Woods Hole
Oceanographic
Institution

VAX-11 PROGRAMS FOR COMPUTING AVAILABLE POTENTIAL ENERGY FROM CTD DATA

by

Nancy Amanda Bray

August 1981

TECHNICAL REPORT

Prepared for the Office of Naval Research under Contracts N00014-76-C-0197; NR 083-400 and N00014-79-C-0071; NR 083-004 and for the National Science Foundation under Grant OCE 77-19403.

Approved for public release; distribution unlimited.

WOODS HOLE, MASSACHUSETTS 02543

WHOI-81-70

VAX-11 PROGRAMS FOR COMPUTING AVAILABLE POTENTIAL ENERGY FROM CTD DATA

by

Nancy Amanda Bray

WOODS HOLE OCEANOGRAPHIC INSTITUTION
Woods Hole, Massachusetts 02543

August 1981

TECHNICAL REPORT

Prepared for the Office of Naval Research under Contracts N00014-76-C-197; NR 083-400 and N00014-79-C-0071; NR 083-004 and by the National Science Foundation under Grant OCE-77-19403.

Reproduction in whole or in part is permitted for any purpose of the United States Government. This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-81-70.

Approved for public release; distribution unlimited.

Approved for Distribution:

Valentine Worthington, Chairman
Department of Physical Oceanography

TABLE OF CONTENTS

Abstract

This report documents the W.H.O.I. VAX-11 programs used to calculate available potential energy and related quantities from CTD data using the technique described in Bray and Fofonoff (1981). The report includes examples of how the programs may be used, as well as complete listings of all the required FORTRAN files.

List of Tables

List of Figures

Introduction

This report describes the structure and usage of programs designed for calculating and displaying available potential energy (APE), adiabatically leveled steric surfaces, and related variables from a group of CTD stations. For a general discussion of the technique it is strongly recommended that the reader refer to Bray and Fofonoff (1981). The programs have an inherent requirement that the input CTD data be an even series in pressure, although the input pressure interval may be specified. This report describes specifically the structure of the programs as used on the W.H.O.I. VAX-11, with input data in the standard CTD78 disc format (Millard, et al (1978)). Other input formats can be accomodated through modification of the data input subroutine as described in section 4.

The calculation and display are divided into separate programs. POTential ENergy (POTEN) reads the input data, calculates the adiabatically leveled reference steric field (see Bray and Fofonoff, 1981) and variables related to the leveled field. Potential Energy PLoT (PEPLT) calculates variables derived from the leveled field variables and displays POTEN output in the form of lists and plots.

This report is divided into four sections. The first, General Structure, covers the non-FORTRAN aspects of the programs: file structure, linkage and general usage. The second and third sections contain detailed documentation for POTEN and PEPLT. The fourth section describes modifications to the data read subroutine in POTEN, to allow input data in other than CTD78 disc format. Documented examples of how to run the programs interactively and in batch mode on the VAX-11 are found in Appendix A. Listings of programs appear in Appendices B and C.

1. General Structure of Programs

Both POTEN and PEPLT are accessed through a short main program which performs initializations of parameters as requested by the user. Control is then transferred to one of three major subroutines, from which point the user is free to access different branches within that subroutine, or request entrance into either of the other two major subroutines. The various branches are described in detail in the following sections. Schematics of POTEN and PEPLT are shown in Figures 1, 2 and 3. The remainder of POTEN and PEPLT consist of secondary subroutines: data read, physical properties of seawater, etc., which are accessed as part of the various branches available to the user in the major subroutines. The file structure reflects the program structure (Table 1). POTEN and PEPLT are linked by linking the object files in Table 1. Accessory files are listed in Table 2.

The input data in CTD78 disc format is accessed using subroutines from CTDATA/LIB, and the plots in PEPLT are created using the NCAR plot package. The plot package creates a file on logical unit 8 which must be read and translated into plot(s) by a Metacode translator. Those translators are available both for the high speed Calcomp plotter and for various screens, for plot previewing. The absolute plot dimensions may be altered after the file is created, and the plots can be plotted as many times as desired. The use of the translators is described at the end of section 3.

The multiple branch structure of the programs provides an extremely powerful and flexible framework for computations which are often not routine; however useful documentation of such programs is correpondingly difficult. It is suggested that the new user begin by studying Figs. 1., 2 and 3. A documented command file (ENERGY.COM) for a routine computation and display is found in Appendix A. This file allows the new user to become familiar gradually with the options available in the programs. After studying and experimenting with the command file, the user may wish to explore other options available by referring to the detailed branch descriptions found in sections 2 and 3 of this report.

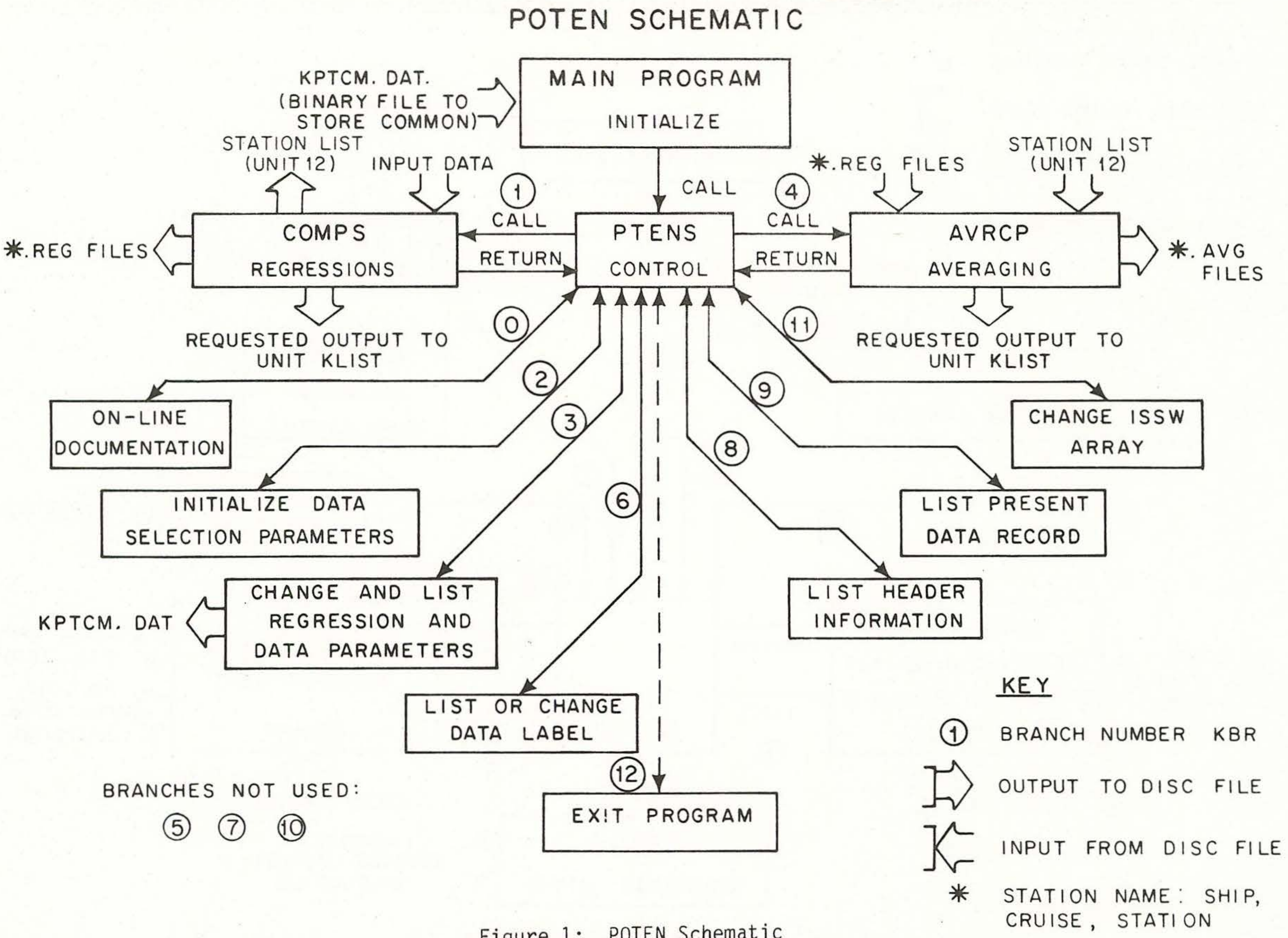


Figure 1: POTEN Schematic

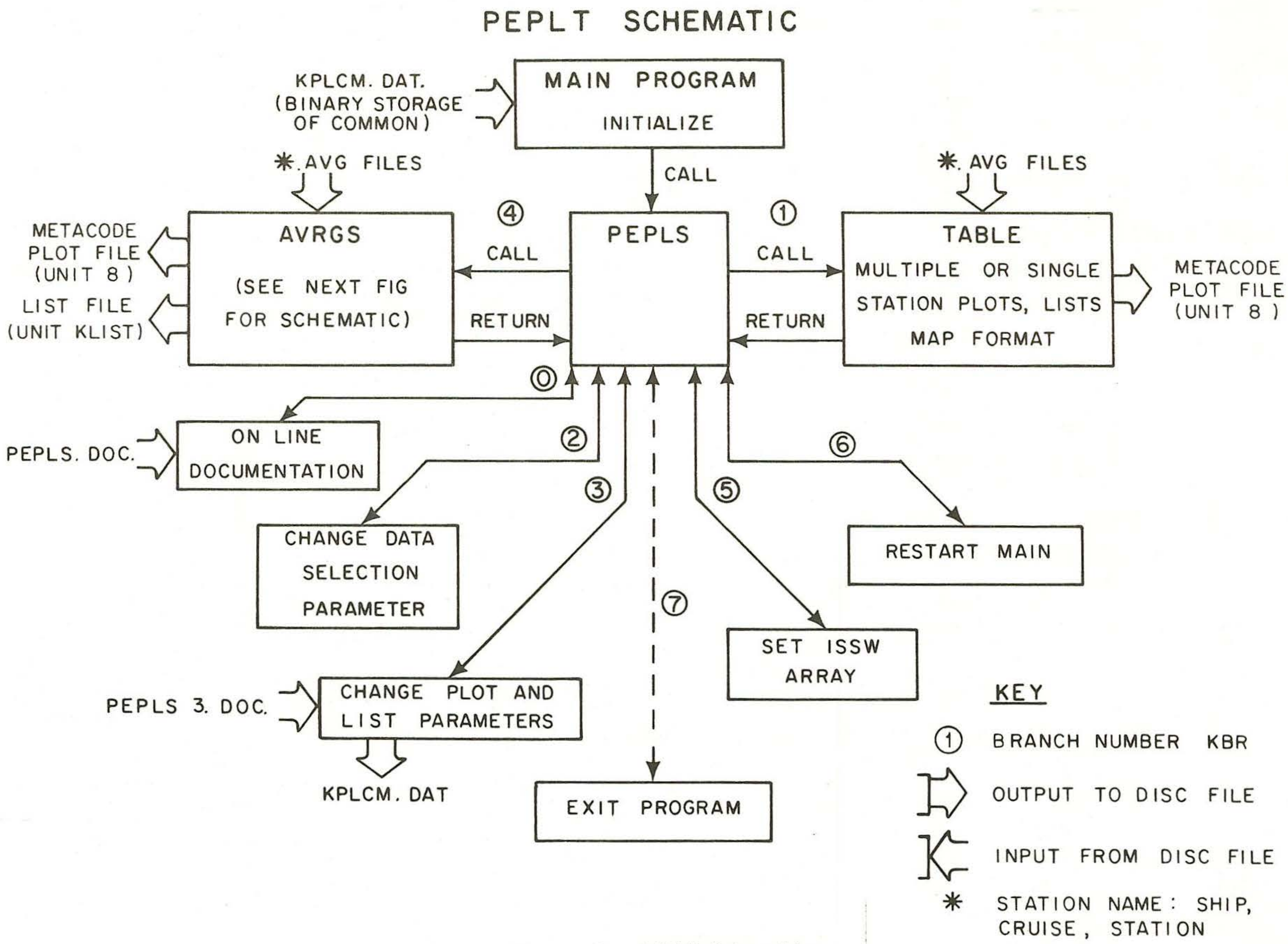


Figure 2: PEPLT Schematic

AVRGS SCHEMATIC

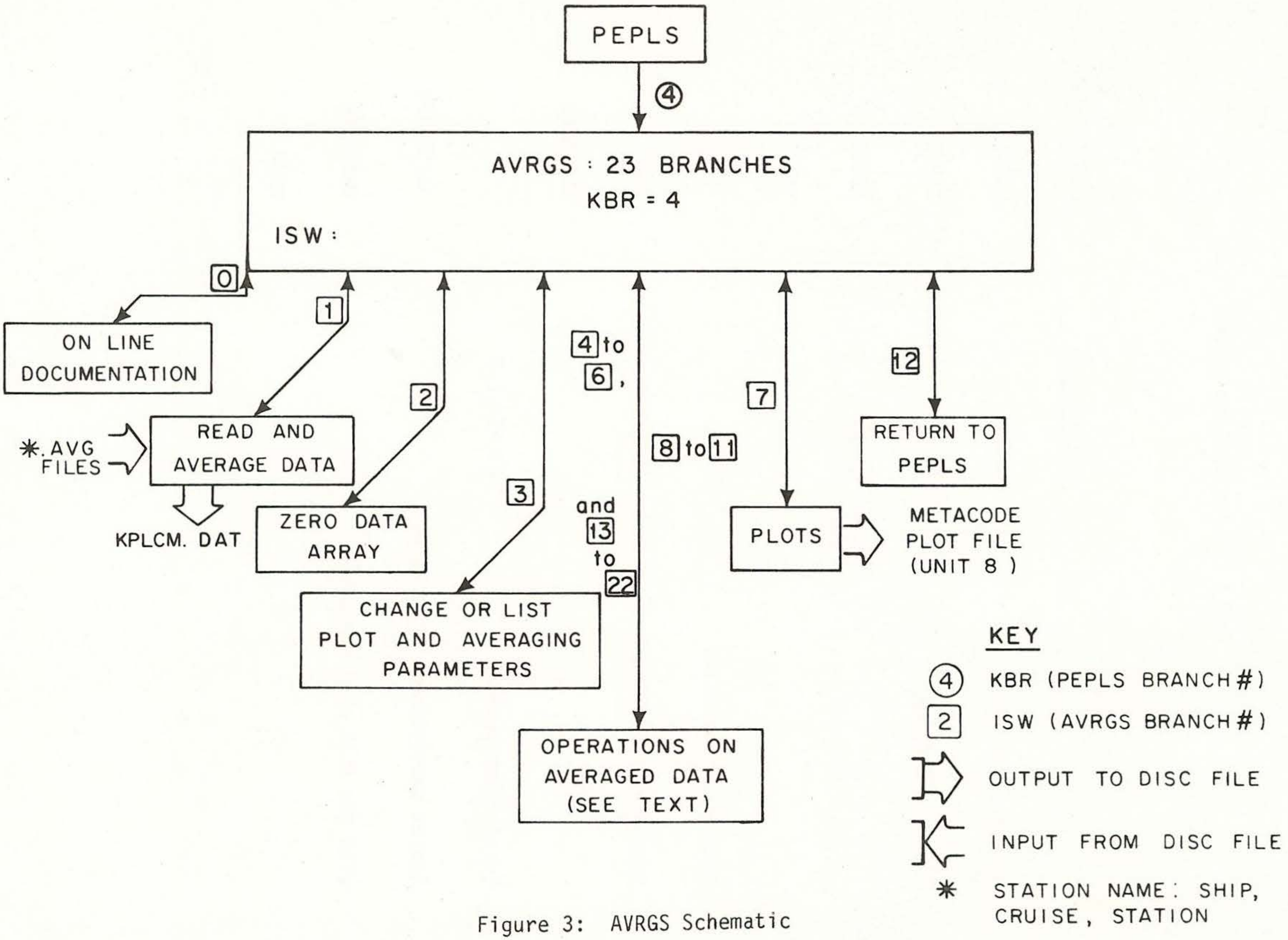


Figure 3: AVRGS Schematic

TABLE 1: FORTRAN and Object Files

	POTEN Files	PEPLT Files
Main program	POTEN	PEPLT
Major subroutines	PTENS	PEPLS
	COMPS	AVRGS
	AVRCP	TABLE
Data read subroutine	DATA	TDATA
Secondary subroutines	POTENSUB	PEPLTSUB
		POTENSUB
Library subroutines	CTDATA/LIB	
System subroutines (associated with the NCAR plot package)		AUTOGRAPH
		DASHCHAR
		NCAR

TABLE 2

Accessory Files	POTEN	PEPLT
Common, dimension and equivalence required for compilation (FORTRAN)	COMPOTEN.FOR	COMPEPLT.FOR
Station lists (JSHP.PTN is generated by COMPS -- unit 12)	JSHP.PTN	JSHP.PTN
Input data	*.CTD	*.AVG
*.CTD is CTD file	*.REG	
*.REG is generated by COMPS		
*.AVG is generated by AVRCP		
* is the station identifier		
Common storage (binary file) POTEN and PEPLT generate these if they do not exist in the directory	KPTCM.DAT	KPLCM.DAT
Documentation files (formatted)	POTEN.DOC	PEPLS.DOC PEPLS3.DOC AVRGS.DOC
Command file used to set up assignments and start an interactive job	POTEN.COM	PEPLT.COM
Command file used to run POTEN and PEPLT sequentially in batch mode, supplying standard output.	ENERGY.COM	
Command file to create plots shown in Fig. 4 using TABLE subroutine.		TABLE.COM
Command file to compute dynamic height station by station and output in map format.		DYNHT.COM

2. Documentation of POTEN

In this section, the major subroutines in POTEN described above are documented in detail. They are all structured around the multiple GO TO statement of the form

GO TO (#1, #2, #3,.... n) BRANCH

with #1 through #n FORTRAN statement numbers, and the BRANCH # an index such that BRANCH # = n transfers control to Statement #n. The BRANCH is input by the user following an appropriate program prompt. The branch numbers are keyed to various computations which may be accessed at the user's option. The keys are listed later in this section, and short versions of them may be obtained on the terminal any time the program prompts the user for branch number input, by typing Ø/.

Within each branch there may be options which are accessible by varying parameters input by the user at the time the branch number is input. These options are also listed in the branch keys.

In addition to input parameter options, there is an array called ISSW with 16 elements found in both programs. Within the different branches, different elements of ISSW may be tested for values of -1 or 0, and options either accessed or skipped depending upon the value. In general, ISSW elements determine whether a given type of output is generated. (Historically, the ISSW array derives from the binary switches available on the shipboard computer, the HP2100 series.) The elements of ISSW may be altered by accessing the appropriate branch in both POTEN and PEPLT as described below.

As described earlier, POTEN is accessed as a short main program which initializes parameters if requested by the user, or reads from a binary file KPTCM.DAT the most recently stored parameters, if no initialization is requested. The main program POTEN then transfers control to the major subroutine PTENS which, as shown in the schematic Fig. 1, controls the various branches available to the user. PTENS is the only component of POTEN in which branches may be accessed. The two remaining major subroutines are COMPS, in which the regressions are performed, and AVRCP, in which the horizontal averaging is performed.

This subsection charts the branches available to the user in detail, and describes briefly the working of COMPS and AVRCP. Short versions of the branch documentation are found in the Appendix, and may also be printed on the screen while the program is (interactively) on line by typing Ø/ whenever the program prompts for branch input.

2a. Main Program:

The main program queries the user 'Initialize common (YES or NO)?'. A NO response causes the present elements of KPTCM.DAT, the binary storage file, to be read into common. (If no file KPTCM.DAT exists in the directory, the program will create a new file named KPTCM.DAT, but if the response to the initialization query was NO, an 'end of file during read' error will result. Therefore, the proper sequence of commands to create a new KPTCM.DAT file is to run POTEN, respond YES to the initialization, thereby creating a new KPTCM.DAT, but not attempting to read from it. Later in the program (in branch 3) common may be stored to the newly created file, for use next time the program is run.) A YES response initializes the data selection parameters (subroutine DATA), and certain other parameters not related to the regressions.

Following this query, control is transferred to PTENS, and the user is asked: 'Initialize regression parameters (YES or NO)?'. A YES response initializes the regression parameters. A NO response reads them from KPTCM.DAT. (Again, with a newly created KPTCM.DAT file, the correct response is YES.) Finally, PTENS asks for the resolution of the input data, before going to branch mode. At this point the user may input up to 7 variables, as listed in the program prompt. The current values of the variables are printed on the screen along with the prompt list. The variables are: KBR, the branch number; ISW and JSW, which may access different options in branch KBR; KLIST, usually the list output logical device number (reset to 6 each time the prompt is printed); KOUT and KTP, the data output and input logical device numbers (note that the program uses named files for data input and output via OPEN statements which use

KTP and KOUT as unit numbers); KIN, the program input device for screen or command file. (Changing the value of KIN to 6 part way through a COM file transfers control to the screen, allowing interactive mode -- see Appendix A for an example command file, POTEN.COM.)

2b. Branches (KBR)

0: Short documentation printed on screen. See Appendix B for a listing of this documentation.

SUMMARY - POTEN:PTENS: KBR = 0

Function: List on terminal the short documentation for PTENS
ISW, JSW options: None
Output device: unit KTTX
Input device: None
ISSW options: None

1: Calls COMPS subroutine, which performs the following sequence:

a. Calls subroutine DATA, which opens the subindex directory for the default file specifications of the input data. (Those specifications may be changed by calling KBR = 13, which is identical to branch 1 except for allowing file specifications to be changed.) Then the header for the ISWth sequential station in that subindex file is examined to see if it meets data selection criteria. If so, a file name is written to file corresponding to logical unit 12. Throughout this report that file is called JSHP.PTN; an example is given in Table 3. The temperature and salinity data are transferred to array DATAX, using Millard subroutine GETDAT. Pressure is stored in the zeroth element of DATAX, which is equivalenced to array PRESS. PRESS is used throught COMPS and AVRCP. The total number of scans (NTOT) is also noted. The above occurs in subroutine DATA, after which control returns to COMPS.

b. COMPS then sets up the regression for the first interval using parameters which may be changed using branch 3 and continues the computation through all the intervals requested,

TABLE 3

Example of a JSHP.PTN file generated by COMPS

Consec. Number	Station I.D.	Weight
1	GY001002	1.0
2	GY001003	1.0
3	GY001004	1.0

or until the end of the data (determined by NTOT) is reached. For each interval potential temperature and steric anomaly referred to p_f (the level pressure) are calculated for each data scan to be used in the regression. Potential temperature is calculated according to Fofonoff (1977), using the polynomial formula of Bryden (1973) for the adiabatic temperature gradient. Steric anomaly δ is calculated as:

$$\delta = 10^5 \times (\alpha(p,\theta(p,T,S,p_f), \delta) - \alpha(p,0,35))$$

with α the specific volume calculated according to the SCOR Working Group 51 new equation of state for seawater (Millero, et al, 1980), for which an algorithm is given by Fofonoff (1981). Within each interval an editing process occurs in which points exceeding three standard deviations of the regression estimate at a given steric anomaly are flagged. Temperature and saliity are then regressed against pressure over the interval. Any points in T or S which exceed three standard deviations are replaced by the regression estimate. The regression of steric anomaly is performed again and rechecked. The number of standard deviations for both tests may be changed -- see KBR = 3. The interpolated scans are printed out on unit KLIST and data scans which are flagged but not interpolated are also listed as such on KLIST if ISSW (3) is set to -1. (ISSW values may be changed using branch 5.) Pressure p and potential temperature θ (referred to the level pressure p_f) are regressed against steric volume anomaly (also referred to p_f) and the coefficients for both p and θ are stored in arrays CP and CT for each interval. Data output occurs if ISSW (13) = -1, and is written into a file with the name *.REG, where * identifies the station, a two character (alpha) ship name, a 3 digit cruise number and a 3 digit station number. The format of the ouput file is a header of 150 words equivalenced to an I*4 array followed by a variable number of data records (each 46 words, also an I*4 array), one record per level

TABLE 4

POTEN Data Output Variables

HEADER RECORD: 150 WORDS

VARIABLE NAME	DESCRIPTION	
LTYPE	Identifies record as header record (LTYPE = 1)	
MHDR	Number of elements in header	
ICON	Sequential number of station (in POTEN calculation)	
ISHP	Ship name (A2 format)	
KCAST	Station number	
IDAY	Julian year day	
IPR	First pressure	
LPR	Last pressure	
XLAT	Latitude of station	
XLONG	Longitude of station	
WGT	Weight	
XLTØ	Latitude of origin for distance computations in kilometers (negative for south latitude)	
XLGØ	Longitude of origin (negative for west)	
LBBL(3)	Short station label (3A4 formt)	
LBL(13)	Run identification label (13A4 format)	
NSC(60)	Regression parameters	} see text
NPR(60)	Regression parameters	
NSECTION	Number of sections in the water column	

DATA RECORD: 46 WORDS

KTYPE	Identifies record as data record (KTYPE = 0)
MBUF	Number of elements in data record
IREC	Level number
N	Polynomial order
NDP	Number of data scans used in regression
KSW	Not used

TABLE 4 (continued)

NAME	DESCRIPTION	
L1	Not used	
L2	Not used	
PF	Level pressure	
TØ,SØ,DVØ	Temperature, salinity and steric anomaly from input data, averaged about PF ± PDIFF (see branch 3 description and Table 5)	
PI	Pressure of the reference steric anomaly (DVF) in the unleveled or initial field	
THF	Local potential temperature (referred to PF) as estimated by the regressions: $\theta_f(P_f)$	
DVI	Steric anomaly corresponding to PF in the initial field	
DVF	Steric anomaly corresponding to PF in the leveled field	
PM,THM,SM,DVM	Average of pressure, potential temperature, salinity and steric anomaly over the regression interval.	
DH	$d\delta/dp$ based on the averaged regression coefficients	
PE	Potential energy anomaly	Recommended that these not be used, but calculated in PEPLT
XPE	Horizontally average PE	
CP(8)	Pressure vs. steric anomaly coefficients	
Z1	Standard deviation of regression pressure estimate (Fofonoff and Bryden, 1975)	
CT(8)	Potential temperature vs. steric anomaly regression coefficients	
Z2	Standard deviation of regression temperature estimate	
F1,F2,F3	Steric volume minimum, maximum and average over regression interval	
XLTØ:	Latitude of origin: default is 40.0	
XLGØ:	Longitude of origin: default is -70.0	

p_f. The output is in binary (unformatted) files. The variables output are identified in Table 4. Some information at each level may be output to unit KLIST if ISSW (12) = -1, for purposes of checking. Header information is ouput to unit KLIST if ISSW (11) = -1. The input data scans are output to unit KLIST if ISSW (5) = -1 and the regression coefficients and residuals are output to unit KLIST if ISSW (10) = -1. If ISSW (6) = -1 statistics of the coefficients are printed on unit KLIST. The ratio of each coefficient to its standard deviation (see Fofonoff and Bryden, 1975, Appendix) is computed. For an infinite number of degrees of freedom, at 95% confidence that ratio should equal or exceed 1.96. The statistic which is listed is (a_i the coefficients):

$$\frac{a_i}{\text{std dev }(a_i)} .$$

When stations with subindex reference number (sequential number) ISW through JSW have been tested for data selection criteria and either been skipped or have gone through the regression calculation, COMPS returns control to PTENS.

SUMMARY - POTEN:PTENS: KBR = 1

Function: calls COMPS subroutine

ISW, JSW Options: ISW to JSW are the station reference numbers

Output device: data goes automatically to *.REG file if ISSW (13) = -1; other information output goes to unit KLIST, as requested by elements of ISSW

ISSW options:
(to unit KLIST)

3 = -1 Print out interpolated scans

5 = -1 Print out input data scans

6 = -1 Print out coefficient statistics

10 = -1 Print out regression coefficients for each scan

11 = -1 Print out header information

12 = -1 Print out selected data following regression

13 = -1 Data output to *.REG

2: Initializes data selection parameters described in Table 5

SUMMARY - POTEN:PTENS: KBR = 2

Function: Initialize data selection parameters

ISW, JSW Options: None

Output device: None

ISSW Options: None

3: Changes or lists regression and data selection parameters described in Table 5. The data selection parameters are straightforward. For the regression parameters the water column is divided into a maximum of nine sections, each of which may have a number of levels whose regression parameters are the same. The regression parameters consist of the total number of sections; in each section, the interval between leveled surfaces, the interval over which the regression is performed, the polynomial order, and start and end pressures for the section. All of these parameters are input using subroutine PARAM, which branch 3 calls. The prompts are (hopefully) self-explanatory. After parameters have been entered for all sections, PARAM translates them into internal parameters which control the way the program performs the regressions. These internal parameters are stored in arrays NPR and NSC. Since the arrays NPR and NSC are included in common stored to KPTCM.DAT, the user form parameters need be entered only once, until a change is required. The old parameters may be retrieved by responding 'NO' to the initial query in PTENS 'Initialize regression parameters?'. Stored common is written to KPTCM.DAT at the end of branch 3, so any changes in regression parameters will overwrite the most recent ones in KPTCM.DAT, provided branch 3 is completed. It is not possible to change only a single regression parameter; if a change is required, all the parameters must be re-entered. (This is because the internal parameters NPR and NSC have elements whose value depends upon parameters for more than

TABLE 5
POTEN Parameters: Branch KBR = 3

Parameter	Definition	Default if Initialized
ICON	Consecutive number	1 for first station. Increments with stations processed
KSW	Not used	1
A2	Number of standard deviations allowed for a regression point in p(δ) before flagging.	3.
A3	Number of standard deviations allowed for a regression point in T(p) and S(p) before interpolation	3.
WGT	Weight	1.
PDIFF	Interval (db) about P_f for averaging TØ,SØ,PØ	6.
DELP	Pressure series interval for input CTD data (db)	2.

REGRESSION Parameters -- as described in program prompts

Data selection parameters: windows such that data inside all windows is accessed; all other data skipped

IDAY1	:	Minimum Julian year day	0
IDAY2	:	Maximum Julian year day	365
JDO	:	Additive constant to actual day	0
XEMN	:	Minimum longitude	-180.0
XEMX	:	Maximum longitude	180.0
XMN	:	Minimum longitude	- 90.0
XNMX	:	Maximum latitude	90.0

one section. PARAM requires that parameters be input sequentially.) It is not necessary to understand how NPR and NSC work in order to run the program (that is the purpose of the PARAM subroutine); however, modifications of the program may require that the programmer know how these arrays function. A brief description is therefore presented here. The pressure p_f for each level is given by:

For IREC less than NPR(section #)

PF = NPR(section # + total number of sections) x (IREC - NPR(section # + 2 x total number of sections))

NPR (section # + total number of sections) contains the interval between pressure levels; NPR(section # + 2 x total number of sections) contains an index which allows the correct p_f to be determined, while NPR(section #) contains the level number at which the section commences. Some care should be taken to assure that the parameters input are consistent. Specifically, the first level of a new section must have a pressure p_f such that p_f is some integral multiple of the pressure interval between leveled surfaces in that section. The use of the total number of sections allows the program to treat NPR as a variable length two-dimensional array, even though it is in fact singly dimensioned. Subroutine PARAM adds an additional 'dummy' section below those input by the user to assure that COMPS does not continue below the desired depth. Thus, the total number of sections (NSECTION) will always be one greater than the number input by the user.

Array NSC contains the remainder of the parameters: start pressure in NSC(section#), polynomial order in NSC(NSECTION + section #), number of data scans in the regression interval in NSC(2*NSECTION + section #).

SUMMARY - POTEN:PTENS: KBR = 3

Function: Change or list regression and data selection parameters

ISW, JSW Options: ISW = 0: short list only

ISW = 1: full list

JSW: no options

Input device: unit KIN

Output device: unit KLIST

ISSW Options: None

4: Call AVRCP - averaging subroutine. The pressure and potential temperature coefficients from the regressions performed in COMPS are averaged horizontally, level by level. The average pressure polynomial at each p_f is set equal to p_f (corresponding to a mass conservation constraint between the initial and leveled fields) and the resultant polynomial is inverted to obtain the reference steric anomaly (δ_f) corresponding to that p_f. (See Bray and Fofonoff, 1981 for a more detailed discussion.)

The averaging is actually done in two 'passes' through the data, but a single call to AVRCP with ISSW(7) = 0 will automatically average and output new station data files based on the leveled field. (Data ouput occurs if ISSW(13) = -1, as in COMPS. The new files are called *.AVG with * as before the station identifier.) Information about the averaged pressure coefficients is output to unit KLIST if ISSW(12) = -1. Information about the averaged steric field is output to unit KLIST if ISSW(11) = -1.

The two averaging 'passes' may be accessed individually, and separately from the data ouput by setting ISSW(7) = -1 and entering KBR = 4, ISW = 1 for the first pass, KBR = 4, ISW = 2 for the second pass and KBR = 4, ISW = 3 to output the new station data files. However, since the second pass must be performed directly after the first, and the output directly after the averaging it is recommended that the automatic access be used (ISSW(7) = 0). If no output is desired, ISSW(13) should be set to 0.

SUMMARY - POTEN:PTENS: KBR = 4

Function: Call AVRCP averaging subroutine

ISW, JSW options: If ISSW(7) = -1 ISW = 1: First averaging pass
ISW = 2: Second averaging pass
ISW = 3: Output of data to *.AVG files if ISSW(13) = -1
If ISSW(7) = 0: ISW = 1: Averaging and output performed automatically.

Input files: *.REG

Output files, data: *.AVG

Output files, lists: unit KLIST

ISSW Options: ISSW(7) = -1: individual access of averaging passes
ISSW(11) = -1: List of averaged steric field on unit KLIST
ISSW(12) = -1: List of averaged pressure coefficients on unit KLIST
ISSW(13) = -1: Leveled field based data output to *.AVG files

5: Not used

6: Print data label. This label is input by the user in branch 3, and is carried in both the *.REG and *.AVG files as an identifier of the group of stations, the version of the POTEN run, etc. Its format is 13 A4 or a total of 52 characters. Branch 6 lists this label to unit KLIST.

<u>SUMMARY</u> - POTEN:PTENS: KBR = 6

Function: Write data label

ISW, JSW options: None

Output device: unit KLIST

ISSW options: None

7: Not used

8: Write header record to unit KLIST: Station label, position, origin, LTYPE, MHDR, ICON, ISHP, ICAST, JDAY, IPR, LPR. This is also done automatically in subroutine DATA when COMPS accesses the station, provided ISSW(11) = -1.

