050      CALL EXEC(6,0,1)
0051      GO TO 5
0052  99   END
0053      END$
```

```

0001  FTN4,L
0002  PROGRAM CTDP1
0003  C
0004  C     REAL TIME PLOTTING PROGRAM FOR CTD.
0005  C     USES VALUE OF IREC FROM COMMON TO PLOT ONE POINT EVERY 8TH
0006  C     BUFFER.  USES VALUE OF ISTOP TO END OR REINITIALIZE.
0007  C     RELOCATABLES: XLRSAL, XR, XPORTMP
0008  C
0009  C     PROGRAMMER: LARRY ROSENBLUM
0010  C     VERSION: SEPT 18, 1979
0011  C
0012  DIMENSION DAT(3), NAMES(4), NAMEP(6), NAMEP(4), NASTA(2), NALOW(2)
0013  DIMENSION IPRAM(5), NAPOT(7)
0014  COMMON IREC, ISTOP, ISCAN, NSCAN, IFLOR
0015  COMMON IBUF1(456), IBUF2(456)
0016  COMMON C1,C2,C3,C4,C5,C6,INSTR
0017  COMMON ISHIP(6), IST, IYR, IMO, IDAY, LOW
0018  COMMON PAXL, PIV, PINCR, TAXL, TIV, TINCR, SAXL, SIV, SINCR
0019  DATA NAMEP/2HPR,2HES,2HSU,2HRE/
0020  DATA NAMES/2HSA,2HLI,2HNI,2HTY/
0021  DATA NAMET/2HTE,2HMP,2HER,2HAT,2HUR,2HE /
0022  DATA NAPOT/2HPO,2HTE,2HNT,2HIA,2HL,2HTE,2HMP/
0023  DATA NASTA/2HST,2HA /
0024  DATA NALOW/2HLO,2HW /
0025  CALL RMPAR(IPRAM)
0026  IANS1=IPRAM(1)
0027  CALL EXEC(22,0)
```

```

0028    7    CALL PLTLU(7)
0029  C      DRAW AXES, LABEL, AND ANNOTATE
0030  NSTIC=INT(SAXL)+1
0031  NPTIC=INT(PAXL)+1
0032  NTTIC=INT(TAXL)+1
0033  C      NEXT LINE TO FUDGE FOR 2647 TERMINAL
0034  PAXL=0.8*PAXL
0035  CALL AX(0.6,0.4,SAXL,NSTIC,3,0.0)
0036  CALL AX(0.6,PAXL+0.4,PAXL,NPTIC,3,-90.)
0037  CALL AX(.6+TAXL,.4+PAXL,TAXL,NTTIC,3,-180.)
0038  CALL LABEL(0.6,0.2,.1,SAXL,NSTIC,SIV,SINCR,2,0.0)
0039  CALL SYMB(3.2,0.0,.1,NAMES,0.0,8)
0040  CALL LABEL(0.3,PAXL+0.4,.1,-PAXL,NPTIC,PIV,PINCR,2,+90.)
0041  CALL SYMB(0.1,2.4,.1,NAMEP,90.0,8)
0042  CALL LABEL(.6,.6+PAXL,.1,TAXL,NTTIC,TIV,TINCR,2,0.)
0043  IF(IANS1.EQ.1)CALL SYMB(3.00,.8+PAXL,.1,NAMET,0.0,12)
0044  IF(IANS1.EQ.2)CALL SYMB(3.00,.8+PAXL,.1,NAPOT,0.0,14)
0045  CALL SYMB(.7+SAXL,-.4+PAXL,.1,NASTA,0.0,4)
0046  CALL SYMB(.7+SAXL,-.7+PAXL,.1,NALOW,0.0,4)
0047  RIST=R(IST)
0048  RLOW=R(LOW)
0049  CALL LABEL(1.2+SAXL,-.4+PAXL,.1,0.,0.,RIST,0.0,-1,0.0)
0050  CALL LABEL(1.2+SAXL,-.7+PAXL,.1,0.,0.,RLOW,0.0,-1,0.0)
0051  CALL PLOT(0.6,0.4,-3)
0052  JREC=3
0053  KOUNT=0
0054  C      RESET PAXL TO ORIG. VALUE FOR COMPUTATION (2647 CORRECT)
0055  PAXL=1.25*PAXL
0056  C      SUSPEND
0057  30  CALL EXEC(6,0,1)
0058  C      CHECK FOR END OR PLOT REINITIALIZATION
0059  IF(ISTOP.EQ.1.OR.ISTOP.EQ.3) GO TO 99
0060  IF(ISTOP.EQ.-2)GO TO 98
0061  RE1=R(IBUF2(12))
0062  RE2=R(IBUF2(10))
0063  RE3=R(IBUF2(11))
0064  DAT(2)=C1*(RE2-C2)
0065  DAT(3)=C3+(C4*RE3)
0066  DAT(1)=C5*RE1*(1+(C6*DAT(2)))
0067  C      GET TEMPERATURE SIGN
0068  C      **** CTD HAS PROBLEM WITH SIGN BYTE ****
0069  C  ISGN1=IAND(IBUF2(13),2B)
0070  C  IF(ISGN1.EQ.2)DAT( 3)=-DAT( 3)
0071  C      COND=DAT1, PRESS=DAT2, TEMP=DAT3
0072  C      CALL SALIN CONVERSION SUBROUTINE. REPLACE COND WITH SALIN IN DAT1
0073  CALL LRSAL(DAT(1),DAT(2),DAT(3),S)
0074  IF(IANS1.EQ.2)WRITE(1,779)DAT(3)
0075  779 FORMAT("TEMP=",F10.4)
0076  DAT(1)=S
0077  C      DAT1=SAL DAT2=PRESS DAT3=TEMP
0078  C      IF REQUIRED, CONVERT TEMP INTO POTENTIAL TEMP
0079  IF(IANS1.EQ.2)CALL POTMP(DAT(2),DAT(3),DAT(1),PTMP)
0080  IF(IANS1.EQ.2)DAT(3)=PTMP
0081  21  CALL PLT(DAT(1),DAT(3),DAT(2))
0082  GO TO 30
0083  98  ISTOP=-1
0084  GO TO 7
0085  99  END
0086  SUBROUTINE PLT(S,T,P)

```

NRL REPORT 8444

```

0087      COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0088      COMMON IBUF1(456),IBUF2(456)
0089      COMMON C1,C2,C3,C4,C5,C6,INSTR
0090      COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0091      COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0092      WRITE(1,40)S,T,P
0093      WRITE(6,40)S,T,P
0094 40    FORMAT(3F10.4)
0095      SMAX=SIV+(SINCR*SAXL)
0096      TMAX=TIV+(TINCR*TAXL)
0097      PMAX=PIV+(PINCR*PAXL)
0098      IF(S.LT.SIV.OR.S.GT.SMAX)RETURN
0099      IF(T.LT.TIV.OR.T.GT.TMAX)RETURN
0100      IF(P.LT.PIV.OR.P.GT.PMAX)RETURN
0101      SSCAL=SAXL/(SMAX-SIV)
0102      SPLT=SSCAL*(S-SIV)
0103      TSCAL=TAXL/(TMAX-TIV)
0104      TPLT=TSCAL*(T-TIV)
0105      PSCAL=PAXL/(PMAX-PIV)
0106      PPLT=PSCAL*(PMAX-P)
0107      NUMB1=1
0108      NUMB2=3
0109  C     ADJUST Y SCALE FOR 2647 TERMINAL
0110      PPLT=0.8*PPLT
0111      CALL SYMB(SPLT,PPLT,.03,NUMB1,0.0,-1)
0112      CALL SYMB(TPLT,PPLT,.03,NUMB2,0.0,-1)
0113      RETURN
0114      END
0115      END$
```

```

0001  FTN4,L
0002      PROGRAM CTDP2
0003  C
0004  C     REAL TIME CTD PLOTTING PROGRAM WHEN CTD USED IN TOWING MODE.
0005  C     PLOTS S,T,P VS. TIME. PLOT IS REINITIALIZED EVERY 30 MINUTES.
0006  C     RELOCATABLES: XLRSL, XR, XPTMP
0007  C
0008  C     PROGRAMMER: LARRY ROSENBLUM
0009  C     VERSION: SEPT 18, 1979
0010  C
0011      DIMENSION DAT(3), NAMES(2), NAMET(2), NAMEP(2), NASTA(2), NALOW(2)
0012      DIMENSION NAMTM(4), NALBL(5), IPRAM(5), NAPOT(4)
0013      COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0014      COMMON IBUF1(456),IBUF2(456)
0015      COMMON C1,C2,C3,C4,C5,C6,INSTR
0016      COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0017      COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0018      DATA NAMEP/2HPR,2HES/
0019      DATA NAMES/2HSA,2HL /
0020      DATA NAMET/2HTE,2HMP/
```

L. J. ROSENBLUM

```

0021      DATA NAPOT/2HPO,2HT,2HTE,2HMP/
0022      DATA NAMTM/2HTI,2HME,2H M,2HIN/
0023      DATA NASTA/2HST,2HA /
0024      DATA NALDW/2HLD,2HW /
0025      DATA NALBL/2HST,2HAR,2HT,2HTI,2HME/
0026      CALL RMPAR(IPRAM)
0027      IANS1=IPRAM(1)
0028      CALL EXEC(22,0)
0029      C      FIX AXES LENGTH AND SET TIME AXIS PARAMETERS
0030      SAXL=5.
0031      PAXL=5.
0032      TAXL=5.
0033      TMAXL=6.0
0034      TMIV=0.0
0035      TMINCR=5.0
0036      7      CALL PLTLU(7)
0037      C      DRAW AXES, LABELS, AND ANNOTATE
0038      LR=1
0039      NSTIC=INT(SAXL)+1
0040      NPTIC=INT(PAXL)+1
0041      NTTIC=INT(TAXL)+1
0042      KNUM1=42
0043      KNUM2=45
0044      KNUM3=46
0045      LNUM1=5
0046      LNUM2=0
0047      LNUM3=3
0048      CALL AX(1.0,0.4,6.,7,4,0,0)
0049      CALL LABEL(1.0,0.2,.1,6.,7,TMIV,TMINCR,-1,0,0)
0050      CALL SYMB(3.4,0.0,.1,NAMTM,0.0,8)
0051      CALL AX(1.0,4.4,4.0,6,4,-90,0)
0052      CALL LABEL(0.9,4.4,.1,-4,.6,PIV,PINCR,2,90,)
0053      CALL SYMB(0.9,0.0,.1,KNUM1,90.,-1)
0054      CALL SYMB(0.1,0.5,.1,LNUM1,90.,-1)
0055      CALL SYMB(0.15,0.6,.1,NAMESP,90.,4)
0056      CALL LABEL(0.7,4.4,.1,-4,.6,SIV,SINCR,2,90,)
0057      CALL SYMB(0.7,0.0,.1,KNUM2,90.,-1)
0058      CALL SYMB(0.1,2.0,.1,LNUM2,90.,-1)
0059      CALL SYMB(0.15,2.1,.1,NAMES,90.,4)
0060      CALL LABEL(0.5,4.4,.1,-4,.6,TIV,TINCR,2,90,)
0061      CALL SYMB(0.5,0.0,.1,KNUM3,90.,-1)
0062      CALL SYMB(0.1,3.5,.1,LNUM3,90.,-1)
0063      IF(IANS1.EQ.1)CALL SYMB(0.15,3.6,.1,NAMET,90.,4)
0064      IF(IANS1.EQ.2)CALL SYMB(0.15,3.6,.1,NAPOT,90.,8)
0065      CALL SYMB(2.5,4.6,.1,NALBL,0.0,10)
0066      CALL SYMB(7.1,3.6,.1,NASTA,0.0,4)
0067      CALL SYMB(7.1,3.3,.1,NALDW,0.0,4)
0068      CALL AX(7.0,0.4,4.,6,4,90,)
0069      RIST=R(IST)
0070      RLOW=R(LOW)
0071      CALL LABEL(7.6,3.6,.1,0.,0.,RIST,0.0,-1,0,0)
0072      CALL LABEL(7.6,3.3,.1,0.,0.,RLOW,0.0,-1,0,0)
0073      CALL PLOT(1.0,0.4,-3)
0074      JREC=3
0075      KOUNT=0
0076      C      SUSPEND
0077      30     CALL EXEC(6,0,1)
0078      IF(ISTOP.EQ.1.OR.ISTOP.EQ.3) GO TO 99
0079      IF(ISTOP.EQ.-2)GO TO 98

```

