

AD-A128 146

A COMPUTER PROGRAM FOR PROCESSING
CONDUCTIVITY-TEMPERATURE-DEPTH (CTD) DATA(U) NAVAL
OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL STATION MS
Z R HALLOCK JAN 83 NORDA-TN-196

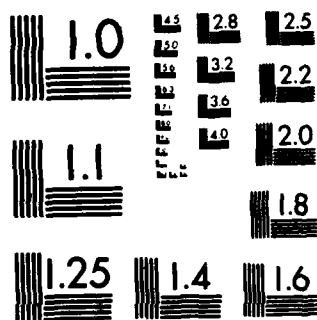
1/1

UNCLASSIFIED

F/G 9/2

NL

END
DATE
FILMED
6-83
DTIC

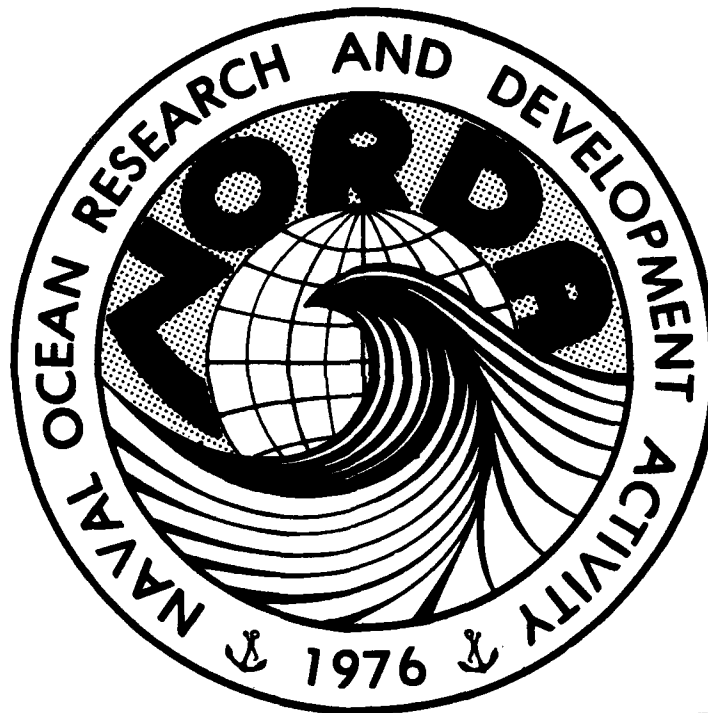


MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Naval Ocean Research and
Development Activity
NSTL Station, Mississippi 39529

A Computer Program for Processing Conductivity-Temperature-Depth (CTD) Data

ADA 128146



DTIC
ELECTE
MAY 17 1983
S A D

DTIC FILE COPY

This document has been approved
for public release and sale; its
distribution is unlimited.

Z.R. Hallock

Ocean Science and Technology Laboratory
Oceanography Division

January 1983

83 05 17 016

ABSTRACT

A software package for processing conductivity-temperature-depth (CTD) data is described. The package includes features for editing, correcting, filtering and pressure sorting to produce working data files for graphic and analysis work.

[illegible]

ACKNOWLEDGMENTS

This work is a description of software developed while the author was employed by the Naval Oceanographic Office (Code 7210). The vast majority of the programming, testing, debugging and use of the described software was performed by Mr. William Teague. Many of the algorithms used were adapted from programs written by Mr. Frank Mulher, who also wrote the preparatory programs. The careful work of Mr. Carlos Mayoral and his colleagues in Code 6320 in determining sensor time-responses and the expert help of Dr. Tom Davis of Code 022 in designing deconvolution filters for temperature, allowed the inclusion of data quality improvement features.

This work was supported by the Ocean Measurements Program through the Ocean Programs Management Office at NORDA, funded by Program Element 63704N, Dr. Rudy Hollman, program manager.

A COMPUTER PROGRAM FOR PROCESSING CONDUCTIVITY-TEMPERATURE-DEPTH (CTD) DATA

I. INTRODUCTION

CTD (conductivity-temperature-depth) data acquired with a Neil Brown Instrument Systems CTD system (or equivalent data) require certain processing steps before they are generally usable for oceanographic analysis. This note describes a processing software package (CTDPRG), which was developed at the Naval Oceanographic Office and has been used in a production mode since 1978. Some minor modifications have been made since that time, but the basic structure has not changed.

Input data for CTDPRG must be in the form of a Fast Easy Binary (FEB) file (Hallock, 1980) and must consist of the physical variables: pressure (dbar), temperature ($^{\circ}\text{C}$) and conductivity (mmho/cm). Hence, a pre-processing step is required which will be specific to the original data recording format. Output is quite flexible ranging from the input data itself through corrected, pressure-sorted and subsampled data with calculated quantities such as salinity and density.

Section II is a general, functional description of algorithms used. Section III provides details for running the programs with several examples of run setups. Section IV contains complete program listings.

II. GENERAL DESCRIPTION

CTDPRG is designed to handle CTD data acquired as vertical profiles, horizontal tows, or a combination of these. The program was developed around vertically profiled data; however, and much of its complexity goes toward converting the time-base CTD data to functions of pressure, when this conversion is required.

The following list is a summary of the processing steps performed by CTDPRG.

- Read input controls from cards or keyboard.
- Read data from mass storage file.
- Correct isolated spikes (wild data points).
- Apply time-domain filters to pressure, temperature and conductivity (this includes T-C matching).
- Correct conductivity values based on water sample comparisons.
- Apply temperature-pressure dependent cell correction.
- Separate up-going and down-going profiles.
- Pressure-sort data by averaging into 0.1 dbar intervals.

- Low-pass filter, in pressure, and subsample to multiple of 0.1 dbar.
- Compute selected quantities (e.g., salinity, sigma-t) from P, T, C.
- Write results to mass-storage files.

All steps (except the first three and last) are optional and can be selected in several combinations.

Input Data Sets

The basic unit of processing is the input data set. This is, typically, a single CTD cast (down and up, if both were recorded). It might be, however, a series of down and up profiles acquired during one instrument deployment. Usually an input data set corresponds to a single input file, but this is not required. There may be more than one data set per file or a data set may span several files. All this is handled by the input controls. The input data set is in a FEB-file containing time, pressure (dbar), temperature (°C) and conductivity (mmho/cm).^{*} Additional variables could be carried with minor modifications. The last FEB-segment in the data set must have IDOCR(1)=1; preceding segments (for multi-segment data sets) must have IDOCR(1)=0. Input files are thus prepared by a pre-processing program that is tailored for the particular instrument and deployment mode.

Time-Domain Filters

One of the most significant problems with CTD data has been the effects of the response mismatch between temperature and conductivity sensors. A number of attempts have been made to correct this through the design and application of filters. The most widely used method is that described by Fofonoff et al. (1974), which assumes an exponential response for the temperature sensor. More recently, as a result of work done at the Naval Oceanographic Office (NOO) involving direct measurement of instrument response, specific filters have been designed for temperature and conductivity, based on experimentally determined sensor response functions (see Mayoral, 1982). In CTDPGR there are three options: no T-C filtering, filtering, the three-point Fofonoff method (the user specifies only the time-constant), or explicitly supplied filters of the user's choice. These are applied to the raw data, in the time domain, prior to other processing.

A smoothing filter is automatically applied to pressure data to remove high-frequency sensor noise.

Conductivity Calibration

With the newer conductivity cells on the MK III CTD systems, coupled with proper cleaning procedures and frequent laboratory calibration adjustments, conductivity data seldom

^{*}See Hallock (1980) for details on FEB-file structure.

has to be adjusted from the recorded values. However, if salinity checks derived from water samples indicate an offset from proper calibration, a multiplicative correction can be supplied by the user. There is also a manufacturer-supplied correction for pressure and temperature distortion which is applied following the filtering step.

Upcasts and Downcasts

If no editing based on pressure is to be done (i.e., no identification of upcasts, downcasts or constant level data), following conductivity correction, selected additional variables are computed, and an output data set is written that contains the same number of scans (cycles) as the input.

A second option is to retain only upcasts and/or downcasts, based on user defined pressure limits, which are then written as separate output data sets, following computation of additional variables.

A third option, which necessitates the second, is to convert each upcast/downcast to a pressure series as opposed to a time series. This case assumes that the instrument descent/ascent rate is 2 m/sec or less. (If this value is exceeded in the mean, some software modifications would be necessary.) At a nominal descent/ascent rate of 1 m/sec, about three scans of data are acquired every 0.1 dbar. If pressure sorting is selected, time, temperature and conductivity values in 0.1 dbar intervals are averaged, creating a pressure series at that resolution. Retrograde (in pressure) data are rejected. The pressure series is then subsampled at some user-defined, integral multiple of 0.1 dbar following the application of a low-pass filter to prevent aliasing. The filter is automatically generated, and its cutoff wavenumber is the new Nyquist wavenumber of the subsampled series. At this point additional variables are computed and data are written to output files. The third option allows the user to select a degree of smoothing/subsampling commensurate with his requirements.

Computed Variables

Including the four input variables, up to twelve are available as output. The user may select any subset of these. The eight computed variables are as follows:

- Salinity (parts per thousand [ppt]) or S: computed with pressure, temperature and conductivity ratio, according to Bennett (1976), Dauphine and Klein (1977) or Lewis (1980).
- Sigma or σ : the in situ density of a water parcel. Computed with pressure, temperature, and salinity according to Fofonoff (1962). ($\text{Sigma} = (\text{density (g/cc)} - 1.) * 1000.$)
- Sigma-t or σ_t : the density of a water parcel with in situ temperature and salinity but at atmospheric pressure.
- Brunt-Väisälä Frequency (cycles per hour) or N: computed with pressure, temperature, and salinity according to Fofonoff

(1962). Essentially a local potential density derivative. The convention is adopted: $N = \sqrt{N^2} \text{sgn}(N^2)$ to allow for instabilities.

- Sound Speed (m/sec): computed with pressure, temperature, and salinity according to Wilson (1960).

- True Depth: for most purposes, pressure in decibars can be interpreted as depth in meters; however, this assumes an average water density of 1.02 g/cc. If depth accurate to more than a few percent are required, this variable can be selected. It is an approximation to:

$$z = \int_0^p \frac{\alpha}{g} dp$$

where α = specific volume,
 g = acceleration of gravity,
 p = pressure,
 z = depth below surface,

- Potential Temperature ($^{\circ}\text{C}$) or θ : The temperature that a parcel of water would have if brought adiabatically to the surface. Computed with pressure, temperature and salinity according to Fofonoff (1962).

- Potential Density or σ_0 : σ_t using θ rather than in situ temperature; the density of a water parcel brought adiabatically to the surface. (σ_0 is calculated using the σ_t routine with θ from the above routine.)

A typical standard processing product might be: retain only downcasts at 1 meter resolution, and store pressure, temperature, salinity and σ_t . This provides a compact version of data for hydrographic analysis work. The best "standard" file is likely to be different for each user group; thus, it is recommended that the input files (edited, raw data) be considered the primary, archived data base.

III. DETAILS OF OPERATION

This section consists of two parts: a tour of CTDPRG and its subroutines; a tutorial on setting up input and running the program.

Functional Program Description

Figure 1 is a block diagram of all the routines in the package. CTDPRG is the driver, whose functions are to read all input controls, to set up filters to be used on input data, and to initiate the processing of each input data set by calling ACCESS.

GENWHF generates a three-point deconvolution filter for temperature based on a user-specified, sensor time-constant. This is described in detail by Fofonoff et al. (1974).

GENFLT is a general filter generating routine (see Brooks, 1976). It is called in CTDPRG to produce a low-pass filter for pressure data to remove sampling noise.

ACCESS controls the majority of the processing. Input data cycles (scans) are accessed, one at a time, via calls to GETREC. If pressure sorting is not selected, output segments are prepared using a user-specified combination of variables from the input as well as quantities computed via DERIVE.

When upcast/downcast separation is specified, each upcast/downcast becomes a separate data set and is written to unit 3/2. Otherwise, all output goes to unit 2 as one data set.

With pressure sorting, which necessitates upcast/downcast separation, input data cycles in 0.1 dB intervals are averaged, producing a pressure series at that resolution. Retrograde data, which result from ship motion or winch reversals, are rejected. For a nominal descent rate of 1 meter per second there should be about 3 cycles per interval. If an interval should turn up empty, it is filled with data from the previous interval. If descent rates systematically exceed 1.5-2 m/s this interval should be increased accordingly.

The pressure sorted data can be subsampled at integral multiples, say n , of 0.1 dB. When this is done, ACCESS calls GENFLT to produce a low-pass filter which is applied to the 0.1 dB data, centered at every n th cycle. The cutoff of the filter is at the new Nyquist of the subsampled series and prevents aliasing. For example: $n = 10$ implies subsampling every 1 dB; cutoff (half-amplitude point) is at 0.5 cycles/dB; the filter is applied only where samples are to be extracted (e.g., every 10th point), thus avoiding unnecessary computation.

Following pressure sorting and subsampling, DERIVE is called for selected computed variables. Segments are then assembled and written to the output file(s).

Subroutine GETREC is the input program. It reads segments from the input file (unit 4), applies conductivity cell corrections, pressure smoothing filters, conductivity-temperature matching filters and pressure limits. Data are then passed to ACCESS, one cycle at a time. Upper and lower pressure limits, specified by the user, are employed by GETREC to distinguish upcast and downcast data.

NBCCC applies a manufacturer supplied cell correction to conductivity which compensates for pressure and temperature distortions of cell geometry. It also applies a user-specified multiplicative conductivity adjustment (usually = 1.) which corrects for salinity offsets determined by field comparisons.