SUMMARY - POTEN:PTENS: KBR = 8

Function: Write station header information

ISW, JSW options: None

Output device: unit KLIST

ISSW options: None

9: Write *.REG or *.AVG single data record to unit KLIST. Of doubtful usefulness, this branch was part of the original program.

SUMMARY - POTEN:PTENS: KBR = 9

Function: Write single output data record to unit KLIST

ISW, JSW options: None

Output device: unit KLIST

ISSW options: None

10: Not used

11: Set the values of the ISSW array. One call allows up to 16 inputs. Each input consists of element number followed by a comma and the value to assign to that element. Whenever input is complete, if less than 16, the branch may be terminated with a /.

SUMMARY - POTEN:PTENS: KBR = 11

Function: Set ISSW array

ISW, JSW options: None

Output device: unit KTTX

Input device: unit KIN

ISSW options: None

12: Exit program. Program queries 'Exit program '. A YES response results in a FORTRAN stop statement execution. A NO response returns the PTENS branch prompt.

SUMMARY - POTEN:PTENS: KBR = 12

Function: Exit program

Input device: unit KIN

If a value of KBR greater than 12 or less than 0 is entered, the short documentaton is printed on the screen.

3. PEPLT Documentation

Like POTEN, PEPLT is accessed through a short main program, which initializes parameters as requested by the user, and then transfers control to a major subroutine, PEPLS. From PEPLS, the user may call subroutine TABLE, which plots and lists station by station, and subroutine AVRGS which computes and displays horizontally averaged quantities as a function of depth. Subroutine AVRGS has its own set of internal branches, one of which returns program control to PEPLS. Subroutine TABLE has no internal branches. As in POTEN, short documentation can be displayed on the screen while the program is running interactively, by typing Ø/ as a response to branch prompts in either PEPLS or AVRGS.

3a. Main Program: PEPLT

The main program queries 'Load in previously stored common?'. A YES response causes the elements of the binary array KPLCM.DAT to be read into common, beginning with the common element KTTX. a 'NO' response causes no action by the program. Control is then transferred to subroutine PEPLS.

3b. Branches - PEPLT

1: Calls subroutine TABLE. TABLE plots and lists station by station. It also outputs requested information in a format appropriate as input to objective mapping programs. The plot section of TABLE is designed to permit a number of stations to be plotted on the same frame, with the origin of each station within the larger frame. Examples are shown in Fig. 4. In Fig. 4a the buoyancy frequency N is plotted as a function of geographical position (relative to an origin at 37°N, 69.65°W), the coordinates of the frame; and, for each station, as a function of depth, where the station axes represent 0 to 3000 db vertically and -3 to 3 cph horizontally. This is accomplished by scaling the buoyancy frequency, and adding it to the X-coordinate (in

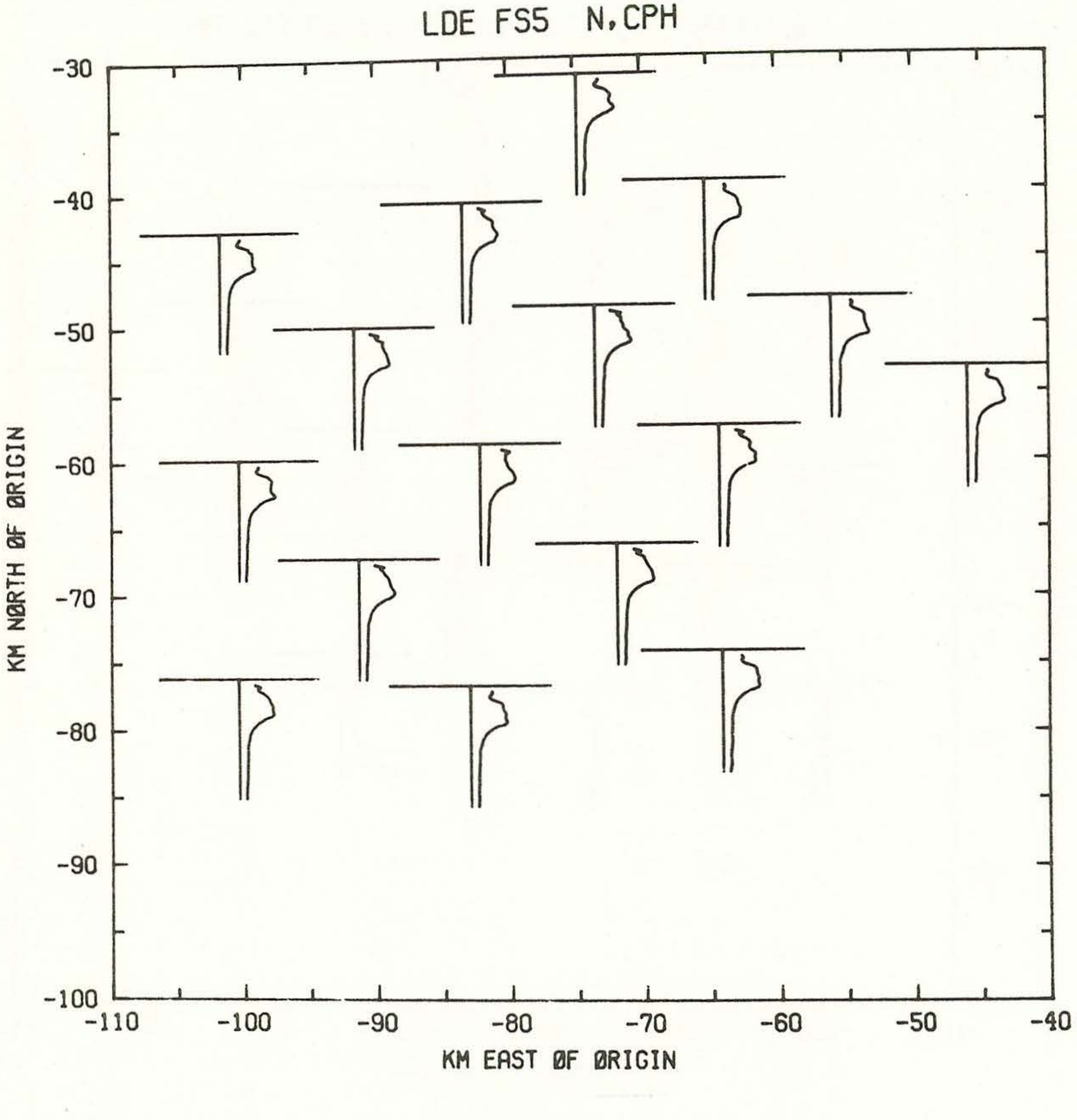
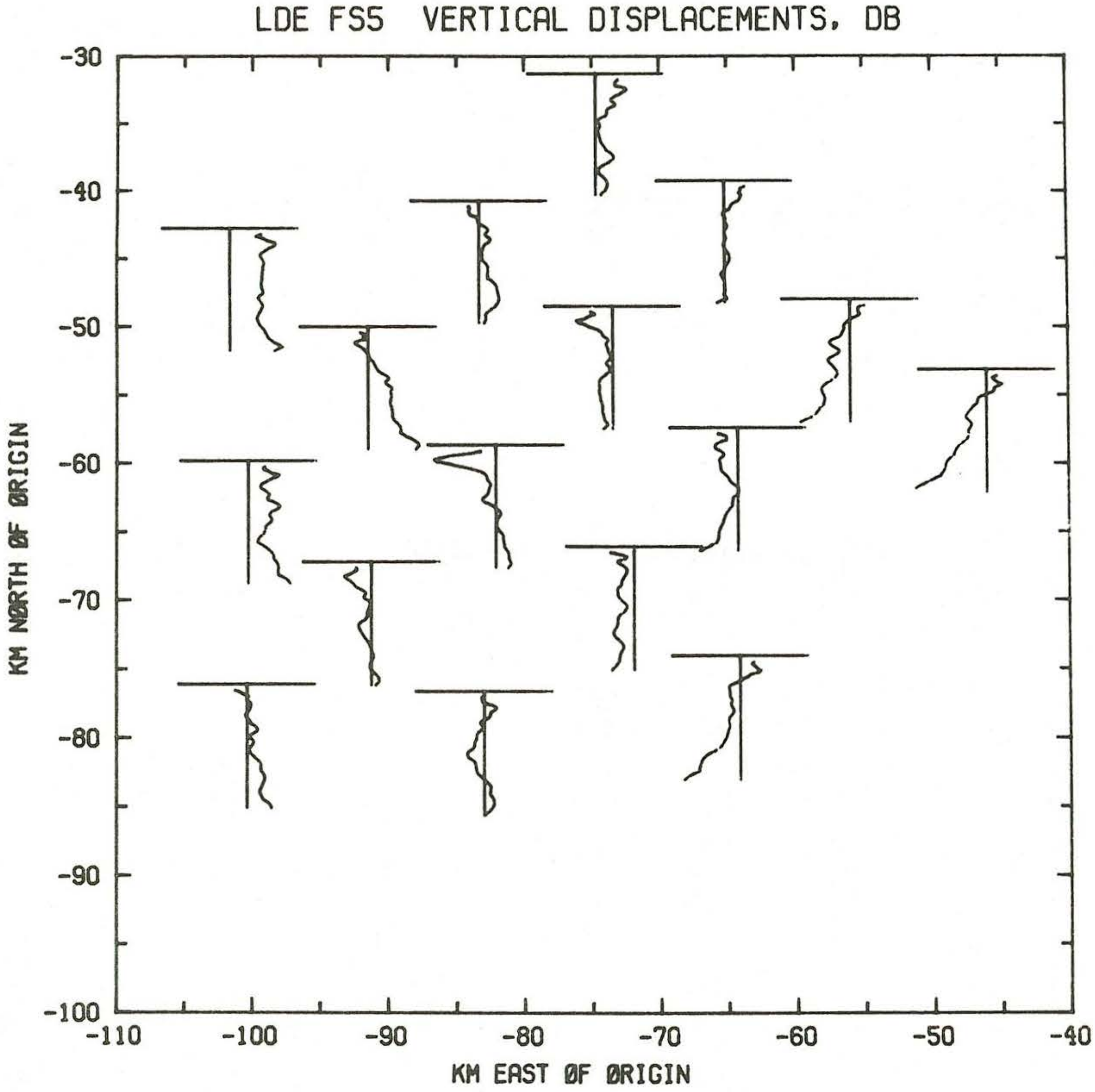


Figure 4: Example plots from TABLE.COM

4a. Buoyancy Frequency, N in cph. Inset axes represent ±3 cph (horizontal), 0 to 3000 db in pressure (vertical). The origin on the inset axes (0,0) represents the station location.



4b. Vertical displacement π in db. Inset axes represent ±100 db (horizontal) and 0 to 3000 db (vertical).

this example, kilometers east of the origin) and scaling the pressure and subtracting from the Y-coordinate (here km north of the origin). The program allows up to four variables to be included in such a sum for X and Y. The length of the station axes are determined by the variables X2DIM and Y2DIM, which are input in user units (i.e. cph in Fig. 4a) for the variables being plotted. In the example, Y2DIM is 3000 (db) and X2DIM is 3 (cph). One frame is created for each call to TABLE; it will encompass ND station plots. The plot parameters may be initialized by calling TABLE (KBR = 1) with ISW $\geq$ 2. Control then returns to PEPLS. Plot parameters may be changed by calling TABLE with ISW = 1. Control again returns to PEPLS. Plotting commences only when TABLE is called with ISW = 0. Figure 4b is the same type of plot as 4a, with vertical displacements plotted instead of N. Both of these plots were created using the documented command file TABLE.COM found in Appendix A.

A number of variables relating to the leveled field, the inital field, and the location, and time of each station may be examined using PEPLT. A list of these variables is found in Table 6; they are computed in function subroutine VRBL, coded by number. Thus a call to VRBL (3) returns the latitude of the station being examined (variable XLAT). Subroutine TABLE plots the following for x and y:

```
                    x = A1*VRBL(NX1) + A2*VRBL(NX2) + A3*VRBL(NX3)
                                     + A4*C(IREC,1)
                    y = B1*VRBL(NY1) + B2*VRBL(NY2) + B3*VRBL(NY3)
                                     + B4*C(IREC,2)
```

Here C(IREC,n) refers to an array which may be filled using AVRGS subroutine (see branch 4). A1 to A4, B1 to B4, NX1 to NX3, and NY1 to NY3 may be changed by accessing branch 3. The default values (initialized by KBR = 1, ISW = 2) are:

TABLE 6 -- PEPLT Variables
(Nomenclature follows that of Bray and Fofonoff, 1981)

VARIABLE NUMBER	NAME or SYMBOL	UNITS	DESCRIPTION
-1	1	None	Returns the number 1 (counts number of observations at each level).
Ø	Ø	None	Returns Ø
1	XPL	km	Zonal distance from origin (XLTØ,XLGØ)
2	YPL	km	Meridional distance from origin
3	XLAT	degrees	Signed decimal latitude (south negative)
4	XLONG	degrees	Signed decimal longitude (west negative)
5	ICON	None	Consecutive station number in POTEN computation
6	WGT	None	Averaging weight
7	JDAY	days	Julian year day
8	ISHP	None	Ship code
9	ICAST	None	Station number
10	N	None	Polynomial order
11	NDP	None	Number of data scans in regression interval
12	PF	db	Level pressure
13	TØ	°C	T, S, δ averaged over the interval PF ± PDIFF
14	SØ	ppt	
15	DVØ	$10^{-5}cm^3 \cdot gm^{-1}$	
16	PI	db	Pressure of reference steric anomaly (δ_f) in the initial field

TABLE 6 (Continued)

VARIABLE NUMBER	NAME or SYMBOL	UNITS	DESCRIPTION
17	THF	°C	Local potential temperature referred to PF. (See 65.)
18	DVI	$10^{-5}cm^3\cdot gm^{-1}$	Initial steric anomaly δ_i on PF
19	DVF	$10^{-5}cm^3\cdot gm^{-1}$	Reference steric anomaly (δ_f) on PF
20	PM	db	Pressure, local θ, salinity and steric anomaly referred to P_f averaged over regression interval
21	THM	°C	
22	SM	ppt	
23	DVM	$10^{-5}cm^3\cdot gm^{-1}$	
24	$\alpha_{p_f}{}^*$	$10^{-5}cm^3\cdot gm^{-1}db^{-1}$	$d\delta/dp$ based on averaged regression coefficients
25	χ		Potential energy anomaly calculated by POTEN
26	APE		APE calculated by POTEN
27 to 34	CP(1) to CP(8)	$(db)^{-i}$	Regression coefficients for pressure
35	Z1	db	Standard deviation of pressure regression estimate
36 to 43	CT(1) to CP(8)	$(°C)^{-i}$	Regression coefficients for local potential temperature
44	Z2	°C	Standard deviation of local potential temperature regression estimate
46,47,48	F1,F2,F3	$10^{-5}cm^3\cdot gm^{-1}$	Minimum, maximum and average values of steric anomaly over regression interval

TABLE 6 (Continued)

VARIABLE	NAME or	UNITS	DESCRIPTION
NUMBER	SYMBOL		
48	π^*	db	'Boussinesq' displacements: $\pi^* = -(\delta_i - \bar{\delta}_i)/(\frac{d\delta}{dP})^*$
49	PI*	db	$\pi^* + PF$
50	π	db	Displacement of initial field from reference field PI–PF. Positive implies downward.
51	$\Delta\delta$	$10^{-5}cm^3 \cdot gm^{-1}$	$\delta_i - \delta_f$
52	Vortex stretching	db	PIX $\frac{Sin(XLAT)}{Sin(XLTØ)}$
53	$\frac{APE^*_B}{g}$		$\frac{1}{g}$ 'Boussinesq' APE with true displacements $-\frac{1}{2g}\alpha^*_p\pi^2$
54			gPFΔD – PE
55	$\frac{APE_B}{g}$		$\frac{1}{g}$ Boussinesq APE with Boussinesq displacements $-\frac{1}{2g}\alpha^*_p\pi^{*2}$
56	EØ	$(10^{-5}cm^3 \cdot gm^{-1})^{-1}db$	Inverse of local specific volume gradient with pressure $(\frac{dp}{d\delta})$
57	1/EØ	$(10^{-5}cm^3 \cdot gm^{-1})db^{-1}$	Local specific volume gradient with pressure
58	N^2	$10^{-6}(rad \cdot sec^{-1})^2$	Squared buoyancy frequency
59	θ_p	°C db^{-1}	Potential temperature gradient $\frac{d\theta_f}{dp}$

TABLE 6 (Continued)

VARIABLE NUMBER	NAME or SYMBOL	UNITS	DESCRIPTION
60	S_p	ppt db^{-1}	Salinity gradient $\frac{dS_f}{dp}$
61	S_f	ppt	Salinity corresponding to θ_f, p_f, δ_f
62	$\frac{dS_f}{d\theta_f}$	$ppt(^{\circ}C)^{-1}$	Gradient of salinity with potential temperature
63	π^2	$(db)^2$	Squared displacement of initial field from reference field
64	N	cph	Buoyancy frequency
65	θ_f	°C	Potential temperature at δ_f referred to zero pressure
66	σ_θ	$10^{-3}gm\text{-}cm^{-3}$	Sigma theta of p_f, S_f, θ_f
67	α_f	$cm^3 \cdot gm^{-1}$	Specific volume anomaly in reference field
68	$-\frac{1}{2}\ \Gamma_k \pi^2$		Vertical gradient of compressibility contribution to GPE
69	Not used		
70	Γ_k	$10^{-5}cm^3\text{-}\ gm^{-1} \cdot db^2$	$\frac{d\kappa}{dp} - (\frac{\partial\kappa}{\partial p})_a$ (see Bray and Fofonoff, 1981)
71	Not used		

TABLE 6 (Continued)

VARIABLE NUMBER	NAME or SYMBOL	UNITS	DESCRIPTION
72	ΔS_f	ppt	Salinity anomaly from cubic spline fit to Worthington-Metcalf and Iselin θ-s curves $\Delta S(p_f,\theta_f,S_f)$ (Armi and Bray, 1981)
73	ΔS_i	ppt	Same as 72 but using the initial field $\Delta S(P_i,\theta(T\emptyset),S(\emptyset))$
74	$\theta_f{}^2$	$(^{\circ}C)^2$	Leveled field potential temperature squared. (Used in calculating horizontal standard deviation using AVRGS branch ISW = 21.)
75	RME	$(10^{-4}\ J\cdot kg^{-1})^2$	Random measurement error (based on pressure error of ±5db magnitude, temperature error of .007°C, salinity error of .005 ppt) for APE. See Bray and Fofonoff, 1981, Appendix, for details of error calculations. $V(\alpha_p{*}\pi^2/2)$
76	Not used		
77	RFE	$(10^{-4}\ J\cdot kg^{-1})^2$	Same as 78 but π
78	RFE*	$(10^{-4} J\cdot kg^{-1})^2$	Random finestructure error: (based on 3xZ1 as error in $\pi{*}$): $V(\alpha_p{*}\pi{*}^2/2)$

TABLE 6 (Continued)

VARIABLE NUMBER	NAME or SYMBOL	UNITS	DESCRIPTION
79	RME	$(10^{-4}\ J\cdot kg^{-1})^2$	Same as 75 but pressure error only
80	RFEC	$(10^{-4}\ J\cdot kg^{-1})^2$	Random finestructure errors in the vertical compressibility term (must be integrated using AVRGS branch ISW = 17) $V(\ \Gamma_k\pi^2/2$
81	RMEC	$(10^{-4}\ J\cdot kg^{-1})^2$	Random measurement error in the vertical compressibility term (pressure error only)
82,83	Not used		
84	κ	$10^{-5}cm^3 - gm^{-1}\cdot db^{-1}$	Compressibility $(\frac{\partial\delta}{\partial p})_a$
85	κ_S	$10^{-5}cm^3 - gm^{-1}\cdot ppt^{-1}$	Derivative of specific volume with respect to salinity; temperature and pressure held constant: $(\frac{\partial\delta}{\partial S})\ P,T$
86	$\kappa\pi$		Contribution to GPE from horizontal gradients of compressibility
87	APE_B	$10^{-4}\ J\cdot kg^{-1}$	'Boussinesq' APE per unit mass with true displacements
88-90	Not used		
91	$(\delta_i - \bar{\delta}_i)$	$10^{-5}cm^3\cdot gm^{-1}$	
92	$(-\Theta_p\pi)^{-1}$	$(^\circ C)^{-1}$	
93	Not used		

TABLE 6 (Continued)

VARIABLE NUMBER	NAME or SYMBOL	UNITS	DESCRIPTION
94	θ_i^2		Potential temperature corresponding to p_f in initial field squared
95	$-(\theta_i - \bar{\theta}_i)/\theta_p \pi$	None	
	(if $\bar{\theta}_i$ is in column 4)		
96	$-(\theta_f - \bar{\theta}_f)/\theta_p \pi$	None	
	(if $\bar{\theta}_f$ is in column 4)		
97	θ_i	°C	Local potential temperature at δ_i
98	Not used		

A1 = 1	B1 = 1	NX1 = 1	NY1 = 2
A2 = 2	B2 = -.003	NX2 = 64	NY2 = 12
A3 = 0	B3 = 0	NX3 = 0	NY3 = 0
A4 = 0	B4 = 0		

These values will cause the buoyancy frequency in cph to be plotted as a function of meridional position (y-axis), time (x-axis) and pressure (station axis). X2DIM defaults to 3 (cph) and Y2DIM to 3000 (db), resulting in station axes representing ±3 cph for the displacements and 0 to 3000 db for the pressure. The default number of stations (variable ND) is 1 and may be changed by calling KBR = 1, ISW = 1. The plot information is stored in the file corresponding to unit 8. It must be read and translated by a Metacode translator. PEPLT may be run on any terminal, but the translators are only available for graphics terminals and the Calcomp plotter. See the last part of this section for instructions on the access of the translators. The origin co-ordinates may be changed in PEPLS branch 2.

Branch 1 with ISW = 0 may be used to change PMIN and PMAX, thereby selecting a range in pressure over which data will be used (all other data is excluded), X2DIM and Y2DIM, described above, JMIN, the level number corresponding to the pressure at which the plot is to start (this allows the user to skip over shallow points which may have anomalous values), and various plot parameters. The plot parameters include PLABL, the overall plot label; XMIN, XMAX, YMIN, YMAX, the axis limits; XLABL and YLABL the x and y-axis labels, respectively.

In addition to plots, if ISSW(10) = -1 TABLE outputs to unit KOUT the following list of variables in format (GF8.3):

PF, XPL, YPL, (VRBL(NV(K)),K = 1,6).

(See Table 6 for descriptions of these variables.)

If ISSW(12) = -1, a short list of variables is output to unit KLIST: pressure (PF), and the variables x, y and z, z given by

```
z = C1*VRBL(NZ1) + C2*VRBL(NZ2) + C3*VRBL(NZ3) + C4*C(IREC,3)
```

SUMMARY - PEPLT:PEPLS: KBR = 1

Function: Call subroutine TABLE - multiple station plots, map format output, lists by station.

ISW, JSW options: ISW = 2 Initialize plot parameters

ISW = 1 Change plot, map format and list parameters

ISW = 0 Plot, list, map format output

JSW No options

ISSW options: ISSW(5) = -1 No interior axes on plot

ISSW(6) = -1 No plot

ISSW(10) = -1 List variables

ISSW(12) = -1 List p, x, y, z.

2: Change data selection variables. Calls subroutine to change time and space windows and origin co-ordinates.

3: Change plot and list parameters. This branch prints a short documentation on the screen each time it is called. Parameters which may be changed and their descriptions are listed in Table 7. This branch has internal branches 1 through 8, which are prompted by '**: PARAMETERS: KBR3, ISW3, KX, MV, MW'. Only KBR3 and ISW3 have any effect in this branch. KX is the total number of parameter input branches (5). To return to PEPLS from branch 3 the user must enter KBR3 = 1, ISW3 = 0 followed by /. This will cause the new parameter values to be written on unit KLIST, and stored common to be written to KPLCM.DAT.

SUMMARY - PEPLT:PEPLS: KBR = 3

Function: Change or list plot and listing parameters

ISW, JSW options: None

Input device: KIN

Output device: KLIST

ISSW options: None

TABLE 7

PEPLT: PEPLS Branch 3 Parameters

VARIABLE	DEFAULT	DESCRIPTION
NX1	12	Variable codes for VRBL used in AVRGS and TABLE computations
NX2	0	
NX3	0	
NY1	19	
NY2	0	
NY3	0	
NZ1	25	
NZ2	0	
NZ3	0	
A1 B1 C1	1.	Scaling factors used in AVRGS and TABLE computations
A2-A6, B2-B6, C2-C6	0	
D1 to D6	1.	
TMIN to YT	None	Not used
SMIN to ST	None	Not used

4: Calls AVRGS subroutine. This subroutine calculates horizontal averages, allows operations such as vertical integration and column addition, multiplication, exponentiation and division. There are 23 internal branches in AVRGS, accessed with different values of ISW(0 to 22). These internal branches are described below, with a summary at the end of each. As an overview, AVRGS reads the requested data from *.AVG files into a two-dimensional array C(100,6). The rows (1 to $\leq$ 100) correspond to the pressure levels and the columns to variables requested by the user and computed in function subroutine VRBL (see PEPLS branch 1 for a description of VRBL). As each successive station is read, the elements of C are added to, forming sums of all data available at all levels. These sums must then be divided by the total number of observations at each level, to obtain the average values. For reasons of flexibility, the reading/summing and division are performed in separate ISW branches within AVRGS. Once the array C is filled (one column of which must be the number of observations) and averaged, then a number of operations can be performed on the averages. The remaining ISW branches of AVRGS are devoted to these operations.

AVRGS has its own prompt 'AVRGS:KBR,ISW,JSW,KLIST', and control does not return to PEPLS unless KBR = 4 ISW = 12 is accessed. Therefore, only four variables (or < 4 followed by a /) need be input following the AVRGS prompt. In order to keep track of the operations performed in AVRGS, if ISSW(2) = -1 the four parameters are written to unit 4 each time an AVRGS branch is accessed, along with other pertinent information. This ISSW option will not be noted in the summaries.

Branches in AVRGS: (ISW)

ISW = 0: Prints short documentation on unit KTTX

ISW = 1: Reads station data into C array. Variables corresponding to NV(JSW) to NV(KLIST) (maximum of six) are read into columns

JSW to KLIST of array C for ND number of stations from file JSHP.PTN (logical unit 12), starting with the first station in that file. All data between PMIN and PMAX is accessed for each station. The array C is stored to KPLCM.DAT before returning to the AVRGS prompt. If ISSW(15) = -1, the weights (WT) from JSHP.PTN file are used; otherwise a weight of 1. is used. Each element of C is a sum of

```
C(IREC,I) = C(IREC,I) + D(I)*WT*(AV*VRBL(NV(I)))
                + (BV+CV*VRBL(NV(I))*VRBL(NX(I)))
```

The default parameters are set such that

```
C(IREC,I) = C(IREC,I) + WT*VRBL(NV(I))
```

Some of the parameters used by this branch may be changed in branch 3 of AVRGS, and some in branch 3 of PEPLS.

<u>SUMMARY</u> - PEPLT:AVRGS: KBR = 4: ISW = 1

Function: Read and store data to C array

JSW, KLIST options: JSW is first column, KLIST last column

Output device: Array is stored to KPLCM.DAT for emergency retrieval. No other output.

ISSW options: None

ISW = 2: Zeros columns JSW to KLIST of array C

ISW = 3: Changes or lists parameters. Parameters involved are listed in Table 8. JSW = 1 initializes the parameters (defaults also in Table 9) before allowing changes; JSW = 0 retains previous values. (The first access to this branch must initialize.)

<u>SUMMARY</u> - PEPLT:AVRGS: KBR = 4: ISW = 3

Function: Change parameters

JSW, KLIST option: JSW = 1 initializes

JSW = 0 prints current values

Output device: KTTX

ISSW options: None

ISW = 4: Average table: divide columns JSW to KLIST by column 6, which should have the number of observations at each level.

TABLE 8

PEPLT: AVRGS Branch ISW = 3 Parameters

VARIABLE	DEFAULT (Initialized)	DESCRIPTION
ND	1	Number of stations to be processed
NV(1)	51	Variables to compute for C array as VRBL(NV(I)) in column I.
NV(2)	68	
NV(3)	86	See Table 6 for VRBL codes.
NV(4)	87	
NV(5)	63	
NV(6)	-1	
JREF	50	Number of levels to be calculated
JMAX	55	Level number corresponding to reference pressure for integrations over pressure
NX(I),I=1,6	0	Optional additive quantities in C array element calculation (see text).
A1	1.	
A2	0.	X and Y scaling factors for plots
A3	0.	
B1	1.	Initialized when PEPLS is called by responding YES to 'Initialize common '
B2	0.	
NX1	12	Optional plot parameters (see text for AVRGS branch ISW = 7).
NX2	0	
NY1	19	
NY2	0	Initialized in PEPLS as above

ISW = 5: Add column JSW vertically, starting from level 2 and going to JMAX:

C(IREC,JSW) = C(IREC - 1,JSW)

ISW = 6: List C array to unit KLIST. Inlcudes data label, parameters, level number and pressure, and C array.

ISW = 7: Plot one frame. Up to six curves allowed per frame. NCAR plot package outputs to unit 8 a file which must be read and translated into a plot by a Metacode translator. PEPLT may be run on any terminal, but the plot files may only be translated on graphics terminals and the Calcomp plotter. Instructions for running the translators are found at the end of this section of the report. The plot branch asks for the number of curves (default 1, maximum 6), the level number for the first point (default 1), the plot label, the minimum and maximum coordinates for x and y (unless the user opts to have the NCAR plot package compute the scales, by responding YES to the query 'Use default axis parameters?'), x and y axis labels, and the column number to be plotted. The program actually plots:

x = B1*C(J,JSW) + B2*C(J,NX2) + B3*PF

y = A1*PF + A2*C(J,NV1) + A3*C(J,NY2)

The default values of the parameters plots

C(J,JSW) vs PF (pressure).

However, if for example the user wished to plot potential temperature θ vs salinity S, with θ (VRBL(65)) in column 1 and S(VRBL(61)) in column 2, then the values of the above parameters should be changed (using AVRGS branch 3)

A1 = 0.	B1 = 1	NY1 = 1
A2 = 1.	B2 = 0	
A3 = 0.	B3 = 0.	

The y-axis runs backwards (maximum at the bottom to minimum at the top) unless A1 is equal to 0. An example is given in ENERGY.COM -- see Appendix A.

Characters of the user's choice which mark the actual data points may also be plotted if ISSW(5) = -1. Note should be made that these are not centered characters, so that the data point actually occurs wherever the plotter commences drawing the character.

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 7

Function: Plot one frame containing up to six curves.

JSW option: JSW is the column number to be plotted. It may be changed while in the plotting branch.

Output device: Plot information goes to Metacode file, unit 8.

ISSW options: ISSW(5) = -1 plots character to mark actual data points. Character is requested while in plot branch.

ISW = 8: Calculates gravitational available potential energy per unit mass (GPE) and per unit area (TGPE), from the horizontal averaged steric volume $\overline{\text{DVI}}$ (VRBL(18)) in column 1 and for the reference steric volume DVF (VRBL(19)) in column 2, except for a constant of integration. GPE and TGPE relative to some reference pressure are calculated by subtracting from GPE and TGPE at each level the value at the level corresponding to the desired reference pressure (denoted by level number JREF) in AVRGS branch ISW = 10. GPE is stored in column 1, TGPE in column 2. The units are 10^{-4} $J.kg^{-1}$ and 10^{+4} $J.m^{-2}$, respectively.

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 8

Function: Calculate GPE and TGPE except for a constant of integration

JSW options: None

Output device: None (GPE and TGPE replace $\overline{\text{DVI}}$ and DVF in columns 1 and 2, respectively, of array C.)

ISSW options: None

ISW = 9: Integrate over pressure columns JSW to KLIST. This is an alternate method for calculating GPE and subsequently TGPE, with $\overline{\text{DVI}}$ - DVF (VRBL(51)) in column JSW. It may also be used to compute the compressibility effects in the GPE calculation (see equation 28 in Bray and Fofonoff, 1981).

The integration is performed starting with the first element in the column, and continuing to the last; the reference value must be subtracted in a separate operation, using AVRGS branch ISW = 10.

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 9

Function: Integration over pressure (except for a constant) of columns JSW to KLIST

JSW options: Columns JSW to KLIST are integrated

Output device: None

ISSW options: None

ISW = 10: Subtract value at reference pressure (level corresponding to JREF) from all other elements in columns JSW to KLIST

Output device: None

ISSW option: None

ISW = 11: Add up to four scaled columns, according to

```
J = IREC
C(J,JC1) = CR1*C(J,JC1) + CR2*C(J,JC2) + CR3*C(J,JC3)
                + CR4*C(JREF,JC4)
```

If JSW = 1, JC1, CR1 to JC4, CR4 are entered; no addition is performed.

If JSW = 0, addition is performed using most recently input parameters.

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 11

Function: Add up to four scaled columns, row by row

JSW Option: 0: perform addition
1: input scaling and column parameters

Output device: None

ISSW option: None

ISW = 12: Return to PEPLS

ISW = 13: Multiply up to three scaled columns, row by row according to C(IREC,I) = CON1*C(REC,I)*{CON2*C(IREC,J)*[CON3*C(IREC,K)]}

If I = -1 no operation is performed.

If J = -1 then the expression in {} is set to 1; if K = -1, the expression in [ ] is set to one, allowing one, two or three scaled columns to be multiplied together. The parameters may be changed when the branch is accessed. The default values are I,J,K = -1; CON1, CON2, CON3 = 1.

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 13

Function: Multiply up to three columns, row by row

JSW options: None

Output device: None

ISSW options: None

ISW = 14: Output in map format to unit KTO. Branch requests output file name and level number (JREC) desired. Variables output are:

IDSTN (station identifier: ship, station), XLAT, XLONG, (VRBL(NV(K)), K = 1,3), (C(JREC,K),K = 4,5) in format (1H ,A5,2(F8.2),5F(8.3)).

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 14

Function: Output in map format

JSW option: None

Output device: Unit KTO (may be changed in branch; default is 60)

ISSW options: None

ISW = 15: Not used

ISW = 16: Take any single column to any power, row by row. Operations are performed on the absolute value of all elements. If JSW = 1, exponent and column inputs are prompted. If JSW = 0, exponentiation is performed. The call to JSW = 0 should immediately follow that to JSW = 1, as the variables used for exponent and column number are not unique to this branch.