```

0080      IF(LR.NE.1) GO TO 32
0081      NDAT1=IBUF2(2)
0082      NDAT2=IBUF2(3)
0083      NTM1=IBUF2(2)/24
0084      NTM2=IBUF2(2)-NTM1*24
0085      NTM3=IBUF2(3)/60
0086      NTM4=IBUF2(3)-NTM3*60
0087      NDAT2=NDAT2-NTM4
0088      NTM4=0
0089      CALL NUMB(3.4,4.2,.1,FLOAT(NTM1),0.0,-1)
0090      CALL NUMB(3.9,4.2,.1,FLOAT(NTM2),0.0,-1)
0091      CALL NUMB(4.3,4.2,.1,FLOAT(NTM3),0.0,-1)
0092      CALL NUMB(4.7,4.2,.1,FLOAT(NTM4),0.0,-1)
0093 32    NDEL1=IBUF2(2)-NDAT1
0094      NDEL2=IBUF2(3)-NDAT2
0095      IF(NDEL1.NE.0)NDEL2=NDEL2+3600
0096      IF(NDEL2.GT.1800) GO TO 98
0097      RE1=R(IBUF2(12))
0098      RE2=R(IBUF2(10))
0099      RE3=R(IBUF2(11))
0100      DAT(2)=C1*(RE2-C2)
0101      DAT(3)=C3+(C4*RE3)
0102      DAT(1)=C5*RE1*(1+(C6*DAT(2)))
0103  C     CHECK TEMPERATURE SIGN
0104      ISGN1=IAND(IBUF2(12),2B)
0105  C     **** CTD HAS TROUBLE IN SIGN BYTE ****
0106  C     IF(ISGN1.EQ.2)DAT(3)--DAT(3)
0107  C     DAT1=COND,DAT2=PRESS,DAT3=TEMP - ALL CONVT TO ENG UNITS
0108  C     CALL SALIN CONVERS SUBROUTINE
0109      CALL LRSAL(DAT(1),DAT(2),DAT(3),S)
0110      DAT(1)=S
0111  C     IF IANS1=2 USE POTENTIAL TEMPERATURE
0112      IF(IANS1.EQ.2)CALL POTMP(DAT(2),DAT(3),DAT(1),PTMP)
0113      IF(IANS1.EQ.2)DAT(3)=PTMP
0114 21    CALL PLT(DAT(1),DAT(3),DAT(2),NDEL2,LR)
0115      LR=LR+1
0116      GO TO 30
0117 98    ISTOP=-1
0118      GO TO 7
0119 99    END
0120      SUBROUTINE PLT(S,T,P,NDEL2,LR)
0121      COMMON IREC,ISTOP,ISCAN,NSCAN,IFLDR
0122      COMMON IBUF1(456),IBUF2(456)
0123      COMMON C1,C2,C3,C4,C5,C6,INSTR
0124      COMMON ISHIP(6),IST,IYR,IMD>IDAY,LOW
0125      COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0126      WRITE(1,40)NDEL2,S,T,P
0127 40    FORMAT(I8,3F10.4)
0128      SMAX=SIV+(SINCR*SAXL)
0129      TMAX=TIV+(TINCR*TAXL)
0130      PMAX=PIV+(PINCR*PAXL)
0131      XIV=60.*0.0
0132      XINCR=60.*5.
0133      XMAX=XIV+XINCR*6.
0134      IF(S.LT.SIV.OR.S.GT.SMAX)RETURN
0135      IF(T.LT.TIV.OR.T.GT.TMAX)RETURN
0136      IF(P.LT.PIV.OR.P.GT.PMAX)RETURN
0137      SSCAL=SAXL/(SMAX-SIV)
0138      SPLT=SSCAL*(SMAX-S)

```

L. J. ROSENBLUM

```
0139      TSCAL=TAXL/(TMAX-TIV)
0140      TPLT=TSCAL*(TMAX-T)
0141      PSCAL=PAXL/(PMAX-PIV)
0142      PPLT=PSCAL*(PMAX-P)
0143      XDEL=R(NDEL2)
0144      XSCAL=6.//(XMAX-XIV)
0145      XPLT=XSCAL*(XDEL-XIV)
0146      NUMB1=0
0147      NUMB2=3
0148      NUMB3=5
0149      SIZE=0.1
0150      IF(LR.NE.1)SIZE=0.03
0151 C      SCALE Y AXIS FOR 2647 TERMINAL
0152      SPLT=0.8*SPLT
0153      TPLT=0.8*TPLT
0154      PPLT=0.8*PPLT
0155      CALL SYMB(XPLT,SPLT,SIZE,NUMB1,0.0,-1)
0156      CALL SYMB(XPLT,TPLT,SIZE,NUMB2,0.0,-1)
0157      CALL SYMB(XPLT,PPLT,SIZE,NUMB3,0.0,-1)
0158      RETURN
0159      END
0160      END$
```

```
0001 FTN4,L
0002      PROGRAM CTDCN(3,14)
0003 C
0004 C      CONRTOL PROGRAM TO CAUSE END OR IDLE.
0005 C      ISTOP CHANGED IN COMMON BY 'RU,CTDCN,,N' WHERE
0006 C      N=-1 NORMAL CONDITION
0007 C      N= 1 NORMAL END
0008 C      N= 2 IDLE MODE
0009 C      N= 3 ABORT
0010 C      N=-2 BEGIN NEW PLOT
0011 C      IBUF1&2(5) USED TO MARK SPECIAL FILE POINTS
0012 C      USING IPARAM(1)
0013 C
0014 C      PROGRAMMER: LARRY ROSENBLUM
0015 C      VERSION: SEPT 22 1979
0016 C
0017      DIMENSION IPARAM(5)
0018      COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0019      COMMON IBUF1(456),IBUF2(456)
0020      COMMON C1,C2,C3,C4,C5,C6,INSTR
0021      COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0022      COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0023      CALL RMPAR(IPARAM)
0024      CALL EXEC(22,0)
0025      IF(IPARAM(2).EQ.0)IPARAM(2)--1
0026      ISTOP=IPARAM(2)
0027      IBUF1(5)=IPARAM(1)
0028      IBUF2(5)=IPARAM(1)
0029      END
0030      END$
```

```

0001 FTN4.L
0002     PROGRAM CTDMS
0003     DIMENSION IPRAM(5),IBACK(5)
0004     CALL RMPAR(IPRAM)
0005     IF(IPRAM(1).EQ.2)GO TO 20
0006     IF(IPRAM(1).EQ.3)GO TO 30
0007   5    WRITE(1,10)
0008   10   FORMAT("PLOT TYPE? 1=S,T VS. P (LOWERING),/\
0009          1           2=S,T,P VS TIME (TOWING)")
0010          READ(1,*)IANS
0011          IF(IANS.NE.1.AND.IANS.NE.2) GO TO 5
0012   12   WRITE(1,15)
0013   15   FORMAT("TEMPATURE TYPE; 1=TEMP., 2=POTENTIAL TEMP.")
0014          READ(1,*)IANS1
0015          IF(IANS1.NE.1.AND.IANS1.NE.2)GO TO 12
0016          IBACK(2)=IANS
0017          IBACK(3)=IANS1
0018          CALL PRTN(IBACK)
0019          CALL EXEC(6)
0020   20   WRITE(1,25)
0021   25   FORMAT("PAUSE - TYPE: 'GO,CTDRT' FOR DATA AQUISITION")
0022          GO TO 99
0023   30   WRITE(1,35)
0024   35   FORMAT("PAUSE - IDLE MODE - TYPE: 'GO,CTDRT' TO CONTINUE")
0025   99   END
0026   ENDS

```

```

0001 FTN4.L
0002     PROGRAM CTDFL(3,16)
0003 C
0004 C THIS PROGRAM SETS THE FLOROMETER ON/OFF SWICTH.
0005 C IFLOR=0 FLOROMETER OFF
0006 C IFLOR=1 FLOROMETER ON
0007 C TYPE: 'RU,CTDFL,N' WHERE N=2 FOR OFF OR N=3 FOR ON
0008 C
0009 C PROGRAMMER: LARRY ROSENBLUM
0010 C VERSION: SEPT. 20, 1979
0011     DIMENSION IPRAM(5)
0012     COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0013     COMMON IBUF1(456),IBUF2(456)
0014     COMMON C1,C2,C3,C4,C5,C6,INSTR
0015     COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0016     COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0017     CALL RMPAR(IPRAM)
0018     IF(IPRAM(1).EQ.2)IFLOR=0
0019     IF(IPRAM(1).EQ.3)IFLOR=1
0020   10   IF(IPRAM(1).NE.2.AND.IPRAM(1).NE.3)WRITE(1,10)
0021          FORMAT("TYPE 'RU,CTDFL,N' WHERE N=2 (OFF) OR =3(ON)")
0022          END
0023   ENDS

```