Time-domain filters are described in Section II.

When an end-of-data set (IDOCR(1) = 1) is detected, control returns to the main program and new input cards are sought. If an end-of-file is encountered, however, GETREC calls IUSER which attempts to dynamically attach another user-specified file name to unit 4. If no file name is supplied, the program terminates. (On some machines dynamic control commands may not be permitted, thus requiring one execution per input file.)

Subroutine DERIVE accepts a cycle of pressure, temperature, conductivity and time from access and, after calling the appropriate variable computation routines, restructures the cycle according to user specifications. The variable computation routines are described in Section II.

Subroutine ZREAD and ZWRIT (described by Hallock, 1980) are input and output routines for FEB-files. A call to ZREAD/ZWRIT transfers one segment of data between main memory and a mass storage file. Typically, a data set (e.g., CTD profile) consists of several FEB segments; each segment is made up of several header blocks followed by a data block. Segments are written sequentially to a mass-storage (disk) file by nonformatted, FORTRAN I/O statements.

Input Control Cards

Essential information for setting up and running CTDPGR is provided through internal documentation (comment cards), which are presented below.

Input Card 1: Printout identification--up to 48 alphanumeric characters. This can be anything.

Input Card 2: N, LN, NWMAX, KCHOP, NINFIL (Free Format).

N: Subsampling interval for pressure sorted data. Integer multiples of 0.1 dB, e.g., N=20 results in data every 2 dB. N=0 results in no pressure sorting.

LN: Sharpness factor for subsampling filter. This is an integer equal to 1 or greater. Larger values require more computer time. ($m = LN \cdot N$ = half-width of filter. See, GENER1 in Brooks, 1976, for a more detailed explanation.)

NWMAX: Maximum no. of cycles in output segments. Typically, 1000.

KCHOP=1: Process all data. No pressure sorting or subsampling. Pressure limits are not applied. Output goes to unit 2.

KCHOP=2: Pressure sort and subsample by N (unless N = 0). Output downcasts to unit 2, upcasts to unit 3.

KCHOP=3: Same as 2 but output only downcasts.

KCHOP=4: Same as 2 but output only upcasts.

NINFIL: No. of input file names to be supplied.
If =0, one input file is assumed on unit 4. Otherwise, supplied names are dynamically attached to unit 4 by IUSET (presently configured for UNIVAC).

Input Card 3: (IND(I), I=1, 12) (Free format) output variable selection. A string of 12 integers (1 for yes, 0 for no) indicating variables to be output from the following list: Time, pressure, temperature, salinity, sigma-t, sigma, Brunt-Väisälä frequency, sound speed, depth, potential temperature, potential density.

Input Filename Cards: up to 24 alphanumeric characters, left-justified. If NINFIL=0, no cards are expected.

The following group of cards, the data set card and pressure limit cards, is repeated for each input data set to be processed. .

Input Data Set Card: NMSER, NPLIMS, CFTR, TIMCON, ISAL
(Free Format)

NMSER: Sequence no. of input data set to be processed. (Usually this is equal to 1, but if there were, say, n data sets to be skipped beginning in the first input file, NMSER would be set to n+1.)

NPLIMS: No. of pressure limit cards for this data set.

CFTR: Conductivity cell correction factor (default=1).

TIMCON>0: Temperature sensor time constant divided by sampling time interval. WHOI (Fofonoff et al. temperature deconvolution.)

=0: No. temperature or conductivity filtering.

=-1: Use same filters that were used for previous data set. (Not permissible for first data set.)

=-5: Read special filters from cards (unit 5) following pressure limit cards.

All Else: Read special filters from file on unit 8.

ISAL = 0: Salinity computed according to Lewis and Perkin (1980)

= 1: Salinity computed according to Bennet-Dauphinee.

Pressure/Cycle Limit Cards: PTOP, PBOT, DELTAP, I1, I2
(Free format)

PTOP, PBOT are upper lower pressure limits to be applied to input data cycles between cycles I1 and I2. Each sequence of cycles spanning this pressure interval is made into an output data set. Pressure limits are irrelevant for KCHOP=1.

DELTAP is a slop factor (usually about 2 dB) that is employed to prevent small pressure fluctuations from tripping up the pressure limit logic. It ensures, for example, that a downcast has ended before the subsequent upcast is recognized. For single-cast data, DELTAP can be set to zero.

I1, I2--There is a global cycle index, IE, which is 1 at the beginning of an input data set. It is incremented for each cycle encountered within the data set. (An input data set typically spans more than one FEB segment and occasionally more than one file.) The first profile start, defined by PTOP, is not sought until IE reaches I1. After each profile is completed, if IE has reached I2, the next pressure limit card is invoked. When the cards are exhausted or the input data set ends, the next data set card is read. This feature allows leading and trailing data within a data set to be skipped, which can save computer time or reject defective data.

Usually, only one pressure limit card is required for each input data set, particularly for a single profile. However, for a series of repeated profiles which are included in a single data set (i.e., a "yo-yo" cast), it may be necessary to change pressure limits several times to maximize data recovery and to reject sections of bad data.

In the simplest case, for a single cast data set, I1 can be set to 1 and I2 to a large number which exceeds the number of cycles in the data set, and everything will be processed. (Default values of 1 and 1,000,000 result for I1 = I2 = 0.)

Special Filter Input: NWTT, NWTC, LAGT, (WT(I), I=1, NWTT), WC(I), I=NWTC) (Free Format)

These are read from cards if TIMCON = -5. If TIMECON = -2, -3, -4, -6, -7,... they are read from a file on unit 8. If TIMCON > -2, this input is not expected. (See TIMCON above.)

NWTT, NWTC: No. of temperature, conductivity, filter weights.

LAGT: No. of cycles to lag temperature relative to pressure and conductivity. (The lag might be built into filter, in which case LAGT = 0.)

WT, WC: Temperature, conductivity filter weights.

Examples of Input Controls

Example 1: Three input files, each containing a downcast and an upcast between about 8 m and 2000 m depth. Only downcasts are to be processed. WHOI (Fofonof, et al., 1974) temperature deconvolution is to be used--time constant is 100 ms, sampling interval is 32 ms. Output pressure resolution is to be 1 dbar. Output variables are to be: P, T, S, σ_t , N. Salinity to be computed by practical salinity scale '78.

Card 1: THIS IS EXAMPLE 1.
Card 2: 10,1,1000,3,3
Card 3: 0,1,1,0,1,1,0,1,0,0,0,0
Card 4: FILE1
Card 5: FILE2
Card 6: FILE3
Card 7: 1,1,1,3.125,0
Card 8: 10,2000,1,1,0

Cards 7 and 8 are repeated for Files 2 and 3.

Example 2: One input file containing a continuous record of two downcasts and two upcasts (i.e., one instrument deployment). The instrument was in the water for one hour before the first downcast began and again following the final upcast. Thus, the first usable data scan is about no. 112,000. The four profiles took about one hour so the last usable scan was around 250,000. Temperature deconvolution filters are to be read from unit 8; a 1.01 conductivity is to be made; salinity is to be computed by Bennett-Dauphinee algorithms. Data are to be subsampled at 50 cm. Both upcasts and downcasts are to be saved, between 100 and 400 dbar.

Card 1: THIS IS EXAMPLE 2.
Card 2: 5,1,1000,2,0 (input assumed on unit 4)
Card 3: 0,1,1,0,1,1,0,1,0,0,0,0
Card 4: 1,1,1.01,-3,1
Card 5: 100,400,2,112000,250000

Example 3: One input file containing 500 downcasts and upcasts between about 90 and 210 dbar ("yo-yo" cast). Data are to be processed as in example 1, saving downcasts at 1 dbar resolution. Prior examination of the data reveals that downcasts 201-208 (scans 1.2×10^6 - 1.248×10^6)

extend only to 190 dbar: three sets of pressure limits are thus required, to recover the maximum amount of data.

Card 1: THIS IS EXAMPLE 3.

Card 2: 10,1,1000,3,0

Card 3: 0,1,1,0,1,1,0,1,0,0,0,0

Card 4: 1,3,1,3.125,0

Card 5: 95,205,2,1,1200000

Card 6: 95,185,2,1200000,1248000

Card 7: 95,205,2,1248000,4000000

Example 4: Input file consists of three downcast-upcast pairs. For diagnostic purposes, it is necessary to process downcast data at original resolution (i.e., no pressure sorting). Output variables are to be: time, pressure, temperature, conductivity and salinity. Pressure limits are 10 to 500 dbar. Special filters for temperature and conductivity are to be supplied on cards.

Card 1: THIS IS EXAMPLE 4.

Card 2: 0,0,1000,3,0

Card 3: 1,1,1,1,1,0,0,0,0,0,0,0

Card 4: 1,1,1,-5,0

Card 5: 10,500,0,1,0

Card 6: 5,5,0,.1,.2,.4,.2,.1,.0625,.125,.625

Card 7: .125,.0625

These are representative examples of how CTDFRG may be used. Many other combinations are possible. Some experimentation may be necessary for unusual requirements.

REFERENCES

Bennett, A.S. (1976). Conversion of In Situ Measurements of Conductivity to Salinity. Deep Sea Res. 23(2), p. 157-165.

Brooks, D.A. (1976). Fast and Easy Time Series Analysis at NCSU. Center for Marine and Coastal Studies, North Carolina State University Report.

Dauphinee, T.M. and H.P. Klein (1977). The Effect of Temperature on the Electrical Conductivity of Seawater. Deep Sea Res. 24, p. 891-902.

Fofonoff, N.P. (1962). Physical Properties of S.W. In: The Sea, v. 1, p. 3-30, M.N. Hill (ed.), Wiley.

Fofonoff, N.P., S.P. Hayes and R.D. Millard (1974). WHOI/Brown CTD Microprofiler: Methods of Calibrating and Data Handling. WHOI TR 74-89.

Hallock, Z.R. (1980). The Fast and Easy Binary File. Naval Oceanographic Office Technical Note (TN)7210-12-80, 29 pp.

Lewis, E.L. (1980). The Practical Salinity Scale 1978 and Its Antecedents. IEEE, J. Oceanic Eng., v. OE-5, p. 3-8.

Mayoral, C.R. (1982). Determination of CTD Sensor Responses for Use in Deconvolution Filtering. Proc., International STD Conference and Workshop, 8-11 February, 1982, Marine Technology Society, San Diego Section.

Wilson W.D. (1980). Speed of Sound in Water as a Function of T.P.S. J. Acoust. Soc. Am., v. 32, p. 641-644.