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 16

Function: Exponentiation of a single column

JSW options:	JSW = 0:	Operation performed
	JSW = 1:	Exponent and column entered
Output device:	None	
ISSW options:	None	

ISW = 17: Integration of error terms: interval pressure squared as the integration variable. This is intended for the calculation of measurement and finestructure errors in GPE and TGPE; as such it may be used on columns containing averaged values of VRBL (75 and 77 through 81) -- see Table 6. This branch uses the same algorithm as AVRGS branch ISW = 9, with ΔP^2 instead of ΔP as the integration variable. See AVRGS branch ISW = 9 for a summary.

ISW = 18: Writes into column 5 the difference in pressure between each pair of levels, beginning at the top.

ISW = 19: Exchange columns JSW and KLIST.

ISW = 20: Input a new single element of C. Branch prompts for column and row of element to be changed.

ISW = 21: Compute the standard deviation and store in column 1 of any quantity X for which $\overline{X}$ (the average value) is stored in column 4 and $\overline{X}^2$ in column 3.

ISW = 22: Compute the dynamic height for each station at any range of levels referred to level JREF and output in map format. Branch prompts for output device (default is 60), and level numbers (JREC1, JREC2) for dynamic height calculation. Reference level JREF may be changed in AVRGS branch ISW = 3. To calculate dynamic height NV(1) must be 18, NV(2) 19. Variables output are:

IDSTN (station identifier), XLAT, XLONG, Dynamic height (in dynamic centimeters), (NV(K),K = 3,6).

Output occurs for ND stations, beginning with the first station in JSHP.PTN (unit 12).

SUMMARY - PEPLT:AVRGS: KBR = 4, ISW = 22

Function: Compute dynamic height relative to JREF for any range of pressure, for each of ND stations and output in map format. Four optional variables are also output, for the same range of pressure. An example command file, DYNHT.COM is found in Appendix A.

ISW options:	None
Output device:	Unit KTO (default 60; may be changed by the user when the branch is accessed).
ISSW options:	None

PEPLT Branches (KBR), continued

5: Set values of elements in the ISSW array. Up to 16 inputs are allowed, each consisting of the element number followed by the element value (-1 or Ø). Terminate before 16 by typing /.

6: Restart main program.

7: Exit program: a YES response to the branch query 'EXIT PROGRAM ' results in the execution of a FORTRAN stop. A NO response returns the PEPLS prompt.

Metacode Translators

The translators for the plot files (written to unit 8) created in AVRGS (branch ISW = 7) and TABLE (PEPLS branch KBR = 1) are device specific. That is, each graphics terminal has its own version. The CALCOMP high speed plotter has two versions: one with default plotting parameters, and one which allows the user to enlarge or stretch the plots, alter their distribution on the plotter paper, etc. The IMLAC and Tektronix terminals also have versions of the translator to allow plot previewing.

For all translators:

If the plot file was written to any other file than that named FOR008.DAT (via an ASSIGN statement before running PEPLT) then you must assign that ouput file name to unit 8 before running the translators. For example, if your plot file is named PLOT.PPT, you must make the following assignment:

```
ASSIGN PLOT.PPT FOR008.
```

For the CALCOMP (both versions) you must also assign terminal TTA4: to FOR061:

```
ASSIGN TTA4:  FOR061
```

Then

```
RUN MCTRNPLOT    (for MetaCode TRaNslator PLOT)
```

plots with default parameters, and

```
RUN MCTRNPLT2
```

prompts the user for changes in the plotting parameters before executing the plots. MCTRNPLT2 asks three questions: first, how many plots in the y-direction (across plotter)? The default is 1, and is retained if a / is entered. Second, what size shall the plots be? The default is 10 by 10 inches. The new dimensions are entered in inches, and need not be equal for x and y. Again a / retains the default values. Finally, the program asks for the distance between plots, in inches. The default is 2 inches in both x and y. All plots in the file assigned to unit 8 are plotted, sequentially.

For the Tektronix (or the IMLAC in Tektronix mode):

```
RUN MCTRNTEK
```

starts the plot previewer. If there is more than one plot, the program prompts for continuing to the next plot by asking 'Option ?' to which the user should respond C for continue, until all plots in the file assigned to unit 8 have been plotted.

For the IMLAC (recommended over the IMLAC in Tektronix mode, since it is simpler, and uses more of the screen):

```
RUN MCTRNDYN1
```

starts the plot previewer. This program also prompts for continuation if there is more than one plot.

This translator information is accurate as of December 1980. If you encounter difficulties you should refer to the current VAX manual.

4. Modification of POTEN to accept input CTD data in other than CTD78 disc format.

This section is intended as a guide to assist users who wish to use POTEN on CTD data with formats other than that read by the standard version. In this section the header information required by POTEN is described in detail, and the procedure for reading data is explained. The only subroutine which must be changed is DATA, providing that the input data is an even series in pressure with no gaps.

DATA requires the following header information for each station:

Description	Variable Name	Format
Ship Name	ISHP	A2
Cruise	ICRUIS	A3
Station	ISTAS	I3
Decimal Latitude (south negative)	XLAT	F
Decimal Longitude (west negative)	XLONG	F
Day	IDA	I2
Month	IMO	I2
Year (last two digits)	IYR	I2
Time (24 hour clock)	ISTME	I4
Station Label	LBBL(3)	3A4
Minimum Pressure	PMIN or IPR	F or I
Maximum Pressure	LPR	I

The CTDATA library subroutines not needed for formats different from the disc version of CTD78 are:

PVER	
CRUISE	Header Information
STATION	
DATIDX	Data Retrieval
GETDAT	

Also, the common file IDXREC.DIM should not be included in DATA -- see the statement INCLUDE 'IDXREC.DIM'. The variable LLREC is the total number of stations in the subindex directory; all statements in DATA and COMPS which refer to LLREC may be deleted. The data are stored in arrays PRESS and DATAX.

Pressure is stored in PRESS,(#), temperature in DATAX(1, #), salinity in DATAX(2, #) with # the data scan number. Subroutine DATA must fill DATAX and PRESS (all scans) when it is called for each station. Finally, DATA must return to COMPS the total number of data scans, JRMAX.

Stations are selected by the call to DATA in COMPS. The call is

CALL DATA (KST,1)

In COMPS, KST is the sequential number in the DO loop from ISW to JSW in branch 1 (or 13). If the input data is on magnetic tape, the user may wish to change the DO loop in COMPS to go from 1 to JSW: that is, start at the beginning of the tape and read through ISW stations.

The section of DATA in which the ship and cruise specification may be changed (NSW = 2) can be readily modified to accept similar information (in branch 13) pertinent to the user's input data.

The header information should be read in following statement # 5, replacing the statements between # 5 and # 54. The data should be read in in statements which replace the calls to DATIDX and GETDAT.

Acknowledgements

Nick Fofonoff was responsible for early versions of most of the programs and subroutines documented here; his contribution to this work is gratefully acknowledged. Jerry Needell, Dan Georgi and Marie-Noelle Houssais used these programs, discovered errors, and suggested improvements. The manuscript was improved by constructive criticism from Bac-Lien Hua, and was typed by Mary Ann Lucas and Audrey Williams.

This work was supported by the Office of Naval Research under contracts N00014-76-C-197, NR083-400 and N00014-79-C-0071, NR083-004 and by NSF grant OCE-77-19403.

References

Armi, L., and N. A. Bray, 1981. A standard analytic curve of potential temperature vs salinity for the western North Atlantic. Deep-Sea Res., submitted.

Bray, N. A., and N. P. Fofonoff, 1981. Available potential energy for MODE eddies. J. Phys. Oceanogr., 11(1), 30-47.

Bryden, H. L., 1973. New polynomials for thermal expansion, adiabatic temperature gradient and potential temperature of sea water. Deep-Sea Res., 20, 401-408.

Fofonoff, N. P., 1977. Computation of potential temperature of seawater for an arbitrary reference pressure. Deep-Sea Res., 24, 489-491.

Fofonoff, 1981. Algorithms for oceanographic computations to appear as a publication of SCOR Working Group 51.

Fofonoff, N. P., and H. L. Bryden, 1975. Specific gravity and density of seawater at atmospheric pressure. J. Mar. Res. (Suppl.), 33, 69-82.

Millard, R. C., Blumer, A., and N. Galbraith, 1978. A digital tape format for Woods Hole Institution CTD Data. W.H.O.I. Technical Report 78-43.

Millero, F. J., Chen, C.-T., Bradshaw, A., and K. Schleicher, 1980. A new high pressure equation of state for seawater. Deep-Sea Res., 27A, 255-264.

Appendix A
Example command files for different routine calculations using POTEN and PEPLT.

In this appendix documented command files which perform various routine calculations are listed. In the order in which they appear they are: ENERGY.COM, POTEN.COM, PEPLT.COM, TABLE.COM, and DYNHT.COM. Brief descriptions of these files are also found in Table 2. The files themselves contain detailed documentation. Example plots from ENERGY.COM are also included: see Figures 5 to 12. Example plots from TABLE.COM are found in Figure 4.

```
   5   $!  ************** ENERGY.COM **************
  10   $!COMMAND FILE TO COMPUTE APE FROM CTD78 FORMAT DATA. CREATES
  18   $!STANDARD PRINTOUT AND PLOTS. INTENDED AS AN AID IN LEARNING
  26   $!TO USE THE VAX-11 PROGRAMS POTEN AND PEPLT.
  34   $!4 JULY 81.  NAN BRAY
  42   $!
  50   $SET DEFAULT DBA2:<316316.LDE>
 100   $ASSIGN JSHPF5.PTN FOR012  !FILE TO BE CREATED BY POTEN CONTAINING
 150   $!                          STATION IDENTIFIERS.
 200   $ASSIGN PRINT.PTN FOR004  !FILE FOR LINEPRINTER OUTPUT FROM POTEN
 300   RUN/NODEB POTEN
 400   YES INITIALIZE COMMON
 500   NO DO NOT INITIALIZE REGRESSION PARAMETERS
 600   2.5  !PRESSURE INTERVAL FOR INPUT DATA
 700   3,1,1,4/  !SET PARAMETERS FOR REGRESSION
 800   ,,,,,2./  !SET PDIFF TO 208--INTERVAL OVER WHICH TO,SO,DVO AVERAGED
 900   /  !DO NOT CHANGE DELP--PRESSURE INTERVAL FOR INPUT DATA
1000   NO DO NOT CHANGE REGRESSION PARAMETERS
1100   /  !LEAVE TIME WINDOW AT DEFAULT 0-365 DAYS
1200   /  !LEAVE EAST-WEST SPACE WINDOW AT DEFAULT: -180,180 DEG
1300   /  !LEAVE NORTH-SOUTH SPACE WINDOW AT DEFAULT: -90,90 DEG
1400   YES CHANGE DATA LABEL
1500     LDE FS5 TEST--STANDARD VERSION, NEW EOS--22 JULY 81
1600   11/  !SET ISSW ARRAY
1800   11,-1/  !LIST STATION INFORMATION TO UNIT KLIST (FILE PRINT.PTN)
1850   11/
1900   12,0/  !DO NOT LIST REGRESSION SUMMARY AT EACH DEPTH FOR EACH STATION
1950   11/
2000   13,-1/  !CREATE *.REG,*.AVG FILES
2050   11/
2100   5,0/  !DO NOT LIST INDIVIDUAL INPUT DATA SCANS
2150   11/
2200   10,0/   !DO NOT LIST REGRESSION COEFFICIENTS FOR EACH LEVEL
2500   1,229,236,4/!USING DEF CRUISE SPECS, REGRESS STA REF # ISW TO JSW
2660   13,201,208,4/ !CHANGE THE CRUISE SPECS, THEN AS IN PREVIOUS COMMAND
2675   W  !SUBDIRECTORY VERSION
2690   IS001003  !SHIP, CRUISE, PROJECT NUMBER
2900   11/  !RESET ISSW ARRAY
3000   12,-1/  !LIST AVERAGED REGRESSION COEFFICIENTS
3100   4,1,0,4/  !PROCEED THROUGH ENTIRE AVERAGING PROCESS. LISTS TO PRINT.PTN
3200   12/  !EXIT PROGRAM?
3300   YES
3400   $ASSIGN PRINT.PPT FOR004  !LISTING FILE FOR PEPLT OUTPUT
3500   $ASSIGN PLOT.PPT FOR008   !NCAR PLOT FILE FOR PEPLT PLOTS
3600   RUN/NODEB PEPLT
3700   NO DO NOT READ IN PREVIOUSLY STORED COMMON
3800   YES INITIALIZE DATA SELECTION PARAMETERS
3900   5/  !SET ISSW ARRAY
4000   2,-1/   !LIST OPERATIONS PERFORMED, IN PRINT.PPT
4050   5/
4100   5,-1/   !PLOT CHARACTERS ON PLOTS FOR IDENTIFICATION
4200   4,2,1,6/  !ZERO C ARRAY IN AVRGS. PROGRAM CONTROL NOW IN AVRGS.
4300   4,3,1/  !SET VARIABLE SELECTION PARAMETERS
4400   1000,,,,,,,50/       !CHANGE 1000 TO # OF STATIONS IF ND < ALL
4500   /  !DO NOT CHANGE AV THROUGH NX(I)
4600   /  !DO NOT CHANGE A1 THROUGH B3
4700   /  !DO NOT CHANGE NX1 THROUGH NY2
4800   4,1,1,6  !READ VARIABLES AS SELECTED INTO C ARRAY.
4900   4,4,1,5  !DIVIDE COLUMNS 1-5 BY 6(NUMBER OF STATIONS) TO AVERAGE
5000   4,6,1,4  !WRITE C ARRAY TO PRINT.PPT
5100   4,9,1,3  !INTEGRATE COLUMNS 1-3 WITH RESPECT TO PRESSURE
```

```
 5200   4,10,1,3  !SUBTRACT FROM ALL LEVELS THE VALUE AT LEVEL JREF
 5325   4,6,1,4  !WRITE C ARRAY TO PRINT.PPT
 5400   4,11,0/  !SUBTRACT FROM COL 1 COLS 2,3; ADD VALUE AT JREF FROM COL 4
 5500   4,6,1,4  !WRITE C ARRAY TO PRINT.PPT
 5600   4,11,1/  !RESET ADDITIVE CONSTANTS
 5700   1,1.,2,1.,3,1.,4,0.  !REPLACE COMPRESSIBILITY TERMS
 5800   4,11,0/
 5900   4,16,1/
 6000   .5,5  !TAKE COLUMN 5 TO THE POWER .5
 6100   4,16,0/
 6200   4,6,1,4  !WRITE C ARRAY TO PRINT.PPT
 6300   4,7/  !CALL PLOT BRANCH
 6400   4,3  !4 PLOTS IN THIS FRAME, STARTING AT LEVEL 3 ON EACH
 6500   YES INPUT NEW PLOT LABEL
 6600   LDE FS5  NEW EOS--22 JULY 81
 6700   NO DO NO USE DEFAULT AXIS PARAMETERS
 6800   -20,200,0,3000  !XMIN,XMAX,YMIN,YMAX
 6900   YES CHANGE X-AXIS LABEL
 7000   APE (CM/SEC)**2
 7100   YES CHANGE Y-AXIS LABEL
 7200   PRESSURE, DB
 7300   1  !PLOT COL 1
 7400   *  !* IS PLOT CHARACTER IDENTIFIER (NOT CENTERED!)
 7500   2  !PLOT COL 2
 7600   +
 7700   3  !PLOT COL 3
 7800   O
 7900   4  !PLOT COL 4
 8000   X
 8100   4,7/  !CALL PLOT BRANCH FOR NEXT PLOT
 8200   1,1/  !I PLOT IN THIS FRAME; STARTING AT LEVEL 1
 8300   NO DO NOT CHANGE PLOT LABEL
 8400   NO DO NOT USE DEFAULT AXIS PARAMETERS
 8500   0,100,0,3000
 8600   YES CHANGE X-AXIS LABEL
 8700   RMS DISPLACEMENTS, DB
 8800   NO DO NOT CHANGE Y-AXIS LABEL
 8900   5  !PLOT COL 5
 9000   +
 9100   4,2,1,5  !ZERO COLUMNS 1-5 OF C ARRAY
 9200   4,3,0/  !RESET SELECTED VARIABLE PARAMETERS, LEAVING OTHERS AS BEFORE
 9300   ,50,61,65,64,19/
 9400   /
 9500   /
 9600   /
 9700   4,1,1,5 !READ VARIABLES INTO COLUMNS 1-5;START AT TOP OF JSHP.PTN LIS
 9800   4,4,1,5  !DIVIDE COLUMNS 1-5 BY NUMBER OF STATIONS
 9900   4,6,1,4  !WRITE C ARRAY TO PRINT.PPT
10000   4,12/  !RETURN CONTROL TO PEPLS
10100   5/  !SET ISSW ARRAY
10200   5,0/  !NO CHARACTERS TO IDENTIFY PLOTS
10300   4,7/  !CALL PLOT BRANCH FOR NEXT FRAME; SEE EARLIER DESCRIPTION
10400   1,1/
10500   NO
10600   NO
10700   34.8,36.8,0,3000
10800   YE
10900   SALINITY, PPT
11000   NO
11100   2
11200   4,7/
```

```
11300   1,1/
11400   NO
11500   NO
11600   2,22,0,3000
11700   YE
11800   POTENTIAL TEMPERATURE, DEG C
11900   NO
12000   3
12100   4,7/
12200   1,1
12300   NO
12400   NO
12500   0,5,0,3000
12600   YE
12700   N, CPH
12800   NO
12900   4
13000   4,7/
13100   1,1
13200   NO
13300   NO
13400   40,220,0,3000
13500   YE
13600   DELTA-F, 1E-5 CM**3/GM
13700   NO
13800   5
13900   4,3,0/  !RESET SELECTED VARIABLE PARAMETERS
14000   /
14100   /
14200   0.,1./  !NOW GOING TO PLOT S(THETA) RATHER THAN S(P), CHANGE A1,A2
14300   ,,3/
14400   4,7/
14500   1,1
14600   NO
14700   NO
14800   34.8,36.8,2,22
14900   YE
15000   SALINITY, PPT
15100   YE
15200   POTENTIAL TEMPERATURE, DEG C
15300   2
15400   4,7/  !CALL PLOT BRANCH FOR FINAL FRAME; DEEP THETA-S
15500   1,37
15600   NO
15700   NO
15800   34.93,35.03,2.6,4.8
15900   NO
16000   NO
16100   2
16200   4,12/  !RETURN CONTROL TO PEPLS
16300   7/  !EXIT PROGRAM?
16400   YES
16500   $PRINT/DEL PRINT.PTN,PRINT.PPT
```

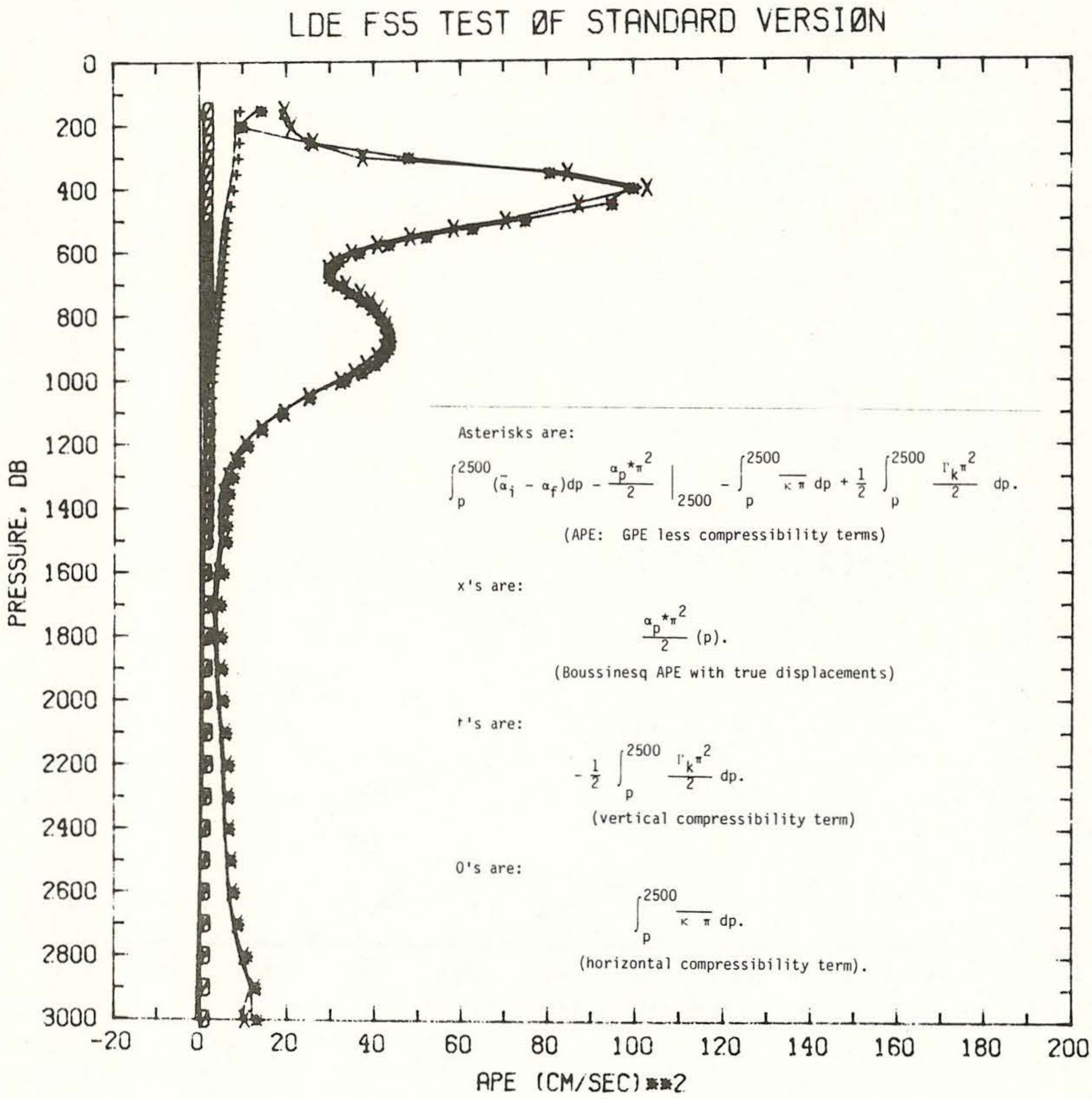


Figure 5: Example plot from ENERGY.COM: APE

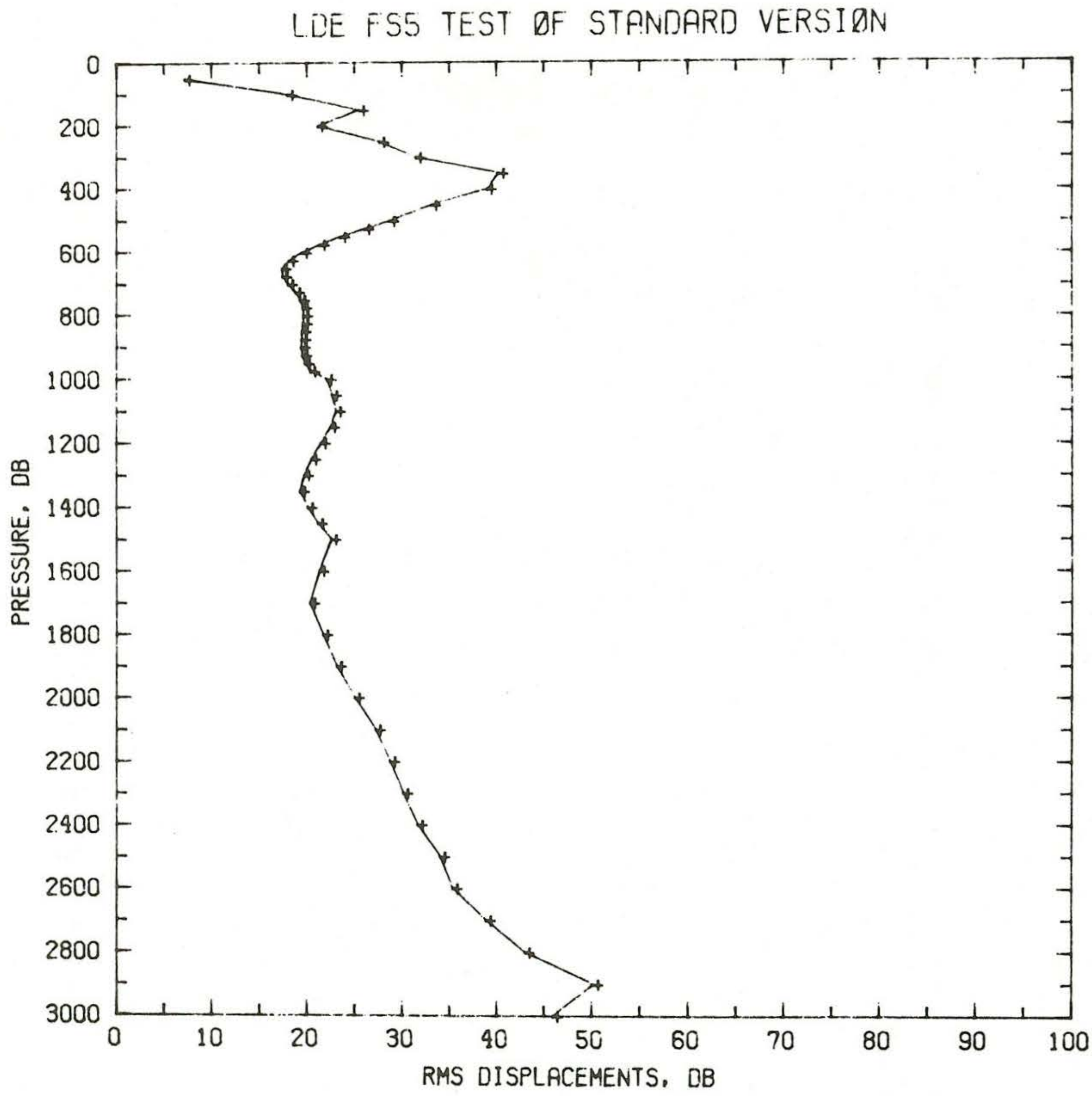


Figure 6: Example plot from ENERGY.COM.
Rms vertical displacements, π, in db.

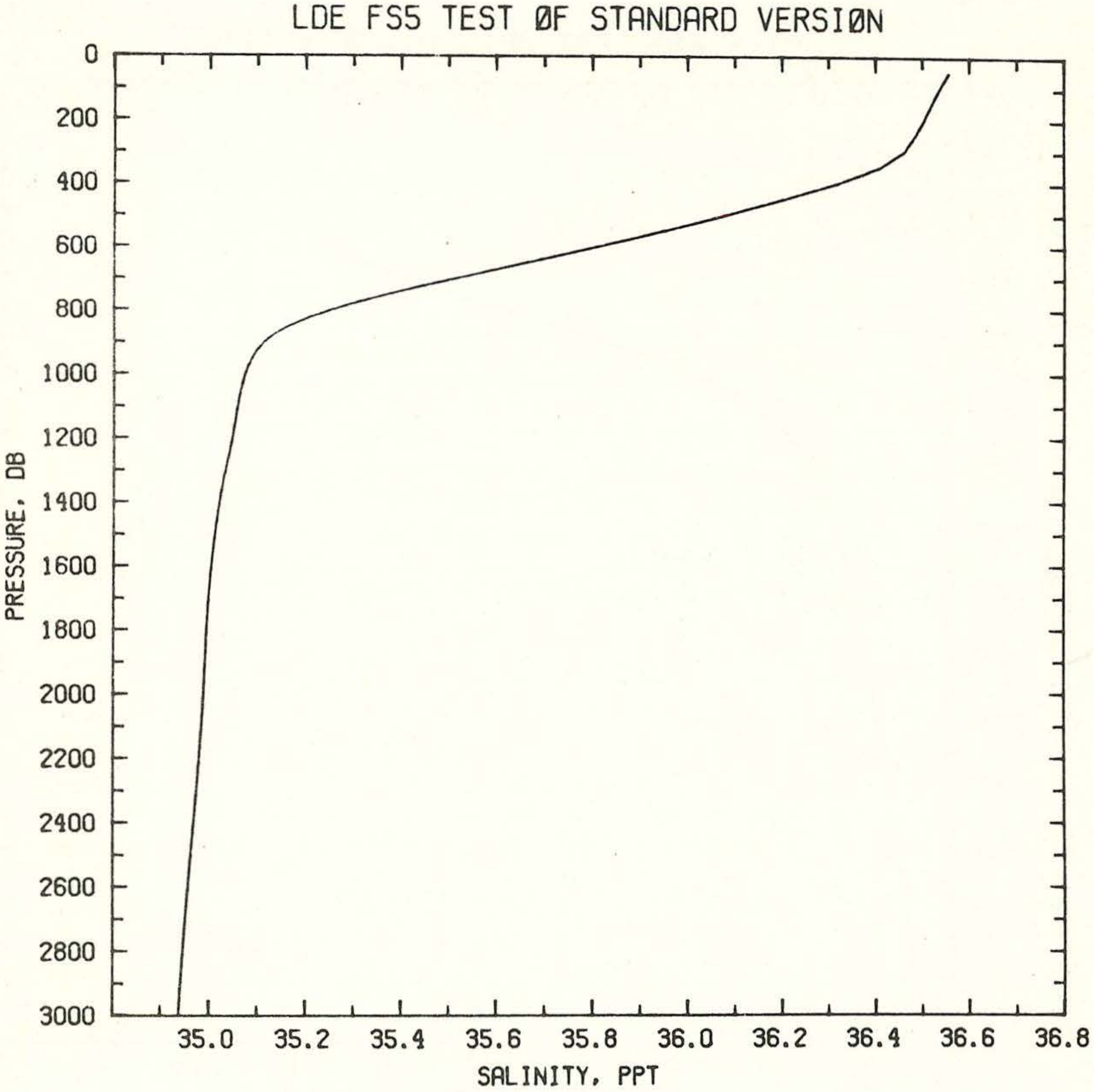


Figure 7: Example plot from ENERGY.COM
Averaged salinity in ppt along adiabatically leveled surfaces.

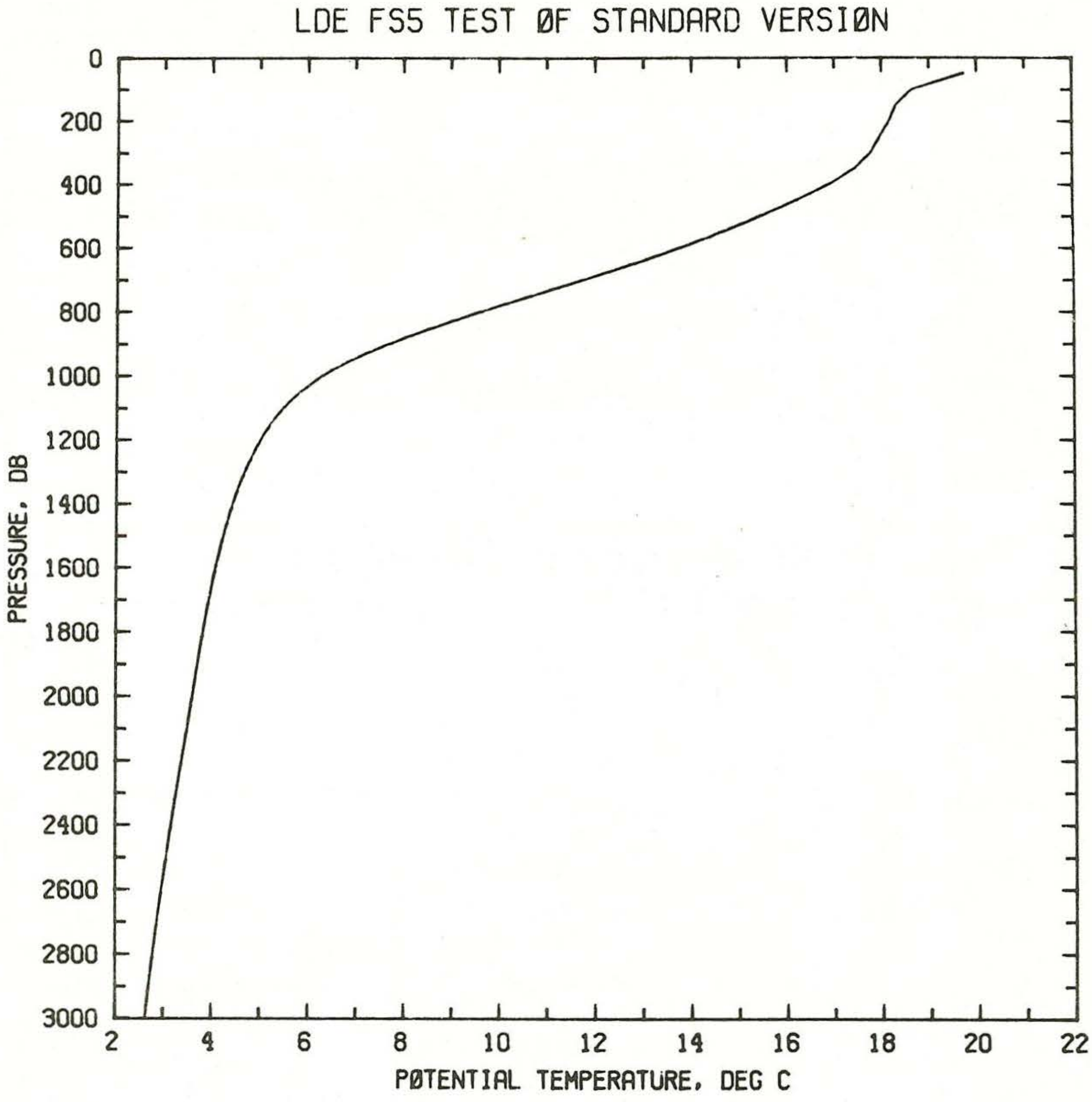


Figure 8: Example plot from ENERGY.COM
Potential temperature in °C averaged along adiabatically leveled surfaces.