```

0001 FTN4,L
0002     PROGRAM CTDTR(3,98)
0003 C
0004 C      PROGRAM TO WRITE TERMINATION FILE AND OPEN PLOTTING PROG. SO
0005 C      THAT IT GETS CLOSED CORRECTLY. TERM IN SCHEDULED BY CTDRT
0006 C      WHEN ISTOP= 1 OR 3 - SEE CTDCN.
0007 C
0008 C      FORMAT OF 8 WORD TERMINATION FILE:
0009 C      WORD 1      -1
0010 C      WORDS 2&3   TIME IN HOURS AND SECONDS
0011 C      WORD 4      ISTOP/3
0012 C      WORDS 5-8    0
0013 C
0014 C      PROGRAMMER: LARRY ROSENBLUM
0015 C      VERSION: SEPT. 20, 1979
0016 C
0017 DIMENSION IBUF4(8),ITIME(5),NAME1(3),NAME2(3),IPRAM(5)
0018 COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0019 COMMON IBUF1(456),IBUF2(456)
0020 COMMON C1,C2,C3,C4,C5,C6,INSTR
0021 COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0022 COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0023 DATA NAME1/2HCT,2HDP,2H1 /,NAME2/2HCT,2HDP,2H2 /
0024 C      GET PLOT VERSION PARAMETER
0025 CALL RMPAR(IPRAM)
0026 IANS=IPRAM(1)
0027 CALL EXEC(22,0)
0028 C      WRITE EOF AFTER DATA
0029 CALL EXEC(3,8+100B)
0030 C      SCHEDULE PLOTTING PROGRAM
0031 IF(IANS.EQ.1)CALL EXEC(10,NAME1)
0032 IF(IANS.EQ.2)CALL EXEC(10,NAME2)
0033 C      WRITE TERMINATION FILE AND DBL EOF
0034 IBUF4(1)--!
0035 CALL EXEC(11,ITIME)
0036 IBUF4(2)=24*ITIME(5)+ITIME(4)
0037 IBUF4(3)=60*ITIME(3)+ITIME(2)
0038 IBUF4(4)=ISTOP/3
0039 IBUF4(5)=0
0040 IBUF4(6)=0
0041 IBUF4(7)=0
0042 IBUF4(8)=0
0043 CALL EXEC(2,8,IBUF4,8)
0044 CALL EXEC(3,8+100B)
0045 CALL EXEC(3,8+100B)
0046 END
0047 ENDS

```