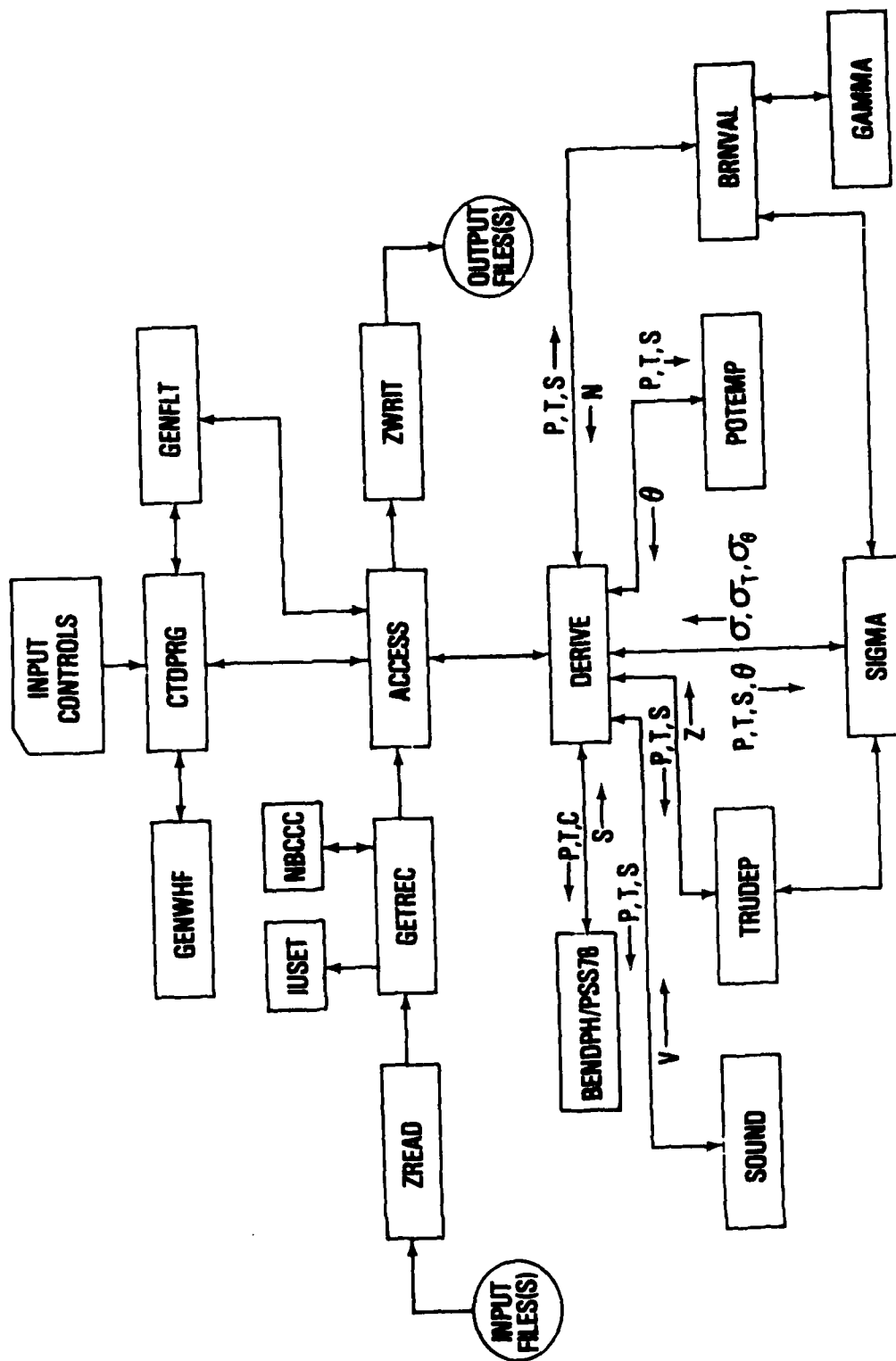


Figure 1. Functional Block Diagram of CTD Processing Programs

APPENDIX

PROGRAM PREPARED BY CTDPRS

```

1      C      CCMPILEID18=3)
2
3      C      BASIC CTD PROCESSING PROGRAM
4
5      C      INPUT FILES (UNIT 4) ARE PROPERLY PREPARED FOR FILES OF
6      C      RAW CTD DATA. VARIABLES ARE TIME, PRESSURE, TEMPERATURE,
7      C      CONDUCTIVITY, HUMIDITY, WIND. WIND SHOULD BE CRUISE AND STATION
8      C      NUMBER, RESPECTIVELY. ISCR(1)=1 MARKS THE END OF AN
9      C      INPUT DATA SET (CAST). THERE CAN BE MORE THAN ONE INPUT
10     C      FILE PER DATA SET (SEE BELOW). OUTPUT GOES TO UNITS
11     C      2 AND 3. UNSEPARATED DATA (NCHOP=1) AND DOWNCASTS GO TO UNIT
12     C      2 WHILE UPCASTS GO TO UNIT 3. FOR FURTHER DETAILS SEE
13     C      EXTERNAL DOCUMENTATION. INPUT CONTROL CARDS ARE DES-
14     C      CRIBED BELOW.
15
16     C      COMMON/WHAT/INNO,INO(20),NCHOP,NPLIMS,PLIS,100),IPLIN12,100),
17     C      * NUMAX,IPMSET(20),IEOLU,NMSEN
18     C      COMMON/SALINE/KSAL
19     C      COMMON/IOFILE/INIFIL,IFEB(120)
20     C      COMMON/CALIB/MP,MT,MC,LAST,MP(60),MT(60),MC(60),POIF.
21     C      * IDIF,CDIF,CFIR
22     C      DIMENSION LABEL(20),IRUN(10)
23     C      DATA (LABEL(I),I=1,12) /'TIME','PRESS','TEMP','COND','SAL','SIGT',
24     C      * 'SIGSTP','BHFCDP','SNOVEL','TDEPTH','TEMPOT','SIGPOT'/
25     C      DATA MP,JA,JB,JC,F/10,2,1,1,1/
26     C      DATA ISW/1/
27     C      DATA POIF,IDIF,CDIF/.5,.2,.2/
28
29     C      ***** INPUT CARD 1: PRINTOUT IDENTIFICATION. UP TO 40 CHARS.
30     C      CAN BE ANYTHING - EVEN BLANK CARD.
31
32     C      READ(5,1020,END=999) IRUN
33     C      1020 FORMAT(8A6)
34     C      WRITE(6,1021)IRUN
35     C      1021 FORMAT(1H1,' CTD PROCESSING RUN: ',8A6//)
36
37     C
38     C
39     C      ***** INPUT CARD 2: N,LN,NUMAX,NCHOP,NINIFIL
40
41     C      N:      SUBSAMPLING INTERVAL FOR PRESSURE SORTING.
42     C      INTEGER MULTIPLES OF .1 DB. I.E. N=20 RESULTS
43     C      IN DATA EVERY 2 DB. N=0 RESULTS IN NO SORTING.
44
45     C      LN:     SHARPNESS FACTOR FOR DECIMATION FILTER.
46     C      THIS IS AN INTEGER EQUAL TO 1 OR GREATER.
47     C      IT DETERMINES THE NO. OF WEIGHTS AND THUS
48     C      THE SHARPNESS OF THE DECIMATION FILTER.
49
50     C      NUMAX:  MAX NO. OF CYCLES IN OUTPUT SEGMENTS.
51     C      (AS A RULE, THIS SHOULD BE ABOUT 1000.)
52
53     C      NCHOP=1: PROCESS ALL DATA; NO PRESSURE SORTING
54     C      OR SUBSAMPLING. PRESSURE LIMITS ARE
55     C      NOT APPLIED.
56
57     C      NCHOP=2: OUTPUT UPCASTS AND DOWNCASTS;
58     C      PRESSURE SORT AND SUBSAMPLING BY N.
59
60     C      NCHOP=3: SAME AS 2 BUT OUTPUT ONLY DOWNCASTS.
61
62     C      NCHOP=4: SAME AS 2 BUT PROCESS ONLY UPCASTS.
63
64     C      NINIFIL: NO. OF INPUT FILENAMES TO BE SUPPLIED.
65     C      IF NINIFIL=0, ONE INPUT FILE IS ASSUMED ON
66     C      UNIT 4. OTHERWISE, SUPPLIED NAMES ARE ASSIGNED
67     C      DYNAMICALLY TO UNIT 4 BY SR IUSE. THIS MAY
68     C      NOT WORK ON ALL MACHINES.
69
70     C      READ(5,1000,END=999)N,LN,NUMAX,NCHOP,NINIFIL
71     C      1000 FORMAT(I)
72     C      WRITE(6,1000)N,LN,NUMAX,NCHOP
73     C      NCHOP=NCHOP
74     C      1000 FORMAT(1X,'N=',I5,1X,'LN=',I5,1X,'NUMAX=',
75     C      I5,1X,'NCHOP=',I5)
76
77     C      ***** INPUT CARD 3: OUTPUT VARIABLE SELECTION. A STRING OF
78     C      12 INTEGERS (1 FOR YES, 0 FOR NO)
79     C      INDICATING VARIABLES TO BE OUTPUT, FROM
80     C      THE FOLLOWING LIST.
81

```

```

82 C TIME,PRESSURE,TEMPERATURE,CONDUCTIVITY,
83 C SALINITY,CIGMA-T, SIGMA, S-V FREQUENCY,
84 C SOUND SPEED, TRUE DEPTH, POTENTIAL
85 C TEMPERATURE,POTENTIAL DENSITY.
86 C
87 READ(5,100,END=999)(IND(I),I=1,12)
88 C
89 C=0
90 DO 11 I=1,20
91 IF(IND(I).EQ.0)GO TO 11
92 J=J+1
93 DMSY(I,J)=LABEL(I)
94 CONTINUE
95 11 NIND=J
96 C
97 WRITE(6,6005)
98 6005 FORMAT(//1X,"VARIABLES REQUESTED")
99 WRITE(6,6006)(DMSY(I),I=1,J)
100 6006 FORMAT(11X,10(1A6,1X))
101 C
102 C ***** INPUT FILENAME CARDS (FORMAT 946)
103 C
104 C ONE CARD FOR EACH INPUT FILENAME OF UP TO
105 C 24 CHARACTERS (INCLUDING *). IF NINFIL=3,
106 C NO CARDS ARE EXPECTED.
107 C
108 CO 993 I=1,4
109 993 IFEB(I):
110 IF(AINFIL.EQ.0) GO TO 990
111 CO 991 I=1,NINFIL
112 I1=(I-1)*4+1
113 I2=I1+3
114 991 READ(5,1001,END=999)(IFEB(I),I=I1,I2)
115 1001 FORMAT(4A6)
116 990 CONTINUE
117 C
118 C
119 C
120 C THE FOLLOWING GROUP OF CARDS, THE DATA SET
121 C CARD AND PRESSURE LIMIT CARDS, IS REPEATED
122 C FOR EACH INPUT DATA SET TO BE PROCESSED.
123 C
124 C ***** INPUT DATA SET CARD: NMSER,NPLINS,CFTR,TINCON,ISAL
125 C
126 C NMSER SEQUENCE NO. OF INPUT DATA SET TO BE PROCESSED.
127 C
128 C NPLINS NO. OF PRESSURE LIMIT CARDS FOR THIS DATA SET.
129 C
130 C CFTR CONDUCTIVITY CELL FACTOR (DEFAULT=1.)
131 C
132 C TINCON GT 0: TEMPERATURE TIME CONSTANT/SAMPLING TIME INTERVAL.
133 C WHOI TEMPERATURE DECONVOLUTION.
134 C
135 C = 0: NO FILTERS ON TEMP OR COND.
136 C
137 C = -1: USE FILTERS FOR PREVIOUS DATA SET. (NOT
138 C PERMISSIBLE ON FIRST DATA SET.)
139 C
140 C = -5: READ SPECIAL FILTERS FROM CARDS FOLLOWING
141 C PRESSURE LIMIT CARDS.
142 C
143 C ELSE: READ SPECIAL FILTER CARD IMAGES FROM UNIT 8.
144 C
145 C ISAL = 0: SALINITY COMPUTED IAW PSS78.
146 C
147 C = 1: SALINITY COMPUTED IAW BENNETT-DAUPHINEE.
148 15 READ(5,1000,END=999)(NMSER,NPLINS,CFTR,TINCON,ISAL
149 NSAL=2
150 IF(15AL.NE.0)NSAL=1
151 C
152 NT=1
153 PC=1
154 NUT=1
155 NUTC=1
156 NT(1)=1.
157 NUT(1)=1.
158 IFILT=2
159 IF(TINCON.LT.0.) IFILT=0
160 IF(TINCON.LT.-1.9)IFILT=3
161 IF(ABS(TINCON).LT..001)IFILT=1
162 C
163 INUT=0
164 IF(ABS(TINCON+.5).LT..1)INUT=5

```

```

165 C
166 IF(CFTR.LT..0)CFTR=1.
167 C
168 WRITE(6,6050)NMSEI,NPLINS,CFTR
169 6050 FORMAT('' SERIES SEQUENCE NO. ',I4,
170 ' ',NO. OF PRESSURE LIMITS =',I4//
171 ' ',CONDUCTIVITY CELL FACTOR =',612.6//)
172 C
173 C ***** PRESSURE/CYCLE LIMIT CARDS *****
174 C
175 C PTOP,PROT,DELTAP,I1,I2
176 C
177 C PTOP/PROT ARE UPPER/LOWER PRESSURE LIMITS TO BE
178 C APPLIED TO INPUT DATA CYCLES BETWEEN
179 C CYCLES I1 AND I2. EACH SEQUENCE OF
180 C CYCLES SPANNING THIS PRESSURE
181 C INTERVAL IS MADE INTO AN OUTPUT DATA
182 C SET OR PROFILE. (PRESSURE LIMITS
183 C ARE IRRELEVANT FOR NCHOP=1.)
184 C
185 C DELTAP IS A SLOP FACTOR (USUALLY ABOUT 2 DB)
186 C WHICH IS EMPLOYED TO PREVENT SMALL PRESSURE FLUC-
187 C TUATIONS FROM TRIPPING UP THE PRESSURE LIMIT
188 C LOGIC. IT ENSURES THAT A DOWNCAST HAS ENDED
189 C BEFORE THE SUBSEQUENT UPCAST IS RECOGNIZED, ETC.
190 C
191 C I1,I2 THERE IS A GLOBAL CYCLE INDEX, SAY IE, WHICH
192 C IS SET TO 1 AT THE BEGINNING OF AN INPUT DATA
193 C SET. IT IS INCREMENTED FOR EACH CYCLE ENCOUNTERED
194 C WITHIN THE DATA SET. (AN INPUT DATA SET TYPICALLY
195 C SPANS MORE THAN ONE SEGMENT AND OCCASIONALLY
196 C MORE THAN ONE FILE.) THE FIRST PROFILE
197 C START, DEFINED BY PTOP, IS NOT SOUGHT
198 C UNTIL IE IS GE I1. AFTER EACH PROFILE
199 C IS COMPLETED, IF IE IS GE I2, THE NEXT PRESSURE
200 C LIMIT CARD IS INVOKED. WHEN THE CARDS ARE EX-
201 C HAUSTED OR INPUT DATA SET ENDS, THE NEXT
202 C DATA SET CARD IS READ. IF THE ENTIRE INPUT
203 C DATA SET IS TO BE SCANNED, SET I1=I2=0.
204 C
205 C
206 I1=1
207 GO 30 L=1,NPLINS
208 READ(5,1000)(PL(I,I1),I=1,3),(IPLIN(I,I1),I=1,2)
209 IF(IPLIN(1,I1).LE.0)IPLIN(1,I1)=1
210 IF(IPLIN(2,I1).LE.0)IPLIN(2,I1)=1000000
211 C
212 WRITE(6,6010)NMSEI,(PL(I,I1),I=1,3),(IPLIN(I,I1),I=1,2)
213 6010 FORMAT('X','SERIES',I4,I2,' ',I3,'X','PTOP=',F5.0,I4,' ',I3,'X','PROT=',
214 ' ',F5.0,' ',I3,'X','DELTAP=',
215 ' ',F5.0,' ',I3,'X','CYCLE1=',I7,' ',I3,'X','CYCLE2=',I7)
216 C
217 I1=I1+1
218 JC CONTINUE
219 C
220 IF(IIFILT.EQ.0.AND.ISH.EQ.1) GO TO 95
221 GO TO (2),22,23,100,IIFILT
222 C
223 C WMO1 FILTER GENERATION SECTION
224 C
225 22 IF(INMT.LT.3)INMT=3
226 CALL G2NMWFINMT,TINCON,MT)
227 C
228 WRITE(6,6075)INMT,TINCON
229 6075 FORMAT('' WOODS HOLE DECONVOLUTION CHOSEN''//
230 ' ',NO. OF HEIGHTS =',I4,
231 ' ',TIME CONSTANT =',611.6//)
232 C
233 NT=(INMT+1)/2
234 PC=NT
235 NUTC=INMT
236 DO 26 I=1,NUTC
237 26 UC(I)=1./NUTC
238 C
239 GO TO 21
240 C
241 C SPECIAL FILTER INPUT SECTION
242 C
243 23 WRITE(6,2310)IIFILT
244 2310 FORMAT('' SUPPLIED TEMPERATURE, CONDUCTIVITY FILTERS''
245 ' ' TO BE READ ON UNIT',I4//)
246 C
247 C ***** SPECIAL FILTERS: NO. TEMP HTS, NO. COND HTS,