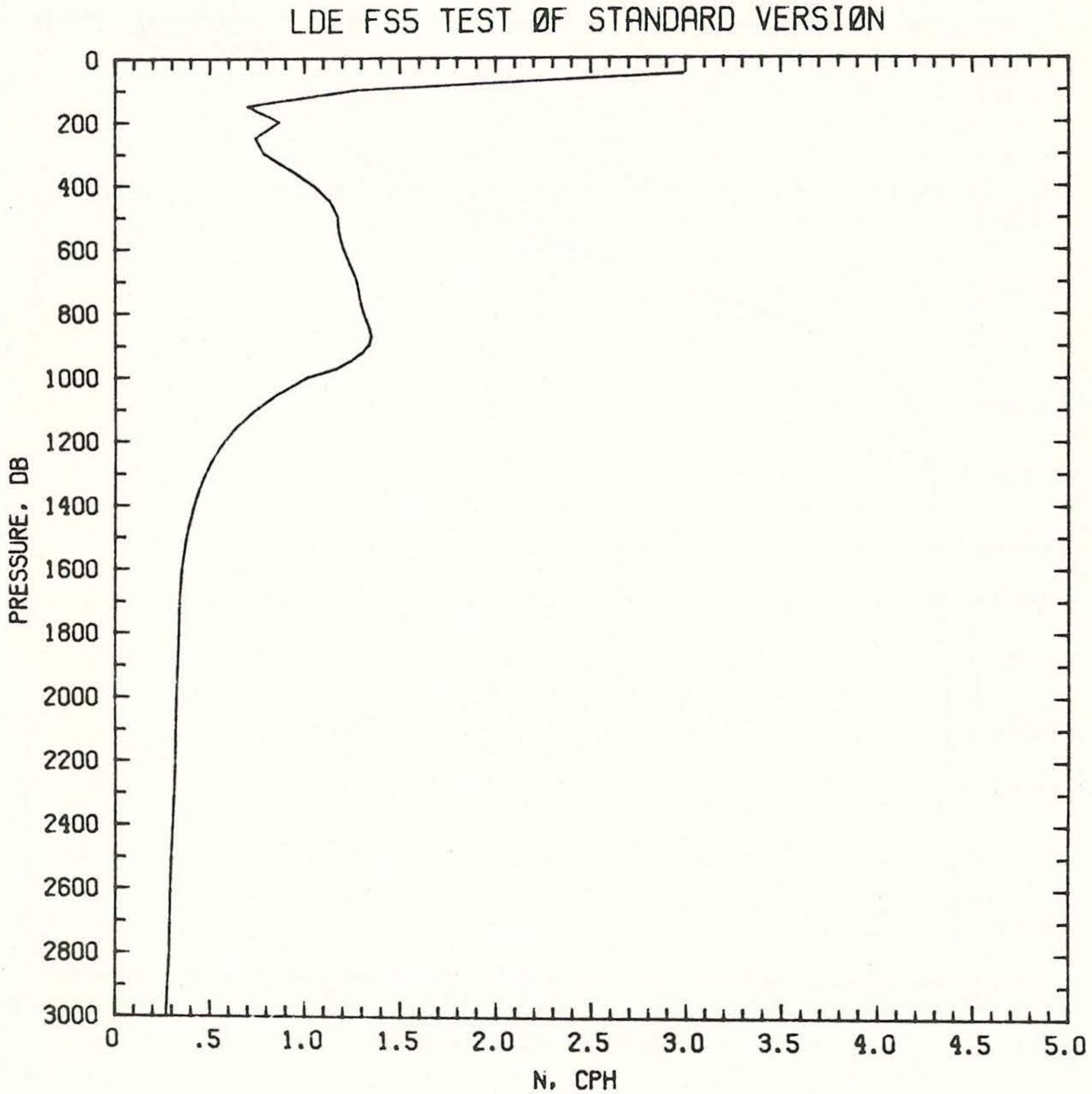


Figure 9: Example plot from ENERGY.COM
Buoyancy frequency N in cph averaged along adiabatically leveled surfaces.

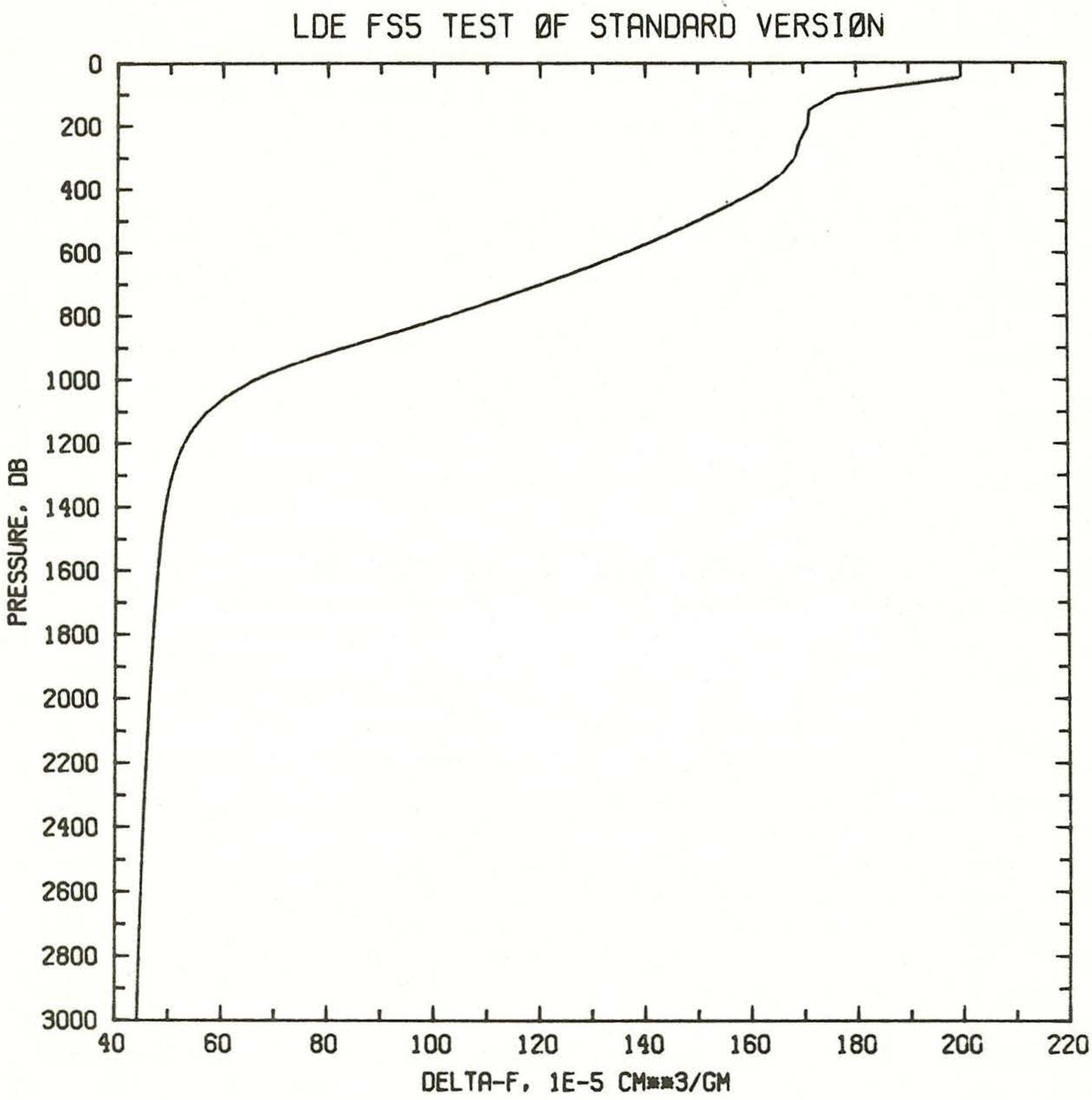


Figure 10: Example plot from ENERGY.COM

Reference (adiabatically leveled) steric anomaly in units of 10^{-5} cm^3/gm.

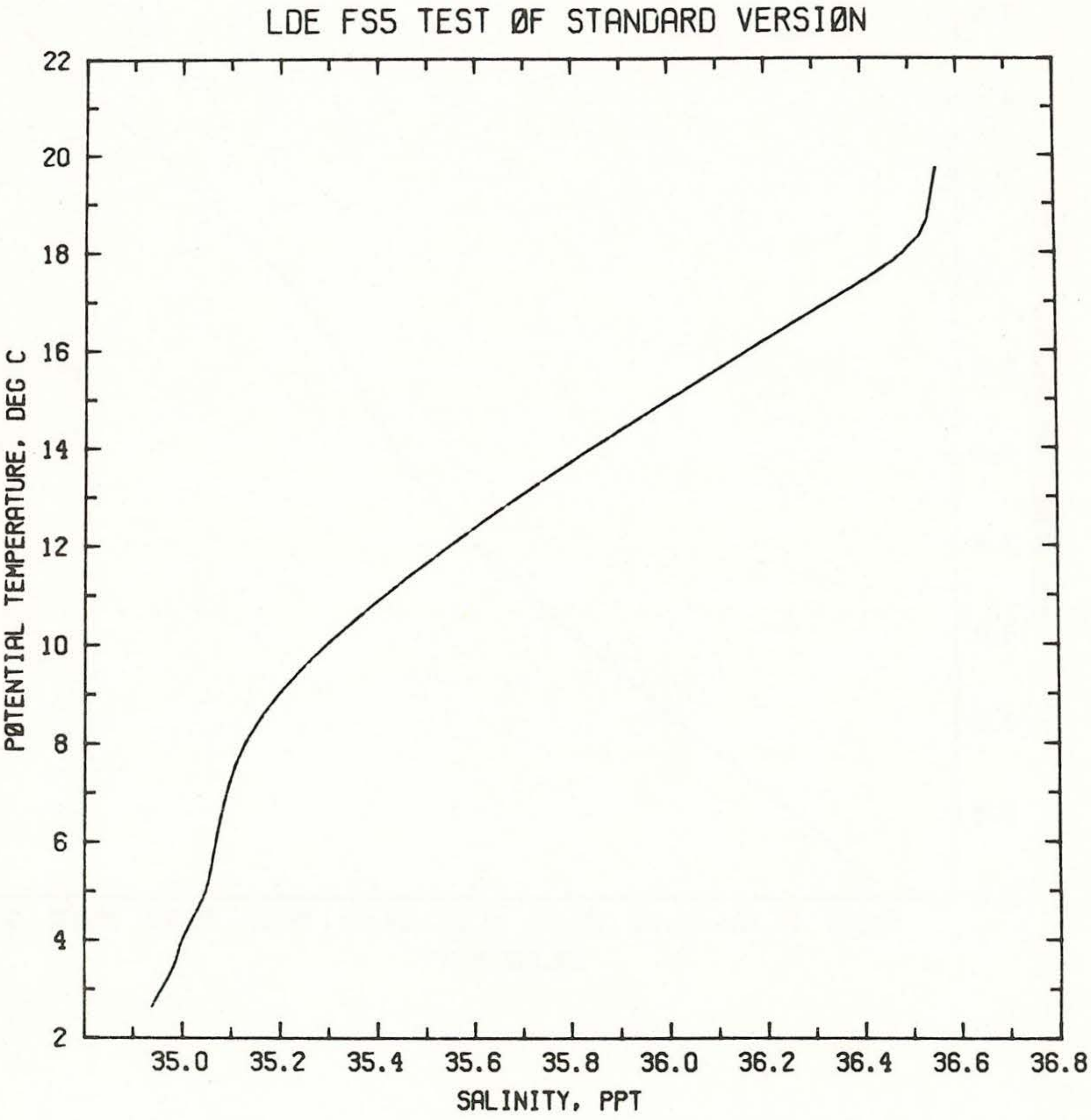


Figure 11: Example plot from ENERGY.COM
Potential temperature vs salinity computed as averages along adiabatically leveled surfaces.

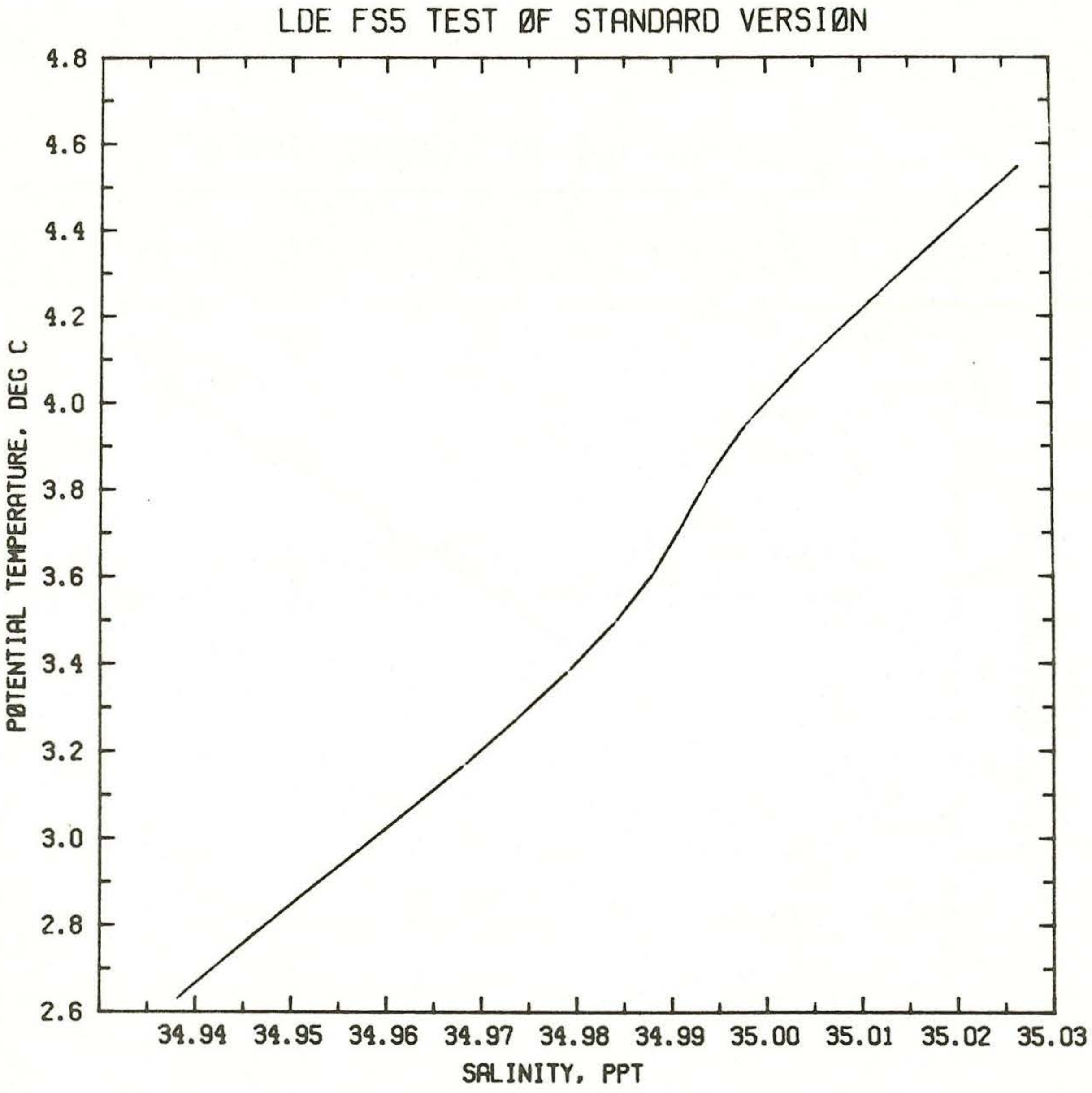


Figure 12: Example plot from ENERGY.COM
Deep potential temperature vs salinity.

```
   50   $!   ************** POTEN.COM ***************
  100   $!TASK IS TO SET UP PARAMETERS FOR AN INTERACTIVE RUN OF POTEN
  110   $!FINAL INSTRUCTION IN THIS COM FILE CHANGES KIN TO 6, THEREBY
  120   $!RETURNING CONTROL TO THE TERMINAL.
  130   $!
  200   $ASSIGN PRINT.PTN FOR004
  300   $ASSIGN JSHPES5.PTN FOR012   !CHANGE JSHPES5 TO CORRECT STATION LIST
  500   RUN/NODEB POTEN
  600   YES INITIALIZE COMMON
  700   NO DO NOT INITIALIZE REGRESSION PARAMETERS
  716   2.5   !PRESSURE INTERVAL FOR INPUT DATA
 1200   0,,,,,,6/   !CONTROL NOW RETURNS TO TERMINAL FOR INTERACTIVE SESSION
```

```
  5    $!  ************** PEPLT.COM ***************
 10    $!TASK IS TO SET UP PARAMETERS FOR AN INTERACTIVE RUN OF PEPLT
 20    $!LAST STATEMENT IN THIS COM FILE CHANGES KIN TO 6, THEREBY
 30    $!RETURNING CONTROL TO THE TERMINAL.
 40    $!
100    $ASSIGN JSHPF5.PTN FOR012    !CHANGE JSHPF5 TO CORRECT STATION LIST
125    $ASSIGN PLOT.PPT FOR008      !PLOT.PPT IS METACODE FILE
150    $ASSIGN PRINT.PPT FOR004     !PRINT.PPT IS LIST FILE
200    RUN/NODEB PEPLT
300    NO DO NOT READ IN PREVIOUSLY STORED COMMON
500    YES INITIALIZE DATA SELECTION PARAMETERS
600    4,2,1,6,,,6/      !CONTROL NOW RETURNS TO TERMINAL FOR INTERACTIVE SESSION
```

```
   50   $!  ************** TABLE.COM  ***************
  100   $!TASK IS TO CREATE TWO PLOTS CORRESPONDING TO FIGURES
  200   $!4 AND 5 IN BRAY(1981) BLUE COVER REPORT. THE FIRST
  300   $!FIGURE IS BUOYANCY FREQUENCY N AS A FUNCTION OF DEPTH
  400   $!AND POSITION. THE SECOND IS VERTICAL DISPLACEMENT.
  500   $!BOTH ARE CREATED USING TABLE SUBROUTINE OF PEPLT;
  600   $!THE FIRST FIGURE USES THE DEFAULT PLOT SPECIFICATIONS.
  700   $ASSIGN PLOT.PPT FOR008
  800   $ASSIGN JSHPF5.PTN FOR012
  900   RUN/NODEB PEPLT
 1000   NO DO NOT READ IN PREVIOUSLY STORED COMMON
 1100   YES INITIALIZE DATA SELECTION PARAMETERS
 1200   1,2/  !INITIALIZE PLOT PARAMETERS IN TABLE
 1300   16/    !ENTER NUMBER OF STATIONS IN PLOT
 1350   3/     !3 IS FIRST LEVEL PLOTTED
 1400   YES INPUT NEW PLOT LABEL
 1500   LDE FS5: N,CPH
 1600   NO DO NOT USE DEFAULT AXIS PARAMETERS
 1700   /  !USE THESE MIN AND MAX
 1800   YES CHANGE X-AXIS LABEL
 1900   KM EAST OF ORIGIN
 2000   YES CHANGE Y-AXIS LABEL
 2100   KM NORTH OF ORIGIN
 2200   1,0/  !PLOT
 2300   1,1/  !CHANGE PARAMETERS FOR SECOND PLOT
 2400   16,,,100,3000/  !100 IS DISPLACEMENT AXIS IN DB
 2450   3/
 2500   YES CHANGE PLOT LABEL
 2600   LDE FS5: VERTICAL DISPLACEMENTS, DB
 2633   NO DO NOT USE DEFAULT AXIS PARAMETERS
 2666   /    !USE THESE MIN,MAX VALUES
 2700   NO DO NOT CHANGE X-AXIS LABEL
 2800   NO DO NOT CHANGE Y-AXIS LABEL
 2816   4,3,0/   !CHANGE PLOT PARAMETERS--AVRGS
 2832   /
 2848   /
 2864   ,.05/  !RESCALE DISPLACEMENTS
 2880   ,50/    !PLOT DISPLACEMENTS RATHER THAN N
 2890   4,12/  !RETURN TO PEPLS
 2900   1,0/  !PLOT
 3000   7/     !EXIT PROGRAM
 3100   YES
```

```
  10    $!  *************** DYNHT.COM ***************
  25    $!TASK IS TO CREATE GPCP COMPATIBLE OUTPUT FROM *.AVG FILES AT
  31    $!SPECIFIED LEVELS PF. VARIABLES OUTPUT ARE STN ID,PF,XLAT,
  37    $!XLONG,DYN HT REF TO PF AT LEVEL 50,TO,SO,REF SPECIFIC VOL,TIME
  43    $!(JULIAN DAYS FROM 1 JAN+DECIMAL HOURS); FORMAT IS
  46    $!1H ,A5,I4,2F(8.2),5F8.3) 28 MAY 81 NAN BRAY
  54    $!NUMBER OF STATIONS OUTPUT, OUTPUT VARIABLES, AND REFERENCE PRESSURE
  58    $!FOR DYNAMIC HEIGHT MAY BE CHANGED IN AVRGS BRANCH 3.
  79    $!DYNAMIC HEIGHT OUTPUT IN DYNAMIC CENTIMETERS.
  89    $!
 100    $ASSIGN JSHPF5.PTN FOR012 !CHANGE JSHPF5 TO APPROPRIATE STATION LIST
 200    RUN/NODEB PEPLT
 300    NO DO NOT READ IN INITIALIZED COMMON
 500    YE INITIALIZE DATA SELECTION PARAMETERS
 600    4,2,1,6/  !ZERO C ARRAY
 700    4,3,1/  !SET DATA VARIABLES
 800    1000,18,19,13,14,19,7,50,55/
 900    /
1000    /
1100    /
1200    4,22,1,6/  !CREATE FILES FOR EACH LEVEL REQUIRED
1250    YES INPUT NEW FILE NAME
1300    TEST.DAT
1400    15,17  !RANGE OF LEVELS FOR WHICH DH WILL BE OUTPUT
3400    4,12/  !RETURN CONTROL TO PEPLS
3500    7/      !EXIT PROGRAM?
3600    YES
3700    $SORT/KEY=(POSITION:7,SIZE:6) TEST.DAT TESTP.DAT !SORTS BY PRESSURE
```