```
0001 FTN,L
0002      PROGRAM F2EOF
0003 C
0004 C      VERSION 7/3/77
0005 C      DEAN CLAMONS
0006 C
0007 C      THIS PROGRAM FINDS THE FIRST DOUBLE EOF ON A TAPE
0008 C      AND POSITION ONE TAPE AFTER THE FIRST OF THE PAIR
0009 C      THE METHOD IS TO BACK UP TWO RECORDS (IN CASE WE
0010 C      ARE ALREADY AT THE EOF'S) AND THEN SPACE FORWARD
0011 C      A FILE AND READ THE NEXT RECORD.
0012 C      IF IT IS AN EOF WE PACK OFF ONE RECOR AND ARE DONE
0013 C      IF NOT AN EOF WE SPACE FORWARD ANOTHER FILE
0014 C      AND REPEAT
0015 C
0016 C      NOTE!!!!!
0017 C      THE TAPE MUST HAVE A DOUBLE EOF FOR THIS
0018 C      PROGRAM TO WORK, IF NOT IT WILL RON AWAY
0019 C
0020 LMT+3
0021 CALL EXEC(3,200B+LMT)
0022 CALL EXEC(3,200B+LMT)
0023 1 CALL EXEC(3,1300B+LMT)
0024 CALL EXEC(1,LMT,1BFF,1)
0025 IF(IEOF(LMT).GE.0)GO TO 1
0026 CALL EXEC(3,200B+LMT)
0027 END
0028 END$
```

USER'S GUIDE TO THE REAL-TIME CTD PROGRAMS

Preparation

Prior to starting the CTD data collection process the operator should ensure that the system is set for CTD data. The operator should:

1. Check that power is on to the CTD deck unit and the unit is displaying reasonable values.
2. Toggle the "hold" switch on the CTD deck unit several times to assure that the sync frame is good. The sync frame indicator should display either 15 or 240 (decimal).
3. Check that the Computer Applications Group CTD Interface has been switched to the "ON" position.
4. If the computer system has been performing other functions, assure that the logical units (LU) assignments are correct. The CTD programs expect that LU-1 is the system console, LU-6 is the line printer, LU-7 is the terminal plotter, LU-8 is the magnetic tape unit, and LU-10 is the CTD. If necessary, logical unit assignments should be changed using the RTE operator "LU" command, so that these logical units will point to the appropriate equipment table entry.
5. Mount a tape on the tape drive and check that the subchannel indicator on the tape unit is set to 0 and that the tape is "write" enabled (the write ring is installed).
6. Obtain the latitude and longitude from a laboratory display, a computer program for this purpose, or the navigation laboratory.

If a CTD lowering (or tow) has just been completed, then, to begin the next lowering (or tow), the operator should get an RTE system prompt * and type "RU, F2EOF." This program will position the tape between the two EOF's which follow the termination record of the previous CTD. Perform step 6 above.

Program Start-Up

The next two sections of this chapter describe the interaction between the operator and CTD real-time programs. For purposes of identification within this report only, all user responses are underlined. All user entries are to be followed by a carriage return. The symbol * is the RTE system prompt and may be obtained in an RTE system by striking any key on the console one time. Answers to questions are by way of example.

* RU, CTDRT

ENTER CTD MODE, 1 = LOWERING, 2 = TOWING

1

ENTER TEMPERATURE TYPE, 1 = TEMP, 2 = POTENTIAL TEMP

2

ENTER SHIP'S NAME (CRUISE) (UP TO 12 CHARACTERS)

HAYES 9999

ENTER DATE (YR, MO, DAY)

1979 10 8

ENTER STATION NUMBER & LOWERING NUMBER

21

ENTER LATITUDE: DEG. MIN. & FRACTION OF MIN xxxx., xxxx.

32. 15.25

ENTER LONGITUDE: DEG, MIN. & FRACTION OF MIN xxxx., xx.xxx

-65. 22.23

At this point the operator is asked

USE DEFAULT VALUES, 1 = YES, 2 = NO

The default values concern the instrument number, the values of the constants C1, C2, . . . , C6 used to convert readings into pressure, temperature, and conductivity and the axes limits for the real-time plots. The conversion constant default values are: C1 = 0.1, C2 = 0.0, C3 = 0.0, C4 = 0.0005, C5 = 0.001, and C6 = 0.0. The only constant with which the operator need be concerned is C1, which is dependent on the depth rating of the pressure sensor. The default is for the 6,500-decibar (65-MPa) pressure sensor. If a different pressure sensor is used, C1 should be changed to the value shown in Table 1.

Table 1—Values of the
Conversion Constant C1 for Each of the
Different CTD Pressure Sensors

Pressure-Sensor Range (decibar)	C1 Value
0-6500	0-65
0-3200	0-32
0-1600	0-16
0-650	0-6.5

The axes lengths are fixed at 6 inches (15.2 cm) for the x axis and 5 inches (12.7 cm) for the y axis. The operator supplies the initial value and the increments per inch for each parameter—pressure, temperature, and salinity. The default values for lowering plots are given in Table 2 and for towing plots in Table 3. In general, except for deep-water lowerings, the operator will want to specify more limited axes scaling using knowledge of the area and depth involved. Returning to the question about default values, let us assume we wish to lower a 1600-decibar (16-MPa) pressure sensor from 200 to 500 decibars (2 to 5 MPa) and that we know that temperature will be in the range 14 to 19°C and salinity in the range 34.5 to 36.0 mmho(mS). In the following the symbol " CR" will denote a single space followed by a carriage return. Then, continuing our sample question and answer, the programs would ask:

USE DEFAULT VALUES, 1 = YES, 2 = NO

2

DEFAULT VALUES GIVEN IN PARENTHESIS

INSTRUMENT # (1)

 CR

CAL CONST PRESSURE C1, C2 (0.1, 0.0)

0.025 0.0

CAL CONST TEMP C3, C4 (0.0, 0.0005)

 CR

CAL CONST C5, C6 (0.001, 0.0)

 CRFOR EACH AXIS THE AXIS LENGTH IS GIVEN IN PARENTHESIS ENTER
THE INITIAL VALUE AND THE INCREMENT/INCH.

PRESSURE (AX LNTH = 5) ENTER INIT VALUE, INCR INCH

200.0 60.0

TEMP (AX LNTH = 6) ENTER INIT VALUE, INCR/INCH

14.0 1.0

SAL (AX LNTH = 6) ENTER INIT VALUE, INCR/INCH

34.5 0.25

L. J. ROSENBLUM

Table 2—Plot Default Values for the Lowering Mode

Parameter	Axis Length (in.)	Axis Length (cm)	Initial Value	Incr (per in.)	Incr (per cm)	Range
Pressure	5.0	12.7	0	1000	394	0–5000 dbar (0–50 MPa)
Temperature	6.0	15.2	1	4	1.6	1–25 °C
Salinity	6.0	15.2	32	1	0.4	32–38 mS

Table 3 - Plot Default Values for the Towing Mode

Parameter	Axis Length (in.)	Axis Length cm	Initial Value	Incr (per in.)	Incr (per cm)	Range
Pressure	5.0	12.7	0	10	3.9	0 – 50 dbar
Temperature	5.0	12.7	1	4	1.6	(0 – 0.5 MPa)
Salinity	5.0	12.7	33	1	0.4	33 – 38 mS
Time	6.0	15.2	Start time of plot	5	2.0	0 – 30 min

The operator entries are printed out on the terminal and written to the line printer. The operator is then given the option of restarting the question process if any mistakes have been made (the answer "3" to the following question is for debug purposes only and should not be used).

ARE INPUT PARAMS O.K.: 1 = YES, 2 = NO, 3 = YES – NO MT FILE

1

PAUSE—TYPE 'GO, CTDRT' FOR DATA ACQUISITION

At this point the plot axes and annotation will be output on the plotting terminal. The operator should verify that these are correct, especially the parameter limits on each axis. Once the plot axes and annotation are drawn the program is prepared to acquire data. The operator now obtains the system prompt (*) and types:

* GO, CTDRT

This completes the start-up procedure.

Special Functions

The program CTDCN is used to

- end data taking,
- abort data taking,
- put the program into an "idle" mode, or
- mark special segments.

The end and abort produce exactly the same result, except that a word in the termination file indicates which way the program was terminated. The idle mode is used to suspend data collection temporarily and the special segment marker changes one word in the data buffers. This last operation can be used to identify data segments for later processing. These operations occur when the operator types:

for a NORMAL TERMINATION

* RU, CTDCN,,1

for an ABORT

* RU, CTDCN,,3

for an IDLE

* RU, CTDCN,,2

The message PAUSE-IDLE MODE-TYPE: 'GO, CTDRT' TO CONTINUE will appear on the system console. To continue the collection process the operator types:

* GO, CTDRT
for marking SPECIAL SEGMENTS
* RU, CTDCN,N

where *N* is any number. This number is placed in Word 5 of each data buffer until it is changed or the programs are terminated.

The program CTDFL sets the fluorometer on/off switch.

The command

* RU,CTDFL, 3

turns the switch on (fluorometer data being taken) and the command

* RU, CTDFL, 2

turns the switch off. It should be noted that the switch is initialized as "off" and that it is used only as a marker; failure to set the switch does not alter the data.

DATA-PROCESSING PROGRAMS

Parameters of Interest

Once a CTD data tape has been collected, many parameters of interest to the oceanographic researcher can be computed from the stored values of conductivity, temperature, and pressure. To assist the researcher a program called PLTBK has been written which will compute values and produce plots of various parameters. To do this the user is offered the following parameter list:

- 1 = Salinity
- 2 = Pressure
- 3 = Temperature
- 4 = Potential Temperature
- 5 = Sound Velocity
- 6 = Density
- 7 = Specific Volume
- 8 = Adiabatic Temperature Gradient
- 9 = Sigma (0)
- 10 = Depth
- 11 = Brunt-Vaisala Frequency
- 12 = Time (minutes)
- 13 = $d(\text{temp}) / d(\text{pressure})$
- 14 = $d(\text{salinity}) / d(\text{pressure})$
- 15 = Conductivity
- 16 = Fluorometer

Program PLTBK allows the user to compute and plot (against each other) any pair of the above parameters or, alternately, to define any one parameter as the y axis and plot any two variables as x-axis parameters against the selected y-axis parameters. Thus, for example, if the user wanted to obtain a sound-speed profile the user would select 5(sound velocity) for the x axis and 10(depth) for the y axis. To obtain a plot of temperature and conductivity vs pressure, the user would select 3(temperature) for

the x axis, 2(pressure) for the y axis and 15(conductivity) as a second x axis. The axes scales for the plot are user selected. Scientific subroutines for calculating these parameters are listed in the appendix. These subroutines were provided by the Woods Hole Oceanographic Institution, except for the Brunt-Vaisala frequency calculation, which uses an approximation formula and is valid for the top several hundred meters.

User's Guide to Program PLTBK

This section describes the interaction between the user and program PLTBK. As before, user responses will be underlined and answers are by way of example.

* RU, PLTBK

ENTER STATION NUMBER AND LOWERING NUMBER

2 1

DO YOU WANT A TIME SEARCH? 1=YES, 2=NO

A time search allows the user to look at only part of an event; otherwise the entire event will be processed.

1

ENTER START TIME—DAY HR MIN SEC

171 5 27 0

ENTER STOP TIME—DAY HR MIN SEC

171 6 31 0

1=SALINITY

2=PRESSURE

3=TEMPERATURE

4=POT. TEMP

5=SOUNDVEL

6=DENSITY

7=SPEC VOL

8=ATG

9=SIGMA

10=DEPTH

11=B-V FREQ

12=TIME

13=D(T) /D(P)

14=D(S) /D(P)

15=CONDUCTIVITY

16=FLUOROMETER

ENTER X AXIS PARAMETER NUMBER

3

ENTER Y AXIS PARAMETER NUMBER

10

DO YOU WANT A SECOND X AXIS, 1=YES, 2=NO

1

ENTER X2 AXIS PARAMETER NUMBER

15

ENTER X AXIS LENGTH, INIT VALUE, INCR/INCH

6 3. 15.

ENTER Y AXIS LENGTH, INIT VALUE, INCR/INCH

5 2000. 400.

ENTER X2 AXIS LENGTH, INIT VALUE, INCR/INCH

6 32. .5

Since using all points usually provides too dense a plot, the user chooses the plotting density by answering the following question:

OF POINTS/4 SEC SCAN; MUST DIVIDE 64

8

The program will now plot the data and stop.

Examples of Program PLTBK Graphics

In this section, several examples of program PLTBK plots are given. The plots were produced using PLTBK from a digital tape recorded on the USNS *Lynch* in April 1978 in the Atlantic off the continental shelf of the United States. Figure 6 shows salinity and temperature plotted against pressure for

NRL REPORT 8444

the full lowering (approx. 100 m to 4600 m) plotting one point every 4 seconds. Figure 7 shows salinity for the top 1000 meters, plotting one point every second. Figure 8 is the sound-velocity profile for the time and site. Figure 9 is a temperature-salinity plot for the lowering. Figure 10 is a density plot and Figure 11 plots density vs potential temperature. Illustrating the ability to plot any two parameters against any other, Figure 12 shows salinity and temperature plotted against sound speed.

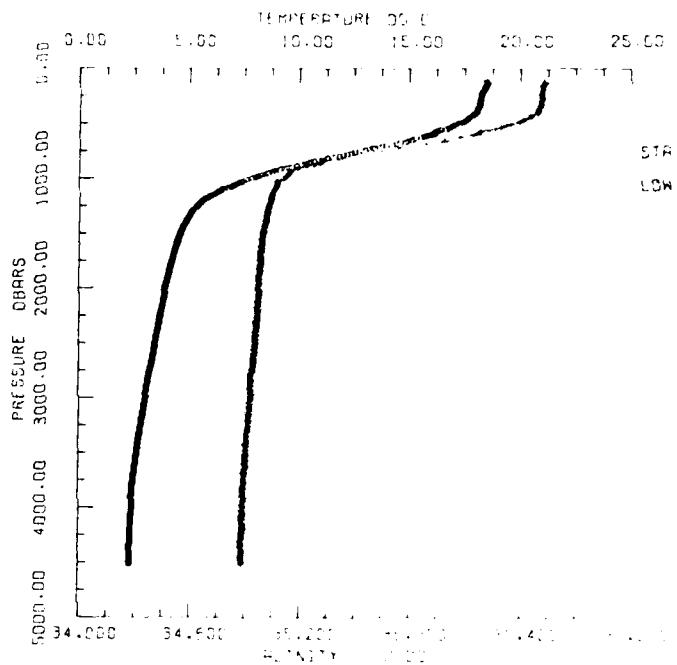


Figure 6 — Plot of salinity and temperature vs pressure for the entire lowering

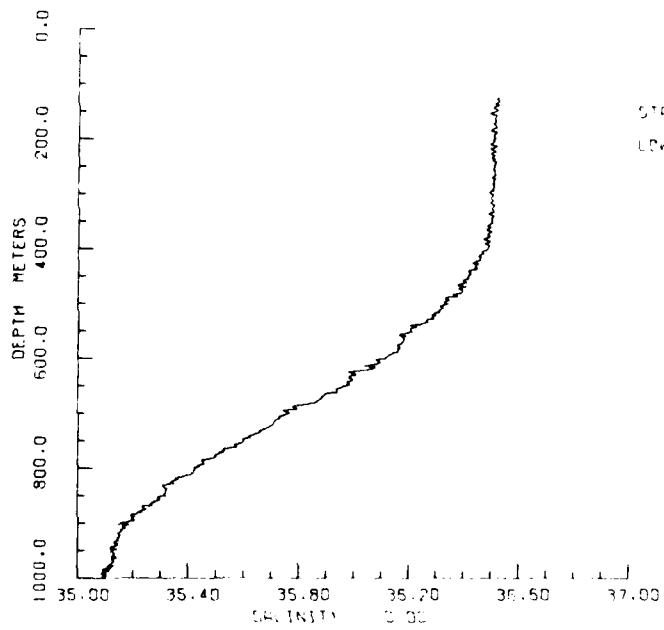


Figure 7 — Expanded salinity plot for top 1000 m

L. J. ROSENBLUM

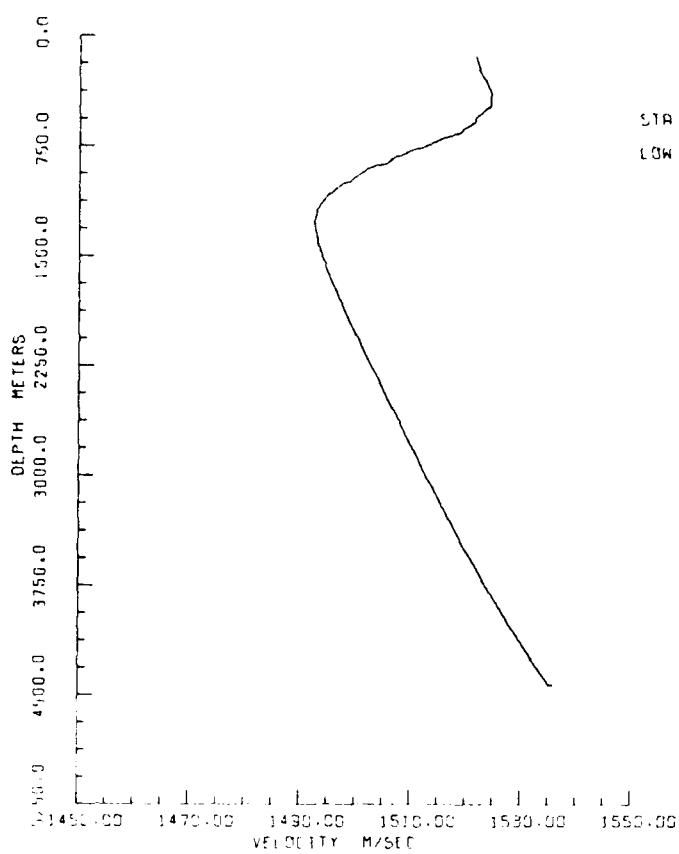


Figure 8 — Sound velocity profile plot for site

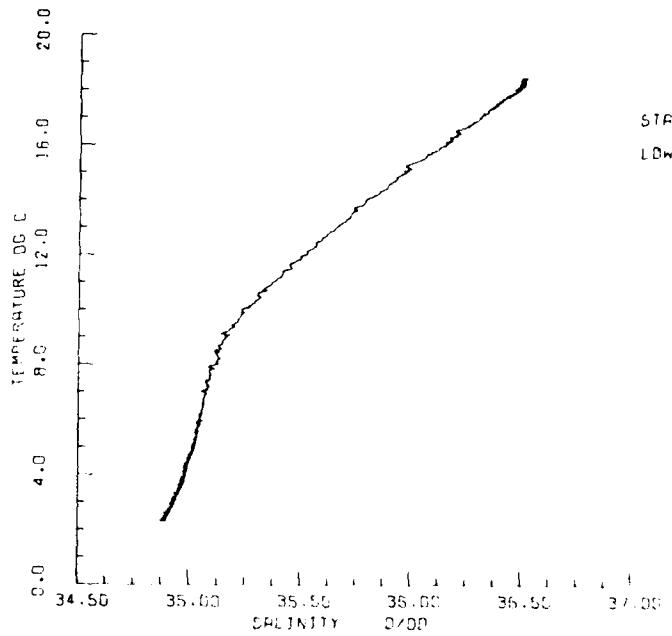


Figure 9 — Temperature - salinity plot for the lowering

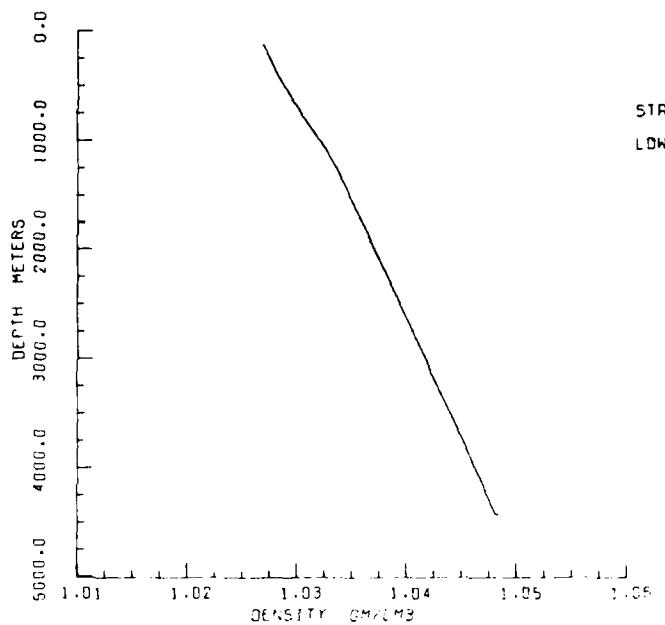


Figure 10 — Density plot for the lowering

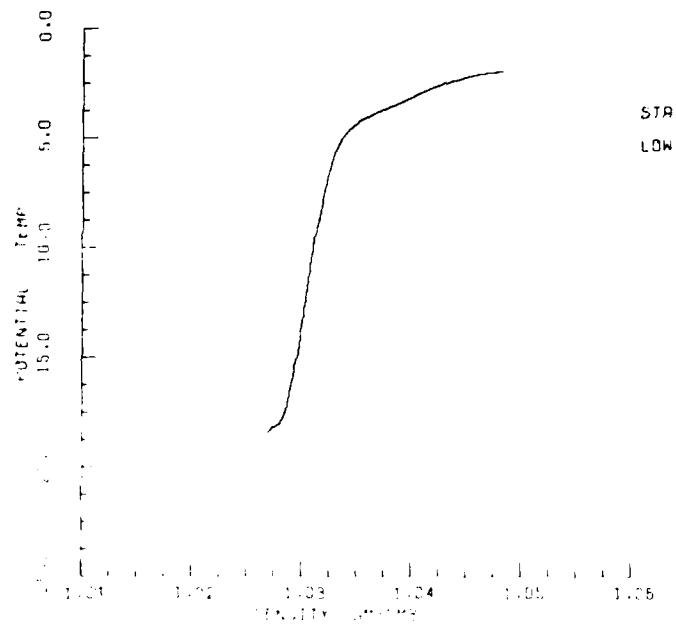


Figure 11 — Plot of density vs potential temperature

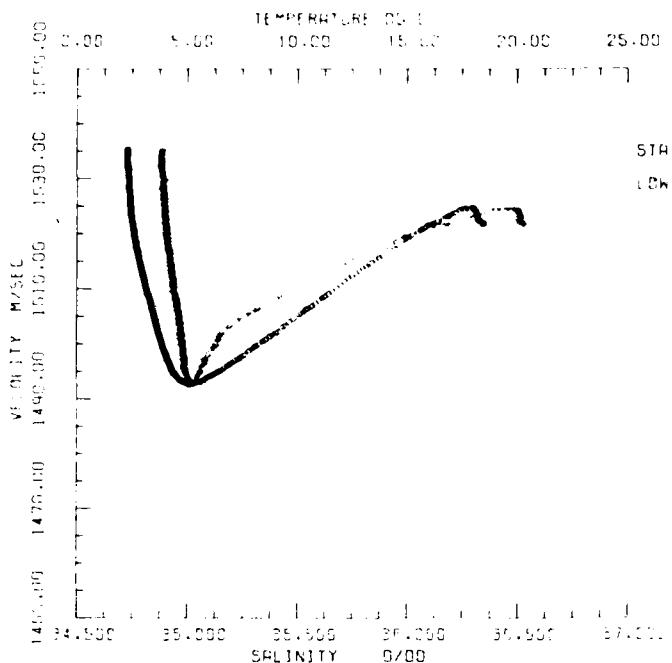


Figure 12 — Plot of salinity and temperature vs sound velocity

Data Processing Programs—Fortran Source Listings