```

```

248 C TEMP LAB, TEMP WTS, COND WTS.
249 C
250 C THESE VALUES ARE READ FROM CARDS IF TINCON=-5.
251 C OTHERWISE (I.E. TINCON=-2,-3,-4,-5,...)
252 C THEY ARE READ FROM UNIT 8. IF TINCON BE -1,
253 C THIS READ STATEMENT IS SKIPPED.
254 C
255 READ(1,1000,END=999) NUTT,NUTC,LABT,(W(I),I=1,NUTT),
256 * (WC(I),I=1,NUTC)
257 C
258 WRITE(6,760)NUTT,NUTC,LABT
259 760 FORMAT(' NO. OF TEMPERATURE HEIGHTS: ',I3//
260 * ' NO. OF CONDUCTIVITY HEIGHTS: ',I3//
261 * ' TEMPERATURE LAB = ',I2,' DATA CYCLES'//)
262 C
263 C
264 NT=INUTT*11/2
265 NC=INUTC*11/2
266 NTCNN=NT*2-1
267 NCCNN=NC*2-1
268 IF(NUTT.EQ.NTCNN.AND.NUTC.EQ.NCCNN) GO TO 21
269 C
270 WRITE(6,2800)
271 2800 FORMAT(' NO. OF HEIGHTS MUST BE ODD. TRY AGAIN.'//)
272 STOP
273 C
274 C CREATE PRESSURE SMOOTHING FILTER.
275 C
276 21 WRITE(6,2100)
277 2100 FORMAT(' PRESSURE SMOOTHING FILTER'//)
278 C
279 CALL GENFLTIMP,JA,JB,JC,F,UP)
280 C
281 NUTP=2*NP-1
282 SUNP=0.
283 SUNC=0.
284 SLAT=0.
285 DO 25 I=1,NUTP
286 25 SUNP=SUNP+WPI(I)
287 DO 27 I=1,NUTT
288 27 SUNT=SUNT+W(I)
289 DO 28 I=1,NUTC
290 28 SUNC=SUNC+WC(I)
291 C
292 WRITE(6,2700)NUTP,SUNP,NUTT,SUNT,NUTC,SUNC
293 2700 FORMAT(' SUM OF ',I3,' PRESSURE HEIGHTS = ',611.5//
294 * ' SUM OF ',I3,' TEMPERATURE HEIGHTS = ',611.5//
295 * ' SUM OF ',I3,' CONDUCTIVITY HEIGHTS = ',611.5//)
296 C
297 C ***** PROCES ONE DATA SET. *****
298 C
299 100 CALL ACCESSIN,LN)
300 ISM=2
301 C
302 C
303 GO TO 15
304 C
305 999 WRITE(6,6040)
306 6040 FORMAT(' *****END OF JOB*****')
307 STOP
308 C
309 95 WRITE(6,9500)
310 9500 FORMAT(' PREVIOUS FILTER SPECIFIED ON INITIAL PASS.'
311 * ' INCONSISTENT.'//)
312 STOP
313 C
314 END

```

APRT,S PF=0074PR80.ACCESS

PF00074P06011). ACCESS

```

1      COMPILERID(48)=3)
2      SUBROUTINE ACCESSIN, LN)
3
4      C
5      COMMON /MNT/MIND,IND(20),KCHOP,NPLINS,PL(3,100),IPLIN(2,100),
6      &NNMAX,IPUS ET(20),IEOLU
7      COMMON /RDOCF/RDOCF(40)
8      COMMON /RDOCI/RDOCI(40)
9      COMMON /RDOCA/RDOCA(100)
10     COMMON / UNDR / LN,NU,NBU,NMBU,NMFU,NFU,NIN,NAN,IPU(20)
11     COMMON / RNDR / LR,NR,NBR,NMBR,NMFR,NMFR
12     COMMON /RDOCF/RDOCF(40) /RDOCI/RDOCI(40) /RDOCA/RDOCA(100)
13     COMMON / MDATA / VM(100,10)
14
15     C
16     COMMON / DIAGS / NISR,NSSU,NNR,NNNU,NNIP,NMF,NNI,NNA,IRST,IUST
17     DATA NNR,NNNU,NNIP,NMF,NNI,NNA/1000,1000,13,40,40,74/
18     DATA NFU,NIN,NAN/40,40,100/
19
20     C
21     DIMENSION M(200),ICT(200),RECBUT(4,200),REMBUT(20)
22     DIMENSION Q(20),SARREC(4)
23     DATA JA,JB,JC/2,1,3/
24     LOGICAL IEOLU
25     KCHOP1=KCHOP
26     IDOC(15)=0
27     IF (N.EQ.1) LN=1
28     IF (KCHOP1.E.1) GO TO 300
29     h=0
30     TLO=2
31
32     30C CONTINUE
33     IDOC(1)=0
34     NATZ=0
35     ICOC(14)=0
36     IF (NMBU.NE.NMBUP) IDOC(15)=0
37     IDOC(15)=IDOC(15)+1
38     JPX=2
39     JN2X=0
40     IF (IND(1).EQ.0) GO TO 325
41     CO J24 IX=1,0
42     IF (IND(IX).NE.0) JN2X=JN2X+1
43     324 CONTINUE
44     NU=0
45     LU=MIND
46     IF (NNMAX.GT.NNNU) NNMAX=NNNU
47     DO 301 J=1,LU
48     IPU(J)=IPUSET(J)
49
50     301 CONTINUE
51     INIT=1
52     ISM=2
53     IF (N.LT.1) ISM=1
54     IEOLU=.FALSE.
55     IC1 GO TO (1,2),ISM
56
57     C STRAIGHT-THROUGH PROCESSING SECTION
58     1 CALL GETREC(190,0,IUD)
59     NUD=IUD+2-3
60
61     C DERIVE OPTION
62     20 IF (IEOLU) RETURN
63     CALL DERIVE(IQ,INIT)
64     INIT=0
65     40 NU=NN+1
66     IJD=(NU-1)*LM
67     GO 71 J=1,LM
68     IJ=IJD+J
69     71 VM(IJ)=Q(IJ)
70     IF (NU.EQ.NNMAX+1) GO TO 95
71     GO TO 1
72
73     C ECCINATION SECTION
74     C GET FIRST RECORD
75     C IUD=1: UP PROFILE IUD=2: DOWN PROFILE
76     2 CALL GETREC(190,0,IUD)
77     NUD=IUD+2-3
78
79     C GET FIRST INTERVAL
80     P=Q(IJPX)
81     IP=P*10.**5
82     IP1=IP
83     IP2=IP1+LN*NU*NU
84     IP2=IP2+LN*NU*NU
85     IPP=IP
86     NI=1
87     N=LN*NU
88     NNT=2*N-1
89     IF (N.EQ.1) GO TO 72
90     IF (N.EQ.NPREV.AND.LN.EQ.LNPREV) GO TO 72

```

```

83      FD=1./N
84      WRITE(6,9311)
85      9311 FORMAT(/' DECIMATION FILTER'/)
86      CALL GENFLTIN,JA,JB,JC,FD,M)
87      WRITE(6,7801)
88      7801 FORMAT(/)
89      NPREV=N
90      LNPREV=LN
91      GO TO 72
92      C GET NEXT RECORD
93      75 CALL GETREC(690,Q,IUD)
94      IUD=IUD+2-3
95      73 P=Q(JPX)
96      3P=P*10.**.5
97      C CHECK FOR P-REVERSAL
98      72 IF((1P-1PP)*KUD.LT.0) GO TO 75
99      1PP=1P
100     C CHECK FOR P OUT OF INTERVAL
101     IF((1P-1P1)*KUD.LE.0) GO TO 75
102     IF((1P-1P2)*KUD.GE.0) GO TO 60
103     C LOAD RECORD INTO FILTER ARRAY
104     JP=(1P-1P1)*KUD
105     ICT(JP)=ICT(JP)+1
106     DO 76 J=3,4
107     76 RECOUNT(J,JP)=RECOUNT(J,JP)+Q(J)
108     RECOUNT(1,JP)=RECOUNT(1,JP)+Q(1)
109     GO TO 75
110     C COMPLETE AVERAGING
111     DO 65 K=1, NMT
112     NCT=ICT(K)
113     IF(NCT.NE.0) GO TO 161
114     NNTHOL=NNTHOL+1
115     NNT2=NNT2+1
116     DO 163 J=3,4
117     163 RECOUNT(J,N)=SAVREC(J)
118     RECOUNT(1,N)=SAVREC(1)
119     GO TO 65
120     161 CONTINUE
121     IF(NNT2.GT.1)WRITE(6,3901)NNT2,PD
122     3901 FORMAT(1X,'GAP OF',I5,1X,'AT P=',G12.6)
123     NNT2=0
124     DO 63 J=3,4
125     RECOUNT(J,N)=RECOUNT(J,N)/NCT
126     63 SAVREC(J)=RECOUNT(J,N)
127     RECOUNT(1,N)=RECOUNT(1,N)/NCT
128     SAVREC(1)=RECOUNT(1,N)
129     65 CONTINUE
130     C IF NO FILTER, SKIP
131     IF(N.EQ.1) GO TO 61
132     C CONVOLVE FILTER
133     USUM=0.
134     JCT=0
135     DO 64 J=3,4
136     64 RECOUNT(J)=0.
137     RECOUNT(1)=0.
138     DO 66 K=1,NMT
139     IF(1CT(K).EQ.0) GO TO 66
140     USUM=USUM+MI(K)
141     JCT=JCT+1
142     DO 67 J=3,4
143     67 RECOUNT(J)=RECOUNT(J)+RECOUNT(J,K)*MI(K)
144     RECOUNT(1)=RECOUNT(1)+RECOUNT(1,K)*MI(K)
145     66 CONTINUE
146     IF(JCT.EQ.0.OR.JCT.EQ.NMT) GO TO 69
147     DO 68 J=3,4
148     68 RECOUNT(J)=RECOUNT(J)/USUM
149     RECOUNT(1)=RECOUNT(1)/USUM
150     GO TO 69
151     61 DO 62 J=3,4
152     62 RECOUNT(J)=RECOUNT(J,1)
153     RECOUNT(1)=RECOUNT(1,1)
154     69 CONTINUE
155     FC=FLOAT(IP0)/10.
156     RECOUNT(2)=PD
157     C RECORD NOW COMPLETE
158     C DERIVE OPTION
159     CALL DERIVE(RECOUNT,INIT)
160     INIT=0
161     C LOAD INTO OUTPUT ARRAY
162     NI=NM+1
163     IJC=(NM-1)*LM
164     DO 51 J=1,LM
165     51 IJ=IJC+J

```



```

166      51 UNITJ:=RECOUNT(J)
167      IF (NUM.EQ.NUMMAX+1) GO TO 95
168 C   RESET ALL SUMS AND COUNTERS
169      48 IP0=IP0+NUMUD
170      IP1=IP0-LNUMUD
171      IP2=IP0+LNUMUD
172      K1=NUMT-N*1
173      NUMTN=NUMT-N
174      DO 45 K=1,NUMTN
175      NM=N+K
176      ICT(K)=ICT(NM)
177      RECOUNT(1,K)=RECOUNT(1,NM)
178      GO 45 J=3,4
179      45 RECOUNT(J,K)=RECOUNT(J,NM)
180      GO 46 K=1,N
181      NM=NUMT-N*1
182      ICT(NM)=0
183      RECOUNT(1,NM)=0.
184      DO 46 J=3,4
185      46 RECOUNT(J,NM)=0.
186      47 IF (ICLUD) RETURN
187      IF (IDOCM1.EQ.1) GO TO 300
188      GO TO 73
189      55 CALL GETRECC(90,0,IUD)
190      NUM=NUMD+2-3
191      46 IDOCM(1)=0
192 C   MORE STUFF HERE **
193      47 CONTINUE
194 C   SHIFT N*2
195      IF (JN2X.EQ.0) GO TO 122
196      NM=NM-1
197      JBVF1=JN2X
198      DO 121 J=1,NM
199      V(LJBVF1)=V(LJBVF1+LW)
200      JBVF1=JBVF1+LW
201      121 CONTINUE
202      122 CONTINUE
203      IF (NUM.GT.NUMMAX) NM=NUMMAX
204      NMNU=NMNR
205      NMFR=NMFR
206      GO TO (43,92),IUD
207      42 IDOCM(21)=0
208      IDOCM(4)=IDOCM(4)*1
209      DO 981 IQ=1,40
210      981 FIDOCM(IQ)=FIDOCR(IQ)
211      IDOCM(2)=IDOCR(2)
212      IDOCM(3)=IDOCR(3)
213      IDOCM(10)=IDOCR(10)
214      IDOCM(11)=IDOCR(11)
215      IDOCM(12)=IDOCR(12)
216      GO 982 IQ=1,NAM
217      982 ADDCM(IQ)=ADOCR(IQ)
218      CALL ZDRIT(2,IF,0)
219      NMNUP=NMNU
220      GO TO 145
221      53 IDOCM(21)=1
222      IDOCM(4)=IDOCM(4)*1
223      GO 983 IQ=1,40
224      983 FIDOCM(IQ)=FIDOCR(IQ)
225      IDOCM(2)=IDOCR(2)
226      IDOCM(3)=IDOCR(3)
227      IDOCM(10)=IDOCR(10)
228      IDOCM(11)=IDOCR(11)
229      IDOCM(12)=IDOCR(12)
230      GO 984 IQ=1,NAM
231      984 ADDCM(IQ)=ADOCR(IQ)
232      CALL ZDRIT(3,IF,0)
233      NMNUP=NMNU
234      145 IF (INTHOL.GT.0) WRITE(6,910)INTHOL,NMU
235      910 FORMAT(1X,15,1X,'HOLES IN DATA REPLACED WITH PRECEDING',
236      1X,'RECORD IN SEGMENT',1X,15)
237      INTHOL=0
238 C
239      94 NM=1
240      IDOCM(1)=0
241 C   SHIFT RECORD NM+1 TO 1
242      NMNXL=NMNXL+LW
243      GO 120 J=1,LN
244      V(LJ)=V(LNMNXL+J)
245      120 CONTINUE
246      IF (IF.GT.1) GO TO 98
247      GO TO (70,98),15
248      90 CONTINUE