Appendix B.
Program Listings for POTEN

```
  50   POTEN: PTENS: SHORT DOCUMENTATION
 100   KBR       ISW      JSW       KLIST    DESCRIPTION
 150    0         -        -          -      SHORT DOCUMENTATION
 200    1         #        -          -      COMPUTE REGRESSIONS FOR ISW
 300                                         SEQUENTIAL STATIONS.
 400    2         -        -          -      INITIALIZE DATA SELECTION
 500                                         PARAMETERS.
 600    3         0        -          -      SET PARAMETERS: SHORT LIST.
 700              1        -          -      SET PARAMETERS: FULL LIST;
 800                                         STORES COMMON IN FILE KPTCM
 900    4         1        -         LU      AVRCP AVERAGING SUBROUTINE.
1000    5         -        -          -      NOT USED
1100    6         -        -          -      LIST LABEL
1300    7         -        -          -      NOT USED
1400    8         -        -          -      LIST HEADER INFORMATION.
1500    9         -        -          -      LIST DATA RECORD.
1600   10         -        -          -      NOT USED
2000   11         -        -          -      SET ISSW (SWITCH) ARRAY.
2100   12         -        -          -      EXIT PROGRAM.
```

```
C COMPOTEN.FOR FILE: DIMENSION,COMMON AND EQUIVALENCE FOR POTEN
C AVAILABLE POTENTIAL ENERGY PROGRAMS. N.BRAY
C
      PARAMETER KCM= 235
      BYTE LB,PROVER
C
C DIMENSION
C
      DIMENSION KHDG(150),KBUF(46),VR(1)
      DIMENSION KPTCM(KCM)
      DIMENSION PRESS(3300)
C
C BLANK COMMON
C
      COMMON KIN
C BEGINNING OF STORED COMMON
      COMMON KTTX,KLIST,KOUT,KTP,ISW,JSW,KBR
C
C BEGINNING OF HEADER
C
      COMMON LTYPE,MHDR,ICON,ISHP,KCAST,DAY,TPR,LPR
      COMMON XLAT,XLONG,WGT,XLTO,XLGO
      COMMON LBBL(3),LBL(13),NSC(60),NPR(60),NSECTION
C
C BEGINNING OF DATA BUFFER KBUF
C
      COMMON KTYPE,MBUF,IREC,N,NDP,KSW,L1,L2
      COMMON PF,TO,SO,DVO
      COMMON PI,THF,SF,DVF
      COMMON PM,THM,SM,DVM
      COMMON DH,PE,XPE
      COMMON CP(8),Z1,CT(8),Z2,F1,F2,F3
C
C END OF KBUF
C
      COMMON DELP,DP
      COMMON A1,A2,A3,N1,N2,N3
      COMMON C(6),ISSW(16)
      COMMON ICRUIS,IPROJ,PROVER
      COMMON JMAX
C
C END OF STORED COMMON
C
      COMMON P(3300),T(3300),S(3300),DV(3300)
      COMMON TH(3300),PT(3300),TT(3300)
      COMMON B(8),BP(8),BT(8),BA(8),CO(36),MR(8)
      COMMON EX(6),JEX(6)
      COMMON WT(600),JSHP(600)
      COMMON DATAX(3300,0:2)
      COMMON JSTN,JRMAX,M1,M2
      COMMON LLREC,KKST
C
C EQUIVALENCE
C
      EQUIVALENCE (KHDG,LTYPE),(KBUF,KTYPE)
      EQUIVALENCE (PDIFF,A1),(VR,PF),(KTTX,KPTCM)
      EQUIVALENCE (PRESS,DATAX)
C
```

```
C  INDEX RECORD FIELDS DECOMPOSED
C  USES LABELLED COMMOM : USE INCLUDE STATEMENT TO MERGE INTO PROGS.
C  RCM MAR 27 1980
C  ARRAY IDXREC CONTAINS THE INDEX RECORD
C  ALWAYS THE FIRST RECORD OF A DATA FILE
C  IDXREC HAS THE SAME STRUCTURE IN THE SUBINDEX FILE
C
      COMMON/INDX/ IDXREC(256)
C*********************************************************************
C        LPGVER IS AN ACSII DESCRIPTION OF LOADING PROGRAM VERSION
      INTEGER LPGVER(4),LSTREC
C*********************************************************************
C LDATE AND LTIME CONTAIN ASCII DATE & TIME FILE CREATED
      INTEGER LDATE(3),LTIME(2)
C  DEVICE AND FILSPEC FORM A COMPLETE FILE SPECIFIER
      INTEGER FILSPEC(8),DEVICE
C  IFHED ARRAY HAS THE SAME STRUCTURE AS CTD78 VAX FORMAT
      INTEGER IFHED(90),ICMNT(35)
C  CNTRL HAS THE LOCATIONS OF THE BEGINNING WORD OF
C INFORMATION FIELDS OF THE INDEX RECORD
C 1=CTD78 HEADER 2=ABREVIATED DATA DESCRIPTORS 3=TAPE 4=FILE SPEC
      INTEGER CNTRL(6)
C VARDES ARRAYS CONTAIN ABREVIATED VARIABLE DESCRIPTORS
C MIN/MAX VALUES IVARDES CONTAINS MNEMONIC IDENTIFIERS
      DIMENSION VARDES(4,16),IVARDES(4,16)
      EQUIVALENCE(KEYWD,IDXREC(1))
      EQUIVALENCE(CNTRL(1),IDXREC(2))
C  LSTREC IS THE NEXT AVAILABLE RECORD IN SUBINDEX FILE 1ST REC. ONLY
      EQUIVALENCE(LSTREC,IDXREC(9))
      EQUIVALENCE(IFHED(1),IDXREC(13))
      EQUIVALENCE (VARDES(1,1),IDXREC(115)),(IVARDES(1,1),IDXREC(115))
      EQUIVALENCE(KSCAN,IDXREC(105))
      EQUIVALENCE (RECLNG,IDXREC(107))
      EQUIVALENCE(MSCAN,IDXREC(106)),(NSCANS,IDXREC(112))
      EQUIVALENCE(PMIN,IDXREC(110)),(PRSINT,IDXREC(111))
      EQUIVALENCE(NTOT,IDXREC(108))
      EQUIVALENCE (IMPVAR,IDXREC(109)),(IPLOC,IDXREC(114))
      EQUIVALENCE
     1 (IFHED(3),ISHIP),(IFHED(4),ICRUZ),(IFHED(5),ISTAS)
     2,(IFHED(7),IYR),(IFHED(8),IMO),(IFHED(9),IDA)
     3,(IFHED(11),ILTSD),(IFHED(12),ILTSM)
     4,(IFHED(13),ILNSD),(IFHED(14),ILNSM)
     5,(IFHED(15),INWPS),(IFHED(16),IHRZ),(IFHED(17),IHRTZ)
     6,(IFHED(19),ILTED),(IFHED(20),ILTEM)
     7,(IFHED(21),ILNED),(IFHED(22),ILNEM)
     8,(IFHED(23),IETME),(IFHED(10),ISTME)
     9,(IFHED(38),ICAST),(IFHED(27),JDAY),(IFHED(28),INST)
     X,(IFHED(55),ICMNT)
C*********************************************************************
      EQUIVALENCE
     1 (DEVICE,IDXREC(193)),(FILSPEC(1),IDXREC(200))
     2 ,(LDATE(1),IDXREC(195)),(LTIME(1),IDXREC(198))
     3,(LPGVER(1),IDXREC(99))
C  RMAX IS THE LAST RECORD OF A DATA FILE
C  IDXLOC IS THE RECORD # OF THE INDEX FILE = 1 FOR SINGLE STATION FILE
     4,(RMAX,IDXREC(209)),(IDXLOC,IDXREC(208))
C*********************************************************************
C*********************************************************************
C END LABELLED COMMON FOR INDEX RECORD
C*********************************************************************
```

```
C POTEN MAIN PROG ************************************************
      PROGRAM POTEN
C ****************************************************************
C
C PROGRAM TO COMPUTE REFERENCE SURFACES RELATIVE TO PF FOR
C CALCULATION OF AVAILABLE POTENTIAL ENERGY. REGRESSION
C FITS ARE MADE TO PRESSURE AND POTENTIAL TEMPERATURE AS
C FUNCTIONS OF SPECIFIC VOLUME ANOMALY AT PF.
C
C JUNE 28 1976 N FOFONOFF
C
      INCLUDE 'COMPOTEN.FOR'
C
C OPEN BINARY FILE FOR STORAGE OF COMMON
C
        OPEN(UNIT=10,NAME='KPTCM.DAT',ACCESS='DIRECT',TYPE='OLD',
     *  RECORDTYPE='FIXED',RECORDSIZE=KCM,ERR=1100)
    1 CONTINUE
C
        KIN = 5
        KTTX = 6
        KLIST = 4
        KOUT = 9
        KTP = 1
   30   WRITE(KTTX,1000)
 1000   FORMAT(1H ,'POTEN: POTENTIAL ENERGY PROGRAM')
        CALL PTENS
       GO TO 50
C
C CREATE NEW BINARY FILE FOR STORAGE OF COMMON IF NO OLD ONE EXISTS
C
 1100   OPEN(UNIT=10,NAME='KPTCM.DAT',ACCESS='DIRECT',TYPE='NEW',
     *  RECORDTYPE='FIXED',RECORDSIZE=KCM,ERR=1100)
       GO TO 1
   50 END
```

```
C PTENS SUBPROG POTEN ***** PTENS.FOR FILE *****
      SUBROUTINE PTENS
C ***********************************************************
C
C PROGRAM TO COMPUTE REFERENCE SURFACES RELATIVE TO PF FOR
C CALCULATION OF AVAILABLE POTENTIAL ENERGY. REGRESSION
C FITS ARE MADE TO PRESSURE AND POTENTIAL TEMPERATURE AS
C FUNCTIONS OF SPECIFIC VOLUME ANOMALY AT PF.
C
C JUNE 28 1976 N FOFONOFF
C
C  MODIFIED TO ACCEPT CTD78 VAX DISC DATA AS INPUT 15DEC80 N.BRAY.
C
      DIMENSION D(5),DOC(10)
C
      INCLUDE 'COMPOTEN.FOR'
C
      CHARACTER*8 DOC
C
        KIN = 5
        KTTX = 6
        WRITE(KTTX,40)
   40   FORMAT(1H ,'INITIALIZE COMMON (YES OR NO)?')
       IF(NOYES(KIN,KTTX).NE.1)GO TO 14
C
C INITIALIZE DATA SELECTION PARAMETERS
C
        CALL DATA(KTP,-1)
       GO TO 30
   14   READ(10'1)KPTCM
   18   WRITE(KTTX,20)
   20   FORMAT(1H ,'INITIALIZE REGRESSION PARAMETERS (YES OR NO)?')
       IF(NOYES(KIN,KTTX).EQ.1)GO TO 15
        READ(10'1,END=10)KPTCM
    5   WRITE(KTTX,25)
   25   FORMAT('OWHAT IS THE RESOLUTION OF THE INPUT DATA, IN DB?')
        READ(KIN,*)DELP
        KLIST = 6
   10   WRITE(KTTX,1005)KBR,ISW,JSW,KLIST,KOUT,KTP,KIN
 1005   FORMAT(1H ,'POTEN:KBR,ISW,JSW,KLIST,KOUT,KTP,KIN',/,7I4)
        READ(KIN,*)KBR,ISW,JSW,KLIST,KOUT,KTP,KIN
       IF(KBR.GT.12)KBR=13
       IF(KBR)1300,1300,12
   12 GO TO(100,200,300,400,500,600,703,800,900,1000,1100,1200,100,
     *1300)KBR
C
C ************INITIALIZATION ******************************
   15   KTYPE = 0
        MHDR = 150
        MBUF = 46
        NSECTION=4
        NPR(1) = 4
        NPR(2) = 12
        NPR(3) = 17
        NPR(4) = 24
        NPR(5) = 50
        NPR(6) = 50
        NPR(7) = 100
        NPR(8) = 200
        NPR(9) = 500
        NPR(10) = 500
        NPR(11) = 0
        NPR(12) = 2
        NPR(13) = 7
        NPR(14) = 13
        NPR(15) = 13
        NSC(1) = 0
        NSC(2) = 400
        NSC(3) = 1000
        NSC(4) = 1800
        NSC(5) = 3500
        NSC(6) = 6
        NSC(7) = 5
        NSC(8) = 4
        NSC(9) = 3
        NSC(10) = 3
        NSC(11) = 20
        NSC(12) = 30
        NSC(13) = 40
        NSC(14) = 50
        NSC(15) = 60
        PROVER = 'W'
        ISHP = 'GY'
        ICRUIS = 1
        IPROJ = 3
       GO TO 5
  30   DELP=2.
      DO 16 J=1,36
   16 VR(J) = 0.0
        PDIFF = 6.0
        A2 = 3.0
        A3 = 3.0
        LTYPE = 1
        ICON = 0
        N = 2
        NDP = 10
        KSW = 1
        WGT = 1.0
      DO 17 J=1,16
        ISSW(J)=0
   17 CONTINUE
       GO TO 18
C ************ SELECT DATA AND COMPUTE #1 *****************
  100   CALL COMPS
       GO TO 10
C
C INITTALIZE DATA SELECTION PARAMETERS #2 ******************
C
  200   CALL DATA(KTP,-1)
C *************SET PARAMETERS #3 ****************************
  300   WRITE(KLIST,3000)ICON,KSW,A2,A3,WGT,PDIFF
        READ(KIN,*)ICON,KSW,A2,A3,WGT,PDIFF
        WRITE(KLIST,3020)DELP
        READ(KIN,*)DELP
       IF(ISW)10,10,310
C
C SUBROUTINE TO ACCEPT REGRESSION PARAMETERS IN ENGLISH AND
C CONVERT TO POTEN PARAMETERS
C
  310   CALL PARAM
        WRITE(KLIST,320)
C PRINT OUT POTEN FORMAT PARAMETERS
```

```
  320   FORMAT('OSECTION LEVEL  LEVEL   INDEX    START   # OF    # OF',/,
     X   1H ,'NUMBER  NUMBER INTERVAL        PRESSURE  TERMS  CYCLES')
        NSF=NSECTION
      DO 330 I=1,NSE
        I1=I+NSE
        I2=I+2*NSE
        WRITE(KLIST,335)I,NPR(I),NPR(I1),NPR(I2),NSC(I),NSC(I1),NSC(I2)
  335   FORMAT(I3,6I8)
  330 CONTINUE
C CHANGE OR LIST DATA SELECTION PARAMETERS
        CALL DATA(KTP,0)
C
C CHANGE OR LIST DATA LABEL (IDENTIFIES THE DATA SOURCE)
C
        WRITE(KTTX,3015)(LBL(I),I=1,13)
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        WRITE(KLIST,3010)
        READ(KIN,6005)(LBL(K),K=1,13)
       ENDIF
C
C STORE COMMON TO BINARY FILE KPTCM
C
       IF(JSW.NE.2)THEN
        WRITE(10'1)KPTCM
       ENDIF
       GO TO 10
C
 3000   FORMAT(1H ,'ICON,KSW,SDP,SDT,WGT,PDIFF',/,2I4,3F6.2,F7.0)
 3005   FORMAT(1H ,'NSC:P,N,NDP',/,5I5,10I3)
 3006   FORMAT(1H ,'PRESSURE CONSTANTS',/,15I4)
 3010   FORMAT(1H ,'INSERT LABEL <27 CHAR.')
 3015   FORMAT(1H ,'INPUT NEW LABEL? OLD LABEL IS:  ',/,2H  ,13A4)
 3020   FORMAT(1H ,'INPUT DATA RESOLUTION',/,F6.1)
C
C ************AVERAGING SUBROUTINE #4 **************************
  400   CALL AVRCP
       GO TO 10
C ************ #5 NOT PRESENTLY USED ***************
  500  GO TO 10
C       NGR = 5
C       KINP = 5
C       JMAX = 23
C       KOUT = 1
C 501 DO 505 M=9,13
C      DO 505 K =1,100
C 505 CR(K,M) = 0.0
C 507   WRITE(KTTX,5010)KOUT,NGR,JMAX,KINP
C5010   FORMAT(1H ,'AVDVF:KOUT,NGR,JMAX,KINP',/,4I4)
C 512   READ(KIN,*)KOUT,NGR,JMAX,KINP
C     DO 530 J=1,NGR
C     DO 520 JR=1,JMAX
C       READ(KINP,*)I,NST,KPR,(D(K),K=1,5)
C       IIPR(I) = KPR
C     DO 520 M=9,13
C 520 CR(I,M) = CR(I,M) + D(M-8)
C 530 CONTINUE
C     DO 540 J=9,13
C     DO 540 I=1,JMAX
C 540 CR(I,J) = CR(I,J)/FLOAT(NGR)
C 545 DO 550 I=1,JMAX
C       WRITE(KOUT,5000)I,NST,IIPR(I),(CR(I,K),K=9,13)
```

```
C 550 CONTINUE
C          KOUT = 24
C          READ(KIN,*)ISW,KOUT
C         GO TO (501,507,545,10)ISW
C5000       FORMAT(2I4,I5,5F10.4)
C *************LIST DATA LABEL #6 **********************************
  600      WRITE(KLIST,6005)(LBL(K),K=1,13)
          GO TO 10
 6005      FORMAT(1H ,13A4)
C ************* #7 NOT PRESENTLY USED ***************************
C 703      MEOF = 0
C 702      CALL READ(KOUT,KBUF,MBUF,IEOF)
C         IF(IEOF)720,704,704
C 704      MEOF = 0
C 705     IF(KTYPE)900,900,710
C 710 DO 715 M=1,MHDR
C 715 KHDG(M) = KBUF(M)
C         GO TO 800
C 720     IF(MEOF)10,725,10
C 725      MEOF = 1
C         GO TO 10
  703    GO TO 10
C *************LIST HEADER RECORD #8 *************************
  800      WRITE(KLIST,8000)(LBBL(K),K=1,3),XLAT,XLONG,XLTO,XLGO
           WRITE(KLIST,8005)LTYPE,MHDR,ICON,ISHP,ICAST,JDAY,IPR,LPR
          GO TO 10
 8000      FORMAT(/,3A4,4F8.3)
 8005      FORMAT(/,' TYPE MDHR ICON SHIP CAST JDAY  IPR  LPR',/,8I5)
C *************LIST DATA RECORD #9 ***************************
  900      WRITE(KLIST,9000)IREC,PF,TO,SO,DVO,PM,THM,SM,DVM,Z1,Z2
  903     IF(ISSW(8))905,10,10
  905      WRITE(KLIST,9005)(CP(K),K=1,N)
           WRITE(KLIST,9010)(CT(K),K=1,N)
          GO TO 10
 9000      FORMAT(1H ,I2,2(F7.1,F7.3,F7.3,F7.2),P6.2,F6.4)
 9005      FORMAT(1H ,3HCP ,6E11.4)
 9010      FORMAT(1H ,3HCT ,6E11.4)
C *************MAG TAPE FUNCTIONS #10 ************************
C1000      CALL PTAPE(ISW,JSW,KLIST)
C          KLIST = 6
 1000     GO TO 10
C5 ****** SET ISSW SWITCHES #11 ******
 1100      WRITE(KTTX,1150)(K,K=1,16),(ISSW(K),K=1,16)
 1150      FORMAT(2(1H ,X,16I4,/),' ENTER K,ISSW(K)')
           READ(KIN,*)(K,ISSW(K),M=1,16)
          GO TO 10
C
C ********** EXIT PROGRAM #12 ******************************
 1200      WRITE(KTTX,1210)
          IF(NOYES(KIN,KTTX).NE.1)GO TO 10
         STOP
 1210      FORMAT(1H ,'EXIT PROGRAM?')
C** POTEN: SHORT DOCUMENTATION--BRANCH 0 ***************
 1300      OPEN(UNIT=50,NAME='POTEN.DOC',TYPE='OLD',READONLY)
         DO 1350 N=1,200
           READ(50,1325,END=1312)(DOC(I),I=1,8)
           WRITE(KTTX,1330)(DOC(I),I=1,8)
 1350 CONTINUE
 1312      CLOSE(UNIT=50)
 1325      FORMAT(8A8)
 1330      FORMAT(1H ,8A8)
```

```
23600          GO TO 10
23700         END
```

```
C COMPS SUBPROG POTEN ***************************************
C ***********************************************************
       SUBROUTINE COMPS
C ************************************************************
C
C TO COMPUTE REGRESSION COEFFICIENTS AT SPECIFIED DEPTHS.
C
C JUNE 28 1976 N FOFONOFF
C MODIFIED FOR CTD78 FORMAT INPUT DATA (VAX DISC VERSION) 15 DEC 80
C N.BRAY
C
       INCLUDE 'COMPOTEN.FOR'
C
C
C  IF OUTPUT IS REQUESTED THEN OUTPUT TO JSHP.PTN FILE THE
C    NUMBER OF STATIONS
C
  100 CONTINUE
C
C  INITIALIZE AND ACCESS INDEX FILE AND CRUISE INFORMATION
C
         CALL DATA(KTP,2)
C
C  BEGIN COMPUTATION FOR ISW TOTAL STATIONS
C
        IF(ISW.GT.LLREC)ISW=LLREC
       DO 106 KST=ISW,JSW
C
C READ STATION HEADER FROM UNIT KTP AND CHECK IF IT MEETS
C SELECTION CRITERIA
C  READ TEMPERATURE AND SALINITY DATA INTO DATA ARRAY.
C
  101    CALL DATA(KST,1)
        GO TO 200
  106 CONTINUE
C
C RETURN TO PTENS
C
  295 RETURN
C
C COMPUTE REGRESSION VERSION OF DATA
C IF TSSW(13)=-1 OUTPUT TO FILE *.REG
C MISCELLANEOUS INFORMATION MAY BE REQUESTED TO BE PRINTED
C TO FILE PRINT.PTN (KLIST=4) BY SETTING ISSW VALUES.
C SEE DETAILED WRITE UP.
C
C  KTYPE DISTINGUISHES BETWEEN HEADER AND DATA RECORDS:
C    O=DATA, 1=HEADER.
C  KF,KT,KM ARE INDICES
C  N IS POLYNOMIAL ORDER
C  NDP IS # OF DATA CYCLES OVER WHICH REGRESSION IS PERFORMED
C  KERR COUNTS THE # OF REPLACEMENTS MADE BY SUBR EDIT
C    IN EACH REGRESSION INTERVAL
C  IPR AND LJP KEEP TRACK OF PRESSURE AS AN INDEX
C  IREC INDEXES THE LEVELS PF
C
  200    CONTINUE
         KF = 2
         N = NSC(NSECTION+1)
         NDP = NSC(2*NSECTION+1)
         XNDP = NDP

        KERR = 0
        IREC = 1
C
C  COMPUTE REGRESSIONS
C
C JMAX IS THE TOTAL NUMBER OF LEVELS.
C
      DO 2155 M =1,JRMAX
        P(M)=PRESS(M)
        T(M)=DATAX(M,1)
        S(M)=DATAX(M,2)
 2155 CONTINUE
  210 DO 270 J=1,JMAX
C
C  SUBR JPR COMPUTES CORRECT PRESSURE PF GIVEN SECTION AND
C    INTERVAL INFORMATION
C
        IP = JPR(IREC,NPR,NSECTION)
       IF(IP.EQ.NSC(KF))THEN
 2152    N = NSC(KF+NSECTION)
         NDP = NSC(KF+2*NSECTION)
         XNDP = NDP
        KF = KF + 1
       ENDIF
        IND = (IP-PRESS(1))/DELP + 1
        M1 = IND-NDP/2
        M2 = IND+NDP/2-1
       IF(M1.LT.1)THEN
        M1=1
        M2=NDP
       ENDIF
        IF(M2.GT.JRMAX)GO TO 280
C  IF ISSW(5)=-1 WRITE OUT SCAN #, SCALED PRESSURE, TEMP,
C    SALINITY.
C
        IF(ISSW(5))216,217,217
  216   WRITE(KLIST,2160)(K,P(K),T(K),S(K),K=M1,M2)
 2160   FORMAT(1H ,I4,3F9.3)
  217 CONTINUE
C
C  PERFORM REGRESSIONS OVER INTERVAL CORRESPONDING TO PF
C
C  FIRST, FIND MEANS OF P,S.
C
  220   PF = IP
        PM = 0.0
        SM = 0.0
        THM = 0.0
        DVM = 0.0
        TO = 0.0
        SO = 0.0
        DVO = 0.0
        XN = 0.0
  231 DO 230 M=M1,M2
        PM = PM + P(M)
  230   SM = SM + S(M)
        PM = PM/XNDP
        SM = SM/XNDP
  235 DO 250 M =M1,M2
C
```

```
C  CALCULATE POTENTIAL TEMP AND SPECIFIC VOLUME ANOMALY
C    REFFRRED TO PF.
C
 2350   TH(M) = THETA(P(M),T(M),S(M),PF)
        DV(M) = DVA(PF,TH(M),S(M))
 2352   THM = THM + TH(M)
        DVM = DVM + DV(M)
        PT(M) = P(M)
        TT(M) = TH(M)
        DVX = DV(M)
C
C  F1,F2 ARE MIN AND MAX SPECIFIC VOLUME ANOMALY WITHIN
C    THE REGRESSION INTERVAL.
C
       IF(M-M1)236,236,237
  236   F1 = DVX
        F2 = DVX
  237  IF(DVX-F1)2372,238,238
 2372   F1 = DVX
  238  IF(F2-DVX)2382,239,239
 2382   F2 = DVX
  239 CONTINUE
        IF(ABS(P(M)-PF)-PDIFF)240,240,250
C
C  AVERAGE T,S,DV OVER PF +- PDIFF
C
  240   TO = TO + T(M)
        SO = SO + S(M)
        DVO = DVO + DV(M)
        XN = XN + 1.0
  250 CONTINUE
        THM = THM/XNDP
        DVM = DVM/XNDP
        DVF = DVM
        TO = TO/XN
        SO = SO/XN
        DVO = DVO/XN
C
C  CALL REGRESSION SUBROUTINE
C
 2503   CALL LSFT
C
C  IF ISSW(10)=-1 PRINT OUT REGRESSION COEFFICIENTS FOR THIS LEVEL
C
 2507  IF(ISSW(10))251,253,253
  251 DO 2510 M=M1,M2
        DVI = DVA(P(M),T(M),S(M))
        PTD = PT(M) - PM
        TTD = TT(M) - THM
 2510   WRITE(KLIST,2511)M,P(M),TH(M),S(M),DVI,DV(M),PTD,TTD
     X  ,Z1,Z2
 2511   FORMAT(1H ,I4,F7.1,2F7.3,3F7.2,F7.3,X,F6.3,F7.4,2F3.0)
        WRITE(KLIST,2515)(CP(M),M=1,N),PM
        WRITE(KLIST,2515)(CT(M),M=1,N),DVM
 2515   FORMAT(1H ,6E11.5)
C
C  IF # OF EDIT ERRORS IS LESS THAN 4, CHECK FOR ANY POINTS EXCEEDIN(
C    A2 TIMES THE STD DEV Z1 (DEFAULT IS 3), AND EXCLUDE.  RE-EDIT.
C
  253  IF(KERR.GT.3)THEN
        WRITE(KLIST,25300)
```

```
25300   FORMAT(1H ,'KERR IS GREATER THAN 3--EDIT GIVES UP')
        GO TO 2536
       ENDIF
       IF(KERR)2536,2532,2532
 2532 DO 2535 M=M1,M2
       IF(ABS(PT(M)-PM)-A2*Z1)2535,2534,2534
C
C CALL EDITING SUBROUTINE
C
 2534   CALL EDIT(KERR)
       IF(KERR)2537,220,220
 2537  IF(ISSW(3).EQ.-1)THEN
        DVI=DVA(PF,THETA(P(M),T(M),S(M),PF),S(M))
        WRITE(KLIST,25370)P(M),DVI
25370   FORMAT(1H ,F9.1,F9.2,'     FLAGGED IN COMPS, BUT NO
     X  INTERPOLATION OF T OR S')
       ENDIF
        GO TO 220
 2535 CONTINUE
 2536   KERR = 0
C
C  IF OUTPUT IS REQUESTED WRITE DATA BUFFER KBUF TO FILE *.REG
C
       IF(ISSW(13))255,260,260
  255   WRITE(KOUT)KBUF
C
C  IF ISSW(12)=-1 WRITE REGRESSION ESTIMATES TO UNIT KLIST
C
  260  IF(ISSW(12))265,267,267
  265   WRITE(KLIST,2650)IREC,PF,TO,SO,DVO,DVM,SM,THM,Z1,Z2,N,NDP
 2650   FORMAT(1H ,I4,F7.1,2F7.3,3F7.2,F7.3,X,F6.3,F7.4,2I4)
  267 CONTINUE
       IREC = IREC + 1
  270 CONTINUE
  280  IF(ISSW(13))285,295,295
  285   CLOSE(UNIT=KOUT)
       GO TO 106
      END
```

```
C AVRCP SUBPROG POTEN ***** AVRCP *****************************
C ***********************************************************
      SUBROUTINE AVRCP
C **********************************************************
C
C SUBROUTINE TO AVERAGE SPECIFIC VOLUME AND COEFFICIENTS.
C
C JUNE 28 1976 N FOFONOFF
C MODIFIED FOR VAX DISC CTD78 FORMAT 15DEC80 N. BRAY
C
      INCLUDE 'COMPOTEN.FOR'
C
C DIMENSION
C
      DIMENSION CR(100,15),SWGT(100),CPM(8),SVA(100)
      DIMENSION APE(100),SVI(100),EOB(100)
      DIMENSION VMIN(100),VMAX(100)
C
C CHARACTER
C
      CHARACTER*12 FNAME(600),GNAME
      CHARACTER*1 IV1,IV2,IV3
C
C EQUIVALENCE
C
      EQUIVALENCE (CR,IHDG),(PF,VR)
      EQUIVALENCE (CR(1,9),SWGT),(CR(1,10),SVA),(CR(1,11),SVI)
      EQUIVALENCE (CR(1,12),APE),(VMIN,CR(1,12)),(VMAX,CR(1,13))
C
C  READ IN STATION #'S TO BE AVERAGED.  ENCODE INTO CORRESPONDING
C    FILE NAMES.
C
       IF (ISW.EQ.1)THEN
       REWIND 12
      DO 61 K=1,1000
        READ(12,610,END=62)M,FNAME(K),WT(K)
  610   FORMAT(I4,A12,F5.2)
   61 CONTINUE
   62 CONTINUE
        JSTN=K-1
        GNAME(9:12)='.AVG'
       ENDIF
C
C  AVERAGING
C
  100   ISW2 = ISW - 2
       IF(ISW2)101,113,113
C
C BRANCH I--INITIAL AVERAGING--BEGINS HERE
  101 DO 110 J=1,100
      DO 110 I=1,13
  110 CR(J,I) = 0.0
  112   IRMX = 0
C
C  BRANCHES 2 AND 3 BEGIN HERE
C
  113   NST = 0
C
C  OPEN APPROPRIATE FILE, READ HEADER
C
      DO 1200 KK=1,JSTN
```

```
        OPEN(UNIT=KOUT,NAME=FNAME(KK),READONLY,TYPE='OLD',
     *  FORM='UNFORMATTED')
        READ(KOUT)KHDG
        NSW=5
C
C  CHECK IF DATA SELECTION PARAMETERS ARE SATISFIED. IF NOT,
C   NSW IS RETURNED FROM DATA AS 4, AND STATION IS SKIPPED.
C
        CALL DATA(KOUT,NSW)
       IF(NSW.NE.5)GO TO 1200
        NST=NST+1
        DELB=0.0
        PPR=0.0
        B2=0.0
        B3=0.0
C
C  IF ISSW(15)=-1 WEIGHTS ARE TAKEN FROM JSHP.PTN FILE; OTHERWISE,
C   THEY ARE SET TO 1.
C
       IF(ISSW(15))1350,1357,1357
 1350   WGT = WT(K)
       GO TO 1370
 1357   WGT = 1.0
 1370  IF(ISW2)120,120,138
C
C  BRANCHES 1 AND 2 CONTINUE HERE FROM STATEMENT #1370
C
  120   READ(KOUT,END=160)KBUF
       GO TO 140
C
C  BRANCH 3 (WRITE OUT AVERAGED FILES) CONTINUES HERE FROM STATEMENT #1370
C
  138  IF(ISSW(13))139,120,120
  139 CONTINUE
C
C  OPEN NEW FILE NAMED *.AVG CORRESPONDING TO INPUT *.REG, FOR OUTPUT
C   ON UNIT 11.
C
        GNAME(1:8)=FNAME(KK)(1:8)
        OPEN(UNIT=11,NAME=GNAME,TYPE='NEW',FORM='UNFORMATTED')
C
C  WRITE HEADER TO *.AVG
C
        WRITE(11)KHDG
       GO TO 120
C
C  BRANCHES 1 AND 2 CONTINUE HERE FROM STATEMENT #120
C
  140  IF(IREC-IRMX)146,142,142
  142   IRMX = IREC
  146  IF(ISW2)155,147,300
C
C  BRANCH 2 CONTINUES HERE FROM STATEMNT #146
C
  147 CONTINUE
 1475  DEL = SVA(IREC) - DVM
        CALL COEFF(1.0,DEL,CP,CPM,N)
        CPM(1) = CPM(1) + PM - PF
      DO 150 J=1,N
  150 CR(IREC,J) = CR(IREC,J) + WGT*CPM(J)
C
```

```
C  BRANCH 2 RETURNS TO READ NEXT DATA RECORD FROM *.REG
C
       GO TO 120
C
C  BRANCH 1 CONTINUES FROM STATEMENT #146
C
  155 CONTINUE
  158   SVA(IREC) = SVA(IREC) + WGT*DVM
        VMIN(IREC) = VMIN(IREC) + WGT*F1
        VMAX(IREC) = VMAX(IREC) + WGT*F2
        SVI(IREC) = SVI(IREC) + WGT*DVZRO(PF,DVM,PM,CP,N,NDP,F1,F2,
     *  ISHP,KCAST,ICON,DELP)
        SWGT(IREC) = SWGT(IREC) + WGT
C
C  BRANCH 1 RETURNS TO READ NEXT DATA RECORD FROM *.REG
C
       GO TO 120
  160   CLOSE(UNIT=KOUT)
        CLOSE(UNIT=11)
       IF(KK.LT.JSTN)GO TO 1200
       IF(ISW2)161,1610,420
C
C  BRANCH 1 CONTINUES FROM PREVIOUS STATEMENT
C
  161 DO 1605 J=1,IRMX
        VMIN(J) = VMIN(J)/SWGT(J)
        VMAX(J) = VMAX(J)/SWGT(J)
        SVI(J) = SVI(J)/SWGT(J)
 1605   SVA(J) = SVA(J)/SWGT(J)
       IF(ISW2)180,180,300
C
C  BRANCH 2 CONTINUES FROM STATEMENT PRECEEDING #161
C  AVERAGE REGRESSION COEFFICIENTS
C
 1610   DELB = 0.0
        PPR = 0.0
        KF = 2
        N = NSC(NSECTION+1)
        NDP = NSC(2*NSECTION+1)
       IF(ISSW(11).EQ.-1)WRITE(KLIST,16230)
      DO 163 J=1,IRMX
       IF(JPR(J,NPR,NSECTION)-NSC(KF))1612,1611,1612
 1611   N = NSC(KF+NSECTION)
        NDP = NSC(KF+2*NSECTION)
        KF = KF + 1
 1612 DO 162 I=1,N
  162 CPM(I) = CR(J,I)/SWGT(J)
        PF = JPR(J,NPR,NSECTION)
       IF(ISSW(11))1621,1624,1624
 1621   WRITE(KLIST,1623)PF,(CPM(I),I=1,N)
 1623   FORMAT(1H ,F6.0,8G11.4)
16230   FORMAT(1H ,'AVERAGED REGRESSION COEFFICIENTS:',/,
     *  'PRESSURE',2X,'CP(1)',5X,'CP(2)',5X,'CP(3)',5X,'CP(4)',
     *  5X,'CP(5)',5X,'CP(6)')
 1624   DVI = SVA(J)
        DVF = DVZRO(PF,DVI,PF,CPM,N,NDP,VMIN(J),VMAX(J),ISHP,KCAST,ICO
        DELA = SVI(J) - DVF
        SVA(J) = DVF
        DELPI = PF - PPR
        EOB(J) = DPDV(DVF,DVI,CPM,N,VMIN(J),VMAX(J))
        FOB(J) = 1./EOB(J)
```

```
            B2 = .50968E-6*(PF*DELA+PPR*DELB)*DELPI
            B3 = 0.5E-5*(DELA+DELB)*DELP1
            IF(J-1)1625,1620,1625
  1620      APE(1) = B2
            VMAX(1) = B3
           GO TO 1626
  1625      APE(J) = APE(J-1) + B2
            VMAX(J) = VMAX(J-1) + B3
  1626      DELB = DELA
            PPR = PF
   163 CONTINUE
C
C  BRANCHES 1 AND 2 CONTINUE HERE FROM STATEMENT #120; END QUALIFIER
C    IMPLIED END OF STATION.
C  IF ISSW(12)=-1 PRINT OUT AVERAGES
   180   IF(ISSW(12))182,400,400
   182    WRITE(KLIST,1832)
  1832    FORMAT(1H ,'IREC NST  PF',6X,'SUM OF',5X,'DVM',5X,'DVF',5X,
       *  'DV',5X,'DV',/,1H ,19X,'WGTS',6X,'BAR',5X,'BAR',5X,'MIN',5X,
       *  'MAX')
          DO 183 I=1,IRMX
            KPR = JPR(I,NPR,NSECTION)
           IF(ISW.EQ.1)WRITE(KLIST,1835)I,NST,KPR,(CR(I,K),K=9,13)
   183   IF(ISW.EQ.2)WRITE(KLIST,1835)I,NST,KPR,(CR(I,K),K=9,11)
  1835    FORMAT(1H ,2I4,I5,5F10.4)
           GO TO 400
C
C  BRANCH 3 CONTINUES HERE FROM STATEMENT #146
C
   300    DVF = SVA(IREC)
          F3 = SVI(IREC)
          PI = POLY(DVF,DVM,CP,N,F1,F2) + PM
         IF(PI)301,302,302
   301    PI = 0.0
   302    THF = POLY(DVF,DVM,CT,N,F1,F2) + THM
   303    SF = DVZRO(PF,DVM,PM,CP,N,NDP,F1,F2,ISHP,KCAST,ICON)
          DELA = SF - DVF
          DELP1 = PF - PPR
         IF(IREC-1)305,304,305
   304    DELB = DELA
   305    B2 = B2 + 0.5E-5*(DELA+DELB)*DELP1
          B3 = B3 + 0.50968E-6*(PF*DELA+PPR*DELB)*DELP1
          DELB = DELA
          PPR = PF
          DH = EOB(IREC)
          PE = B3
          XPE = APE(IREC)
C
C  IF OUTPUT REQUESTED WRITE DATA TO FILE *.AVG
C
         IF(ISSW(13))310,316,316
   310    WRITE(11)KBUF
   316   IF(JSW)317,320,3245
C
C  IF MAP FORMAT OUTPUT REQUESTED INITIALIZE AND REQUEST INPUT
C
   317   IF(ISSW(14).EQ.-1)THEN
  3170    FORMAT(1H ,'N1,N2,N3,KTO,IYR,ITM,IV1,IV2,IV3',6I5,3(X,A1))
          KTO = 1
          IYR = 73
          ITM = 0
```

```
             IV1='X'
             IV2='Y'
             IV3='Z'
             N1 = 1
             N2 = 2
             N3 = 3
             NV1 = 6
             NV2 = 4
             NV3 = 8
             WRITE(KTTX,3170)N1,N2,N3,KTO,IYR,ITM,IV1,IV2,IV3
             READ(KIN,*)N1,N2,N3,KTO,IYR,ITM
             READ(KIN,3175)IV1,IV2,IV3
             WRITE(KTTX,3176)NV1,NV2,NV3
             READ(KIN,*)NV1,NV2,NV3
       3175  FORMAT(3A1)
       3176  FORMAT(1H ,'NV1,NV2,NV3',3I5)
            ENDIF
             JSW = 0
C
C  N1 TO N3 ARE EFFECTIVELY IGNORED, UNLESS ISSW(8)=-1. SEE STATEMENT #450.
C
        320 IF(IREC-N1)321,450,321
        321 IF(IREC-N2)322,470,322
        322 IF(IREC-N3)120,324,120
        324 CONTINUE
C
C  SET VALUES FROM MAP FORMAT VARIABLES
C
       3245  VR1 = VR(NV1)
             VR2 = VR(NV2)
             VR3 = VR(NV3)
C
C  IF ISSW(10)=-1 WRITE MAP VARIABLES TO UNIT KLIST (NOT IN MAP FORMAT--FOR
C    CHECK PURPOSES ONLY).
C
        325 IF(ISSW(10))326,327,327
        326  WRITE(KLIST,3150)ICON,IREC,ISHP,ICAST,PF,PI,TO,VR1,VR2,VR3,
          X  DH,PE
C
C  IF MAP FORMAT NOT REQUESTED, RETURN TO STATEMENT #120 TO READ NEXT DATA
C    RECORD.
C
C  MAP FORMAT WRITTEN TO UNIT KTO
C
        327 IF(ISSW(14))330,120,120
        330  LFILE = 0
             IPF = PF
             XLG = XLONG
        335  WRITE(KTO,3300)(LBBL(M),M=1,2),ISHP,ICAST,LFILE,IPF,
          X  XLAT,XLG,IYR,JDAY,ITM,IV1,VR1,IV2,VR2,IV3,VR3
             IF(LFILE)550,120,550
       3150  FORMAT(1H ,I3,X,2I2,I4,X,2F5.0,2F7.3,2F8.3,F6.3,F7.2)
       3300  FORMAT(2A4,I1,I4,X,I1,I5,F7.2,F8.2,X,I2,I4,I2,X,3(A1,F9.4))
C
C  ISSW(8)=-1 ALLOWS SPECIAL FUNCTIONS TO BE COMPUTED--SUBTRACTING VALUES
C    AT ONE LEVEL FROM ANOTHER BEFORE OUTPUTING IN MAP FORMAT.
C
        450 IF(ISSW(8))455,324,324
        455  VR1 = VR(NV1)
             VR2 = VR(NV2)
             VR3 = VR(NV3)
```

```
      GO TO 120
  470 IF(ISSW(8))475,324,324
  475  VR1 = VR(NV1) - VR1
       VR2 = VR(NV2) - VR2
       VR3 = VR(NV3) - VR3
      GO TO 325
C
C  BRANCHES 1 AND 2 CONTINUE FROM 183 OR 180
C
C  IF ISSW(7)=0 CONTINUE THROUGH BRANCH 3 AUTOMATICALLY.
C
  400 IF(ISSW(7))550,410,410
  410  ISW = ISW + 1
      IF(ISW-3)100,100,420
C
C  BRANCH 3 CONTINUES HERE FROM STATEMENT PRECEEDING #161
C
C IF MAP OUTPUT REQUESTED SET LFILE TO 1 TO INDICATE EOF IN MAP FORMAT.
C
  420 IF(ISSW(14))422,560,560
  422  LFILE = 1
      GO TO 335
  550 CONTINUE
 1200 CONTINUE
      IF(NSW.NE.5.AND.ISW.EQ.1)GO TO 161
      IF(NSW.NE.5.AND.ISW.EQ.2)GO TO 1610
      IF(NSW.NE.5.AND.ISW.EQ.3)GO TO 420
  560  RETURN
     END
```

```
C DATA SUBR POTEN ***** DATA ***********
C ****************************************
      SUBROUTINE DATA(KUN,NSW)
C ****************************************
C
C TO SELECT AND ACCESS CTD78 FORMAT DATA FROM VAX DISC FORMAT
C  ACCESSES VARIOUS MILLARD SUBROUTINES FOUND IN CTDATA/LIB
C
C
C JAN 6 1976 N FOFONOFF
C MODIFIED FOR CTD78 FORMAT INPUT 15 DEC 80.  N BRAY
C
      INCLUDE 'COMPOTEN.FOR'
C
C INCLUDE MILLARD DIMENSION STATEMENTS
C
      INCLUDE 'IDXREC.DIM'
C
C CHARACTER
C
      CHARACTER*12 GNAME
C
C PROGRAM
C
       IF(NSW.EQ.5) GO TO 80
       IF(NSW.EQ.2) GO TO 30
       IF(NSW)1,20,5
C
C  NSW LESS THAN ZERO:  INITIALIZE SELECTION PARAMETERS
C
    1   CONTINUE
        JDO = 0
        DAY1 = 0.
        DAY2 = 365.
        XEMN = -180.0
        XEMX =  180.0
        XNMN = -90.0
        XNMX =  90.0
        XLTO = 40.00
        XLGO = 70.00
        JSTN = 1
      RETURN
C
C  NSW=0: LIST OR CHANGE SELECTION PARAMETERS
C
   20 CONTINUE
  172   WRITE(KLIST,173)DAY1,DAY2,JDO
  173   FORMAT(1H ,5HDAY1:F8.3,X,5HDAY2:,F8.3,X,4HJDO:,I4)
        READ(KIN,*)DAY1,DAY2,JDO
  174   WRITE(KLIST,175)XEMN,XEMX,XNMN,XNMX
  175   FORMAT(1H ,7HE-N LIM,4F7.2)
        READ(KIN,*)XEMN,XEMX,XNMN,XNMX
        WRITE(KLIST,177)XLTO,XLGO
  177   FORMAT(1H ,8HORIGIN: ,2(X,F8.3))
        READ(KIN,*)XLTO,XLGO
      RETURN
C
C NSW = 2:  READ FROM FILE STATIONS.PTN INFORMATION  TO IDENTIFY
C    STATIONS
C
   30 CONTINUE
```

```
       IF(KBR.EQ.13)THEN
        WRITE(KTTX,310)
        READ(KIN,300)PROVER
        WRITE(KTTX,320)
        READ(KIN,330)ISHP,ICRUIS,IPROJ
  300   FORMAT(A)
  310   FORMAT(1H ,'ENTER SUBDIRECTORY VERSION #')
  320   FORMAT(1H ,'ENTER SHIP CODE,CRUISE #,PROJ #')
  330   FORMAT(A2,2I3)
       ENDIF
C
C MILLARD HEADER RELATED SUBROUTINES
C
        CALL PVER(PROVER)
        CALL CRUISE(ISHP,ICRUIS,IPROJ)
        CALL STATION(0,0,KTP)
        CALL INDEX(11)
        LREC = IDXREC(9)
        LLREC = LREC
        KKST = 0
        M=0
      RETURN
C
C NSW = 1: READ STATION HEADER, CHECK AGAINST DATA SELECTION CRITERIA,
C   AND READ TEMPERATURE AND SALINITY INTO DATA ARRAY.
C
    5 CONTINUE
       IF(KUN.GT.LLREC) GO TO 620
        CALL RECIDX(KUN)
        XLAT=SLAT()
        XLONG=SLNG()
        IPR = PMIN
        XN = NTOT-1
        LPR = XN*PRSINT+PMIN
        LBBL(1)=IFHED(3)
        ENCODE(4,53,LBBL(2))IFHED(4)
        ENCODE(4,54,LBBL(3))IFHED(5)
   53   FORMAT(I3,'-')
   54   FORMAT(I4)
C
C  COMPUTE JULIAN YEAR DAY
C  CHECK AGAINST SELECTION PARAMETERS
C
        IDAY=KDAY(IDA,IMO,IYR)-KDAY(31,12,IYR-1)
        DAY = FLOAT(IDAY) + FLOAT(ISTME)/2400.
       IF(DAY-DAY1)620,602,602
  602  IF(DAY-DAY2)604,604,620
  604 CONTINUE
C
C  CHECK LAT AND LONG AGIANST SELECTION PARAMETERS.
C
       IF(XLONG-XEMN)620,606,606
  606  IF(XLONG-XEMX)608,608,620
  608  IF(XLAT-XNMN)620,610,610
  610  IF(XLAT-XNMX)616,616,620
  616   LTYPE = 1
        ICON = ICAST
        DAY = DAY + JDO
        KCAST = ISTAS
        JRMAX = NTOT
C
```

```
C  IF ISSW(11)=-1 WRITE OUT HEADER INFORMATION ON UNIT KLIST
C
            IF(ISSW(11))6160,620,620
 6160        WRITE(KLIST,8000)(LBBL(K),K=1,3),XLAT,XLONG,XLTO,XLGO
             WRITE(KLIST,8005)LTYPE,MHDR,ISHP,ISTAS,ICON,DAY,IPR
C
C  IF ISSW(12)=-1, WRITE OUT HEADINGS FOR OUTPUT WHICH IS WRITTEN IN COMPS.
C
            IF(ISSW(12).EQ.-1)WRITE(KLIST,8010)
 8010        FORMAT('0IREC    PF      TO      SO     DVO     DVM      SM      THM
         X   Z1      Z2      N   NDP')
 8000        FORMAT(/,1H ,3A4,4F8.3)
 8005        FORMAT(1H ,' TYPE MDHR  SHIP ISTN CAST DAY  IPR',/,
         *   2I5,2X,A2,2X,2I5,F8.3,2I5)
 6005        FORMAT(I5,X,3A4,I2,I4,X,I5,I5,F7.2,F8.2,X,I2,I4,F8.5)
C
C  MILLARD SUBROUTINE TO FILL DATA ARRAY WITH TEMP AND SALINITY
C   DATA FOR ALL OBSERVATIONS.
C
             WGT=1.
            IF(ISSW(13).EQ.-1)THEN
             M=M+1
             JICR=ICRUIS
             JIST=ISTAS
             CALL LZ(JICR)
             CALL LZ(JIST)
             ENCODE(12,52,GNAME)ISHP,JICR,JIST
   52        FORMAT(A2,A3,A3,'.REG')
             WRITE(12,8020)M,GNAME,WGT
 8020        FORMAT(I4,A12,F5.2)
             OPEN(UNIT=KOUT,NAME=GNAME,TYPE='NEW',FORM='UNFORMATTED')
             WRITE(KOUT)KHDG
            ENDIF
             CALL  DATIDX(KUN)
             CALL  GETDAT(KTP,DATAX,3300,2)
  620 RETURN
C
C  NSW=5: CHECK ONLY LAT AND LONG OF HEADER ALREADY READ AGAINST
C   SELECTION PARAMETERS.
C
   80  IF(XLONG-XFMN)85,87,87
   87  IF(XLONG-XFMX)89,89,85
   89  IF(XLAT-XNMN)85,83,83
   83  IF(XLAT-XNMX)82,82,85
   82 CONTINUE
      RETURN
C
C  NSW=4 IMPLIES SELECTION CRITERIA ABOVE NOT MET.
C
   85   NSW=4
      RETURN
      END
C **************** SLAT FUNCTION *****************************
      REAL FUNCTION SLAT
C ************************************************************
      INCLUDE 'IDXREC.DIM'
C FUNCTION RETURNS DECIMAL DEGREE VALUE SIGNED - FOR SOUTH & WEST
      XLAT=ILTSD
      XLATM=ILTSM
        XLATM=XLATM/6000.
      SLAT=XLAT+SIGN(XLATM,XLAT)
```

```
15449            RETURN
15456            ENTRY SLNG
15463            XLAT=ILNSD
15470            XLATM=ILNSM
15473              XLATM=XLATM/6000.
15477            SLNG=XLAT+SIGN(XLATM,XLAT)
15484            RETURN
15491            END
15500      C **************** SUBROUTINE LZ(IA) ********************************
15503            SUBROUTINE LZ(IA)
15504      C ******************************************************************
15506            INTEGER IA(1),IW(1)
15512            IF(IA(1).GE.100) GO TO 100
15515            IF(IA(1).GE.10) GO TO 10
15518            IF(IA(1).GE.0) GO TO 1
15521            RETURN
15524        100 CONTINUE
15527            ENCODE(3,2,IW(1)) IA(1)
15530            IA(1)=IW(1)
15533          2 FORMAT(1I3)
15536            RETURN
15539         10 CONTINUE
15542            ENCODE(3,3,IW(1)) IA(1)
15545            IA(1)=IW(1)
15548          3 FORMAT(1H0,I2)
15551            RETURN
15554          1 CONTINUE
15557            ENCODE(3,4,IW(1)) IA(1)
15560          4 FORMAT(2H00,I1)
15563            IA(1)=IW(1)
15566            RETURN
15584            END
```

```
C POTENSUB.FOR FILE: SUBROUTINES FOR POTEN,PEPLT POTENTIAL ENERGY
C PROGRAMS. VAX VERSION. N.BRAY.
C *****************************************************************
      SUBROUTINE SMINV(A,N,P,Q,MR,IFAIL)
C *****************************************************************
C
C TO INVERT SYMMETRIC MATRIX FOR TRIANGULAR SECTION ARRANGED
C IN A LINEAR ARRAY A(J).
C FROM SYMINV2,,,CACM #150 BY RUTISHAUSER VIA J. MALTAIS.
C
C APRIL 27 1975 N. FOFONOFF
C
      DIMENSION A(1),P(1),Q(1),MR(1)
C
      IFAIL = 0
      DO 10 I=1,N
   10 MR(I) = 0
C SEARCH FOR PIVOT
      DO 100 I=1,N
      BIGAJ = 0.0
      JJ = -N
      DO 20 J=1,N
      JJ = JJ+N-J+2
      B = ABS(A(JJ))
      IF(MR(J))20,12,20
   12 IF(B-BIGAJ)20,20,14
   14 BIGAJ = B
      K = J
      KK = JJ
   20 CONTINUE
      IF(BIGAJ)16,15,16
   15 IFAIL = 1
      RETURN
C PREPARATION OF ELIMINATION
   16 MR(K) = 1
      Q(K) = 1./A(KK)
      P(K) = 1.0
      A(KK) = 0.0
      KM1 = K-1
      IF(KM1)15,19,160
  160 JK = K - N
      DO 30 J=1,KM1
      JK = JK+N-J+1
      P(J) = A(JK)
      IF(MR(J))18,17,18
   17 Q(J) = -A(JK)*Q(K)
      GO TO 30
   18 Q(J) = A(JK)*Q(K)
   30 A(JK) = 0.0
   19 KP1 = K+1
      KJ = KK
      IF(KP1-N)21,21,41
   21 DO 40 J=KP1,N
      KJ = KJ + 1
      IF(MR(J))34,32,34
   32 P(J) = A(KJ)
      GO TO 35
   34 P(J) = -A(KJ)
   35 Q(J) = -A(KJ)*Q(K)
   40 A(KJ) = 0.0
C ELIMINATION PROPER
```

```
   41 JK = 0
      DO 50 J=1,N
      DO 50 K =J,N
      JK = JK + 1
   50 A(JK) = A(JK) + P(J)*Q(K)
  100 CONTINUE
  150 RETURN
      END
C SUBROUTINE TO INPUT PARAMETERS FOR REGRESSION IN POTEN
C POTENTIAL ENERGY PROGRAM
C
C**********************************************************
      SUBROUTINE PARAM
C**********************************************************
C
      INCLUDE 'COMPOTEN.FOR'
C
      INTEGER PFDELTA,PDELTA
C
C
      WRITE(KTTX,10)
 10   FORMAT('OINPUT NEW PARAMETERS?')
      IF(NOYES(KIN,KTTX).NE.1)RETURN
 15   WRITE(KTTX,20)
 20   FORMAT('OENTER THE NUMBER OF SECTIONS (GROUPS OF LEVELS WITH
     X THE SAME PARAMETERS)')
      READ(KIN,*)NSECTION
     IF(NSECTION.GT.19)THEN
      WRITE(KTTX,22)
 22   FORMAT('OMAXIMUM ALLOWED IS 19')
      GO TO 15
     ENDIF
      WRITE(KTTX,25)
 25   FORMAT('OENTER THE PRESSURE FOR THE FIRST LEVEL:')
      READ(KIN,*)INITIALP
      NLEVP=1
      NPREV=-1
       NSE=NSECTION+1
      DO 1000 I=1,NSECTION
      I2=I+NSE
      I3=I+2*NSE
       WRITE(KTTX,100)I
 100  FORMAT('OFOR SECTION',I4,'  ENTER THE INTERVAL IN DB BETWEEN
     X LEVELED SURFACES:')
      READ(KIN,*)PFDELTA
       WRITE(KTTX,120)
 120   FORMAT('OENTER THE INTERVAL SIZE IN DB FOR THE REGRESSION:')
       READ(KIN,*)PDELTA
       WRITE(KTTX,140)
 140   FORMAT('OENTER THE FIRST PRESSURE IN THE NEXT SECTION:')
        READ(KIN,*)IP2
       NLEVEL=(IP2-INITIALP)/PFDELTA
       WRITE(KTTX,160)
 160  FORMAT('OENTER THE NUMBER OF TERMS IN THE REGRESSION:',/,
     X '(N=2 IMPLIES A LINEAR FIT; MAXIMUM N IS 8)')
        READ(KIN,*)N
     IF(N.GT.8)N=8
C****** COMPUTE NPR(I) ******
        NPR(I)=NLEVEL+NLEVP
C****** COMPUTE NPR(I2)
        NPR(I2)=PFDELTA
```

```
C****** COMPUTE NSC(I) ******
      IF(I.FQ.1)THEN
         NSC(I)=INITIALP-PFDELTA
         IPREV=INITIALP
      ENDIF
        NSC(I+1)=IPREV+NLEVEL*PFDELTA
C****** COMPUTE NPR(I3) ******
        NPR(I3)=NLEVP-(IPREV/PFDELTA)
        NPREV=NPR(I3)
        IPREV=NSC(I+1)
         NLEVP=NPR(I)
        INITIALP=IPREV
C****** COMPUTE NSC(I2)
        NSC(I2)=N
C****** COMPUTE NSC(I3) ******
        NSC(I3)=PDELTA/DELP
 1000 CONTINUE
        ITOTAL=IPREV
        JMAX=NLEVP
        WRITE(KTTX,200)NSECTION,NLEVP,ITOTAL
  200   FORMAT('OA TOTAL OF',I4,'SECTIONS;',I6,'LEVELS; THE DEEPEST
     X  LEVEL IS AT',I6,'DB.')
        WRITE(KTTX,220)
  220   FORMAT('OENTER MAXIMUM DEPTH OF THE DATA:')
        READ(KIN,*)ITMAX
        I=NSECTION+1
        I2=2*I
        I3=3*I
        NPR(I)=NLEVP+5
        NPR(I2)=(ITMAX+500)/(NLEVP-NPREV)
        NPR(I3)=NPREV
        NSC(I)=IPREV
        NSC(I2)=N
        NSC(I3)=PDELTA/DELP
        NSECTION=NSECTION+1
      RETURN
      END
C COEFF SUBR ***** PTSB1 ***************************************
      SUBROUTINE COEFF(A,B,C,D,N)
C **************************************************************
C
C COMPUTES COEFFICIENTS FOR A LINEAR TRANSFORMATION X=AX+B
C FOR POLYNOMIAL OF ORDER N-1. INPUT ARRAY C, OUTPUT D.
C
C OCT 22 1975 N. FOFONOFF
C
      DIMENSION C(1),D(1)
C
      DO 25 I=1,N
      R = 1.0
      S = C(I)
      NMI = N - I
      IF(NMI)12,12,5
    5 DO 10 J=1,NMI
      IPJ = I + J
      R = (FLOAT(IPJ-1)/FLOAT(J))*B*R
   10 S = S + R*C(IPJ)
   12 IM1 = I - 1
      IF(IM1)15,15,20
   15 R = 1.0
      GO TO 25
   20 R = A**IM1
   25 D(I) = S*R
      RETURN
      END
C
C R(N) FCN ***** PTSB1 *******
      FUNCTION R(N)
C ***************************
C
      INTEGER*2 IA,N
      IA = "77777
      IF(N)1,2,2
    1 R = FLOATI(IIAND(N,IA)) + 32768.
      RETURN
    2 R = FLOAT(N)
      RETURN
      END
C
C *********************************
      SUBROUTINE EDIT(JERR)
C *********************************
C
C EDIT TEMP AND SALINITY IN REGRESSION TABLE.
C
C JAN 28 1975 N FOFONOFF
C
      INCLUDE 'COMPOTEN.FOR'
      EQUIVALENCE (PDIFF,A1)
C
        IERR = 0
        DVPMAX = -.12
      DO 10 M=M1,M2
        DV(M) = P(M)
        P(M) = S(M)
   10   TH(M) = T(M)
   15   DVM = 0.0
        PM = 0.0
        THM = 0.0
        XNDP = NDP
      DO 20 M=M1,M2
        DVM = DVM + DV(M)
        PM = PM + P(M)
        THM = THM + TH(M)
        PT(M) = P(M)
        TT(M) = TH(M)
   20 CONTINUE
        DVM = DVM/XNDP
        PM = PM/XNDP
        THM = THM/XNDP
       CALL LSFT
        KERR = 0
      DO 60 M=M1,M2
       IF(ABS(PT(M)-PM)-A3*Z1)40,30,30
   30   CORR = POLY(DV(M),DVM,CP,N,0.0,6000.0) + PM
        KERR = 1
       IF(ISSW(3))31,32,32
   31   DELTA = CORR-P(M)
        S1 = DATAX(M-1,2)
        S2 = DATAX(M,2)
        S3 = DATAX(M+1,2)
        THI = THETA(DV(M-1),DATAX(M-1,1),S1,PF)
        TH2 = THETA(DV(M),DATAX(M,1),S2,PF)
        TH3 = THETA(DV(M+1),DATAX(M+1,1),S3,PF)
        DV1 = DVA(PF,TH1,S1)
        DV2 = DVA(PF,TH2,S2)
        DV3 = DVA(PF,TH3,S3)
        DVP1 = (DV1-DV2)/DELP
        DVP2 = (DV2-DV3)/DELP
        WRITE(KLIST,3100)DV(M),P(M),CORR,DELTA,DVP1,DVP2
   32   P(M) = CORR
   40  IF(ABS(TT(M)-THM)-A3*Z2)55,50,50
   50   CORR = POLY(DV(M),DVM,CT,N,0.0,6000.0) + THM
        KERR = 1
       IF(ISSW(3))51,52,52
   51   DELTA = CORR-TH(M)
        S1 = DATAX(M-1,2)
        S2 = DATAX(M,2)
        S3 = DATAX(M+1,2)
        TH1 = THETA(DV(M-1),DATAX(M-1,1),S1,PF)
        TH2 = THETA(DV(M),DATAX(M,1),S2,PF)
        TH3 = THETA(DV(M+1),DATAX(M+1,1),S3,PF)
        DV1 = DVA(PF,TH1,S1)
        DV2 = DVA(PF,TH2,S2)
        DV3 = DVA(PF,TH3,S3)
        DVP1 = (DV1-DV2)/DELP
        DVP2 = (DV2-DV3)/DELP
        WRITE(KLIST,3100)DV(M),TH(M),CORR,DELTA,DVP1,DVP2
   52   TH(M) = CORR
   55  IF(KERR.EQ.0.AND.IERR.EQ.0)THEN
        JERR=-2
       ENDIF
        IERR = IERR + 1
   60 CONTINUE
        IF(KERR)70,70,15
   70 DO 75 M=M1,M2
        S(M) = P(M)
        T(M) = TH(M)
   75   P(M) = DV(M)
        JERR = JERR + 1
   80 RETURN
 3100   FORMAT(F7.1,F9.3,'  REPLACED BY: ',F9.3,'   CHANGE IS:',F9.3,
     *  '    SP. VOL. GRADIENTS: ABOVE = ',F9.3,'  BELOW = ',F9.3)
      END
C
C ******************************
      SUBROUTINE LSFT
C ******************************
C
C LEAST SQUARES REGRESSION SUBROUTINE FOR POTEN.
C
C MAR 6 1976 N FOFONOFF
C
      INCLUDE 'COMPOTEN.FOR'
C
    1   NA = N*(N+1)/2
       L = 1
      DO 10 I=1,NA
   10 CO(I) = 0.0
      DO 12 I=1,N
        CP(I) = 0.0
   12 CT(I) = 0.0
   15 DO 20 I=1,N
```

```
        BP(I) = 0.0
        BT(I) = 0.0
   20 CONTINUE
      DO 8 I=M1,M2
        X = DV(I) - DVM
      DO 100 J=1,N
       IF(J-1)90,90,95
   90   B(J) = 1.0
       GO TO 100
   95   B(J) = X**(J-1)
  100 CONTINUE
        JK = 0
        X = PT(I) - PM
        XT = TT(I) - THM
      DO 8 J=1,N
        BP(J) = BP(J) + B(J)*X
        BT(J) = BT(J) + B(J)*XT
       IF(L-1)105,105,8
  105 DO 7 K=J,N
        JK = JK + 1
    7 CO(JK) = CO(JK) + B(J)*B(K)
    8 CONTINUE
       IF(L-1)173,173,174
  173   CALL SMINV(CO,N,B,BA,MR,IFAIL)
  174 DO 200 M=1,N
        SP = 0.0
        ST = 0.0
        JM = M-N
      DO 1745 J=1,N
       IF(J-M)1740,1740,1742
 1740   JM = JM + N - J + 1
       GO TO 1744
 1742   JM = JM + 1
 1744   SP = SP + CO(JM)*BP(J)
 1745 ST = ST + CO(JM)*BT(J)
        CP(M) = CP(M) + SP
  200   CT(M) = CT(M) + ST
C COMPUTE RESIDUALS
  175   RP = 0.0
        RT = 0.0
      DO 185 I=M1,M2
        FP = 0.0
        FT = 0.0
        X = DV(I) - DVM
      DO 180 J=1,N
        NJ = N-J+1
        FP = FP*X + CP(NJ)
  180 FT = FT*X + CT(NJ)
        SP = P(I) - FP
        PT(I) = SP
        ST = TH(I) - FT
        TT(I) = ST
       IF(L-KSW)185,183,183
  183   RP = (SP-PM)**2 + RP
        RT = (ST-THM)**2 + RT
  185 CONTINUE
        L = L + 1
       IF(L-KSW)15,15,195
  195   XN = NDP - N
        Z1 = SQRT(RP/XN)
        Z2 = SQRT(RT/XN)
```

```
          IF(ISSW(6))300,350,350
    300 DO 310 I=1,N
          II = I + ((2*N-I)*(I-1))/2
          B(I) = ABS(CP(I))/(Z1*SQRT(ABS(CQ(II))))
          BP(I) = ABS(CT(I))/(Z2*SQRT(ABS(CQ(II))))
    310 CONTINUE
          WRITE(KLIST,3100)PF,N,NDP,(B(K),K=1,N)
          WRITE(KLIST,3110)(BP(K),K=1,N)
    350 RETURN
   3100   FORMAT(F6.0,I4,I3,8F8.3)
   3110   FORMAT(13X,8F8.3)
        END
```