```

0001  FTN4,L
0002  PROGRAM PLTBK
0003  C
0004  C      THIS VERSION REVISED TO WORK FOR FLUOROMETER DATA SET 10 /79
0005  C      THIS PROGRAM WILL PRODUCE PLOTS FROM THE COLLECTED CTD DATA
0006  C      TAPE.  ANY PARAMETER FROM THE PARAMETER LIST CAN BE PLOTTED
0007  C      AGAINST ANY OTHER PARAMETER OR AGAINST ANY PAIR OF PARAMETERS
0008  C      FROM THE LIST.  TAPE MUST BE POSITIONED AT THE DATA.  SCIENTIFIC
0009  C      SUBROUTINES USED WERE SUPPLIED BY W.H.O.I. EXCEPT FOR THE APPROX.
0010  C      USED FOR BRUNT-V AISALA FREQUENCY.
0011  C
0012  C      RELOCATABLE SUBROUTINES: %LRSAL,%R,%POTMP,%SVEL,%ATG,%SIGMT,
0013  C                           %SPVOL,%BRVA1
0014  C
0015  C      THIS VERSION ADDS THE FLUOROMETER.
0016  C
0017  C      PROGRAMMER: LARRY ROSENBLUM      VERSION: SEPT. 1979
0018  C
0019  DIMENSION DAT(16),NAMEP(8),NAMES(8),NAMET(8),NAMPT(8),NAME(8,16)
0020  DIMENSION NAMEV(8),NASTA(2),NALOW(2),NAHEN(8),NASPV(8),NAATG(8)
0021  DIMENSION NASMT(8),NADPH(8),NABVF(8),NATME(8),NALBL(5),NADLT(8)
0022  DIMENSION NADLS(8),NAFLR(8),NACON(8)
0023  DIMENSION IREG(2),XBV(500),IAx(3)
0024  DOUBLE PRECISION XALPH,XBV,DENS,SUMRH,XSD,XST
0025  COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0026  COMMON IBUF1(456),IBUF2(456)
0027  COMMON C1,C2,C3,C4,C5,C6,INSTR
0028  COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW

```

```

0029      COMMON X1AXL,X1IV,X1INCR,YAXL,YIV,YINCR,X2AXL,X2IV,X2INCR
0030      EQUIVALENCE (NAME$,NAME(1,1)),(NAMEP,NAME(1,2))
0031      EQUIVALENCE (NAME$,NAME(1,3)),(NAMPT,NAME(1,4))
0032      EQUIVALENCE (NAMEV,NAME(1,5)),(NADEN,NAME(1,6))
0033      EQUIVALENCE (NASPV,NAME(1,7)),(NAATG,NAME(1,8))
0034      EQUIVALENCE (NASMT,NAME(1,9)),(NADPH,NAME(1,10))
0035      EQUIVALENCE (NABVF,NAME(1,11)),(NATME,NAME(1,12))
0036      EQUIVALENCE (NADLT,NAME(1,13)),(NADLS,NAME(1,14))
0037      EQUIVALENCE (NACON,NAME(1,15)),(NAFLR,NAME(1,16))
0038      EQUIVALENCE (REG,IREG)
0039      DATA NAMEP/2HPR,2HES,2HSU,2HRE,2H ,2HDB,2HAR,2HS /
0040      DATA NAMES/2HSA,2HLI,2HNI,2HTY,2H ,2H ,2H0/,2H00/
0041      DATA NAMET/2HTE,2HMP,2HER,2HAT,2HUR,2HE ,2HDG,2H C/
0042      DATA NAMPT/2HPO,2HTE,2HNT,2HIA,2HL ,2H ,2HTE,2HMP/
0043      DATA NAMEV/2HVE,2HLO,2HCl,2HTY,2H ,2HM/,2HSE,2HC /
0044      DATA NADEN/2HDE,2HNS,2HIT,2HY ,2H G,2HM/,2HCM,2H3 /
0045      DATA NASPV/2HSP,2H ,2HVO,2HL ,2H ,2HCM,2H3/,2HGM/
0046      DATA NAATG/2HAD,2HIA,2HBT,2H T,2HEM,2HP ,2HGR,2HAD/
0047      DATA NASMT/2HSI,2HGM,2HA ,2H T,2H ,2HGM,2H/C,2HM3/
0048      DATA NADPH/2H ,2HDE,2HPT,2HH ,2H M,2HET,2HER,2HS /
0049      DATA NABVF/2HBR,2H-V,2HA ,2HFR,2HEQ,2H C,2HY/,2HHR/
0050      DATA NATME/2H ,2HTI,2HME,2H ,2H ,2HMI,2HN ,2H /
0051      DATA NADLT/2H ,2HDE,2HL (,2HT),2H/D,2HEL,2H(P,2H) /
0052      DATA NADLS/2H ,2HDE,2HL (,2HS),2H/D,2HEL,2H(P,2H) /
0053      DATA NACON/2H ,2HCO,2HND,2HUC,2HTI,2HVI,2HTY,2H /
0054      DATA NAFLR/2H ,2H F,2HLO,2HRO,2HME,2HTE,2HR ,2H /
0055      DATA NASTA/2HST,2HA /
0056      DATA NALOW/2HLO,2HW /
0057      DATA NALBL/2HST,2HAR,2HT ,2HTI,2HME/
0058      C
0059      LUMT=8
0060      POLD=0.0
0061      WRITE(1,111)
0062      111 FORMAT("ENTER STATION NUMBER AND LOWERING NUMBER")
0063      READ(1,*)IST,LOW
0064      WRITE(1,200)
0065      200 FORMAT("DO YOU WANT A TIME SEARCH? 1=YES, 2=NO")
0066      READ(1,*)IANS1
0067      IF(IANS1.NE.1) GO TO 206
0068      WRITE(1,202)
0069      202 FORMAT("ENTER START TIME - DAY HRS MIN SECS")
0070      READ(1,*)IST1,IST2,IST3,IST4
0071      WRITE(1,204)
0072      204 FORMAT("ENTER STOP TIME - DAY HRS MIN SECS")
0073      READ(1,*)ISP1,ISP2,ISP3,ISP4
0074      ISP5=24*ISP1+ISP2
0075      IST5=24*IST1+IST2
0076      IST6=60*IST3+IST4
0077      ISP6=60*ISP3+ISP4
0078      206 WRITE(1,12)
0079      12 FORMAT(" 1=SALINITY      2=PRESSURE      3=TEMPERATURE",/
0080      1     " 4=POT. TEMP      5 =SOUND VEL.    6=DENSITY",/
0081      2     " 7=SPEC. VOL.      8=ATG          9=SIGMA T(MOD)",/
0082      3     "10=DEPTH          11=B-V FREQ.    12=TIME(MIN)",/
0083      4     "13=D(T)/D(P)      14=D(S)/D(P)    15=CONDUCTIVITY",/
0084      5     "16=FLOROMETER")
0085      WRITE(1,101)
0086      101 FORMAT("ENTER X AXIS PARAMETER NUMBER")
0087      READ(1,*)IAX(1)

```