```

```

249      IDOCN1=1
250      IDOCN1=1
251      I=1
252      GO 405 IN99=1,NMT
253      ICI(IK99)=0
254      DO 405 IN4=1,4
255      405  RECOUT(IK4,IK99)=0
256      IF (NM.EQ.0)GO TO 47
257      GO TO 97
258      10  WRIT(16,9800) IF
259      9800 FORMAT('/// PROBLEM IN ZURIT')
260      STOP
261      EN1

```

APRT,3 PF0U074P:60.6ETREC

```

PF0U074P:60.6ETREC
1      C CRPILERIDIA6=3)
2      SUBROUTINE GETREC(4,0,1UD)
3
4      C
5      DIMENSION Q(20)
6      COMMON/UMAT/NIND,IND(20),NCMP,NPLIMS,PL(3,100),IPLIM(2,100),
7      *  NMAX,IPHSET(20),IEOLU,NMER
8      COMMON/CALB/MP,MT,NC,LAGT,UP(60),UT(60),UC(60),PDIF,
9      *  TDIF,CDIF,CFTR
10     COMMON/UMDR/LR,NR,NBR,NMNR,NMFR,NFR,NIR,NAR,IPR(20)
11     COMMON /RDATA/ VR(4,1100)
12     COMMON/RDOCF/RDOCR(40)
13     COMMON/RDOCT/IDOCR(40)
14     COMMON/RDOCA/ADOCR(100)
15     COMMON/DIAGS/MSGR,MSGV,NNNR,NNNV,NNIP,NNF,NNI,NNR,IRST,IUST
16     DATA NNNR,NNIP,NNF,NNI,NNR/100,20,40,40,100/
17     DATA ISM/1,JM,JP,JT,JC/1,2,3,4/
18     DATA IIU/4/
19     DATA ISET,INEXT/1,1/
20     LOGICAL IEOLU
21     GO TO (6,7),ISET
22     6  CCONTINUE
23     IDOCR(1)=0
24     IIT=MT+LAGT
25     II=PC
26     IF (IIT.GT.NC)II=IIT
27     IF (IMP.GT.II)II=IMP
28     NMT=2*MT-1
29     PNC=2*PC-1
30     PHP=2*MP-1
31     ISET=2
32     IRST=2*II-1
33     7  CONTINUE
34     C INITIALIZE
35     C
36     GO TO (1,2),ISM
37     1  NSM=4
38     CALL IUSEY
39     701  ANSER1=NI) 1+1
40     IF (ANSER1.EQ.NMSER)GO TO 701
41     CALL Z6EAD(IIU,IF,0)
42     IF (IF.EQ.1)GO TO 702
43     IF (IDOCR(1).EQ.1)NID1=NID1+1
44     GO TO 700
45     702  CALL IUSEY
46     GO TO 700
47     701  CONTINUE
48     IDOCR(1)=0
49     ISM=2
50     N=11-1
51     ICPF=0
52     IDL1=-1
53     GO TO 5
54     4  NSM=1
55     703  JE=IE-1

```

22

```

139      Q(4)=Q(4)+VR(JC,IC)*NC(I)
140      13      CONTINUE
141      DO 14 I=1,NMP
142      IP=IEE+1-MP
143      Q(2)=Q(2)+VR(JP,IP)*MP(I)
144      14      CONTINUE
145      C
146      C APPLY MANUFACTURERS CELL CORRECTION AND CFTR
147      C
148      CALL NBCCC(10,2),Q(3),Q(4),CFTR)
149      C
150      IEE=IEE+1
151      C
152      C KCHOP=1:ALL DATA,      KCHOP=2:UP AND DOWN CAST
153      C KCHOP=3 DOWN CAST ONLY      KCHOP=4:UP CAST ONLY
154      GO TO (119,100,100,100),KCHOP
155      C
156      15      IF(I.NEXT.GT.NPLIMS)GO TO 200
157      IE1=IPLIM1,I.NEXT)
158      IENEXT=IPLIM(2,I.NEXT)
159      IF(IE.LT.IE1)GO TO 2
160      IF(IE.LT.IENEXT)GO TO 25
161      IENEXT=IENEXT+1
162      GO TO 19
163      C END OF LU OR FILE
164      C
165      20C IENEXT=1
166      20 ISW=1
167      ISET=1
168      IEOLU=.TRUE.
169      MIDIP1=MID1*1
170      IF(1DOCR(1).EQ.1)MIDIP1=MID1
171      IF(1NMSE.NE.MIDIP1)WRITE(6,9432)MIDIP1,NMSE
172      9432 FORMAT(///1X,"CONGRATULATIONS TURKEY!!! YOU HAVE TRIED TO PROCESS
173      & SERIES",I3,"INSTEAD OF SERIES",I3,1X/" CHECK PRESSURE LIMITS"///)
174      RETURN
175      25 RETURN
176      C
177      100 F=Q(1JP)
178      C
179      GO TO (110,110,110,122),NSW
180      C
181      C SET UP P-LIMITS
182      C
183      112 PIT=PL(1,1)
184      IF(PIT.LE.0.)PIT=AIMT(P+1,1)
185      P2T=PIT*PL(3,1)
186      P10=PL(2,1)
187      IF(P10.LE.0.)P10=99000.
188      P20=P10-PL(3,1)
189      IE1=IPLIM1,1)
190      IENEXT=IPLIM(2,1)
191      INEXT=2
192      IF(IE.LT.IE1)GO TO 2
193      C
194      C SEARCH FOR FIRST PROFILE START
195      114 GO TO (140,140,140,141),KCHOP
196      140 IF(P.LT.P1)GO TO 117
197      GO TO (141,141,142,141),KCHOP
198      141 IF(P.GT.P10)GO TO 118
199      142 GO TO 2
200      C
201      C SHIP FOLLOWING RUBBISH IF PLIM EXHAUSTED
202      C
203      C
204      C
205      110 IF(IE.LT.IE1)GO TO 2
206      C
207      GO TO (111,117,118,119),NSW
208      C
209      C
210      C CHECK FOR P INSIDE LIMITS
211      C
212      111 IF(P.GT.P10) GO TO 113
213      IF(P.LE.P1T) GO TO 112
214      GO TO (116,115),IUD
215      C
216      C END OF PROFILE
217      C
218      112 NSW=2
219      GO TO (151,151,2,151),KCHOP
220      113 NSW=3
221      GO TO (151,151,151,2),KCHOP

```

```

222      151  CONTINUE
223          ICOLU=.FALSE.
224          IF (IE.GE.IPLIM12,NPLIMS) GO TO 20
225          IF (IDOCR11.EQ.1) GO TO 20
226          IF (IE.LE.IENEXT) GO TO 124
227          IF (INEXT.GT.NPLIMS) GO TO 124
228      C INCREMENT PLIMS
229          F1=PL(1,INEXT)
230          P1B=PL(2,INEXT)
231          P2T=P1T*PL(3,INEXT)
232          F2B=P1B*PL(3,INEXT)
233          IE1=IPLIM1,INEXT)
234          IENEXT=IPLIM12,INEXT)
235          INEXT=INEXT+1
236      124  CONTINUE
237          RETURN
238      C
239      C A1 TCP
240      C
241          117  NSW=2
242              IF (P.LY.P21) GO TO 2
243              JUD=2
244              NSW=1
245          115  GO TO (25,25,25,2),NCHOP
246      C
247      C A1 BOTTOM
248      C
249          118  NSW=3
250              IF (P.GT.P2B) GO TO 2
251              JUD=1
252              NSW=1
253          116  GO TO (25,25,2,25),NCHOP
254      C
255          END

```

APRT,5 PF=U074PR6D.IUSET

```

PF=U074PR6D11.IUSET
1      COMPILER(DIAG=3)
2      SUBROUTINE IUSET
3      COMMON /IOFILE,NINFIL,IFEB(120)
4      DIMENSION LABL(6)
5      DATA JFILE/D/
6      IFILE=IFILE+1
7      I=(IFILE-1)*4+1
8      I1=I+3
9      IF (IFILE.GT.NINFIL) GO TO 900
10     ENCODE(36,12,LABL)(IFEB(J),J=1,I13)
11     12  FORMAT('DUSE 4.,'A6,' . ')
12     CALL EXTRAN(6,'FREE,A 4. . ')
13     CALL EXTRAN(6,LABL)
14     WRITE(6,6000)(IFEB(J),J=1,I13)
15     6000  FORMAT(1X,'FILE ASSIGNED',1X,A6)
16     RETURN
17     900  IF (IFILE.EQ.1) GO TO 901
18     I=I-4
19     I13=I13-4
20     WRITE(6,9000)(IFEB(K),K=1,I13)
21     9000  FORMAT(1X,' PROGRAM TERMINATED BEFORE REACHING LIMIT,'
22     * ' AT END OF INPUT FILE 'A6)
23     STOP
24     901  WRITE(6,9001)
25     9001  FORMAT(1X,' NO INPUT FILE NAMES SUPPLIED.//
26     * ' INPUT FILE ASSUMED ON UNIT 4'//)
27     RETURN
28     END

```

APRT,5 PF=U074PR6D.6E4NHF

```

PF0UD74PRGDI1).GENHMF
1      SUBROUTINE GENHMF(MNT,TINCON,M)
2      C
3      Coo  MNT = WOODS HOLE FILTER SIZE ,
4      Coo  TINCON = TIME CONSTANT/SAMPLING TIME INTERVAL OF INSTRUMENT .
5      Coo  M = WOODS HOLE FILTER ARRAY NAME
6      C
7      DIMENSION M(1)
8      FMT=FLOAT(MNT)
9      IF(MNT.EQ.0) RETURN
10     DO 10 I=1,MNT
11     F=FLOAT(I)
12     J(I)=(1./FMT)*((TINCON)*(12.*F-6.*(FMT+1.)))/
13     I  (FMT*(FMT+2.-1.))
14     10 CONTINUE
15     RETURN
16     END

```

APRT,5 PF0UD74PRGDI1).GENFLT

```

PF0UD74PRGDI1).GENFLT
1      C*****
2      C
3      C GENFLT IS ESSENTIALLY GENR1 FROM THE FESTSA PACKAGE.
4      C (SEE: FAST AND EASY TIME SERIES ANALYSIS AT NCSU -
5      C CITED BY D. A. BROOKS - AN NCSU INTERNAL REPORT)
6      C
7      C  - HALF-WIDTH OF FILTER
8      C  - 1,-1 FOR ONE-SIDED FILTER FADING TO THE RIGHT,LEFT
9      C  - 2 FOR A SYMMETRICAL FILTER OF (2M-1) TERMS.
10     C  - 0 FOR LAG WINDOW APPLICATION, LARGEST TERM=1.
11     C  - 1 FOR LOW PASS FILTER.
12     C  - 2 FOR HIGH PASS FILTER.
13     C  - 1 BARTLETT TAPER
14     C  - 2 PARZEN TAPER
15     C  - 3 COSINE (TUKEY) TAPER
16     C  - 4 LANZOS TAPER
17     C  - 5 LANZOS SQUARED TAPER
18     C
19     C FILTER WEIGHTS ARE RETURNED IN SER
20     C
21     C*****
22     C
23     COMPILER(OTAG=1)
24     SUBROUTINE GENFLT(M,JA,JB,JC,F,SER)
25     WRITE(6,66666)M,JA,JB,JC,F
26     6666 FORMAT(1 GENFLT: M=,16, JA=,13, JB=,13,
27     6 JC=,13, F=,612.6)
28     DIMENSION SER(1)
29     F1=3.14159265
30     FM=M
31     DO 100 I=1,M
32     F1=1
33     L=(F1-1.)/FM
34     GO TO(40,50,60,61,61),JC
35     41 IF(JC-.00000001)40,40,63
36     43 IF(JC-4)64,64,65
37     44 SER(I)= SIN(PI*U) / (PI*U)
38     40 TO 70
39     45 SER(I)=(SIN(PI*U)/(PI*U))*2
40     40 TO 70
41     40 SER(I)=1.-U
42     40 TO 70
43     40 SER(I)=.5*(1.+COS(PI*U))
44     40 TO 70
45     50 IF(ABS(U)-.5)51,52,52
46     51 SER(I)=1.-6.*U**2+.6.*ABS(U**3)
47     40 TO 70
48     52 SER(I)=2.*(1.-ABS(U))**3
49     70 IF(F-.00000001)100,100,71
50     71 IF(ABS(PI*(F1-1.)*F)-.00001)100,100,73
51     73 SER(I)=SER(I)*SIN(PI*(F1-1.)*F)/(PI*(F1-1.)*F)