Appendix C.
Program Listings for PEPLT

```
 100    PEPLT/PEPLS:  SHORT DOCUMENTATION
 200    KBR       ISW       JSW       KLIST    DESCRIPTION
 250    0         -         -         -        SHORT DOCUMENTATION
 300    1         0         -         -        CALL TABLE SUBROUTINE: LIST,
 400                                           PLOT, OUTPUT IN MAP FORMAT
 450                                           PLOT IS:
 466            X = A1*VRBL(NX1)+A2*VRBL(NX2)+A3*VRBL(NX3)+A4*C(IREC,1)
 482            Y = B1*VRBL(NY1)+B2*VRBL(NY2)+B3*VRBL(NY3)+B4*C(IREC,2)
 488    1         1         -         -        CHANGE PARAMETERS FOR PLOT
 494              2         -         -        INITIALIZE PARAMETERS FOR PLOT
 500    2         -         -         -        CHANGE DATA SELECTION VARIABLES
 600    3         -         -         -        CHANGE PLOT PARAMETERS
 700    4         -         -         -        CALL AVRGS SUBROUTINE: HORIZONTAL
 800                                           AVERAGES. FOR DETAILED DOCUMENTA-
 900                                           TION, ACCESS KBR=0 AFTER ENTERING
 950                                           AVRGS BRANCH.
1000    5         -         -         -        SET ISSW (SWITCH) ARRAY
1100    6         -         -         -        RESTART MAIN PROGRAM
1200    7         -         -         -        EXIT PROGRAM
```

```
 100   PEPLT/PEPLS: BRANCH 3--PARAMETERS--SHORT DOCUMENTATION
 300   KBR3     ISW3    DESCRIPTION
 400   1        0       PRINT OUT PARAMETERS ON KLIST; STORE COMMON TO
 500                    FILE KPLCM.  RETURN TO PEPLS.
 600            1       INPUT VARIABLE SELECTORS NX1 TO NZ3
 700            2       ENTER A1 TO A6
1100   2        2       ENTER B1 TO B6
1200   3        2       ENTER C1 TO C6
1300   4        2       ENTER D1 TO D6
```

```
 100    PEPLT/AVRGS: SHORT DOCUMENTATION
 200    KBR   ISW   JSW   KLIST    DESCRIPTION
 250      4     0     -     -      SHORT DOCUMENTATION
 300      4     1     #     #      READ FROM DATA FILES VARIABLES IN COLUMNS
 400                               JSW TO KLIST
 500            2     #     #      ZERO COLUMNS JSW TO KLIST
 600            3     1     -      INITIALIZE AND INPUT PARAMETERS
 700                  0     -      INPUT PARAMETERS--NO INITIALIZATION
 800            4     #     #      DIVIDE COLUMNS JSW TO KLIST BY COLUMN 6
 900            5     #     -      ADD COLUMN JSW VERTICALLY FROM THE TOP
1000            6     -    LU      PRINT OUT DATA ARRAY ON UNIT KLIST
1100            7     #     -      CALL NCAR PLOT PACKAGE TO PLOT ONE FRAME.
1200                               DEFAULT IS COLUMN JSW AGAINST PRESSURE.
1300                               GENERAL PLOTS:
1400                               X=B1*C(I,JSW)+B2*C(I,NX2)+B3*PR
1500                               Y=A1*PR+A2*C(I,NY1)+A3*C(I,NY2)
1600                               MULTIPLE PLOTS ON ONE FRAME ALLOWED
1700    KBR    ISW    JSW   KLIST
1800            8     -     -      COMPUTE DYNAMIC HEIGHT AND POTENTIAL ENERGY:
1900                               ASSUMES DVI IN COLUMN 1 (NV(1)=18) AND DVF
2000                               IN COLUMN 2 (NV(2)=19).
2100            9     #     #      INTEGRATE COLUMNS JSW TO KLIST AS A FUNCTION
2200                               OF PRESSURE
2300           10     #     #      SUBTRACT REFERNCE LEVEL VALUE C(JREF,#) FROM
2400                               COLUMNS JSW TO KLIST
2500           11     1     -      INPUT JC1,CR1 TO JC4,CR4
2600           11     0     -      PERFORM THE FOLLOWING COLUMN ADDITION:
2700                               C(I,JC1)=CR1*C(I,JC1)+CR2*C(I,JC2)+
2800                               CR3*C(I,JC3)+CR4*C(JREF,JC4)
2900           12     -     -      RETURN TO PEPLS
3000           13     -     -      INPUT COLUMN #'S AND CONSTANTS TO PERFORM
3100                               THE FOLLOWING COLUMN MULTIPLICATION:
3200                               C(IREC,I)=CON1*C(IREC,I)*CON2*C(IREC,J)*
3300                               CON3*C(IREC,K).  INPUT ORDER: I,J,K,CON1
3400                               CON2,CON3; AN INDEX (I,J,OR K) OF VALUE -1
3500                               PREVENTS THE INCLUSION OF THE ASSOCIATED
3600                               AND FOLLOWING COLUMN(S).
3700    KBR     ISW   JSW KLIST
3800           14     -     -      OUTPUT FIRST THREE COLUMNS IN MAP FORMAT:
3900                               ACCESS TO THIS BRANCH QUERIES WHAT HORIZONTAL
4000                               LEVEL # IS DESIRED
4100           15     -     -      NOT USED
4200           16     1     -      INPUT X,J
4300                  0     -      C(IREC,J)=C(IREC,J)**X.  (SHOULD FOLLOW
4400                               4,16,1 IMMEDIATELY IN EXECUTION.)
4500           17     -     -      ERROR SUMMATION: VERTICAL INTEGRATION WITH
4600                               (DELTA P)**2 AS THE INCREMENT
4700           18     -     -      INPUT DELTA P INTO C(IREC,5)
4800           19     -     -      EXCHANGE TWO COLUMNS OF C
4900           20     -     -      CHANGE A SINGLE ELEMENT OF C
5000           21     -     -      COMPUTE STANDARD DEVIATION OF X GIVEN
5100                               X-BAR IN C(IREC,4) AND X*X-BAR IN C(IREC,3).
5200                               RESULT IS STORED IN C(IREC,1).
5300           22     1     1      CALCULATE DYNAMIC HEIGHT AT A GIVEN LEVEL
5400                               RELATIVE TO PRESSURE CORRESPONDING TO JREF
5500                               AND OUTPUT IN MAP FORMAT,ALONG WITH VARIABLES
5600                               FROM COLUMNS 3 AND 4 AT THAT PRESSURE.
5700                               NV(1) MUST BE = 18 AND NV(2)=19.
```

```
    5    C COMPEPLT.FOR FILE: DIMENSION,COMMON AND EQUIVALENCE FOR PEPLT
    7    C DISPLAY PROGRAM. N.BRAY
   10          PARAMETER KCM=943
   32          PARAMETER JDIM=100
   55          CHARACTER*8 DOC
   77          CHARACTER*12 GNAME
  100          DIMENSION CST(6,4),VR(35)
  200          DIMENSION KHDG(150),KBUF(46)
  300          DIMENSION KPLCM(KCM)
  378    C
  391    C COMMON
  404    C
  450          COMMON KIN
  475    C BEGINNING OF STORED COMMON
  500          COMMON KTTX,KLIST,KTP,KOUT,KBR
  600          COMMON NX1,NX2,NX3,NY1,NY2,NY3,NZ1,NZ2,NZ3
  700          COMMON ISW,JSW,MV,MW,IN,ND
  800    C
 1100          COMMON XMIN,XMAX,YMIN,YMAX
 1300          COMMON A1,A2,A3,A4,A5,A6
 1400          COMMON B1,B2,B3,B4,B5,B6
 1500          COMMON C1,C2,C3,C4,C5,C6
 1600          COMMON D1,D2,D3,D4,D5,D6
 1700          COMMON X1,ZLTO,ZLGO,DAY,XPL,YPL
 1800          COMMON WT
 1900          COMMON NV(6),NX(6),AV,BV,CV
 2000          COMMON JC1,JC2,JC3,JC4
 2100          COMMON CR1,CR2,CR3,CR4
 2200          COMMON JMAX,JREF
 2300          COMMON NX4,NY4,NZ4
 2400          COMMON IV1,IV2,IV3,JBUF,JHDR,JDO
 2500          COMMON JSHP(6),DAY1,DAY2,PMIN,PMAX
 2600          COMMON XEMN,XEMX,XNMN,XNMX
 2633          COMMON C(100,6)
 2666          COMMON ISSW(16)
 2683          COMMON PLABL(10),XLABL(10),YLABL(10)
 2700    C
 2800          COMMON LTYPE,MHDR,ICON,ISHP,ICAST,XDAY,IPR,LPR
 2900          COMMON XLAT,XLONG,WGT,XLTO,XLGO
 3000          COMMON LBBL(3),LBL(13),NSC(60),NPR(60),NSECTION
 3100          COMMON KTYPE,MBUF,IREC,N,NDP,KSW,L1,L2
 3200          COMMON PF,TO,SO,DVO
 3300          COMMON PI,THF,SF,DVF
 3400          COMMON PM,THM,SM,DVM
 3500          COMMON DH,PE,XPE
 3600          COMMON CP(8),Z1,CT(8),Z2,F1,F2,F3
 3616    C END OF STORED COMMON
 3632          COMMON XDAT(100,7),YDAT(100,7)
 3650          COMMON IDELP,DP
 4100    C
 4200    C CHAR
 4300          COMMON/CHARACTER/ GNAME(200),DOC(10)
 4400    C
 4425          EQUIVALENCE (A1,CST),(VR,PF)
 4450          EQUIVALENCE (KHDG,LTYPE),(KBUF,KTYPE)
 4475          EQUIVALENCE (KTTX,KPLCM)
 4500    C
```

```
C PEPLT PROG ***********  SEPT 24 1977 *****
C ******************
      PROGRAM PEPLT
C ***********************************************************
C
C PROGRAM TO PLOT POTEN VARIABLES.
C JUNE 27 1976 N FOFONOFF
C VAX VERSION
C NOV 1980 N.BRAY
      INCLUDE 'COMPEPLT.FOR'
C
C
        OPEN(UNIT=10,NAME='KPLCM.DAT',ACCESS='DIRECT',TYPE='OLD',
     *  RECORDTYPE='FIXED',RECORDSIZE=KCM,ERR=1100)
C
   10   KIN = 5
        KTTX = 6
        KLIST = 6
        KOUT = 9
        KTP = 11
        KBR = 3
        WRITE(KTTX,1000)
       IF(NOYES(KIN,KTTX).EQ.1)THEN
   20   READ(10'1,ERR=1100)KPLCM
       ELSE
C
      DO 107 I = 1,6
      DO 107 J = 1,9
  107 CST(I,J) = 0.0
      DO 108 I=1,100
      DO 108 J=1,6
  108 C(I,J) = 0.0
      DO 109 J=1,16
        ISSW(J)=0
  109 CONTINUE
C
       ENDIF
        CALL PEPLS
       GO TO 20
 1000  FORMAT(1H ,'PEPLT: LOAD IN PREVIOUSLY STORED COMMON?')
 1100   OPEN(UNIT=10,NAME='KPLCM.DAT',ACCESS='DIRECT',TYPE='NEW',
     *  RECORDTYPE='FIXED',RECORDSIZE=KCM,ERR=1100)
       GO TO 10
      END
```

```
C PEPLS SUBROUTINE *********  SEPT 24 1977 *****
      SUBROUTINE PEPES
C ************************************************************
C
C PROGRAM TO PLOT POTEN VARIABLES.
C JUNE 27 1976 N FOFONOFF
C VAX VERSION--NOV 1980
      INCLUDE 'COMPEPLT.FOR'
C
        WRITE(KTTX,40)
   40   FORMAT(1H ,  'INITIALIZE DATA,  VARIABLE SELECTION PARAMETERS
     * (YES OR NO)?')
       IF(NOYES(KIN,KTTX).EQ.1)GO TO 106
  120   WRITE(KTTX,1200)KBR,ISW,JSW,KLIST,KTP,KOUT,KIN
 1200   FORMAT('*PEPLT:KBR,ISW,JSW,KLIST,KTP,KOUT,KIN',/,
     *  6X,3I3,I5,I3,I4,I3)
        KLIST = 6
        KOUT = 8
        KTP = 11
        READ(KIN,*)KBR,ISW,JSW,KLIST,KTP,KOUT,KIN
       IF(KBR.GT.7)KBR=7
       IF(KBR)120,800,130
  130   GO TO (200,20,30,400,500,600,700)KBR
C  *******************************************************************
C INITIALIZE
C
  106   XMIN=-20
        XMAX = 100.0
        YMIN = 0.0
        YMAX = 5000.0
        A1 = 1.0
        B1 = 1.0
        C1 = 1.0
        D1 = 1.0
        D2 = 1.0
        D3 = 1.0
        D4 = 1.0
        D5 = 1.0
        D6 = 1.0
        NX1 = 12
        NX2 = 0
        NX3 = 0
        NY1 = 19
        NY2 = 0
        NY3 = 0
        NZ1 = 25
        NZ2 = 0
        NZ3 = 0
        MV = 3
        MW = 0
        KTP = 11
        CALL DATA(-1,IEOF)
       GO TO 120
C*************** BRANCH 1--CALL TABLE SUBROUTINE ***************
  200   CALL TABLE
       GO TO 120
C ************** BRANCH 2--CHANGE DATA SELECTION VARIABLES ******
   20   CALL DATA(0,IEOF)
       GO TO 120
C ************** BRANCH 3--CHANGE OR LIST COEFFICIENTS **********
   30   OPEN(UNIT=50,NAME='PEPLS3.DOC',TYPE='OLD',READONLY)
```

```
      DO 3350 N=1,200
        READ(50,3325,END=3340)(DOC(I),I=1,9)
        WRITE(KTTX,3330)(DOC(I),I=1,9)
 3350 CONTINUE
 3325   FORMAT(9A8)
 3330   FORMAT(1H ,9A8)
 3340   CLOSE (UNIT=90)
  300   WRITE(KTTX,3000)
 3000   FORMAT(IH ,'**: PARAMETERS:  KBR3,ISW3,KX,MV,MW')
        KX = 4
        READ(KIN,*)KBR3,ISW3,KX,MV,MW
       IF(KBR3)30,350,31
   31  IF(ISW3-1)32,32,34
   32   WRITE(KLIST,3200)NX1,NX2,NX3,NY1,NY2,NY3,NZ1,NZ2,NZ3
 3200   FORMAT(1H ,'NX1,NX2,NX3,NY1,NY2,NY3,NZ1,NZ2,NZ3',/,9I4)
       IF(ISW3-1)34,33,30
   33   READ(KIN,*)NX1,NX2,NX3,NY1,NY2,NY3,NZ1,NZ2,NZ3
       GO TO 300
C
   34   WRITE(KLIST,3400)KBR3,(CST(JC,KBR3),JC=1,6)
 3400   FORMAT(I2,6(X,E10.4))
       IF(ISW3)30,38,35
   35   READ(KIN,*)(CST(JC,KBR3),JC=1,6)
   37  GO TO 300
   38   KBR3 = KBR3 + 1
       IF(KBR3-KX)34,34,350
C
  350   WRITE(10'1)KPLCM
       GO TO 120
C
C *************AVERAGES #4 **************************
  400   CALL AVRGS
       GO TO 120
C5 ****** SET ISSW SWITCHES ******
  500   WRITE(KTTX,5000)(K,K=1,16),(ISSW(K),K=1,16)
 5000   FORMAT(2(1H ,X,16I4,/),' ENTER K,ISSW(K)')
        READ(KIN,*)(K,ISSW(K),M=1,16)
       GO TO 120
C ************* RETURN TO MAIN PROGRAM **********************
  600 RETURN
C *********** EXIT PROGRAM ***********************************
  700   WRITE(KTTX,7000)
       IF(NOYES(KIN,KTTX).NE.1)GO TO 120
      STOP
 7000   FORMAT(1H ,'EXIT PROGRAM?')
C** PEPLS: SHORT DOCUMENTATION--BRANCH 0 ***************
  800   OPEN(UNIT=50,NAME='PEPLS.DOC',TYPE='OLD',READONLY)
      DO 850 N=1,200
        READ(50,825,END=812)(DOC(I),I=1,9)
        WRITE(KTTX,830)(DOC(I),I=1,9)
  850 CONTINUE
  812   CLOSE(UNIT=50)
  825   FORMAT(9A8)
  830   FORMAT(1H ,9A8)
       GO TO 120
      END
```

```
C AVRGS SUBPROG ******************************************
      SUBROUTINE AVRGS
C ***********************************************************
C
C FOR HORIZONTAL AVERAGES COMPUTE LIST AND PLOT.
C JUNE 27 1976 N FOFONOFF
C VAX VERSION--NOV 1980. N.BRAY
      DIMENSION D(6)
      DIMENSION ICHAR(6)
      DIMENSION XYM(4)
      CHARACTER*12 OUTNAME
      INCLUDE 'COMPEPLT.FOR'
C
      EQUIVALENCE (D1,D)
C
        CHARACTER*5 IDSTN
C
   10 GOTO(100,200,300,400,500,600,700,800,900,1000,1100,1200,1300,
     *  1400,1550,1600,1700,1800,1900,2000,2100,2200,1500)ISW
C
C *************#1 READ DATA TO C-TABLE ****************
  100 CONTINUE
      DO 101 K=1,ND
        READ(12,1011,END=1012)M,GNAME(K),WT
 1011   FORMAT(I4,A12,F5.2)
        GNAME(K)(9:12)='.AVG'
  101 CONTINUE
      GO TO 1013
 1012 CONTINUE
        ND = K-1
        REWIND 12
 1013   CONTINUE
      DO 170 NST=1,ND
        IEOF = 0
        OPEN(UNIT=KTP,NAME=GNAME(NST),READONLY,TYPE='OLD',FORM=
     *  'UNFORMATTED',ERR=168)
  102   CALL DATA(1,IEOF)
       IF(IEOF)165,105,105
  105   WT = 1.0
       IF(ISSW(15))110,115,115
  110   WT = WGT
  115 DO 160 I=JSW,KLIST
  120   XT = VRBL(NV(I))
        C(IREC,I)=C(IREC,I) + D(I)*WT*(AV*XT+(BV+CV*XT)*
     X  VRBL(NX(I)))
  160 CONTINUE
       GO TO 102
  165  IF(ISW.EQ.22)GO TO 800
        GO TO 170
  168   WRITE(KTTX,*)'ERROR READING',GNAME(NST)
  170 CONTINUE
        WRITE(10'1)KPLCM
  130 GO TO 1500
C ************ #2 ZERO TABLE SET PARAMETERS **********
  200 DO 210 I=JSW,KLIST
      DO 210 J=1,100
  210 C(J,I) = 0.0
       IF(ISW.EQ.22.AND.LFILE.EQ.0)GO TO 170
       GO TO 1500
C ************ #3 SET PARAMETERS ***************************
  300 IF(JSW)320,320,310
 310   ND = 1
         NV(1) = 51
         NV(2) = 68
         NV(3) = 86
         NV(4) = 87
         NV(5) = 63
         NV(6) = -1
       DO 312 I=1,6
 312 NX(I)=0
         JC1 = 1
         CR1 = 1.0
         JC2 = 2
         CR2 =-1.0
         JC3 = 3
         CR3 =-1.0
         JC4 = 4
         CR4 = 1.0
         AV=1.
         BV=0.
         CV=0.
         JMAX = 55
         JREF = 50
 320   IF(JSW)340,325,325
 325    WRITE(KTTX,3200)ND,(NV(K),K=1,6),JREF,JMAX
        READ(KIN,*)ND,(NV(K),K=1,6),JREF,JMAX
 330    WRITE(KTTX,3300)AV,BV,CV,(NX(I),I=1,6)
        READ(KIN,*)AV,BV,CV,(NX(I),I=1,6)
 340    WRITE(KTTX,3400)A1,A2,A3,B1,B2,B3
        READ(KIN,*)A1,A2,A3,B1,B2,B3
        WRITE(KTTX,3500)NX1,NX2,NY1,NY2
        READ(KIN,*)NX1,NX2,NY1,NY2
 225   GO TO 1500
3200    FORMAT(1H ,'ND,NV(6),JREF,JMAX',/,9I3)
3300    FORMAT(1H ,'AV,BV,CV,NX(6)',/,3F6.3,6I3)
3400    FORMAT(1H ,'PLOT PARAMETERS:   A1         A2         A3
    *        B1         B2         B3',/,16X,6F9.3)
3500    FORMAT(1H ,'    NX1     NX2     NY1     NY2',/,4I5)
C *********** #4 AVERAGE TABLE ***************************
 400 DO 410 J=1,100
       IF(C(J,6))405,415,405
 405 DO 410 I=JSW,KLIST
 410 C(J,I) = C(J,I)/C(J,6)
 415   JMAX = J - 1
       IF(ISSW(2))420,1500,1500
 420    WRITE(4,425)KBR,ISW,JSW,KLIST
 425    FORMAT(1H ,4(I3,2X))
       GO TO 1500
C **************** #5 ADD COLUMN JSW ***********
 500 DO 510 J=2,JMAX
 510 C(J,JSW) = C(J-1,JSW) + C(J,JSW)
       GO TO 1500
C *********** #6 LIST TABLE ***************************
 600    WRITE(KLIST,6000)(LBL(K),K=1,13)
        WRITE(KLIST,6055)ND,JREF,JMAX,(NV(K),K=1,6)
        WRITE(KLIST,6056)AV,BV,CV,(NX(I),I=1,6)
      DO 610 J=1,JMAX
 605    KP = JPR(J,NPR,NSECTION)
        WRITE(KLIST,6050)J,KP,(C(J,K),K=1,6)
 610 CONTINUE
       GO TO 1500
6000    FORMAT(1H ,13A4)
```

```
 6050   FORMAT(I3,I5,2X,6F10.4)
 6055   FORMAT(1H ,3I4,6I8)
 6056   FORMAT(1H ,3F4.2,6I8)
C ************ #7 PLOT TABLE ****************************
  700 CONTINUE
        JMIN=1
        NCURV=1
        WRITE(KTTX,7000)
        READ(KIN,*)NCURV,JMIN
        IF(NCURV.GT.6)NCURV=6
        WRITE(KTTX,7010)PLABL
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        READ(KIN,7020)PLABL
        CALL STRIP(PLABL)
       ENDIF
        WRITE(KTTX,7030)
       IF(NOYES(KIN,KTTX).EQ.-1)THEN
        WRITE(KTTX,7040)XMIN,XMAX,YMIN,YMAX
        READ(KIN,*)XMIN,XMAX,YMIN,YMAX
        CALL AGSETF(6HX/MIN.,XMIN)
        CALL AGSETF(6HX/MAX.,XMAX)
        CALL AGSETF(6HY/MIN.,YMIN)
        CALL AGSETF(6HY/MAX.,YMAX)
       ENDIF
        WRITE(KTTX,7050)XLABL
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        READ(KIN,7020)XLABL
        CALL STRIP(XLABL)
       ENDIF
        WRITE(KTTX,7060)YLABL
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        READ(KIN,7020)YLABL
        CALL STRIP(YLABL)
       ENDIF
C SET UP PLOT LABEL
C         CALL AGSETF(11HLABEL/NAME.,1HT)
C         CALL AGSETF(12HLINE/NUMBER.,.85)
C SET PARAMETERS FOR EZMXY PLOT
C         CALL AGSETF(17HTOP/NUMERIC/TYPE.,1.E36)
        CALL AGSETF(4HROW.,2.)
        CALL AGSETF(6HFRAME.,2)
C READ DATA INTO PLOT ARRAYS
      DO 710 K=1,NCURV
        WRITE(KTTX,7070)JSW
        READ(KIN,*)JSW
       IF(ISSW(5).EQ.-1)THEN
        WRITE(KTTX,7080)
        READ(KIN,7090)ICHAR(K)
       ENDIF
  705   IJM=0
      DO 710 J=JMIN,JMAX
        IJM=IJM+1
        PR=FLOAT(JPR(J,NPR,NSECTION))
        XDAT(IJM,K)=B1*C(J,JSW) + B2*C(J,NX2) + B3*PR
        YDAT(IJM,K)=A1*PR + A2*C(J,NY1) + A3*C(J,NY2)
  710 CONTINUE
       IF(A1.NE.0.)THEN
        CALL AGSETF(8HY/ORDER.,1.)
       ELSE
        CALL AGSETF(8HY/ORDER.,0.)
       ENDIF
```

```
        CALL ANOTAT(XLABL,YLABL,0,0,0,0.)
        CALL EZMXY(XDAT,YDAT,JDIM,NCURV,IJM,PLABL)
        CALL AGGETP(15HSECONDARY/USER.,XYM,4)
       IF(0.GT.XYM(1).AND.0.LT.XYM(2))THEN
        CALL LINENCAR(0.,XYM(3),0.,XYM(4))
       ENDIF
       IF(ISSW(5).EQ.-1)THEN
      DO 720 I=1,NCURV
        CALL POINTS(XDAT(1,I),YDAT(1,I),IJM,ICHAR(I),0)
  720 CONTINUE
       ENDIF
        CALL AGSETF(6HX/MIN.,1.E36)
        CALL AGSETF(6HX/MAX.,1.E36)
        CALL AGSETF(6HY/MIN.,1.E36)
        CALL AGSETF(6HY/MAX.,1.E36)
        CALL FRAME
       GO TO 1500
C FORMATS
 7000   FORMAT(1H ,'INPUT # OF CURVES IN THIS PLOT (MAX IS 6);
     *  AND INDEX OF FIRST POINT:')
 7010   FORMAT(1H ,'CHANGE PLOT LABEL? OLD LABEL IS:',/,4H       ,10A4)
 7020   FORMAT(10A4)
 7030   FORMAT(1H ,'USE DEFAULT AXIS PARAMETERS?')
 7040   FORMAT(1H ,'CURRENT VALUES OF XMIN,XMAX,YMIN,YMAX:',/,4F10.3)
 7050   FORMAT(1H ,'CHANGE X-AXIS LABEL? OLD LABEL IS:',/,4H       ,10A4)
 7060   FORMAT(1H ,'CHANGE Y-AXIS LABEL? OLD LABEL IS:',/,4H       ,10A4)
 7070   FORMAT(1H ,'INPUT COLUMN # (1 TO 6) TO BE PLOTTED',I3)
 7080   FORMAT(1H ,'INPUT IDENTIFYING CHARACTER')
 7090   FORMAT(A1)
C *********** #8 COMPUTE DH AND PE ******************
  800   PPR = 0.0
        DELA = C(1,1)-C(1,2)
        DELB = DELA
      DO 820 J=1,JMAX
  805 PR = JPR(J,NPR,NSECTION)
        DELP = PR-PPR
        DELA = C(J,1)-C(J,2)
        DHX = 0.5*(DELA+DELB)*DELP
        PEX = 0.50968E-1*(PR*DELA+PPR*DELB)*DELP
       IF(J-1)815,810,815
  810   C(1,I) = DHX
        C(1,2) = PEX
       GO TO 817
  815   C(J,1) = C(J-1,1) + DHX
        C(J,2) = C(J-1,2) + PEX
  817   DELB = DELA
  820   PPR = PR
       IF(ISW.EQ.22)THEN
        KLIST=2
        GO TO 1000
       ENDIF
       IF(ISSW(2))825,1500 ,1500
  825   WRITE(4,425)KBR,ISW,JSW,KLIST
       GO TO 1500
C *********** #9 INTEGRATE OVER PRESSURE *************
  900 DO 950 I=JSW,KLIST
        PPR = 0.0
        CPR = C(1,I)
      DO 940 J=1,JMAX
  910   PR = JPR(J,NPR,NSECTION)
       IF(ISW.EQ.17) GO TO 917
```

```
        PEX = 0.5*(C(J,I)+CPR)*(PR-PPR)
       GO TO 918
  917   PEX = .5*(C(J,I) + CPR)*(PR-PPR)**2
  918  IF(J-1)930,920,930
  920   C(I,I) = PEX
       GO TO 940
  930   CPR = C(J,I)
        C(J,I) = C(J-1,I) + PEX
  940   PPR = PR
  950 CONTINUE
       IF(ISSW(2))975,1500,1500
  975   WRITE(4,425)KBR,ISW,JSW,KLIST
       GO TO 1500
C ************ #10 SUBTRACT REFERENCE VALUE **********
 1000 DO 1050 I=JSW,KLIST
        CREF = C(JREF,I)
      DO 1040 J=1,JMAX
 1010   C(J,I) = CREF - C(J,I)
 1040 CONTINUE
       IF(ISW.EQ.22) GO TO 1425
 1050 CONTINUE
       IF(ISSW(2))1075,1500,1500
 1075   WRITE(4,425)KBR,ISW,JSW,KLIST
       GO TO 1500
C ************* #11 ADD COLUMNS ********************
 1100  IF(JSW)1110,1120,1110
 1110   WRITE(KTTX,1115)JC1,CR1,JC2,CR2,JC3,CR3,JC4,CR4
        READ(KIN,*)JC1,CR1,JC2,CR2,JC3,CR3,JC4,CR4
       GO TO 1500
 1115   FORMAT(1H ,'JC1,CR1,JC2,CR2,JC3,CR3,JC4,CR4',/,4(I4,E12.4))
 1120 DO 1125 J=1,JMAX
 1125 C(J,JC1) = CR1*C(J,JC1)+CR2*C(J,JC2)+CR3*C(J,JC3)
     Y  +CR4*C(JREF,JC4)
       IF(ISSW(2))1150,1500,1500
 1150   WRITE(4,425)KBR,ISW,JSW,KLIST
        WRITE(4,1115)JC1,CR1,JC2,CR2,JC3,CR3,JC4,CR4
       GO TO 1500
C *********
 1200 RETURN
C ******************** #13 MULTIPLY UP TO3 COLUMNS *****************
 1300   I=-1
        J=-1
        K=-1
        CON1=1
        CON2=1
        CON3=1
        WRITE (KTTX,1310)
 1310   FORMAT(1H ,'INPUT COLUMN NUMBERS UP TO 3 VALUES, AND CORRESPONDING
     X  MULTIPLICATIVE CONSTANTS')
        READ(KIN,*)I,J,K,CON1,CON2,CON3
      DO 1390 IREC=1,JMAX
       IF(I.LE.0)GO TO 1500
        A=C(IREC,I)*CON1
       IF(J.LE.0)GO TO 1380
        B=C(IREC,J)*CON2
       IF(K.LE.0)GO TO 1381
        CO=C(IREC,K)*CON3
       GO TO 1385
 1380 CONTINUE
        B=1.
 1381 CONTINUE
```

```
        CO=1.
 1385   C(IREC,I)=A*B*CO
 1390 CONTINUE
       IF(ISSW(2))1395,1500,1500
 1395   WRITE(4,425)KBR,ISW,JSW,KLIST
        WRITE(4,1311)I,J,K,CON1,CON2,CON3
 1311   FORMAT(1H ,3(I3,2X),3(F12.6,2X))
       GO TO 1500
C **************** BRANCH 14--OUTPUT IN MAP FORMAT ***********
 1400   KTO = 60
        JREC = 1
        WRITE(KTTX,1404)OUTNAME
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        READ(KIN,1402)OUTNAME
       ENDIF
 1402   FORMAT(A12)
        WRITE(KTTX,1403)JREC1,JREC2
        READ(KIN,*)JREC1,JREC2
        WRITE(KTTX,1401)OUTNAME
        OPEN(UNIT=KTO,NAME=OUTNAME,TYPE='NEW')
        JSW=1
        KLIST=6
       GO TO 100
 142   IF(IREC.NE.JREC1)GO TO 102
 1425   LFILE = 0
        IDSTN(1:2)=GNAME(NST)(1:2)
        IDSTN(3:5)=GNAME(NST)(6:8)
       IF(ISW.NE.22) GO TO 1430
      DO 145 KREC=JREC1,JREC2
        VR1 = C(KREC,JSW)*1.E-3
        VR2=C(KREC,3)
        VR3=C(KREC,4)
        VR4=C(KREC,5)
        VR5=C(KREC,6)
        KP=JPR(KREC,NPR,NSECTION)
        WRITE(KTO,1421)IDSTN,KP,XLAT,XLONG,VR1,VR2,VR3,VR4,VR5
  145 CONTINUE
       IF(NST.EQ.ND)LFILE=1
       GO TO 1490
 1430   VR1 = VRBL(NV(1))
        VR2 = VRBL(NV(2))
        VR3 = VRBL(NV(3))
 1490   CONTINUE
        KP=JPR(KREC,NPR,NSECTION)
        IDSTN(1:2)=GNAME(NST)(1:2)
        IDSTN(3:5)=GNAME(NST)(6:8)
        WRITE(KTO,1421)IDSTN,KP,XLAT,XLONG,VR1,VR2,VR3,VR4,VR5
 1421   FORMAT(1H ,A5,I6,2(F8.2),4(F8.3),F8.3)
       IF((ISW.EQ.22).AND.(LFILE.EQ.0)) THEN
        KLIST=6
        GO TO 200
       ENDIF
       IF(ISW.EQ.22.AND.LFILE.EQ.1)WRITE(KTO,1422)
 1422   FORMAT(/)
        CLOSE(UNIT=KTO)
       IF(LFILE)200,102,200
 1401   FORMAT(1H ,'NEW OUTPUT FILE NAME IS  ',A12)
 1403   FORMAT(1H ,'LEVEL NUMBERS ARE  ',2I3)
 1404   FORMAT(1H ,'INPUT NEW OUTPUT FILE NAME (YE OR NO)?  ',A12)
C **************** BRANCH 0--SHORT DOCUMENTATION **********
 1550 GO TO 1500
```

```
C *************************
 1500   WRITE(KTTX,1505)KBR,ISW,JSW,KLIST
 1505   FORMAT(1H ,'AVRGS:KBR,ISW,JSW,KLIST',/,5X,3I4,I5)
        KLIST = 6
        READ(KIN,*)KBR,ISW,JSW,KLIST
       IF(KBR.EQ.0) GO TO 2300
       IF(KBR.GT.22)GO TO 1500
       GO TO 10
C ******************* #16 TAKE C(IREC,J)**X ******************
 1600  IF(JSW)1610,1620,1610
 1610   WRITE(KTTX,1611)
        READ(KIN,*)X,J
       GO TO 1500
 1620 DO 1630 IREC=1,JMAX
        C(IREC,J)=ABS(C(IREC,J))**X
 1630 CONTINUE
       IF(ISSW(2))1650,1500,1500
 1650   WRITE(4,425)KBR,ISW,JSW,KLIST
        WRITE(4,1651)X,J
 1651   FORMAT(1H ,F12.6,2X,I2)
       GO TO 1500
 1611   FORMAT(1H ,'INPUT EXPONENT,COLUMN')
C ******************** BRANCH 17--SUMMATION OF ERRORS OVER P **************
 1700  GO TO 900
 1755   FORMAT(1H ,2(I2,2X))
C ******************** '#18--DELP INTO C(IREC,5) ******************
 1800   PPR=0.0
      DO 1810 J=1,JMAX
        PR=JPR(J,NPR,NSECTION)
        DELP=PR-PPR
        PPR=PR
        C(J,5)=DELP
 1810 CONTINUE
       IF(ISSW(2))1825,1500,1500
 1825   WRITE(4,425)KBR,ISW,JSW,KLIST
       GO TO 1500
C ****************** #19-- EXCHANGE COLUMNS ******************
 1900   WRITE(KTTX,1910)
 1910   FORMAT(1H ,'INPUT COLUMN NUMBERS TO BE EXCHANGED')
        READ(KIN,*)I,J
      DO 1920 IREC=1,JMAX
        CIREC = C(IREC,I)
        C(IREC,I) = C(IREC,J)
        C(IREC,J) = CIREC
 1920 CONTINUE
       IF(ISSW(2))1925,1500,1500
 1925   WRITE(4,425)KBR,ISW,JSW,KLIST
        WRITE(4,1755)I,J
       GO TO 1500
C ****************** CHANGE SINGLE ELEMENT OF C -- #20
 2000   WRITE(KTTX,2010)
 2010   FORMAT(1H ,'INPUT COLUMN, ROW, NEW VALUE')
        READ(KIN,*)I,J,XCHG
        C(J,I) = XCHG
       IF(ISSW(2))2025,1500,1500
 2025   WRITE(4,425)KBR,ISW,JSW,KLIST
        WRITE(4,2030)I,J,XCHG
 2030   FORMAT(1H ,2(I2,2X),F12.6)
       GO TO 1500
C **********COMPUTE STD.DEV.(X) IN COL 1 FOR X-BAR,X*X-BAR IN COL 4,3 ***
 2100   IF(ISSW(2))2110,2120,2120
```

```
      2110   WRITE(4,425)KBR,ISW,JSW,KLIST
      2120 DO 2130 IREC = 1,JMAX
             Z = C(IREC,4)
             C(IREC,1) = SQRT(ND*(C(IREC,3)-Z*Z)/(ND-1))
      2130 CONTINUE
             GO TO 1500
C *****************#22--CALCULATE DH FOR EACH STATION--OUTPUT IN MAP FORMAT
      2200   GO TO 1400
C ***************** BRANCH 0--SHORT DOCUMENTATION--AVRGS **********
      2300   OPEN(UNIT=50,NAME='AVRGS.DOC',TYPE='OLD',READONLY)
           DO 2350 N=1,200
             READ(50,2325,END=2355)(DOC(I),I=1,9)
             WRITE(KTTX,2330)(DOC(I),I=1,9)
      2350 CONTINUE
      2325   FORMAT(9A8)