```

0088      WRITE(1,102)
0089  102  FORMAT("ENTER Y AXIS PARAMETER NUMBER")
0090      READ(1,*)IAX(2)
0091      WRITE(1,105)
0092  105  FORMAT("DO YOU WANT A 2ND X AXIS: 1=YES, 2=NO")
0093      READ(1,*)IOPT
0094      IF(IOPT.EQ.2) GO TO 123
0095      WRITE(1,106)
0096  106  FORMAT("ENTER X2 AXIS PARAMETER NUMBER")
0097      READ(1,*)IAX(3)
0098  123  IF(IAX(1).NE.11.AND.IAX(3).NE.11.AND.IAX(2).NE.11)GO TO 113
0099      WRITE(1,127)
0100  127  FORMAT("BE SURE THAT B-V FREQ. IS PLOTTED VS. DEPTH ONLY")
0101      WRITE(1,124)
0102  124  FORMAT("ENTER B-V FREQ.DEPTH BIN SIZE, BETWEEN 1M. & 20 M.")
0103      READ(1,*)AVDPT
0104      IF(AVDPT.LT.1.0.OR.AVDPT.GT.20.0)WRITE(1,126)
0105  126  FORMAT("FREQ. BIN NOT WITHIN REQUIRED RANGE")
0106      IF(AVDPT.LT.1.0.OR.AVDPT.GT.20.0)GO TO 123
0107      WRITE(1,125)
0108  125  FORMAT("ENTER SURFACE DENSITY-OBTAINT FROM DENSITY PLOT")
0109      READ(1,*)RH00
0110  C      INITIALIZE BRUNT-VAISALA AVERAGING COUNTERS
0111      NUMBV=1
0112      DPTBV=0.0+AVDPT
0113      DLAST=0.0
0114  113  IF(IAX(1).NE.12.AND.IAX(3).NE.12.AND.IAX(2).NE.12.) GO TO 107
0115      WRITE(1,110)
0116  110  FORMAT("ENTER PLOT START TIME: DAYS,HRS,MIN,SECS")
0117      READ(1,*)NT1,NT2,NT3,NT4
0118      NDAT1=24*NT1+NT2
0119      NDAT2=60*NT3+NT4
0120  107  WRITE(1,103)
0121  103  FORMAT("ENTER:X AXIS LENGTH, INIT. VALUE, INCR/INCH")
0122      READ(1,*)X1AXL,X1IV,X1INCR
0123      WRITE(1,104)
0124  104  FORMAT("ENTER: Y AXIS LENGTH, INIT. VALUE, INCR/INCH")
0125      READ(1,*)YAXL,YIV,YINCR
0126      IF(IOPT.EQ.2) GO TO 109
0127      WRITE(1,108)
0128  108  FORMAT("ENTER: X2 AXIS LENGTH, INIT. VALUE, INCR/INCH")
0129      READ(1,*)X2AXL,X2IV,X2INCR
0130  109  CALL INIT1
0131      WRITE(1,11)
0132  11   FORMAT("# OF PTS/4 SEC SCAN; MUST DIVIDE 2**? ")
0133      READ(1,*)INUM
0134      JSKIP=448/INUM
0135      CALL PLTLU(?)
0136      NYTIC=INT(YAXL)+1
0137      NX1TIC=INT(X1AXL)+1
0138      NX2TIC=INT(X2AXL)+1
0139      CALL AX(0.6,0.4,X1AXL,NX1TIC,3,0.0)
0140      CALL AX(0.6,YAXL+0.4,YAXL,NYTIC,3,-90.)
0141      CALL LABEL(0.6,0.2,.1,X1AXL,NX1TIC,X1IV,X1INCR,3,0.0)
0142      CALL SYMB(-.3+X1AXL/2.,0.0,.1,NAME(1,IAX(1)),0.0,16)
0143      CALL LABEL(0.3,YAXL+0.4,.1,-YAXL,NYTIC,YIV,YINCR,2,+90.)
0144      CALL SYMB(0.1,-.3+YAXL/2.,.1,NAME(1,IAX(2)),90.0,16)
0145      CALL SYMB(.7+X1AXL,-.4+YAXL,.1,NASTA,0.0,4)

```

NRL REPORT 8444

```

0146      CALL SYMB(.7+X1AXL,-.7+YAXL,.1,NALOW,0.0,4)
0147      RIST=R(IST)
0148      RLOW=R(LOW)
0149      CALL LABEL(1.2+X1AXL,-.4+YAXL,.1,0.,0.,RIST,0.0,-1,0.0)
0150      CALL LABEL(1.2+X1AXL,-.7+YAXL,.1,0.,0.,RLOW,0.0,-1,0.0)
0151      IF(IOPT.EQ.2) GO TO 19
0152      CALL AX(.6+X2AXL,.4+YAXL,X2AXL,NX2TIC,3,-180.)
0153      CALL LABEL(.6,.6+YAXL,.1,X2AXL,NX2TIC,X2IV,X2INCR,2,0.)
0154      CALL SYMB(-.3+X2AXL/2.,.8+YAXL,.1,NAME(1,IAX(3)),0.0,16)
0155      19   CALL PLOT(0.6,0.4,-3)
0156      IF(IAX(1).NE.12.AND.IAX(2).NE.12.AND.IAX(3).NE.12) GO TO 15
0157      CALL SYMB(2.1,5.2,.1,NALBL,0.0,10)
0158      CALL NUMB(3.4,5.2,.1,FLOAT(NT1),0.0,-1)
0159      CALL NUMB(3.9,5.2,.1,FLOAT(NT2),0.0,-1)
0160      CALL NUMB(4.3,5.2,.1,FLOAT(NT3),0.0,-1)
0161      CALL NUMB(4.7,5.2,.1,FLOAT(NT4),0.0,-1)
0162      C      READ IN DATA
0163      15   REG= EXEC(1,LUMT,IBUF2,456)
0164      ILEN=IREG(2)
0165      IF(IANS1.NE.1) GO TO 208
0166      IF(IEOF(LUMT).LT.0) GO TO 15
0167      IF(IBUF2(2).LT.IST5) GO TO 15
0168      IF(ILEN.NE.456) GO TO 15
0169      IF(IBUF2(2).EQ.IST5.AND.IBUF2(3).LT.IST6) GO TO 15
0170      IF(IBUF2(2).GT.ISP5)GO TO 99
0171      IF(IBUF2(2).EQ.ISP5.AND.IBUF2(3).GT.ISP6) GO TO 99
0172      208   IF(IEOF(LUMT).LT.0) GO TO 99
0173      DO 20 I=9,456,JSKIP
0174      RE1=R(IBUF2(I+3))
0175      RE2=R(IBUF2(I+1))
0176      RE3=R(IBUF2(I+2))
0177      C      CALCULATE CONDUCTIVITY, PRESSURE, AND TEMPERATURE
0178      DAT1=COND, DAT2=PRESS, DAT3=TEMP
0179      THEN CONVERT TO SALINITY AND PUT DAT1=SAL
0180      STORE COND IN DAT(15)
0181      C
0182      DAT(2)=C1*(RE2-C2)
0183      DAT(3)=C3+(C4*RE3)
0184      DAT(1)=C5*RE1*(1+(C6*DAT(2)))
0185      GET SIGN OF TEMPERATURE
0186      **** MAY BE PROBLEM IN CTD WITH STATUS BIT ****
0187      ITST1=IAND(IBUF2(I+4),2B)
0188      IF(ITST1.EQ.2) DAT(3)=-DAT(3)
0189      DAT(15)=DAT(1)
0190      CALL LRSAL(DAT(1),DAT(2),DAT(3),S)
0191      DAT(1)=S
0192      WRITE(6,992)DAT(2),DAT(3),DAT(1)
0193      992   FORMAT(1X,3(F12.2,2X))
0194      C      CALCULATE FLOROMETER READING
0195      IF(IAX(1).NE.16.AND.IAX(2).NE.16.AND.IAX(3).NE.16)GO TO 21
0196      DAT(16)=FLOAT(IBUF2(I+5))*FLOAT(IBUF2(I+6))
0197      C      CALCULATE POTENTIAL TEMPERATURE
0198      21   IF(IAX(1).NE.4.AND.IAX(2).NE.4.AND.IAX(3).NE.4) GO TO 24
0199      CALL POTMP(DAT(2),DAT(3),DAT(1),PTMP)
0200      DAT(4)=PTMP
0201
0202
0203
0204

```

```

0205 C
0206 C      CALCULATE SOUND VELOCITY
0207 C
0208 24 IF(IAX(1).NE.5.AND.IAX(2).NE.5.AND.IAX(3).NE.5) GO TO 26
0209 CALL SVEL(DAT(2),DAT(3),DAT(1),DAT(5))
0210 C
0211 C      CALCULATE SPECIFIC VOLUME AND DENSITY
0212 C
0213 26 IF(IAX(1).EQ.7.OR.IAX(3).EQ.7.OR.IAX(2).EQ.7) GO TO 29
0214 IF(IAX(1).EQ.6.OR.IAX(3).EQ.6.OR.IAX(2).EQ.6) GO TO 29
0215 IF(IAX(1).EQ.9.OR.IAX(3).EQ.9.OR.IAX(2).EQ.9) GO TO 29
0216 IF(IAX(1).EQ.11.OR.IAX(3).EQ.11.OR.IAX(2).EQ.11) GO TO 29
0217 GO TO 28
0218 29 CALL SIGMT(DAT(3),DAT(1),X50,XST)
0219 CALL SPVOL(DAT(2),DAT(3),X50,XST,XALPH)
0220 DAT(7)=XALPH
0221 DENS=1.0/XALPH
0222 DAT(6)=1./DAT(7)
0223 DAT(9)=1000.*(DAT(6)-1.0)
0224 C
0225 C      CALCULATE ADIABATIC TEMPERATURE GRADIENT
0226 C
0227 28 IF(IAX(1).NE.8.AND.IAX(3).NE.8.AND.IAX(2).NE.8) GO TO 30
0228 CALL ATG(DAT(2),DAT(3),DAT(1),GAMMA)
0229 DAT(8)=GAMMA
0230 C
0231 C      CALCULATE DEPTH FROM PRESSURE
0232 C
0233 30 IF(IAX(1).EQ.10.OR.IAX(2).EQ.10.OR.IAX(3).EQ.10) GO TO 32
0234 IF(IAX(1).EQ.11.OR.IAX(2).EQ.11.OR.IAX(3).EQ.11) GO TO 32
0235 GO TO 31
0236 32 IF(DAT(2).LE.600.0) DAT(10)=DAT(2)/1.009
0237 IF(DAT(2).LE.600.0) GO TO 31
0238 A=.980665*2.022365
0239 A2=1.E-6
0240 A=A*A2
0241 B=.980665*1.02940
0242 C=-DAT(2)
0243 ROOTZ=SQRT((B**2)-4*A*C)
0244 DAT(10)=(-B+ROOTZ)/(2*A)
0245 C      PRES(DBARS)=.980665*(1.02940Z+2.022365*10**-6*Z**2) Z IN METERS
0246 C
0247 C      CALCULATE BRUNT-VAISALA FREQUENCY
0248 C      AVERAGE DENSITIES WITHIN EACH BIN AND THEN COMPUTE BV
0249 C
0250 31 IF(IAX(1).NE.11.AND.IAX(2).NE.11.AND.IAX(3).NE.11) GO TO 756
0251 C      CHECK FOR 1ST POINT OF NEW SEGMENT
0252 IF(NUMBV.NE.1) GOTO 33
0253 DMX2=DPTBV
0254 DMN2=DPTBV-AVDPT
0255 IF(DAT(10).GE.DMN2.AND.DAT(10).LE.DMX2) DFRST=DAT(10)
0256 IF(DAT(10).GE.DMN2.AND.DAT(10).LE.DMX2) GO TO 252
0257 DLAST=DAT(10)
0258 DPTBV=DLAST+AVDPT
0259 GO TO 20
0260 C      SEE IF POINT WITHIN BOUNDS, IFSO ADD TO AVERAGE
0261 33 DMX1=DPTBV
0262 DMN1=DPTBV-AVDPT
0263 IF(DAT(10).GT.DMX1.OR.DAT(10).LT.DMN1)GO TO 250

```