```

```

52      100 CONTINUE
53      102 IF (J0) 100,130,103
54      103 SUM=.5*SER(I)
55      DO 104 I=2,M
56      104 SUM=SUM+SER(I)
57      GO 120 I=1,M
58      110 SER(I)=SER(I)/(2.*SUM)
59      IF (J0-1) 120,120,120
60      120 SER(I)=-SER(I)
61      IF (I-1) 120,121,120
62      121 SER(I)=SER(I)*1.
63      120 CONTINUE
64      130 M=M+2
65      GO TO (150,150,140,100),J0
66      140 CONTINUE
67      RETURN
68      150 M2=M/2
69      DO 155 I=1,M2
70      155 X=SER(I)
71      Y=X+1
72      SER(I)=SER(I)
73      SER(I+1)=X
74      RETURN
75      160 MM=2*M-1
76      CO WRITE(6,6900)(I,SER(I),I=1,MM)
77      6900 FORMAT(15,612.6)
78      DO 105 I=M+1,-1
79      105 SER(I)=SER(I)
80      MM=M-1
81      DO 101 I=1,MM
82      101 II=MM-I+1
83      SER(I)=SER(II)
84      RETURN
85      END
86

```

APRT,5 PF0UC74PR60.DERIVE

```

PF0UD74PR6D(1).DERIVE
1      C(Compiler(DIAG=3)
2      SUMROUTINE DERIVE(I,INIT)
3      C
4      DIMENSION Q(20),MM(20)
5      COMMON /MMAT/MIND,IND(20)
6      COMMON /SALINE/MSAL
7      DATA CF,ISKIP /A-.1/
8      LOGICAL IOBV,IOPT,IOPD,IOSV,IODT,IOSIG
9      C
10     F=Q(1)
11     T=Q(1)
12     C=Q(1)
13     GO TO (21,22),ISKIP
14     C
15     21 M=0
16     ISKIP=2
17     DO 1 I=1,20
18     IF (IND(I).EQ.0) GO TO 1
19     ILAST=I
20     N=N+1
21     NAD(N)=I
22     IOBV=(I.EQ.8.OR.10BV)
23     IOSV=(I.EQ.9.OR.10SV)
24     IOI=(I.EQ.10.OR.10DT)
25     IOPT=(I.EQ.11.OR.1.EQ.12.OR.10PT)
26     IOPD=(I.EQ.12.OR.10PD)
27     IOSIG=(I.EQ.6.OR.1.EQ.7.OR.10SIG)
28     1 CONTINUE
29     C
30     MSU=0
31     IF (ILAST.LT.5) MSU=1
32     IF (ILAST.EQ.5) MSU=2
33     IF (10SIG) MSU=3

```

```

34      IF(IOSIG.AND.(ILAST.GE.7)) NSU=S
35      C
36      I2 GO TO (50,51,51,51,51),NSU
37      C
38      I1 GO TO (511,512),NSAL
39      S12 CALL BENDPH(P,T,C,S)
40      GO TO 513
41      S12 CALL PSS70(P,T,C,S)
42      S13 Q15=S
43      C
44      GO TO (50,50,52,53,52),NSU
45      C
46      I2 CALL SIGMA(P,T,S,SIGT,SIG)
47      C46=SIGT
48      C17=SIG
49      C
50      GO TO (50,50,50,53,53),NSU
51      C
52      I3 IF(IOSV)CALL SOUNDIT,S,P,SV)
53      Q19=SV
54      IF(IOPT)CALL POTEMP(P,T,S,TPOT)
55      Q11)=TPOT
56      IF(IOPD)CALL SIGNAIP,TPOT,S,SIGPOT,HX)
57      C11)=SIGPOT
58      IF(IOSV)CALL ORNVAL(T,P,S,G,INIT,EN,PRIGHT,INBV)
59      C18)=EN
60      IF(IODT)CALL TRUDEP(P,SIG,INIT,DEP)
61      C1C)=DEP
62      C
63      SO CONTINUE
64      C
65      GO 2 I=1,NIND
66      N=NAD(I)
67      2 Q11)=Q(IN)
68      C
69      RETLKN
70      C
71      END

```

6PRT,S PF0U074PR6D,NBCCC

```

PF0U074PR6D(11).NBCCC
1      COMPILER(DTAG=3)
2      C
3      SUBROUTINE NBCCC (P,T,C,CF)
4      C
5      C NBCCC CORRECTS FOR TEMPERATURE AND PRESSURE EFFECTS ON CELL
6      C AND APPLIES CALIBRATION CORRECTION, CF.
7      C
8      DATA ALPHA/6.5DE-6/,FK/67.5E6/
9      C
10     C=CF+C*(1.-ALPHA*(T-15.))*(P01.45D38)/FK)
11     C
12     RETURN
13     END

```

6PRT,S PF0U074PR6D,BENDPH


```

PFOU074PRG011).BENDPH
1      COMPILER(DIAG=3)
2      C
3      SUBROUTINE BENDPH(P1,T1,C1,SPPT)
4      C
5      C THIS ROUTINE USES THE BENNETT(1976), AND GAMPHREE(1976),
6      C EQUATIONS TO COMPUTE SALINITY FROM PRESSURE, TEMPERATURE,
7      C AND CONDUCTIVITY.
8      C
9      C P1=PRESSURE IN DECIBARS
10     C T1=TEMPERATURE IN DEGREES CELSIUS
11     C C1=CONDUCTIVITY IN MHMO/CM
12     C RPRM=CONDUCTIVITY RATIO
13     C
14     C COMPUTE CONDUCTIVITY RATIO
15     RPRM=C1/42.906
16     C
17     C CORRECT RPRM TO ZERO PRESSURE
18     F=(16.166E-150P1-5.4405E-10)0P1+1.60836E-5)0P1/
19     1 (13.169E-40T1+3.0706E-2)0T1+1.)
20     R1=RPRM/(1.+F)
21     C
22     C CORRECT R1 FOR TEMPERATURE EFFECTS AT 35PPT WHERE
23     R1=C 35,T,0/C 35,15,0
24     C
25     R1=(1411.35060E-90T1-7.26682E-7)0T1+1.11099E-4)0T1+2.005294E-2)
26     1 0T1+6.76552E
27     R1=R1/R1H
28     C
29     C CALCULATE SALINITY
30     RSP=(1111-1.3231)0RS+5.98620)0RS-10.61869)0RS+12.18882)0RS+20.0567)
31     10RS+.00996
32     RSRM1=RS0(RS-1.10(14.42E-2-.46E-30T1-0.E-30RS)0T1
33     1 +11.25E-4-2.9E-60T1)0P1)
34     SPPT=RSP+RSRM1
35     C
36     RETURN
37     END

```

APRT,5 PFOU074PRG0.PSS78

```

PFOU074PRG011).PSS78
1      COMPILER(DIAG=3)
2      SUBROUTINE PSS78IP,T,C,SALD
3      C THIS ROUTINE USES THE PRACTICAL SALINITY SCALE (1978)
4      C EQUATIONS TO COMPUTE SALINITY FROM PRESSURE,
5      C TEMPERATURE AND CONDUCTIVITY.
6      C IEEE JOURNAL (JAN 1980)
7      C
8      C P1=PRESSURE (DB)
9      C T1=TEMPERATURE (DEG-C)
10     C C1=CONDUCTIVITY (MHMO/CM)
11     C RP=CONDUCTIVITY RATIO.
12     C
13     C COMPUTE CONDUCTIVITY RATIO
14     C
15     RP=C/42.914
16     C
17     ALPH=(12.070E-50P-4.370E-10)00+2.+3.989E-150P003.)
18     1 /11+3.426E-20T+4.464E-40T0+2.+
19     1 4.215E-10RP-3.107E-30RP0T)
20     C
21     C ALPH=F(R,T,P) IS THE FRACTIONAL INCREASE IN COND DUE TO PRESS.
22     C
23     SHTCO=6.766097E-1+2.00540E-20T+1.104259E-40T0+2.
24     1 -6.9698E-70T003.+1.0031E-90T004.
25     C
26     C THIS IS : C(35,T,0)/C(35,15,0)
27     C
28     RRT=RP/(SHTCO+(1.+ALPH))
29     C

```

```

30      SAL=10.0000-0.1692*ART*0.5*25.3651*ART
31      C +14.0941*ART*0.15-7.0261*ART*0.2*2.7081*ART*0.25)
32      C +1117-15.1/11.+10.0162*(T-15.)))))
33      C (0.0005-0.0056*ART*0.5-0.0066*ART
34      C -0.0375*ART*0.15+0.0636*ART*0.2.
35      C -0.0144*ART*0.25))
36      C
37      RETURN
38      END

```

MPRT,S PF0U074PFGD.SIGNA

PF0U074PRGD(11).SIGP

```

1      SUBROUTINE SIGNA(P,T,S,SIGT,SIGSTP)
2      TM398=T-3.98
3      SIGSIG=(1-TM398*TM398*(283.+T))/(503.57*(T+67.26))
4      SIGO=S*(5+.67678614E-5+S-.00249614E-3)+.01487058)
5      I=.934458632E-1
6      AT=1+1.47867E-2-T*1.98105E-4-T*0.10893E-5)
7      BT=1+1.18030E-4-T*1.8164E-6-T*0.1667E-7)
8      SIGT=SIGSIG*(5.350+.1324)*I)-AT*BT*(5160-.1324)
9      C THIS IS SIGMA-T
10     SIG20=SIGO-28.
11     T1=4006./I)+1.03E-5*P)
12     T2=227.+T*(20.33-T*1.551-T*0.004)
13     T3=1.E-4*P*(105.5+T*(9.50-T*0.168-1.E-4*P))
14     T4=.10516280*(147.3-T*(2.72-0.0007)
15     I-1.E-4*P*(32.4-T*1.87-.02*P))
16     T5=1.E-2*(516280516280*(4.5-.1*P-P*(1.0E-4-.6E-5*P))
17     XHU=1.E-9*(T1-T2+T3-T*P)
18     ALFA=(1.-XHU*P)/(SIGT*.001+.1)
19     RHOSTP=1./ALFA
20     SIGSTP=(RHOSTP-1.)*1000.
21     C THIS IS SIGMA-STP
22     RETURN
23     END

```

MPRT,S PF0U074PFGD.SOUND

PF0U074PRGD(13).SOUND

```

1      SUBROUTINE SOUND(T,SAL,PRES,SVEL)
2      C REVISID PRESSURE CONVERSION JMO
3      C REFERENCE - WILSON, W.D., 1960. EQUATION FOR THE SPEED OF
4      C SOUND IN SEA WATER, JOUR. ACUST. SOC. AMER., 32(13),1357.
5      C
6      C PRES= PRESSURE IN DECIBARS FROM SEA SURFACE .
7      C P = TOTAL PRESSURE IN KG/CM2 ABSOLUTE .
8      C SAL = SALINITY IN PARTS PER 1000 .
9      C T = TEMPERATURE IN DEGREES CELSIUS .
10     C SOUND = SOUND VELOCITY IN METERS PER SECOND .
11     C
12     P = (PRES + 10.1325) * 0.1019716
13     S = SAL - 35.
14     C
15     VT = 1404.5721+T*1-0.4532E-2+T*1-2.6045E-4+7.9851E-6*P))
16     VS = 5011.39799+1.69202E-3*S)
17     VP = P*(10.160272+P*(1.0260E-5+P*(3.5216E-9-3.3603E-12*P)))
18     C
19     VSTP = 5017*(1-1.1244E-2+7.7711E-7*P)+P*(17.7016E-5-1.2943E-7*P+
20     1.013-1580E-8+1.5790E-9*P))

```

```

21      2P01101-1.0607E-4*1017.4012E-6*4.5283E-8*111)*
22      3P0P01101-2.5294E-7*1.8563E-9*1-1.9646E-10*P1)
23      C
24      SOUND = 1449.10 * VT * VP * VS * VSTP
25      C
26      SVEL = SOUND
27      RETURN
28      END

```

APRT,S PF0U079P160.1RUDEP

PF0U079PRG0111.1RUCEP

```

1      COMPILER(DIAG=3)
2      SUBROUTINE 1RUDEP(PRESS,SIGSTP,INIT,TRUDP)
3      COMMON/UDOC1/IDOCR(1)
4      COMMON/UDOCF/FDOCR(1)
5      C
6      IF(INIT.NE.1) GO TO 2
7      XLAT=ABS(FDOCR(1))
8      GLAT = XLAT*0.017453293
9      TGRAV=9806.32.3-2586.157*0.05*GLAT*2.0)+2.882446*0.05*GLAT*4.0)
10     GRAV=1.0E-6*TGRAV
11     TRUDP=0.
12     PRVPRS=0.
13     RHOP=0.
14     C
15     2    RHQ=(SIGSTP/1000.)*1.
16     PRS=PRESS
17     IF(RHOP.EQ.0.1RHOP=RHQ
18     GRAVZ=GRAV*1.1*2.28E-7*TRUDP)
19     TRUDP=TRUDP+2.0*(PRS-PRVPRS)/(GRAVZ*1RHOP+RHQ)
20     C
21     PRVPRS=PRS
22     RHOP=RHQ
23     RETURN
24     END