      2330   FORMAT(1H ,9A8)
      2355   CLOSE (UNIT=50)
             GO TO 1500
           END
```

```
C TABLE SUBPROG PEPLT *****  OCT 27 1977 *****
      SUBROUTINE TABLE
C ******************************************************************
C
C TO COMPUTE AND PLOT POTEN VARIABLES.
C JUNE 27 1976 N FOFONOFF
C VAX VERSION--NOV 1980 N.BRAY
C
      INCLUDE 'COMPEPLT.FOR'
      DIMENSION VAR(100,9)
C
C PROG
       IF(ISW-1)600,15,10
   10   NOSTN = 0
        ND = 1
        NX1=1
        NX2=64
        NY1=2
        NY2=12
        A1=1.
        A2=2
        B1=1.
        B2=-.003
        X2DIM=3.
        Y2DIM=3000.
        XMIN=-110.
        XMAX=-40.
        YMIN=-100.
        YMAX=-30
   15   WRITE(KTTX,1505)ND,PMIN,PMAX,X2DIM,Y2DIM
 1505   FORMAT(1H ,'NO. STATIONS?:ND,PMIN,PMAX,X2DIM,Y2DIM: ',I4,4F7.
        READ(KIN,*)ND,PMIN,PMAX,X2DIM,Y2DIM
        JMIN=1
        WRITE(KTTX,7000)
        READ(KIN,*)JMIN
        WRITE(KTTX,7010)PLABL
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        READ(KIN,7020)PLABL
        CALL STRIP(PLABL)
       ENDIF
        WRITE(KTTX,7040)XMIN,XMAX,YMIN,YMAX
        READ(KIN,*)XMIN,XMAX,YMIN,YMAX
        CALL AGSETF(6HX/MIN.,XMIN)
        CALL AGSETF(6HX/MAX.,XMAX)
        CALL AGSETF(6HY/MIN.,YMIN)
        CALL AGSETF(6HY/MAX.,YMAX)
        WRITE(KTTX,7050)XLABL
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        READ(KIN,7020)XLABL
        CALL STRIP(XLABL)
       ENDIF
        WRITE(KTTX,7060)YLABL
       IF(NOYES(KIN,KTTX).EQ.1)THEN
        READ(KIN,7020)YLABL
        CALL STRIP(YLABL)
       ENDIF
        CALL AGSETF(6HFRAME.,2.)
        RETURN
C ********** PLOT RELATED FORMATS **********************************
 7000   FORMAT(1H ,'INPUT INDEX OF FIRST POINT:')
 7010   FORMAT(1H ,'CHANGE PLOT LABEL? OLD LABEL IS:',/,4H    ,10A4)
 7020   FORMAT(10A4)
 7040   FORMAT(1H ,'CURRENT VALUES OF XMIN,XMAX,YMIN,YMAX:',/,4F10.3)
 7050   FORMAT(1H ,'CHANGE X-AXIS LABEL? OLD LABEL IS:',/,4H    ,10A4)
 7060   FORMAT(1H ,'CHANGE Y-AXIS LABEL? OLD LABEL IS:',/,4H    ,10A4)
  600   CONTINUE
C
   18 DO 180 J=4,6
      DO 180 I=1,100
  180 C(I,J) = 0.0
      DO 101 K=1,ND
        READ(12,1010,END=106)M,GNAME(K),WT
 1010   FORMAT(I4,A12,F5.2)
        GNAME(K)(9:12)='.AVG'
  101 CONTINUE
  106  IF(ND.GT.K-1)ND=K-1
        REWIND 12
   90 DO 175 JST=1,ND
        IEOF=0
        OPEN(UNIT=KTP,NAME=GNAME(JST),READONLY,TYPE='OLD',FORM=
     *  'UNFORMATTED',ERR=175)
   95   CALL DATA(1,IEOF)
       IF(IEOF.EQ.-1)GO TO 111
        X = A1*VRBL(NX1)+A2*VRBL(NX2)+A3*VRBL(NX3)+A4*C(IREC,1)
        Y = B1*VRBL(NY1)+B2*VRBL(NY2)+B3*VRBL(NY3)+B4*C(IREC,2)
        Z = C1*VRBL(NZ1)+C2*VRBL(NZ2)+C3*VRBL(NZ3)+C4*C(IREC,3)
        XPR=A1*VRBL(NX1)
        YPR=B1*VRBL(NY1)
        WT = 1.0
        VAR(IREC,1)=PF
        VAR(IREC,2)=XPL
        VAR(IREC,3)=YPL
      DO 950 M=1,6
        VAR(IREC,M+3)=VRBL(NV(M))
  950 CONTINUE
       IF(ISSW(10).EQ.1)THEN
        WRITE(KOUT,1421)(VAR(IREC,K),K=1,9)
       ENDIF
 1421   FORMAT(9F8.3)
 1422   FORMAT(/)
  940  IF(ISSW(15))980,985,985
  980   WT = WGT
  985   C(IREC,4)=X
        C(IREC,5)=Y
        C(IREC,6)=Z
  105  IF(ISSW(12))110,113,113
  110   WRITE(KLIST,*)PF,X,Y,Z
 1100   FORMAT(2I4,A2,I4,F7.0,3F10.4)
  113 GO TO 95
  111   CONTINUE
       IF(ISSW(10).EQ.-1.AND.JST.EQ.ND)
     *WRITE(KOUT,1422)
       IF(ISSW(6).EQ.-1)GO TO 175
  171   IJM = 0
      DO 172 K=JMIN,IREC
        IJM=IJM+1
        XDAT(IJM,1) = C(K,4)
        YDAT(IJM,1) = C(K,5)
  172 CONTINUE
       IF(JST.GT.1)THEN
        CALL AGSETF(11HBACKGROUND.,4.)
       ENDIF
      CALL FRSTPT(XDAT(1,1),YDAT(1,1))
      CALL ANOTAT(XLABL,YLABL,0,0,0,0.)
      CALL EZMXY(XDAT,YDAT,JDIM,1,IJM,PLABL)
     IF(ISSW(5).EQ.-1)GO TO 175
      YPR2=YPR+B2*Y2DIM
      XPR2=XPR-A2*X2DIM
      XPR3=XPR+A2*X2DIM
      CALL LINENCAR(XPR,YPR,XPR,YPR2)
      CALL LINENCAR(XPR2,YPR,XPR3,YPR)
175 CONTINUE
     IF(ISSW(6).EQ.-1)GO TO 178
      CALL FRAME
      CALL AGSETF(6HX/MIN.,1.E36)
      CALL AGSETF(6HX/MAX.,1.E36)
      CALL AGSETF(6HY/MIN.,1.E36)
      CALL AGSETF(6HY/MAX.,1.E36)
      CALL AGSETF(11HBACKGROUND.,1.)
178  IF(ISSW(10).EQ.-1)CLOSE(UNIT=KOUT)
    RETURN
    END
```

```
C DATA SUBR PEPLT *****  SEPT 15 1977 *****
      SUBROUTINE DATA(NSW,IEOF)
C ********************************************************************
C
C PROGRAM TO READ AND SELECT POTEN DATA.
C JUNE 27 1976 N FOFONOFF
C VAX VERSION--NOV 1980. N.BRAY
      INCLUDE 'COMPEPLT.FOR'
      REAL*4 JDO
C
C
        MW = 1
       IF(NSW)1,20,200
    1 CONTINUE
        JBUF = 46
        JHDR = 150
        JDO = 0.
        PMIN = 0.0
        PMAX = 6000.0
        DAY1 = 0.
        DAY2 = 365.
        XEMN = -180.0
        XEMX =  180.0
        XNMN = -90.0
        XNMX =  90.0
        ZLTO = 31.0
        ZLGO = 69.50
        IFLAG=0
      RETURN
C
   20 CONTINUE
  172   WRITE(KTTX,173)DAY1,DAY2
  173   FORMAT(1H ,5HDAY1:F8.3,X,5HDAY2:,F8.3)
        READ(KIN,*)DAY1,DAY2
  174   WRITE(KTTX,175)XEMN,XEMX,XNMN,XNMX
  175   FORMAT(1H ,7HE-N LIM,4F7.2)
        READ(KIN,*)XEMN,XEMX,XNMN,XNMX
        WRITE(KTTX,177)ZLTO,ZLGO,JDO,PMIN,PMAX
  177   FORMAT(9H ORIGIN: ,2(X,F8.3),X,4HJDO:,F8.2,10HPMIN,PMAX ,2F7.1)
        READ(KIN,*)ZLTO,ZLGO,JDO,PMIN,PMAX
      RETURN
C
  200  IF(IFLAG.EQ.1)GO TO 212
        READ(KTP,END=280)KHDG
        IFLAG=1
C
  251  IF(XDAY-DAY1)280,252,252
  252  IF(DAY2-XDAY)280,254,254
  254  IF(XLONG-XEMN)280,256,256
  256  IF(XEMX-XLONG)280,258,258
  258  IF(XLAT-XNMN)280,260,260
  260  IF(XNMX-XLAT)280,262,262
  262   XPL = -111.12*(XLONG-ZLGO)*COS((XLAT+ZLTO)/114.592)
     *  + JDO*FLOAT(ICON)
        YPL = 111.12*(XLAT-ZLTO)
        DAY = XDAY
  212   READ(KTP,END=280)KBUF
 2615  IF(ISSW(13))2620,263,263
 2620  IF(IREC -1)2630,2625,2630
 2625   WRITE(KOUT)KHDG
 2630   WRITE(KOUT)KBUF
```

```
263   CONTINUE
270  IF(PF-PMIN)200,272,272
272  IF(PMAX-PF)200,274,274
274 RETURN
280   IEOF=-I
      CLOSE(UNIT=KTP)
      IFLAG=0
    RETURN
    END
```

```
C PEPLT SUBR ***** PEPLTSUB SEPT 15 1977*********************
C VAX VERSION NOV 1980. N.BRAY.
      FUNCTION POLY(VO,DVM,CP,N,VMN,VMX)
C **********************************************************
C
C TO EVALUATE POLYNOMIAL OF ORDER N-1 WITH COEFF CP(I).
C
C JAN 28 1976 N. FOFONOFF
C
      DIMENSION CP(1)
C
      V = VO
      IF(V-VMN)1,2,2
    1 V = VMN
    2 IF(VMX-V)3,4,4
    3 V = VMX
    4 POLY = 0.0
      X = V - DVM
      DO 10 I=1,N
      NI = N - I + 1
   10 POLY = POLY*X + CP(NI)
      RETURN
      END
C DPDV FCN **************************************
      FUNCTION DPDV(VO,DVM,CP,N,VMN,VMX)
C **************************************************
C
C TO COMPUTE DERIVATIVE OF POLYNOMIAL
C
C JAN 28 1976 N. FOFONOFF
C
      DIMENSION CP(1)
C
      V = VO
      IF(V-VMN)1,2,2
    1 V = VMN
    2 IF(VMX-V)3,4,4
    3 V = VMX
    4 NM1 = N - 1
      X = V - DVM
      DPDV = 0.0
      DO 20 I =1,NM1
      NMI = N - I
   20 DPDV = DPDV*X + FLOAT(NMI)*CP(NMI+1)
      RETURN
      END
C BND FCN *********************************
      FUNCTION BND(Z,ZMIN,ZMAX)
C **********************************************
C
C TEST AND LIMIT VARIABLES.
C
      BND = Z
      IF(Z-ZMIN)10,20,20
   10 BND = ZMIN
      RETURN
   20 IF(ZMAX-Z)30,40,40
   30 BND = ZMAX
   40 RETURN
      END
C JPR FCN ***** PTSBI *******
```

```
      FUNCTION JPR(IREC,NPR,NS)
C ****************************
C
C GENERATES PRESSURES CORRESPONDING TO IREC.
C
C OCT 28 1975 N FOFONOFF
C
      DIMENSION NPR(1)
C
      DO 100 J=1,NS
       IF(IREC.LT.NPR(J))THEN
        JPR=NPR(NS+J)*(IREC-NPR(2*NS+J))
        RETURN
       ENDIF
  100 CONTINUE
        JPR=NPR(2*NS)*(IREC-NPR(3*NS))
      RETURN
      END
C*********************************************************
      SUBROUTINE STRIP(A)
C*********************************************************
C
C STRIPS TRAILING BLANKS AND PUTS A $ AT THE END OF CHARACTERS
C IN ARRAY A FOR CONFORMANCE WITH NCAR PLOT PACKAGE LABELS
C N.BRAY 17NOV80
C
      DIMENSION A(1)
C
        B='    '
      DO 100 J=1,10
       K=10-J+1
       IF(A(K)-B)200,100,200
  100 CONTINUE
  200   NCH=K+1
       IF(NCH.GT.10)NCH=10
        A(NCH)='$   '
        RETURN
      END
C PEPLT SUBR =*****  PESBZ 4 MAY 1979 ***********
C D2PDV FN--SECOND DERIVATIVE OF POLYNOMIAL
      FUNCTION D2PDV(VO,DVM,CP,N,VMN,VMX)
C ****************************************************
C
C
C JAN 28 1976 N. FOFONOFF
C
      DIMENSION CP(I)
C
      V = VO
      IF(V-VMN)1,2,2
    1 V = VMN
    2 IF(VMX-V)3,4,4
    3 V = VMX
    4 NM1 = N - 2
      X = V - DVM
      D2PDV = 0.0
      DO 20 I =1,NM1
      NMI = N - I
      NMI2 = NMI - 1
   20 D2PDV = D2PDV*X + FLOAT(NMI*NMI2)*CP(NMI+1)
      RETURN
      END
C SEAWATER PROPERTIES ********************************
C
C SGO *****
      FUNCTION SGO(S)
C *********************
C SIGMA-0 KNUDSEN
C FEB 15 1976 N. FOFONOFF
C
      SGO = ((6.76786136E-6*S-4.82496I4E-4)*S+0.814876577)*S
     X -0.0934458632
      RETURN
      END
C SGT FCN **********
      FUNCTION SGT(T,S,SG)
C **********************
C SIGMA-T KNUDSEN
C FEB 15 1976 N FOFONOFF
C
      SG = SGO(S)
   20 SGT = ((((-1.43803061E-7*T-1.98248399E-3)*T-0.545939111)*T
     X +4.53168426)*T)/(T+67.26)+((((1.667E-8*T-8.164E-7)*T
     X +1.803E-5)*T)*SG+((-1.0843E-6*T+9.8185E-5)*T-4.7867E-3)*T
     X +1.0)*SG
      RETURN
      END
C EQUATION OF STATE FOR SEAWATER EOS80
C ***************************************************
      REAL FUNCTION EOS80(P1,T,S)
C ****************************************
C EQUATION OF STATE FOR SEAWATER PROPOSED BY JPOTS 1980
C REFERENCES
C MILLERO ET AL 1980, DEEP-SEA RES.,27A,255-264
C JPOTS NINTH REPORT 1978,TENTH REPORT 1980
C UNITS:
C        PRESSURE         P         BARS
C        INPUT PRESSURE   P1        DECIBARS
C        TEMPERATURE      T         DEG CELSIUS (IPTS-68)
C        SALINITY         S         NSU (IPSS-78)
C        DENSITY          RHO       KG/M**3
C        SPEC. VOL.       EOS80     M**3/KG
C CHECK VALUE: EOS80 = 9.435561E-4 M**3/KG FOR S = 40 NSU,
C T = 40 DEG C, P = 1000 BARS.
C
C N FOFONOFF REVISED OCT 7 1980
C MODIFIED TO TAKE DB INPUT PRESSURE, AND OUTPUT IN CM**3/GM 28NOV80
C N.BRAY
      REAL P1,P,T,S,RHO,SR,R1,R2,R3,R4
      REAL A,B,C,D,E,A1,B1,AW,BW,K,KO,KW
C EQUIV
      EQUIVALENCE (E,D,B1,R4),(BW,B,R3),(C,A1,R2)
      EQUIVALENCE (AW,A,R1,RO),(KW,KO,K)
C CONVERT PRESSURE TO BARS AND SQUARE ROOT SALINITY.
      P = P1*.1
      SR = SQRT(ABS(S))
C COMPUTE DENSITY PURE WATER AT ATM PRESSURE
      R1 = ((((6.536332E-9*T-1.120083E-6)*T+1.001685E-4)*T
     X-9.095290E-3)*T+6.793952E-2)*T+999.842594
C SEAWATER DENSITY ATM PRESS.
      R2 = (((5.3875E-9*T-8.2467E-7)*T+7.6438E-5)*T-4.0899E-3)*T
     X+8.24493E-1
```

```
      R3 = (-1.6546E-6*T+1.0227E-4)*T-5.72466E-3
      R4 = 4.8314E-4
      RHO = (R4*S + R3*SR + R2)*S + R1
C SPECIFIC VOLUME AT ATMOSPHERIC PRESSURE
      ALPHA = 1.E+3/RHO
      EOS80 = ALPHA
      IF(P.EQ.0.0)RETURN
C COMPUTE COMPRESSION TERMS
      E = (9.1697E-10*T+2.0816E-8)*T-9.9348E-7
      BW = (5.2787E-8*T-6.12293E-6)*T+8.50935E-5
      B = BW + E*S
C
      D = 1.91075E-4
      C = (-1.6078E-6*T-1.0981E-5)*T+2.2838E-3
      AW = ((-5.77905E-7*T+1.16092E-4)*T+1.43713E-3)*T
     X+3.239908
      A = (D*SR + C)*S + AW
C
      B1 = (-5.3009E-4*T+1.6483E-2)*T+7.944E-2
      A1 = ((-6.1670E-5*T+1.09987E-2)*T-0.603459)*T+54.6746
      KW = (((-5.155288E-5*T+1.360477E-2)*T-2.327105)*T
     X+148.4206)*T+19652.21
      KO = (B1*SR + A1)*S + KW
C
      K = (B*P + A)*P + KO
      ALPHA = ALPHA*(1.0 - P/K)
      EOS80 = ALPHA
      RETURN
      END
C V350P FCN ***** OCT 7 1980 *****
      REAL FUNCTION V350P(P1)
C ********************************
C SPECIFIC VOLUME (CM**3/GM) FOR S = 35 NSU (IPSS-78),
C TEMPERATURE 0 DEG CELSIUS (IPTS-68) AND PRESSURE IN DECIBARS.
C EQUATION DERIVED FROM EOS80
C CHECK VALUE: V350P = 9.337431E-4 M**3/KG FOR P = 1000 BARS.
C MODIFIED TO ACCEPT INPUT PRESSURE IN DB AND OUTPUT SP.VOL IN
C CM**3/GM 28 NOV 80.  N BRAY.
      P = P1*.1
      ALPHA = 9.72662E-4*(1.0-P/(21582.27+(3.35941+5.032E-5*P)*P))
      ALPHA = 1.E+3*ALPHA
      V350P = ALPHA
      RETURN
      END
C DEPTH FCN ***** OCT 7 1980 *****
      REAL FUNCTION DEPTH(P1,LAT)
C ********************************
C DEPTH IN METERS FROM PRESSURE IN DECIBARS USING
C SAUNDERS AND FOFONOFF'S METHOD.
C DEEP-SEA RES., 1976,23,109-111.
C FORMULA REFITTED FOR EOS80
C
      REAL LAT
C
      P = P1*.1
      X = SIN(LAT/57.29578)
      X = X*X
      GR = 9.780318*(1.0+(5.2788E-3+2.36E-5*X)*X) + 1.092E-5*P
      DEPTH = (((-1.82E-11*P+2.279E-7)*P-2.2512E-3)*P+97.2659)*P
      DEPTH = DEPTH/GR
      RETURN
      END
C ATG FCN ***********
      FUNCTION ATG(P,T,S)
C ***************************
C
C ADIABATIC TEMPERATURE GRADIENT. BRYDEN 1973.
C
      DS = S - 35.0
      ATG = (((-2.1687E-16*T+1.8676E-14)*T-4.6206E-13)*P
     X+((2.7759E-12*T-1.1351E-10)*DS+((-5.4481E-14*T
     X+8.733E-12)*T-6.7795E-10)*T+1.8741E-8))*P
     X+(-4.2393E-8*T+1.8932E-6)*DS
     X+((6.6228E-10*T-6.836E-8)*T+8.5258E-6)*T+3.5803E-5
      RETURN
      END
C DVA FCN ***** PTSB1 *****
      FUNCTION DVA(P,T,S)
C ***************************
C
C SPECIFIC VOLUME ANOMALY
C
      DVA = SVAN(P,T,S,SPV)
      RETURN
      END
C
C SVAN FCN ***********
      FUNCTION SVAN(P,T,S,V)
C ***************************
C SPECIFIC VOLUME ANOMALY*1E5
C FEB 15 1976 N FOFONOFF
      V = EOS80(P,T,S)
      SVAN = 1.0E5*(V - V350P(P))
      RETURN
      END
C THETA FCN ****************
      FUNCTION THETA(PO,TO,S,PF)
C *******************)**************
C
C TO COMPUTE LOCAL POTENTIAL TEMPERATURE AT PF
C FOURTH-ORDER RUNGE-KUTTA INTEGRATION USING STEPS OF 100 DB
C OR LESS.(RALSTON-WILF VOL 1 P115, EQ 26)
C
C OCT 12 1975 N. FOFONOFF
C
      P = PO
      T = TO
      H = PF - P
      N = ABS(H)/1000.0 + 1.0
      H = H/FLOAT(N)
      DO 10 I=1,N
      XK = H*ATG(P,T,S)
      T = T + 0.5*XK
      Q = XK
      P = P + 0.5*H
      XK = H*ATG(P,T,S)
      T = T + 0.29289322*(XK-Q)
      Q = 0.58578644*XK + 0.121320344*Q
      XK = H*ATG(P,T,S)
      T = T + 1.707106781*(XK-Q)
      Q = 3.414213562*XK - 4.121320344*Q
      P = P + 0.5*H
      XK = H*ATG(P,T,S)
      T = T + (XK-2.0*Q)/6.0
   10 CONTINUE
      THETA = T
      RETURN
      END
C T68 FCN **********
      FUNCTION T68(T)
C **************************
C TO CONVERT T-48 TO T-68 TEMPERATURE SCALE
C FEB 15 1976 N FOFONOFF
C
      T68 = T - 4.4E-6*T*(100.0-T)
      RETURN
      END
C T48 FCN **********
      FUNCTION T48(T)
C **************************
C TO CONVERT T-68 TO T-48 TEMPERATURE SCALE
C FEB 15 1976 N FOFONOFF
C
      T48 = T + 4.4E-6*T*(100.0-T)
      RETURN
      END
C DVDT FCN **********
      FUNCTION DVDT(P,T,S)
C **************************
C DERIVATIVE OF SPECIFIC VOL. WITH TEMPERATURE*1E5
C FEB 20 1976 N FOFONOFF
C
      H = 0.25
      DVDT = (5.0E4/H)*(EOS80(P,T+H,S)-EOS80(P,T-H,S))
      RETURN
      END
C DVDS FCN **********
      FUNCTION DVDS(P,T,S)
C **************************
C DERIVATIVE OF SPECIFIC VOL. WITH SALINITY*1E5
C FEB 20 1976 N FOFONOFF
C
      H = 0.5
      DVDS = (5.0E4/H)*(EOS80(P,T,S+H)-EOS80(P,T,S-H))
      RETURN
      END
C DVDP FCN **********
      FUNCTION DVDP(P,T,S)
C **************************
C ADIABATIC DERIVATIVE OF SPEC. VOL. WITH PRESSURE*1E5
C FEB 20 1976 N FOFONOFF
C
      H = 6.0
      DVDP =(5.0E4/H)*(EOS80(P+H,T,S)-EOS80(P-H,T,S))
     X + ATG(P,T,S)*DVDT(P,T,S)
      RETURN
      END
C DKDT FCN **********
      FUNCTION DKDT(P,T,S)
C **************************
C ADIABATIC COMPRESSIBILITY TEMP DERIVATIVE
C FEB 20 1976 N FOFONOFF
C
```

```
      H = 1.0
      DKDT = (0.5/H)*(DVDP(P,T+H,S) - DVDP(P,T-H,S))
      RETURN
      END
C DKDS FCN ***********
      FUNCTION DKDS(P,T,S)
C ****************************
C ADIABATIC COMPRESSIBILITY SALINITY DERIVATIVE.
C FEB 20 1976 N FOFONOFF
C
      H = 2.0
      DKDS = (0.5/H)*(DVDP(P,T,S+H) - DVDP(P,T,S-H))
      RETURN
      END
C SAL FCN ***********
      FUNCTION SAL(P,T,D)
C ****************************
C COMPUTE SALINITY GIVEN PRESSURE,TEMPERATURE AND SPECIFIC
C VOLUME ANOMALY(10**5*DELTA)
C FEB 16 1976 N FOFONOFF
C
      K = 0
      SAL = 35.0
   10 S = SAL
      SAL = S + (D-SVAN(P,T,S,V))/DVDS(P,T,S)
      K = K+1
      IF(K-50)20,30,30
   20 IF(ABS(SAL-S)-0.0005)30,10,10
   30 RETURN
      END
C NOYES FUNCTION **** DEC 3 1979 ******
      FUNCTION NOYES(KIN,KTTX)
C ****************************
C RETURNS 1 FOR YES -1 FOR NO
      NOYES = 0
    1 READ(KIN,10)LB
   10 FORMAT(A2)
C
      IF(LB.EQ.2HYE)NOYES=1
      IF(LB.EQ.2HNO)NOYES=-1
      IF(NOYES)30,20,30
C ERROR
   20 WRITE(KTTX,100)
  100 FORMAT('$ YES OR NO? ')
      GO TO 1
C
   30 RETURN
      END
C CTDSD FILE ******* JULY 15 1977 *******************
C THSAL FCN ******** JULY 6 1977 *******************
      FUNCTION THSAL(KIN,T)
C ***********************
C
C TAKES UP TO 25 CUBIC SPLINES TO GENERATE A SALINITY FROM
C POTENTIAL TEMPERATURE REFERRED TO THE SURFACE.INPUT DATA
C CONSISTS OF LOWER SPLINE BOUNDARY FOLLOWED BY FOUR COEFFICIENTS.
C COEFFICIENTS ARE FROM THE FIT OF ARMI AND BRAY (1981) TO
C ISELIN AND WORTHINGTON METCALF THETA-SAL DATA.
C
      DIMENSION C(5,25)
C DATA
```

```
      DATA C/0.00,34.738063,0.0,0.0,0.0,
      *0.50,34.738053,.107290,.584849E-02,-.253429E-02,
      *1.20,34.815152,.111753,.523726E-03,.582151E-01,
      *1.50,34.850297,.127785,.529320E-01,-.135379,
      *1.75,34.883436,.128868,-.485828E-01,-.129913,
      *2.00,34.910587,.802174E-01,-.146093,.228920,
      *2.25,34.925087,.500936E-01,.255484E-01,-.267382E-01,
      *2.50,34.938790,.578544E-01,.552526E-02,-.359945E-01,
      *2.75,34.953036,.538681E-01,-.214953E-01,-.374594E-01,
      *3.00,34.964575,.360969E-01,-.495364E-01,.509274E-01,
      *3.20,34.970220,.223936E-01,-.189292E-01,.580683E-01,
      *3.40,34.974406,.217901E-01,.157868E-01,.479730E-02,
      *3.60,34.979434,.286805E-01,.185975E-01,-.294172E-01,
      *3.80,34.985679,.325895E-01,.102958E-02,-.279688E-01,
      *4.00,34.992014,.296450E-01,-.157123E-01,.643397E-02,
      *5.00,35.01238,.175223E-01,.357759E-02,.114377E-02,
      *7.00,35.07089,.455579E-01,.104386E-01,.865592E-05,
      *10.00,35.30174,.108423,.105172E-01,-.763343E-03,
      *13.00,35.70106,.150916,.364790E-02,.310805E-04,
      *16.00,36.18748,.173643,.392926E-02,-.689782E-02,
      *19.00,36.55753,.109775E-01,-.581443E-01,.696380E-01,
      *21.00,36.9040118,0.0,0.0,0.0,15*0.0/
C
      DATA KNOTS/22/
C
  250 X = 0.0
      DO 310 I=1,KNOTS
      DT = C(1,I) - T
      IF(DT)305,320,320
  305 X = -DT
  310 CONTINUF
  320 D = X
      ID = I-1
      IF(ID)325,325,330
  325 ID = 1
      D = 0.0
  330 THSAL =((C(5,ID)*D+C(4,ID))*D+C(3,ID))*D+C(2,ID)
      RETURN
      END
C
C VRBL FUNCTION: PEPLT ****************************************
      FUNCTION VRBL(NVR)
C ************************************************************
C
C PROGRAM TO SELECT POTEN VARIABLES
C MAR 25 1976 N FOFONOFF
C VAX VERSION--INDIAN OCEAN DATA (1976). NOV 1980. N.BRAY
      INCLUDE 'COMPEPLT.FOR'
C
C
      IF(NVR)10,20,30
   10 VRBL = 1.0
      RETURN
   20 VRBL = 0.0
      RETURN
   30 GO TO (31,32,33,34,35,36,37,38,39,40,41,42)NVR
      IF(NVR.GT.12)GO TO 42
   31 VRBL = XPL
      RETURN
   32 VRBL = YPL
      RETURN
   33 VRBL = XLAT
      RETURN
   34 VRBL = XLONG
      RETURN
   35 VRBL = ICON
      RETURN
   36 VRBL = WGT
      RETURN
   37 VRBL = DAY
      RETURN
   38 VRBL = ISHP
      RETURN
   39 VRBL = ICAST
      RETURN
   40 VRBL = N
      RETURN
   41 VRBL = NDP
      RETURN
   42 IF(NVR-48)420,43,43
  420 VRBL = VR(NVR-11)
      RETURN
   43 E0 = DPDV(DVF,DVM,CP,N,F1,F2)
      PDF = PI - PF
      E1 = -.050968*PDF**2/E0
      E2 = (F3-SF)*E0
      E3 = -.050968*E2*E2/E0
      E5 = DPDV(SF,DVM,CP,N,F1,F2)
      NVRX = NVR - 47
      GO TO(48,49,50,51,52,53,54,55,56,57,58)NVRX
      IF (NVRX.GT.11)GO TO 58
   48 VRBL = E2
      RETURN
   49 VRBL = E2+PF
      RETURN
   50 VRBL = PDF
      RETURN
   51 VRBL = SF - DVF
      RETURN
   52 VRBL = PI*SIN(XLTO/57.296)/SIN(XLAT/57.296)
      RETURN
   53 VRBL = E1
      RETURN
   54 VRBL = 0.101937*PF*DH - PE
      RETURN
   55 VRBL = E3
      RETURN
   56 VRBL = E0
      RETURN
   57 VRBL = 1.0/E0
      RETURN
   58 SHF = SAL(PF,THF,DVF)
      VF = EOS80(PF,THF,SHF) + 1.0
      THP = DPDV(DVF,DVM,CT,N,F1,F2)/E0
      SHP = (1.0/E0 - DVDT(PF,THF,SHF)*THP)/DVDS(PF,THF,SHF)
      GR = -(.981/VF)**2
      BV1 = 100.0*GR/E0
        E6 = VR(13)
      THI = POLY(SF,DVM,CT,N,F1,F2) + THM
      SI = SAL(PF,THI,SF)
      IF(NVR-67)585,582,582
  582 GK = (DKDT(PF,THF,SHF)*THP+DKDS(PF,THF,SHF)*SHP)
```

```
      E4 = -50.968*GK*PDF**2
  585 NVRX = NVR - 57
      GO TO (580,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,
     *76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,
     *97,98,20)NVRX
      IF(NVRX.GT.39)GO TO 20
  580 VRBL = BV1
      RETURN
   59 VRBL = THP
      RETURN
   60 VRBL = SHP
      RETURN
   61 VRBL = SHF
      RETURN
   62 VRBL = SHP/THP
      RETURN
   63 VRBL = PDF*PDF
      RETURN
   64 E5 = 0.572958*SQRT(ABS(BV1))
      VRBL = SIGN(E5,BV1)
      RETURN
   65 VRBL = THETA(PF,THF,SHF,0.0)
      RETURN
   66 E5 = THETA(PF,THF,SHF,0.0)
      VRBL = SGT(E5,SHF,SG)
      RETURN
   67 VRBL = VF
      RETURN
   68 VRBL = -.5*GK*PDF*PDF
      RETURN
   69 VRBL = BV1 + GR*PF*GK
      RETURN
   70 VRBL = GK
      RETURN
   71 VRBL = GR*PF*GK
      RETURN
   72 Z = THETA(PF,THF,SHF,0)
      Z0=SHF
      GO TO 720
   73  Z=THETA(PI,T0,S0,0.0)
      Z0=S0
  720 Z1=THSAL(1,Z)
      VRBL=Z0-Z1
      RETURN
   74 Z = THETA(PF,THF,SHF,0.0)
      VRBL = Z*Z
      RETURN
   75 X = DVDP(PF,THF,SHF)
      Y = DVDT(PF,THF,SHF)
      Z = DVDS(PF,THF,SHF)
      VRPA =16.*X*X + 49.0E-06*Y*Y + 25.0E-06*Z*Z
      VRPI = VRPA*E0*E0
      GO TO 770
   76 Z = DVDP(PF,THF,SHF)
      VRBL = 16.*Z**2
      RETURN
   77 CONTINUE
      VRPI = Z1*Z1
  770 Y = VRPI + 2*PDF*PDF
      VRBL = .5*VRPI*E6*E6*Y
      RETURN
   78 VRPI = 3.*Z1*Z1/NDP
      GO TO 770
   79 VRPI = 16.
      GO TO 770
   80 VRPI = Z1*Z1/NDP
  800 Y = .5*VRPI*(VRPI + 2*PDF)
       VRBL = GK**2*Y
      RETURN
   81 VRPI = 16.
      GO TO 800
   82 VRBL = -1
      RETURN
   83 VRBL = -1
      RETURN
   84 VRBL = DVDP(PF,THF,SHF)
      RETURN
   85 VRBL = DVDS(PF,THF,SHF)
      RETURN
   86 VRBL = DVDP(PF,THF,SHF)*PDF
      RETURN
   87 VRBL = -.5*PDF*PDF*E6
      RETURN
   88 VRBL = -1
      RETURN
   89 VRBL = -1
      RETURN
   90 VRBL= -1
      RETURN
   91 VRBL = (SF-F3)**2
      RETURN
   92 Z = -THP*PDF
      VRBL = 1/Z
      RETURN
   93 VRBL = 1/(E0*E0)
      RETURN
   94 VRBL = THI*THI
      RETURN
   95 THMM = C(IREC,4)
      VRBL = -(THI-THMM)/(THP*PDF)
      RETURN
   96 THMM = C(IREC,4)
      VRBL = -(THF-THMM)/(THP*PDF)
      RETURN
   97 VRBL = THI
      RETURN
   98 E0 = 1.0/E0
      VRBL = -.5*E6*D2PDV(DVF,DVM,CP,N,FI,F2)*(PDF*E0)**2
      RETURN
 2000   END
C KDAY FCN ***** KDAY'S JULY 6 1977 ******
      FUNCTION KDAY(ID,IMO,IYR)
C ****************************************
C CONVERT GREGORIAN DATE TO JULIAN DAY
C USES LAST 4 DIGITS OF JULIAN DAY. ADD 2440000 TO GET
C FULL JULIAN DAY.
C
C JULY 12 1975
C
      IY = IYR - 68
      IF(2-IMO)10,20,20
   10 M = IMO - 3
```

```
      GO TO 30
   20 M = IMO + 9
      IY = IY - 1
   30 KDAY = (1461*IY)/4+(153*M+2)/5 + ID - 84
      RETURN
      END
C KDATE ****** CTDSB JULY 7 1977 *****
      SUBROUTINE KDATE(KD,ID,M,IY)
C ************************************
C CONVERT JULIAN DAY TO GREGORIAN DATE
C
      K=KD+84
      IY=(4*K-1)/1461
      ID=4*K-1-1461*IY
      IY = IY + 68
      ID=(ID+4)/4
      M=(5*ID-3)/153
      ID=5*ID-3-153*M
      ID=(ID+5)/5
      IF(M-10)20,10,10
   10 M=M-9
      IY=IY+1
      RETURN
   20 M=M+3
      RETURN
      END
C VKE FCN ***** SWPR1 *****
      FUNCTION VKE(P,T,S)
C *************************
C SPECIFIC VOLUME KNUDSEN/EKMAN
C FEB 15 1976 N FOFONOFF
C
      VO = 0.001*SGT(T,S,SG)
      VO = -VO/(1.0 + VO)
   20 VKE = (-4.886E-6*P/(1.0+1.83E-5*P) +((1.5E-17*T*P
     X +((-6.0E-17*T+1.8E-15)*SG+(-2.0E-16*T+1.206E-14)*T
     X -4.248E-13)*SG+(2.14E-14*T-1.24064E-12)*T-6.68E-14)*P
     X +((1.0E-12*T-4.5E-11)*SG
     X +(4.0E-12*T-3.28E-10)*T+1.725E-8)*SG
     X +((4.0E-12*T-6.63E-10)*T+3.673E-8)*T-2.2072E-7)*P)
      VKE = VO + VKE*(1.0 + VO)
      RETURN
      END
C DVZRO FCN ***********************************
      FUNCTION DVZRO(PO,DVM,PM,CP,N,NDP,VMN,VMX,ISHP,KCASY,ICON,DELP)
C **************************************************
C
C TO INVERT POLYNOMIAL FOR INDEPENDENT VARIABLE.
C
C  FEB 1 1976 N. FOFONOFF
C
      DIMENSION CP(1)
C
      XN = 0.0
      VR = 0.0
      PDF = PO
      KN = 0
      DV = (VMX-VMN)/FLOAT(NDP-1)
      DO 50 J=1,NDP
      V = VMX - DV*FLOAT(J-1)
      P = POLY(V,DVM,CP,N,VMN,VMX) + PM
      DP = DPDV(V,DVM,CP,N,VMN,VMX)
       IF (J.EQ.NDP/2)THEN
        DPO=DP
        VO=V
       ENDIF
      IF(DP)5,50,50
    5 PD = PO - P
      IF(KN)7,7,9
    7 PPD = PD
      KN = KN + 1
    9 IF(ABS(PD)-PDF)12,10,10
   12 PDF = ABS(PD)
      VS = V
   10 IF(PPD*PD)15,50,50
   15 VR = VR + V
      XN = XN + 1.0
      PPD = PD
   50 CONTINUE
   60 IF(XN - 1.0)66,70,65
   65 DVZRO = VR/XN
      GO TO 90
   66 DVZRO = VS
      GO TO 90
   70 K = 0
      V = VR
   75 VP = V
      P = POLY(V,DVM,CP,N,VMN,VMX) + PM
      V = VP + (PO-P)/DPDV(V,DVM,CP,N,VMN,VMX)
      K = K + 1
      IF(K-100)80,85,85
   80 IF(ABS(PO-P)-0.05)85,75,75
   85 DVZRO = V
   90 IF(DVZRO-VMN)95,100,100
   95 CONTINUE
        PO=ABS((DVZRO-VO)*DPO)
        PPO=ABS(PO-NDP*DELP/2)
        WRITE(KTTX,1000)ISHP,KCAST,ICON,PO,PPO
 1000   FORMAT(1H ,'FOR STATION ',A2,2I3,' AT',F8.0,' DB LEVEL,
     *  YOU SHOULD INCREASE REGRESSION INTERVAL BY ',F8.0,' DB.')
      DVZRO = VMN
  100 IF(VMX-DVZRO)105,110,110
  105 CONTINUE
        PO=ABS((DVZRO-VO)*DPO)
        PPO=ABS(PO-NDP*DELP/2)
        WRITE(KTTX,1000)ISHP,KCAST,ICON,PO,PPO
        DVZRO = VMX
  110 RETURN
      END
```