```

0264      IF(NUMBV.GT.500)GO TO 250
0265  C      DEPTHS TOO CLOSE TOGETHER CAUSES ARITH. PROBS IN
0266  C      DIFFERENCE METHOD, SO DON'T ALLOW
0267      IF(DAT(10)-DLAST.LT.0.05)GO TO 20
0268  252  XBV(NUMBV)=DENS
0269      DLAST=DAT(10)
0270      NUMBV=NUMBV+1
0271      GO TO 20
0272  C      AVERAGE DENSITIES, COMPUTE BV, AND REINITIALIZE
0273  250  SUMRH=0.0
0274  C      DO NOT USE INTERVAL UNLESS HAS CONTAINED AT LEAST 4 VALID PTS
0275      IF(NUMBV.GT.5) GO TO 255
0276      NUMBV=1
0277      DPTBV=DLAST+AVDPT
0278      GO TO 20
0279  255  DO 260 J=1,NUMBV-1
0280      SUMRH=SUMRH+XBV(J)
0281  260  CONTINUE
0282      SUMRH=SUMRH/FLOAT(NUMBV-1)
0283      CALL BRVA1(DPTBV,SUMRH,RH00,BV)
0284      NUMBV=1
0285      DPTBV=DLAST+AVDPT
0286      DAT(11)=BV
0287  C      FOR PLOTTING PURPOSES PUT DAT(10) = AVG FOR INTERVAL
0288      DAT(10)=(DLAST+DFRST)/2.0
0289      WRITE(6,270)DAT(10),SUMRH,DAT(11)
0290  270  FORMAT(1X,"DEPTH=",F10.2,4X,"RHO=",F10.6,4X,"BV FREQ=",F10.2)
0291  C
0292  C      TIME PLOTS
0293  C
0294  756  IF(IAX(1).NE.12.AND.IAX(3).NE.12.AND.IAX(2).NE.12) GO TO 84
0295      IF(NDAT1.GT.IBUF2(2))GO TO 15
0296      IF(NDAT1.EQ.IBUF2(2).AND.NDAT2.GT.IBUF2(3))GO TO 15
0297      NDEL1=IBUF2(2)-NDAT1
0298      NDEL2=IBUF2(3)-NDAT2+(NDEL1*3600)
0299      DAT(12)=(FLOAT(NDEL2)/60.)+((FLOAT(I-8))/5760.)
0300      IF(IAX(1).EQ.12)TX=X1AXL*X1INCR
0301      IF(IAX(3).EQ.12)TX=X2AXL*X2INCR
0302      IF(IAX(2).EQ.12)TX=YAXL*YINCR
0303      IF(DAT(12).GT.TX) GO TO 99
0304  C
0305  C      COMPUTE DEL(T)/DEL(P) AND DEL(S)/DEL(P)
0306  C
0307  84  IF(IAX(1).EQ.13.OR.IAX(2).EQ.13.OR.IAX(3).EQ.13)GO TO 85
0308      IF(IAX(1).EO.14.OR.IAX(2).EO.14.OR.IAX(3).EO.14)GO TO 85
0309      GO TO 23
0310  85  DTMP=DAT(3)-TOLD
0311      DSAL=DAT(1)-SOLD
0312      DPRS=DAT(2)-POLD
0313  C      WRITE(6,977)DAT(3),DAT(1),DAT(2),DTMP,DSAL,DPRS
0314  C977  FORMAT(1X,6(F12.4,2X))
0315      IF(ABS(DPRS).LT.0.001)GO TO 20
0316      DAT(13)=DTMP/DPRS
0317      DAT(14)=DSAL/DPRS
0318  C      WRITE(6,978)DAT(13),DAT(14)
0319  C978  FORMAT(1X,2(F12.4,2X))
0320      TOLD=DAT(3)
0321      SOLD=DAT(1)
0322      IF(POLD.NE.0.0)ITMPS=0

```

```

0323      IF(POLD.EQ.0.0) ITMPS=1
0324      POLD=DAT(2)
0325      IF(ITMPS.EQ.1) GO TO 20
0326      C
0327      C      PLOT OUT THE DATA
0328      C
0329      23  CALL PLT(DAT(IAX(1)),DAT(IAX(2)),DAT(IAX(3)),IOPT)
0330      20  CONTINUE
0331      GO TO 15
0332      99  ICOPY=15427B
0333      WRITE(7,754) ICOPY
0334      754  FORMAT(A2)
0335      END
0336      SUBROUTINE PLT(X1,Y,X2,IOPT)
0337      COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0338      COMMON IBUF1(456),IBUF2(456)
0339      COMMON C1,C2,C3,C4,C5,C6,INSTR
0340      COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0341      COMMON X1AXL,X1IV,X1INCR,YAXL,YIV,YINCR,X2AXL,X2IV,X2INCR
0342      X1MAX=X1IV+(X1INCR*X1AXL)
0343      YEXT=YIV+(YINCR*YAXL)
0344      YMAX=AMAX1(YEXT,YIV)
0345      YMIN=AMIN1(YEXT,YIV)
0346      X2MAX=X2IV+(X2INCR*X2AXL)
0347      IF(X1.LT.X1IV.OR.X1.GT.X1MAX)RETURN
0348      IF(IOPT.EQ.2) GO TO 118
0349      IF(X2.LT.X2IV.OR.X2.GT.X2MAX)RETURN
0350      118  IF(Y.LT.YMIN.OR.Y.GT.YMAX)RETURN
0351      X1SCAL=X1AXL/(X1MAX-X1IV)
0352      X1PLT=X1SCAL*(X1-X1IV)
0353      X2SCAL=X2AXL/(X2MAX-X2IV)
0354      X2PLT=X2SCAL*(X2-X2IV)
0355      YSCAL=YAXL/(YMAX-YMIN)
0356      IF(YINCR.GE.0) YPLT=YSCAL*(YMAX-Y)
0357      IF(YINCR.LT.0) YPLT=YSCAL*(Y-YMIN)
0358      NUMB1=3
0359      NUMB2=1
0360      CALL SYMB(X1PLT,YPLT,.04,NUMB1,0.0,-1)
0361      IF(IOPT.EQ.2) GO TO 117
0362      CALL SYMB(X2PLT,YPLT,.04,NUMB2,0.0,-1)
0363      117  RETURN
0364      END
0365      SUBROUTINE INIT1
0366      COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0367      COMMON IBUF1(456),IBUF2(456)
0368      COMMON C1,C2,C3,C4,C5,C6,INSTR
0369      COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0370      COMMON X1AXL,X1IV,X1INCR,YAXL,YIV,YINCR,X2AXL,X2IV,X2INCR
0371      INSTR=1
0372      TCTP=0.0
0373      C1=0.1
0374      C2=0.0
0375      C3=0.0
0376      C4=.0005
0377      C5=.0010
0378      C6=0.0
0379      INTCP1=4
0380      LAG=0
0381      WRITE(1,18)

```

```

0382 18 FORMAT("USE DEFAULT VALUES, 1=YES 2=NO")
0383 READ(1,*)KANS
0384 IF(KANS.EQ.1)GO TO 99
0385 WRITE(1,20)
0386 20 FORMAT("DEFAULT VALUES GIVEN IN PARENTHS")
0387 WRITE(1,22)
0388 22 FORMAT("INSTRUMENT # (1)")
0389 READ(1,*)INSTR
0390 WRITE(1,24)
0391 24 FORMAT("TEMP PROBE TIME CONST,SECONDS (0.0)")
0392 READ(1,*)TCTP
0393 WRITE(1,26)
0394 26 FORMAT("CAL CONST PRESSURE C1,C2 (0.1,0.0)")
0395 READ(1,*)C1,C2
0396 WRITE(1,28)
0397 28 FORMAT("CAL CONST TEMP C3,C4 (0.0,0.0005)")
0398 READ(1,*)C3,C4
0399 WRITE(1,30)
0400 30 FORMAT("CAL CONST COND C5,C6 (0.001,0.0)")
0401 READ(1,*)C5,C6
0402 WRITE(1,40)
0403 40 FORMAT("LAG OF TEMP PROBE #OF SCANS (0)")
0404 READ(1,*)LAG
0405 C      WRITE OUT VALUES
0406 WRITE(1,64)INSTR,TCTP
0407 64 FORMAT(1X,"INSTR ",I2,"TCTP ",F8.4)
0408 WRITE(1,66)C1,C2,C3,C4,C5,C6
0409 66 FORMAT(1X,"CAL CONSTS ",6F8.4)
0410 WRITE(1,68)X1AXL,X1IV,X1INCR,YAXL,YIV,YINCR,X2AXL,X2IV,X2INCR
0411 68 FORMAT(1X,"AXES CONSTS ",9F6.2)
0412 WRITE(1,70)INTCPI,LAG
0413 70 FORMAT(1X,"PLT TM INT ",I4,"LAG IN SCANS ",I4)
0414 99 RETURN
0415 END
0416 END$

```

REFERENCES

1. N.L. Brown and G.K. Morrison, "W.H.O.I/Brown Conductivity, Temperature and Depth Microprofiler," Woods Hole Oceanographic Institution Technical Report 78-23 (unpublished manuscript), 1978.
2. Neil L. Brown, "A Precision CTD Microprofiler," in *Ocean '74: IEEE International Conference on Engineering in the Ocean Environment Record*, Vol. 2, p. 270, 1974.
3. N.P. Fofonoff, S.P. Hayes, and R.C. Millard Jr., "W.H.O.I./ Brown CTD Microprofiler: Methods of Calibration and Data Handling," Woods Hole Oceanographic Institution Technical Report 74-89 (unpublished manuscript), 1974.
4. A. Bradshaw, and K.E. Schleicher, "The Effect of Pressure on the Electrical Conductance of Sea Water," *Deep-Sea Res.* **12**, 151-162 (1965).
5. N. Brown and B. Allentoft, "Salinity, Conductivity and Temperature Relationships of Sea Water over the Range of 0 to 60 p.p.t.," Bissett-Berman Corp. manuscript report, Mar. 1, 1966.

Appendix
SCIENTIFIC SUBROUTINES—FORTRAN SOURCE LISTINGS