```

APRT,S PF0U079PRG0.P01ENP

PF0U079PRG011.POTENP

```

1      COMPILER(DIAG=3)
2      SUBROUTINE POTENP(P,T,S,PTEMP)
3      C
4      C      THIS SUBROUTINE COMPUTES DELT, THE CHANGE IN TEMPERATURE
5      C      DUE TO ADIABATIC CHANGE IN SEA PRESSURE FROM INSITU POSITION
6      C      TO THE SURFACE. THE POTENTIAL TEMPERATURE, PTEMP, IS THEN
7      C      PTEMP = T-DELT ( IN DEG CELCIUS )
8      C      REFERENCE: THE SEA, 1962, VOL 1, PG 17
9      C
10     DEL1=-1.6E-5*1011.014E-5*101-1.27E-7*2.7E-9*101)
11     DEL2=1.322E-6-2.62E-8*101+1E-9*5
12     DEL3=9.14E-9-1.55E-13*101+2.77E-10*9.5E-13*101)
13     C
14     DEL1=P*DEL1+S*DEL2+P*DEL3)
15     PTEMP=T-DELT
16     C
17     RETURN
18     END

```

APRT,S PF0U079P160.DENVAL

```

PF=UD74PRGD(1).BRNVAL
1      COMPILER(DIAG=3)
2      SUBROUTINE BRNVAL (TJP1,PJP1,SJP1,G,INIT,ENCPH,PJ,ISBV)
3
4      C
5      E=9.00665
6      IF (INIT.EQ.1) GO TO 100
7      SBARJ=(SJP1+SJP1)/2.
8      TBARJ=(TJP1+TJP1)/2.
9      PBARJ=(PJP1+PJP1)/2.
10     CELPJ=(PJP1-PJP1)/2.
11     GAMBAR=GAMMA(TBARJ,PBARJ,SBARJ)+.001
12     GANDEL=GANBAR*DELPJ
13
14     C
15     CALL SIGMA(PBARJ,TJP1-GANDEL,SJP1,DUM,SSTP1)
16     ALPHA1=1./ (SSTP1+.001+.1)
17
18     C
19     CALL SIGMA(PBARJ,TJP1+GANDEL,SJP1,DUM,SSTP2)
20     ALPHA2=1./ (SSTP2+.001+.1)
21     E=(ALPHA2-ALPHA1)/(2.*DELPJ)
22     RHOJPH=(1/SSTP1+.001+.1)*(1/SSTP2+.001+.1)/2.
23     EN2RPS=.1*(1/RHOJPH+.002+.1)
24     IF (ISBV.EQ.0) GO TO 80
25     ENRPS=ABS(EN2RPS)+.5
26     ENCPH=(ENRPS/6.2832)*3600.
27     GO TO 90
28
29     80     ENCPH=EN2RPS
30     IF (E.LT.0.) ENCPH=-ENCPH
31
32     C
33     S1 TJP=TJPH
34     PJ=PJPH
35     SJ=SJP1
36     TJPH=TJP1
37     PJPH=PJP1
38     SJP1=SJP1
39     RETLRN
40
41     C
42     100 TJPH=TJP1
43     PJPH=PJP1
44     SJP1=SJP1
45     GO TO 91
46     END

```

BPRT,S PF=UD74P,GD.GAMMA

```

PF=UD74PRGD(1).EAMPA
1      FUNCTION GAMMA(T,P,S)
2      T2=T*T
3      P2=P*P
4      S2=S*S
5      T4=T2*T2
6      GAMMA=-4.21E-2+1.022E-2*T-1.478E-4*T2+3.45E-6*T2*T
7      1-3.3E-8*T4+2.427E-3*S-5.0E-7*S*T+9.0E-8*S*T2
8      2-6.5E-6*S2+2.291E-5*S*P-7.62E-7*P*T
9      3+7.5E-9*P*T2-1.06E-7*P*S+1.97E-9*P*S*T
10     4-4.53E-10*P2+1.56E-11*P2*T
11     RETURN
12     END

```

BPRT,S F.ZREAD

```

PFOUCT04PR6D(11).JREAD
1  C  COMPILER/DIAG=30
2  SUBROUTINE ZREAD(IU,IF,IDL)
3
4  C  THIS SUBROUTINE IS THE READ HALF OF AN INPUT-OUTPUT
5  C  PACKAGE FOR HANDLING NON-FORMATTED, FORTRAN
6  C  WRITTEN DATA FILES, COMMONLY REFERRED TO AS
7  C  FIDIFAST & EASY BINARY FILES.
8  C
9  COMMON / RMDR / LR,NR,NBR,NMBR,NMFR,NFR,NIR,NAR,IPR(11)
10 COMMON / RDOCF/FDOCR(11) / RDOC1/IDOCR(11) / RDOC4/ADOCR(11)
11 COMMON / RDATA / VRT(11)
12
13 C  COMMON / DIAGS / MSGR,MSGW,MNBR,MNHW,MNIP,MNF,MNI,MNA,IRST,INST
14 C  COMMON / JPOS / JUNIT(30)
15 C  COMMON / DRDCOM / JFLG,ISECR(30)
16 DIMENSION IUNIT(30)
17 LOGICAL B1,B210,B10,B35,B45,B69
18 DATA MSGR / 2 /
19 DATA LLSW/1/,IRST/1/
20
21 C
22 B1=MSGR.EQ.1
23 B210=MSGR.GE.2.AND.MSGR.LE.10
24 B10=MSGR.EQ.10
25 B35=MSGR.EQ.3.OR.MSGR.EQ.5.OR.MSGR.EQ.7.OR.MSGR.EQ.9.OR.MSGR.EQ.10
26 B45=MSGR.EQ.4.OR.MSGR.EQ.5.OR.MSGR.GE.8.AND.MSGR.LE.10
27 B69=MSGR.GE.6.AND.MSGR.LE.9
28
29 C
30 IBLK=IDL
31 IPOS=JUNIT(IU)
32 IF(IPOS.EQ.0) IPOS=1
33 ISEC=ISECR(IU)
34 IF(IBL.EQ.0) IBL=JUNIT(IU)
35 IF (IBLK.LT.IPOS) GO TO 5
36 4 IF (IBLK.EQ.IPOS) GO TO 3
37
38 C
39 READ(IU,END=99,ERR=98)LR,NR,NBR,NMBR,NMFR,IPR(11),I=1,LR)
40 9 NFR,NIR,NAR
41 ICC=0
42 IC1=0
43 IC2=0
44 IC3=0
45 IC4=0
46 IC0=(LR*8+30)/28
47 IF (NFR.NE.0) IC1=(NFR+((NFR-1)/22)*3)*30/28
48 IF (NAR.NE.0) IC2=(NAR+((NAR-1)/22)*3)*30/28
49 IF (NIR.NE.0) IC3=(NIR+((NIR-1)/22)*3)*30/28
50 IF (NBR.NE.0) IC4=(NBR+((NBR-1)/22)*3)*30/28
51 ISEC=ISEC+IC1+IC2+IC3+IC4+IC0
52 CALL SETADR(IU,ISEC)
53
54 C1
55 IPOS=IPOS+1
56 JUNIT(IU)=IPOS
57 JUNIT(IU)=IPOS
58 GO TO 4
59
60 C
61 5 IF=0
62 REMIND IU
63 IF (IBL.EQ.0) IBL=J
64 IPOS=1
65 IUNIT(IU)=IPOS
66 JUNIT(IU)=IPOS
67 ISECR(IU)=0
68 ISEC=0
69 GO TO 4
70
71 3 CONTINUE
72
73 C
74 READ(IU,END=99,ERR=98)LR,NR,NBR,NMBR,NMFR,IPR(11),I=1,LR)
75 9 NFR,NIR,NAR
76 ICC=0
77 IC1=0
78 IC2=0
79 IC3=0
80 IC4=0
81 IC0=(LR*8+30)/28
82 IF (NFR.NE.0) IC1=(NFR+((NFR-1)/22)*3)*30/28
83 IF (NAR.NE.0) IC2=(NAR+((NAR-1)/22)*3)*30/28
84 IF (NIR.NE.0) IC3=(NIR+((NIR-1)/22)*3)*30/28
85 IF (NBR.NE.0) IC4=(NBR+((NBR-1)/22)*3)*30/28
86 ISEC=ISEC+IC1+IC2+IC3+IC4+IC0
87 IF (NR.GT.NNBR.OR.LR.GT.NNIP.OR.NFR.GT.NMF)

```

```

83      * OR,NIR,GT,NMI,OR,NAR,GT,NNA) GO TO 95
84      IF (NFR+NIR+NAR).EQ.0) GO TO 11
85      IF (NFR,GT,0) READ(IU,END=99,ERR=98) (FDOCR(I),I=1,NFR)
86      IF (NIR,GT,0) READ(IU,END=99,ERR=98) (IDOCR(I),I=1,NIR)
87      IF (NAR,GT,0) READ(IU,END=99,ERR=98) (ADOCR(I),I=1,NAR)
88      11 CONTINUE
89      C1
90      C1
91      NL=NR+LR
92      N1=(IRST-1)*LR+1
93      N2=N1+NL-1
94      IF (JFLG.NE.1) GO TO 8
95      CALL SETADR(IU,ISEC)
96      GO TO 9
97      8 READ(IU,END=99,ERR=98) (VR(I),I=N1,N2)
98      9 CONTINUE
99      C
100     C
101     IPO5=IPO5+1
102     IUNIT(IU)=IPO5
103     JUNIT(IU)=IPO5
104     ISECR(IU)=ISEC
105     C
106     IF (NSER.EQ.0) GO TO 108
107     IF (0210) WRITE(6,1000) IU,NMFR,NBR,NMNR,NR,LR,NFR,NIR,NAR
108     1000 FORMAT(' READ UNIT',I3,' FILE ',A6,
109     * ' SEGNUM',I4,' SEGNUM ',A6,' N=',I6,
110     * ' L=',I4,' NF=',I4,' NI=',I4,' NA=',I4)
111     C
112     IF (01) WRITE(6,1011) IU,NMFR,NBR,NMNR,NR,LR,NFR,NIR,NAR
113     1011 FORMAT(' RD ',I4,2X,A6,2X,I4,2X,A6,2X,I6,4I4)
114     C
115     IF (035) WRITE(6,1012) (IPR(I),I=1,LR)
116     1012 FORMAT(' PARAMETERS: ',I2(2X,A6)/(13X,12(2X,A6)))
117     C
118     IF (.NOT.B45) GO TO 110
119     IF (NFR+NIR+NAR).EQ.0) GO TO 110
120     WRITE(6,1013)
121     1013 FORMAT(' ADDL DATA:')
122     IF (NFR,GT,0) WRITE(6,1100) (FDOCR(I),I=1,NFR)
123     IF (NIR,GT,0) WRITE(6,1101) (IDOCR(I),I=1,NIR)
124     IF (NAR,GT,0) WRITE(6,1102) (ADOCR(I),I=1,NAR)
125     1100 FORMAT(10611.5)
126     1101 FORMAT(1X,12I6)
127     1102 FORMAT(1X,12A6)
128     C
129     110 IF (.NOT.B69) GO TO 107
130     JL=IRST*LR
131     J1=JL-LR+1
132     WRITE(6,1014) (VR(I),I=J1,JL)
133     JL=INR*IRST-1*LR
134     J1=JL+1-LR
135     WRITE(6,1015) (VR(I),I=J1,JL)
136     1014 FORMAT(' FIRST CYCLE: ',10611.5/(13X,10611.5))
137     1015 FORMAT(' LAST CYCLE: ',10611.5/(13X,10611.5))
138     C
139     107 IF (.NOT.B10) GO TO 108
140     WRITE(6,1017)
141     1017 FORMAT(' I=1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165')
142     IQ2=IQ1+NR-1
143     CO 106 I=IQ1,IQ2
144     JL=IQ2*LR
145     J1=JL+1-LR
146     WRITE(6,1016) (VR(I),I=J1,JL)
147     1016 CONTINUE
148     1016 FORMAT(15X,15,3X,10612.6)
149     1017 FORMAT(' LISTING OF DATA'///)
150     C
151     108 IF=0
152     IUP=IU
153     RETURN
154     C
155     C
156     95 IF=5
157     WRITE(6,1005) NMNR,NMIP,NMF,NMI,NNA,
158     * NR,LR,NFR,NIR,NAR
159     1005 FORMAT(' A DIMENSION IS TOO SMALL.///
160     * NMNR=',I6,' NMIP=',I6,' NMF=',I6,
161     * ' NMI=',I6,' NNA=',I6,' NR=',I6,
162     * ' LR=',I6,' NFR=',I6,' NIR=',I6,' NAR=',I6///)
163     RETURN
164     98 IF=2
165     WRITE(6,1002) IU

```

```

166      10G2 FORMAT(' READ ERROR ON UNIT ',I3)
167      GO TO 90
168      99 IF=1
169      WRITE(6,J001) IU
170      10C1 FORMAT(' EOF ON UNIT ',I3)
171      90 REWIND IU
172      IPOS=0
173      IUNIT(IU)=IPOS
174      JUNIT(IU)=IPOS
175      ISEC(IU)=0
176      RETURN
177      END