MANDATORY DISTRIBUTION LIST

FOR UNCLASSIFIED TECHNICAL REPORTS, REPRINTS, AND FINAL REPORTS PUBLISHED BY OCEANOGRAPHIC CONTRACTORS OF THE OCEAN SCIENCE AND TECHNOLOGY DIVISION OF THE OFFICE OF NAVAL RESEARCH

(REVISED NOVEMBER 1978)

1 Deputy Under Secretary of Defense
(Research and Advanced Technology)
Military Assistant for Environmental Science
Room 3D129
Washington, D.C. 20301

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217
3 ATTN: Code 483
1 ATTN: Code 460
2 ATTN: 102B

1 CDR Joe Spigai, (USN)
ONR Representative
Woods Hole Oceanographic Inst.
Woods Hole, MA 02543

Commanding Officer
Naval Research Laboratory
Washington, D.C. 20375
6 ATTN: Library, Code 2627

12 Defense Technical Information Center
Cameron Station
Alexandria, VA 22314
ATTN: DCA

Commander
Naval Oceanographic Office
NSTL Station
Bay St. Louis, MS 39522
1 ATTN: Code 8100
1 ATTN: Code 6000
1 ATTN: Code 3300

1 NODC/NOAA
Code D781
Wisconsin Avenue, N.W.
Washington, D.C. 20235

1 Mr. Michael H. Kelly
Administrative Contracting Officer
Department of the Navy
Office of Naval Research
Eastern/Central Regional Office
Building 114, Section D
666 Summer Street
Boston, MA 02210

UNCLASSIFIED 8/81

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER WHOI-81-70	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* VAX-11 PROGRAMS FOR COMPUTING AVAILABLE POTENTIAL ENERGY FROM CTD DATA		5. TYPE OF REPORT & PERIOD COVERED Technical 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Nancy Amanda Bray		8. CONTRACT OR GRANT NUMBER*(s)* * N00014-76-C-0197 ** N00014-79-C-0071 OCE-77-19403
9. PERFORMING ORGANIZATION NAME AND ADDRESS Woods Hole Oceanographic Institution Woods Hole, Massachusetts 02543		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS * NR 083-400 ** NR 083-004
11. CONTROLLING OFFICE NAME AND ADDRESS NORDA/National Space Technology Laboratory Bay St. Louis, MS 39529		12. REPORT DATE August 1981 13. NUMBER OF PAGES 135 pages
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)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

This report should be cited as: Woods Hole Oceanog. Inst. Tech. Rept. WHOI-81-70.

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

1. Oceanic available potential energy
2. Definitions of horizontal density surfaces
3. Hydrographic data

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This report documents the W.H.O.I. VAX-11 programs used to calculate available potential energy and related quantities from CTD data using the technique described in Bray and Fofonoff (1981). The report includes examples of how the programs may be used, as well as complete listings of all the required FORTRAN files.

Woods Hole Oceanographic Institution
WHOI-81-70

VAX-11 PROGRAMS FOR COMPUTING AVAILABLE POTENTIAL ENERGY FROM CTD DATA by Nancy Amanda Bray. 135 pages August 1981. Prepared for the Office of Naval Research under Contracts N00014-76-C-0197; NR 083-400 and N00014-79-C-0071; NR 083-004 and for the National Science Foundation under Grant OCE-77-19403

This report documents the W.H.O.I. VAX-11 programs used to calculate available potential energy and related quantities from CTD data using the technique described in Bray and Fofonoff (1981). The report includes examples of how the program may be used, as well as complete listings of all the required FORTRAN files.

1. Oceanic available potential energy
2. Definitions of horizontal density surfaces
3. Hydrographic data

I. Bray, Nancy Amanda
II. N00014-76-C-0197; NR 083-400
III. N00014-79-C-0071; NR 083-004
IV. OCE-77-19403

This card is UNCLASSIFIED

Woods Hole Oceanographic Institution
WHOI-81-70

VAX-11 PROGRAMS FOR COMPUTING AVAILABLE POTENTIAL ENERGY FROM CTD DATA by Nancy Amanda Bray. 135 pages August 1981. Prepared for the Office of Naval Research under Contracts N00014-76-C-0197; NR 083-400 and N00014-79-C-0071; NR 083-004 and for the National Science Foundation under Grant OCE-77-19403

This report documents the W.H.O.I. VAX-11 programs used to calculate available potential energy and related quantities from CTD data using the technique described in Bray and Fofonoff (1981). The report includes examples of how the program may be used, as well as complete listings of all the required FORTRAN files.

1. Oceanic available potential energy
2. Definitions of horizontal density surfaces
3. Hydrographic data

I. Bray, Nancy Amanda
II. N00014-76-C-0197; NR 083-400
III. N00014-79-C-0071; NR 083-004
IV. OCE-77-19403

This card is UNCLASSIFIED

Woods Hole Oceanographic Institution
WHOI-81-70

VAX-11 PROGRAMS FOR COMPUTING AVAILABLE POTENTIAL ENERGY FROM CTD DATA by Nancy Amanda Bray. 135 pages August 1981. Prepared for the Office of Naval Research under Contracts N00014-76-C-0197; NR 083-400 and N00014-79-C-0071; NR 083-004 and for the National Science Foundation under Grant OCE-77-19403

This report documents the W.H.O.I. VAX-11 programs used to calculate available potential energy and related quantities from CTD data using the technique described in Bray and Fofonoff (1981). The report includes examples of how the program may be used, as well as complete listings of all the required FORTRAN files.

1. Oceanic available potential energy
2. Definitions of horizontal density surfaces
3. Hydrographic data

I. Bray, Nancy Amanda
II. N00014-76-C-0197; NR 083-400
III. N00014-79-C-0071; NR 083-004
IV. OCE-77-19403

This card is UNCLASSIFIED

Woods Hole Oceanographic Institution
WHOI-81-70

VAX-11 PROGRAMS FOR COMPUTING AVAILABLE POTENTIAL ENERGY FROM CTD DATA by Nancy Amanda Bray. 135 pages August 1981. Prepared for the Office of Naval Research under Contracts N00014-76-C-0197; NR 083-400 and N00014-79-C-0071; NR 083-004 and for the National Science Foundation under Grant OCE-77-19403

This report documents the W.H.O.I. VAX-11 programs used to calculate available potential energy and related quantities from CTD data using the technique described in Bray and Fofonoff (1981). The report includes examples of how the program may be used, as well as complete listings of all the required FORTRAN files.

1. Oceanic available potential energy
2. Definitions of horizontal density surfaces
3. Hydrographic data

I. Bray, Nancy Amanda
II. N00014-76-C-0197; NR 083-400
III. N00014-79-C-0071; NR 083-004
IV. OCE-77-19403

This card is UNCLASSIFIED