```

0001  FTN4.L
0002      SUBROUTINE LRSAL(G,P,T,S)
0003  C      CTD PACKAGE SUBROUTINE TO OBTAIN SALINITY FROM C,T,P
0004  C      MODIFIED TO FIT INTO MY PLOT PACKAGE. 18JULY77 L. ROSENBLUM
0005  COMMON IREC,ISTOP,ISCAN,NSCAN,IFLOR
0006  COMMON IBUF1(456),IBUF2(456)
0007  COMMON C1,C2,C3,C4,C5,C6,INSTR
0008  COMMON ISHIP(6),IST,IYR,IMO,IDAY,LOW
0009  COMMON PAXL,PIV,PINCR,TAXL,TIV,TINCR,SAXL,SIV,SINCR
0010 C DEG CELSIUS; PRESSURE IN DECIBARS
0011      IF(ABS(S-35.0)-35.0) 3,3,2
0012      2 S=35.0
0013 C SCALE CTD TEMPERATURE TO THE 1948 STANDARD
0014      3 CONTINUE
0015      XT=T+4.4E-6*T*(100.-T)
0016      XC=G
0017      XP=P
0018      XTO = XT-15.
0019      RC=XC/42.909
0020      AL = -7.6E-5*XT+6.95E-3
0021      AJ = ((-1.657E-4*XT+0.276E-3)*XT-1.535E-1)*XT+1.0
0022      RT = (((-.67249142E-8*XT-.19426015E-6)*XT+.99886585E-4)*XT
0023      1+.20131661E-1)*XT+.67652453
0024      GA= ((-7.9E-6*XT+8.3089E-4)*XT-4.5302E-2)*XT+1.5192
0025      H=(-2.492E-9*XP+2.577E-5)*XP+4.0E-4
0026      F=((3.3E-13*XP-3.3913E-8)*XP+1.042E-3)*XP
0027      SP =GA*F +H*AJ
0028      RCT=RC/RT
0029      RP0 = 1. +1.0E-2*SP
0030      RP1=1.0E-2*SP*AL
0031      5 SO=S
0032      RP=RP0+RP1*(35.0-S)
0033      RS=RCT/RP
0034      R15 = ((-.21E-5*RS*RS-.63E-5)*XT0+(37.3E-5*RS-72.E-5)*RS
0035      1+96.7E-5)*XT0*RS*(RS-1.0) + RS
0036      S=((((-1.32311*R15+5.98624)*R15-10.67869)*R15+12.80832)
0037      1*R15+28.2972)*R15-0.08996
0038      IF(ABS(S-S0)-5E-2) 10,10,5
0039      10 CONTINUE
0040      END
0041      END$
```

```

0001 FTN4,L
0002      SUBROUTINE POTMP(PRS, TEMP, SAL, PTMP)
0003 C POTENTIAL TEMPERATURE ACCORDING TO BRYDEN
0004 C CONVERT TO 1948 TEMPERATURE SCALE
0005      TMP = TEMP + 4.4E-6*TEMP*(100.-TEMP)
0006      S0=SAL-35.
0007      A3=.50484E-14*TMP-0.16056E-12
0008      A2=(.21987E-11*TMP-0.31628E-9)*TMP+.89309E-8
0009      1-.41057E-10*S0
0010      A11=-.29778E-7*TMP+.17439E-5
0011      A10=((-4.0274E-9*TMP-.54065E-7)*TMP
0012      1+.83198E-5)*TMP+.36504E-4
0013      A1=A11*S0+A10
0014      PTMP=TEMP-((A3*PRS+A2)*PRS+A1)*PRS
0015      RETURN
0016      END

```

```

0001 FTN4,L
0002      SUBROUTINE SVEL(P,T,S,V)
0003 C J WEBSTER          SVEL SUBROUTINE
0004 C COMPUTES VELOCITY OF SOUND IN METERS/SEC FROM PRESSURE IN DECIBARS
0005 C TEMPERATURES IN DEGREES C AND SALINITY IN PARTS PER THOUSAND
0006 C CHANGE PRESSURE TO TOTAL PRESSURE IN KG/CM**2
0007 PKC = P * .1019716 + 1.03323
0008 SM = S - 35.0
0009 C00 = (((7.9851E-6*T - 2.6045E-4)*T - 4.4532E-2)*T + 4.5721 )*T
0010 X   + 1449.14
0011 C01 = (7.7711E-7*T - 1.1244E-2)*T + 1.39799
0012 C10 = ((4.5283E-8*T + 7.4812E-6)*T - 1.8607E-4)*T + 1.60272E-1
0013 C11 = (1.5790E-9*T + 3.1580E-8)*T + 7.7016E-5
0014 C20 = (1.8563E-9*T - 2.5294E-7)*T + 1.0268E-5
0015 B3 = -1.9646E-10*T + 3.5216E-9
0016 B0 = (1.69202E-3*SM + C01)*SM + C00
0017 B1 = C11*SM + C10
0018 B2 = -1.2943E-7*SM + C20
0019 V = ((((-3.3603E-12 * PKC + B3)*PKC+B2)*PKC+B1)*PKC+B0
0020 RETURN
0021 END
0022 END$
```

L. J. ROSENBLUM

```

0001  FTN4,L
0002  C
0003      SUBROUTINE ATG(PRS,TEMP,SAL,GAMMA)
0004  C COMPUTES ADIABATIC TEMPERATURE GRADIENT IN DEG CELSIUS/1000 DB
0005  C CONVERT TO 1948 TEMPERATURE SCALE
0006      TMP = TEMP + 4.4E-6*TEMP*(100.-TEMP)
0007      S = SAL - 35.
0008      A01 = -.42393E-4*TMP+.18932E-2
0009      A00 = ((.66228E-6*TMP-.6836E-4)*TMP+.85258E-2)*TMP+.35803E-1
0010      A10 = ((-.54481E-10*TMP+.8733E-8)*TMP-.67795E-6)*TMP+.18741E-4
0011      A11 = .27759E-8*TMP-.11351E-6
0012      A2 = (-.21687E-12*TMP+.18676E-10)*TMP-.46206E-9
0013      A1 = A11*S + A10
0014      A0 = A01*S + A00
0015      GAMMA = (A2*PRS+A1)*PRS + A0
0016      RETURN
0017      END
0018      END$
```

```

0001  FTN4,L
0002  C
0003      SUBROUTINE SIGMT(TMP,XS,XS0,XST)
0004  DOUBLE PRECISION XT,XS0,E1,B1,B2,E2,XST
0005  C CONVERT TO 1948 TEMPERATURE SCALE
0006      XT = TMP + 4.4D-6*TMP*(100.-TMP)
0007      XS0 = ((6.76786136D-6*XS-4.82496140D-4)*XS+8.14876577D-1)*XS
0008      X  -9.34458632D-2
0009      E1 = (((-1.43803061D-7*XT-1.98248399D-3)*XT-5.45939111D-1)*XT
0010      X  +4.53168426D0)*XT
0011      B1 = (((-1.0843D-6*XT+9.8185D-5)*XT-4.7867D-3)*XT + 1.0
0012      B2 = (((1.667D-8*XT-8.164D-7)*XT + 1.803D-5)*XT
0013      E2=(B2 *XS0+B1) * XS0
0014      XST = E1/(XT+6.726D1 )+E2
0015      RETURN
0016      END
0017      END$
```

NRL REPORT 8444

```

0001 FTN4,L
0002 C
0003      SUBROUTINE SPVOL(PRS,TEMP,XSO,XST,XALPH)
0004      DOUBLE PRECISION A0,E2,XALPH,XSO,XST,TMP,B10,B11,B12,B20,B21,
0005      1           B22,C3,C1,C2,E1
0006 C
0007 C SPECIFIC VOLUME IN CM**3/GM
0008 C
0009 C CONVERT TO 1948 TEMPERATURE SCALE
0010 TMP = TEMP + 4.4E-6*TEMP*(100.-TEMP)
0011 B10=((4.0E-12*TMP-6.63E-10)*TMP+3.673E-8)*TMP-2.2072E-7
0012 B11=(4.0E-12*TMP-3.28E-10)*TMP+1.725E-8
0013 B12=1.0E-12*TMP-4.5E-11
0014 B20=(2.14E-14*TMP-1.24064E-12)*TMP-6.68E-14
0015 B21=(-2.0E-16*TMP+1.206E-14)*TMP-4.248E-13
0016 B22=-6.0E-17*TMP+1.0E-15
0017 C3=1.5E-17*TMP
0018 C1=(B12*XSO+B11)*XSO+B10
0019 C2=(B22*XSO+B21)*XSO+B20
0020 E1=((C3*PRS+C2)*PRS+C1)*PRS
0021 E2=1.0-4.886E-6*PRS/(1.0+1.83E-5*PRS)+E1
0022 A0=1.0/(1.0+.001*XST)
0023 XALPH=A0*E2
0024 RETURN
0025 END
0026 END$
```

```

0001 FTN4,L
0002 C
0003      SUBROUTINE BRVA1(DPT,RHO,RH00,BV)
0004      DOUBLE PRECISION RDIF,E,RHO,RHOL,DDIF,GSPVO
0005 C      SUBROUTINE COMPUTES BRUNT-VAISALA FREQ. IN CYCLES/MR
0006 C      USING APPROX. GIVEN IN PHILLIPS P. 17
0007 C      PROGRAMMER L. ROSENBLUM 10/78
0008 C
0009 C      CONST IS GRAVITY IN M/SEC/SEC
0010 GSPVO=9.81/RH00
0011 DDIF=DPT-DPTL
0012 RDIF=RHO-RHOL
0013 E=GSPVO*RDIF/DDIF
0014 C      CONST CONVERTS FROM RADIAN/S/SEC TO CYCLES/MR
0015 BVV=572.9578*DSQRT(DABS(E))
0016 BV=SIGN(BVV,E)
0017 RHOL=RHO
0018 DPTL=DPT
0019 RETURN
0020 END
0021 END$
```