```

APRT,S P.ZWRIT

PF00074PRG011).ZWRIT

```

1      COMPILER(DIAG=3)
2      SUBROUTINE ZWRIT(IU,IF,IOL)
3
4      C THIS SUBROUTINE IS THE WRITE HALF OF AN INPUT-OUTPUT
5      C PACKAGE FOR HANDLING NON-FORMATTED, FORTRAN
6      C WRITTEN DATA FILES, COMMONLY REFERRED TO AS
7      C FEFIFAST & EASY BINARY) FILES.
8
9      C
10     COMMON / WHDR / LW,MW,NBW,MMW,MFW,NFW,NIW,NAW,IPW(1)
11     COMMON / WDOCF / FDOCF(1) / WDOI / IDOCW(1) / WDOCA / ADOCF(1)
12     COMMON / WDATA / VMI(1)
13
14     COMMON / RHDR / LR,NR,NDR,MMR,MFR,NFR,NIR,NAR,IPR(1)
15     COMMON / RDOCF / FDOCR(1) / RDOI / IDOCR(1) / RDOCA / ADOCR(1)
16     COMMON / RDATA / VRI(1)
17
18     C
19     COMMON / DIAGS / MSGR,MSGW,MMNR,MMNW,MNTP,MNF,MNI,MNA,IRST,IWST
20     LOGICAL B1,B210,B10,B35,B45,B69
21     COMMON / JPOS / JUNIT(30)
22     COMMON / DRDCOM / JFL6,ISECR(30)
23     DIMENSION IUNIT(30),KUNIT(30)
24     DATA MSGW / 2 /
25     DATA LLSW / 1 /, IRST, IWST / 1, 1 /
26
27     C
28     IW=1
29     IF (IU.LT.0) IW=2
30     IU=ABS(IU)
31
32     C
33     B1=MSGW.EQ.1
34     B210=MSGW.EQ.2.AND.MSGW.LE.10
35     B10=MSGW.EQ.10
36     B35=MSGW.EQ.3.OR.MSGW.EQ.5.OR.MSGW.EQ.7.OR.MSGW.EQ.9.OR.MSGW.EQ.10
37     B45=MSGW.EQ.4.OR.MSGW.EQ.5.OR.MSGW.EQ.8.AND.MSGW.LE.10
38     B69=MSGW.EQ.6.AND.MSGW.LE.9
39
40     C
41     IOLK=IOL
42     IPOS=JUNIT(IU)
43     IF (IPOS.EQ.0) IPOS=1
44     IF (IOLK.EQ.0) IOLK=JUNIT(IU)
45     IF (IOLK.LT.IPOS) GO TO 5
46     4 IF (IOLK.EQ.IPOS) GO TO 3
47
48     C
49     READ(IU,END=99,ERR=90) L0,N0,I0,J0,I1,J1,I2,J2,L1,MFW,NIW,NAW
50     IC0=0
51     IC1=0
52     IC2=0
53     IC3=0
54     IC4=0
55     IC5=(ILO*30)/20
56     IF (MFW.NE.0) IC1=(MFW+1*(MFW-1)/22)*31*30/20
57     IF (NIW.NE.0) IC2=(NIW+1*(NIW-1)/22)*31*30/20
58     IF (NAW.NE.0) IC3=(NAW+1*(NAW-1)/22)*31*30/20

```

```

56      IF (INQ.NE.O) IC4=((INQ+LQ)+1*((INQ+LQ)-1)/221)*31+301/28
57      ISEC=ISEC+ICD+IC1+IC2+IC3+IC4
58      CALL SETADR(IU,ISEC)
59      IPQS=IPQS+1
60      IUNIT(IU)=IPOS
61      JUNIT(IU)=IPOS
62      GO TO 4
63  C
64  5 REMIND IU
65      IPOS=1
66      ISEC=0
67      ISECR(IU)=0
68      IUNIT(IU)=IPOS
69      JUNIT(IU)=IPOS
70      IF (IDL.NE.J) GO TO 4
71  C
72  2 READ(IU,END=4,ERR=0) LQ,NQ,(IX,I=1,3),(IXQ,J=1,LQ),NFO,NIO,NAQ
73      ICD=0
74      IC1=0
75      IC2=0
76      IC3=0
77      IC4=0
78      ICQ=((LQ+301)/28
79      IF (NFO.NE.O) IC1=(NFO+1*((NFO-1)/221)*31+301)/28
80      IF (NIO.NE.O) IC2=(NIO+1*((NIO-1)/221)*31+301)/28
81      IF (NAQ.NE.O) IC3=(NAQ+1*((NAQ-1)/221)*31+301)/28
82      IF (INQ.NE.O) IC4=((INQ+LQ)+1*((INQ+LQ)-1)/221)*31+301/28
83      ISEC=ISEC+ICD+IC1+IC2+IC3+IC4
84      CALL SETADR(IU,ISEC)
85      IPOS=IPOS+1
86      IUNIT(IU)=IPOS
87      JUNIT(IU)=IPOS
88      IF (IPQS.EQ.NUNIT(IU)) GO TO 3
89  C2
90      GO TO 2
91  C
92  8 CONTINUE
93      IUNIT(IU)=IPOS
94      IDLK=1
95      IPQS=1
96      REMIND IU
97      ISEC=0
98      ISECR(IU)=0
99      IF (IUNIT(IU).EQ.J) GO TO 3
100     GO TO 2
101  C
102  6 CONTINUE
103      BACKSPACE IU
104      WRITE(6,1001)IU
105  C
106  C
107  3 CONTINUE
108      NU=IPOS
109      GO TO (01,02),IU
110  01 IF (INW.GT.NMW.OR.LV.GT.NNIP.OR.NFW.GT.NNF.
111     *OR.NIW.GT.NMI.OR.NAL.GT.NNA) GO TO 95
112      WRITE(IU,ERR=97) LV,NU,NW,NM,NMW,NMF,NIPW(I),I=1,LV),NFW,NIW,NAW
113      IF (NFW.NIW.NAW).EQ.0) GO TO 87
114      IF (INFW.GT.0) WRITE(IU,ERR=97) (FDOCR(I),I=1,NFW)
115      IF (NIW.GT.0) WRITE(IU,ERR=97) (IDOCR(I),I=1,NIW)
116      IF (INAW.GT.0) WRITE(IU,ERR=97) (ADOCR(I),I=1,NAW)
117  87 CONTINUE
118  C1
119  C1
120      NL=NU*LV
121      N1=(INST-1)*LV+1
122      N2=N1+NL-1
123      WRITE(IU,ERR=98) (VR(J),J=N1,N2)
124  C
125      GO TO 83
126  82 WRITE(IU,ERR=97) LR,NR,NW,NMR,NMNR,NIPR(I),I=1,LR),NFR,NIR,NAR
127      IF (INFR.NIR.NAR).EQ.0) GO TO 88
128      IF (INFR.GT.0) WRITE(IU,ERR=97) (FDOCR(I),I=1,NFR)
129      IF (NIR.GT.0) WRITE(IU,ERR=97) (IDOCR(I),I=1,NIR)
130      IF (NAR.GT.0) WRITE(IU,ERR=97) (ADOCR(I),I=1,NAR)
131  88 CONTINUE
132  C1
133  C1
134      NL=NR*LR
135      N1=(INST-1)*LR+1
136      N2=N1+NL-1
137      WRITE(IU,ERR=97) (VR(J),J=N1,N2)
138  C

```



```

139      23 CONTINUE
140      C
141      C
142      IPO5=IPOS+1
143      ISEC(IU)=ISEC
144      IUNIT(IU)=IPOS
145      IUNIT(IU)=IPOS
146      IUNIT(IU)=IPOS
147      GO TO 104,85,110
148      24 IF(8210) WRITE(6,1000)IU,NMFU,NBU,NMBU,NV,LU,NFU,NIV,NAM
149      1000 FORMAT(' WRITE UNIT',I3,' FILE ',A6,
150      * ' SEGNUM',I4,' SEGNAME',A6,' N=',I6,
151      * ' L=',I4,' NF=',I4,' NI=',I4,' NA=',I4)
152      C
153      C
154      IF(81) WRITE(6,1011) IU,NMFU,NBU,NMBU,NV,LU,NFU,NIV,NAM
155      1011 FORMAT(' WRT ',I4,2X,A6,2X,I4,2X,A6,2X,I6,4I4)
156      C
157      IF(835) WRITE(6,1012)((IPW(I),I=1,LU)
158      1012 FORMAT(' PARAMETERS: ',2(2X,A6)/(13X,12(2X,A6)))
159      C
160      IF(1.NOT.B45) GO TO 110
161      IF(1.NFU.NIV.NAM).EQ.0) GO TO 110
162      WRITE(6,1013)
163      1013 FORMAT(' ADD DATA:')
164      IF(1.NFU,67.C)WRITE(6,1100)((FDCR(I),I=1,NFU)
165      IF(1.NIV,67.C)WRITE(6,1101)((IDCR(I),I=1,NIV)
166      IF(1.NAM,67.C)WRITE(6,1102)((ADCR(I),I=1,NAM)
167      1100 FORMAT(10G11.5)
168      1101 FORMAT(1X,12I6)
169      1102 FORMAT(1X,12A6)
170      C
171      110 IF(1.NOT.B49) GO TO 107
172      JL=INSTLM
173      J1=JL-LM+1
174      WRITE(6,1014)((VM(I),I=J1,JL)
175      JL=(NM+INST-1)*LM
176      J1=JL+1-LM
177      WRITE(6,1015)((VM(J),J=J1,JL)
178      1014 FORMAT(' FIRST CYCLE:',10G11.5/(13X,10G11.5)
179      1015 FORMAT(' LAST CYCLE:',10G11.5/(13X,10G11.5)
180      C
181      107 IF(1.NOT.B10) GO TO 108
182      WRITE(6,1017)
183      J01=J05T
184      I02=I01+NM-1
185      GO 106 I=I01,I02
186      JL=1*LM
187      J1=JL+1-LM
188      WRITE(6,1016) I,((VM(J),J=J1,JL)
189      106 CONTINUE
190      1016 FORMAT(5X,15,3X,10G12.6)
191      1017 FORMAT('/// LISTING OF DATA'///)
192      C
193      GO TO 86
194      85 IF(8210) WRITE(6,1000)IU,NMFR,NBU,NMBR,NR,LR,NFR,NIR,NAR
195      C
196      IF(81) WRITE(6,1011) IU,NMFR,NBU,NMBR,NR,LR,NFR,NIR,NAR
197      C
198      IF(835) WRITE(6,1012)((IPR(I),I=1,LR)
199      C
200      IF(1.NOT.B45) GO TO 109
201      IF(1.NFR.NIR.NAR).EQ.0) GO TO 109
202      WRITE(6,1013)
203      IF(1.NFR,67.C)WRITE(6,1100)((FDCR(I),I=1,NFR)
204      IF(1.NIR,67.C)WRITE(6,1101)((IDCR(I),I=1,NIR)
205      IF(1.NAR,67.C)WRITE(6,1102)((ADCR(I),I=1,NAR)
206      C
207      109 IF(1.NOT.B49) GO TO 117
208      JL=IRSTLMR
209      J1=JL+1-LR
210      WRITE(6,1014)((VR(I),I=J1,JL)
211      JL=(NR+IRST-1)*LR
212      J1=JL+1-LR
213      WRITE(6,1015)((VR(J),J=J1,JL)
214      C
215      117 IF(1.NOT.B10) GO TO 108
216      WRITE(6,1017)
217      J01=IRST
218      I02=I01+NR-1
219      GO 116 I=I01,I02
220      JL=1*LR
221      J1=JL+1-LR

```

```

222      WRITE(6,1016) I,(VR(J),J=J1,JL)
223      116 CONTINUE
224      C
225      C
226      86 CONTINUE
227      108 IF=0
228      IUP=IU
229      RETURN
230      C
231      C
232      95 IF=5
233      WRITE(6,1005)NNNU,NNIP,NMF,NMI,MNA,
234      *   NU,LU,NFU,NIU,NAN
235      1005 FORMAT(// 'A DIMENSION IS TOO SMALL-')
236      *   NNU=*,16,*   NNIP=*,16,*   NMF=*,16,*
237      *   NMI=*,16,*   MNA=*,16,*   NU=*,16,*
238      *   LU=*,16,*   NFU=*,16,*   NIU=*,16,*   NAN=*,16//)
239      RETURN
240      97 IF=1
241      WRITE(6,1003) IU
242      1003 FORMAT(1' WRITE ERROR ON UNIT ',I3)
243      LU=IU
244      GO TO 90
245      98 IF=2
246      WRITE(6,1002) IU
247      1002 FORMAT(1' READ ERROR ON UNIT ',I3)
248      LU=IU
249      GO TO 90
250      99 IF=1
251      WRITE(6,1001) IU
252      1001 FORMAT(1' EOF ON UNIT ',I3)
253      LU=IU
254      GO TO 90
255      C
256      ENTRY RESET(MU)
257      LU=MU
258      50 FEWNO LU
259      IPO5=0
260      IUNIT(LU)=0
261      JUNIT(LU)=0
262      ISECR(LU)=C
263      RETURN
264      END

```

ABRPT PRINTS

UNCLASSIFIED

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
NORDA Technical Note 196	AD-A128146	
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED
A Computer Program for Processing Conductivity-Temperature-Depth (CTD) Data		Final
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
Z.R. Hallock		
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Naval Ocean Research and Development Activity NSTL Station, Mississippi 39529		P.E. 63704N
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
Naval Ocean Research and Development Activity NSTL Station, Mississippi 39529		January 1983
		13. NUMBER OF PAGES
		39
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
		UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
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)		
CTD Data Processing		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
A software package for processing conductivity-temperature-depth (CTD) data is described. The package includes features for editing, correcting, filtering and pressure sorting to produce working data files for graphic and analysis work.		

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)