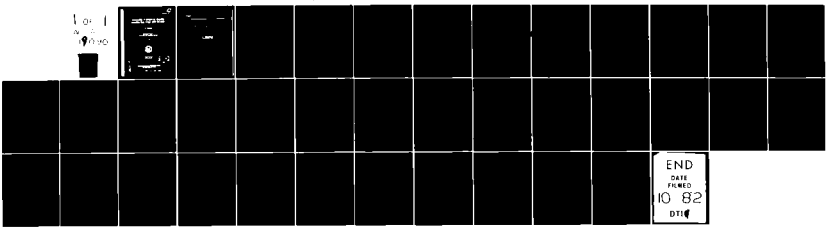
AD-A119 090 NAVAL OCEAN RESEARCH AND DEVELOPMENT ACTIVITY NSTL S--ETC F/G 8/10
DOCUMENTATION OF SOFTWARE FOR ESTIMATING EXPENDABLE SHEAR PROFI--ETC(U)
JUN 82 K D SAUNDERS
UNCLASSIFIED NORDA-50 NL



Foreword

Oceanographic research is the primary mission of [illegible]. In order to pursue this mission it is necessary to have well calibrated instruments.

This report [illegible] calibration [illegible]

[illegible]

[illegible]

Executive Summary

Expendable Current Profilers (XCPs) do not transmit a pressure signal from the instrument to the ship. The depth is determined either by an empirical equation or by comparison with a reference oceanic variable. This report provides documentation of a computer program for the determination of a least squares best fitted polynomial for determining the depth as a function of time, using temperature as measured from a CTD at the time of the profiler drop.

Acknowledgments

This work was supported by NORDA Code 500, Dr. Rudolph Hollman project manager.

Contents

Documentation of Software for Estimating Expendable Shear Profiler (XCP) Fall Rates

I. Introduction

A serious problem encountered when analyzing XCP (Expendable Current Profiler) data is the error in true depth of the instrument. The XCP is free falling and does not transmit a pressure signal to the ship. Because there is no direct measurement of pressure, the pressure is estimated through the use of an empirical equation relating time (starting when the profiler is dropped) to pressure. Variations in launching procedures, initial conditions and oceanic conditions may result in significant departures of the true pressure from that estimated by the equation.

The XCP does return a temperature signal to the ship. If the true temperature profile (versus pressure) is known, then it should be possible to obtain an accurate estimate of the position of the instrument as a function of time through appropriate comparison of the true profile of temperature and that returned by the XCP.

A program to accomplish this has been written and is documented in this report. The theory and some particular results are given in the journal (Oceans, September 1981) reprint in the following section; the detailed program documentation follows.

II. Reprint from Oceans, September 1981

A.W. Green and K.D. Saunders

Naval Ocean Research & Development Activity, NSTL Station, MS 39529

ABSTRACT

Recent intercomparisons of depths of expendable bathythermographs (XBT) and electromagnetic current profilers with more accurate shipboard profilers have shown that the empirical formulae currently used are in error. A theoretical model for the bulk dynamics of XBT's (T-7) and XCP's is proposed and assessed with recent intercomparisons between XCP's with accurate Conductivity-Temperature-Depth profilers (CTD's). The records for the intercomparisons were obtained in the Gulf of Mexico, where a rich temperature fine-structure provides numerous features in the profiles that quantitatively improve the XBT/CTD and XCP/CTD comparisons. The theory yields fall rate and total depth versus time predictions that agree with observations somewhat better than the manufacturer's empirical formulae. The theory also furnishes a rational basis for establishing deployment techniques and probe modifications that can improve depth accuracy and fall rate.

1. INTRODUCTION

During the past decade the production technology for expendable oceanographic profiling instruments has improved significantly, consequently, expendables are becoming increasingly important in oceanographic survey work. As the measurement consistency of these devices has improved, the users have increased their demands of quality of the data obtained by the probes. Probe fall rate and depth as a function of time have received relatively little attention in technical literature to date. The expendable probes such as XBT's (bathythermographs), XCP's (Current plus BT), XSV's (Sound speed + BT) are not currently equipped with pressure sensors; thus, the probe depth must be estimated from empirical formulas based on observed fall rates and depths as a function of time. These empirical formulas do not in general provide the best possible estimates of probe fall rate or depth. Motivated by our own needs to improve probe fall rate and depth estimates, we have undertaken development of experimental methods and theories that will help reduce fall rate and depth errors. In this paper we present results of a simple dynamical model for streamlined bodies, such as expendable profilers, falling through water. The model is applied specifically to XCP's in this paper, but we note that it is also applicable to other types of expendables (Green, 1981). Next we describe calibration methods that reduce depth error by analytical treatments of XCP data and profiles from reference instruments, such as conductivity-temperature-pressure profilers (CTD's). We show that the dynamical model is in general an excellent predictor of probe depth as a function of time. We conclude our discussion with some recommendations that could improve XCP probe performance.

2. THE DYNAMICAL MODEL

A body falling vertically through water is subject to the bulk forces of hydrodynamic drag and buoyancy, which are in balance described by the following equation:

$$M dv/dt = \tfrac{1}{2}\rho_W C_D S v^2 - g(M - M_W) \quad (2.1)$$

The bulk drag force is proportional to the square of the fall velocity (v), and the buoyancy is determined by the difference between the body mass and the mass of the water displaced. (A list of definitions of the symbols are in Appendix A.) The expendables currently in use operate over a range of Reynolds number (R) between 10^5 and 10^6, where R is defined for a streamlined body as the product of probe length and fall speed divided by the kinematic viscosity of the water. The kinematic viscosity increases with depth in the ocean and the fall speed is relatively constant after initial transient adjustment, consequently, the R tends to decrease with increasing depth. If the probes were perfect streamlined bodies moving through quiescent sea water, then we should expect that the drag coefficient (C_D) would vary significantly with depth and alter the drag force. This tendency appears to be attenuated by probe rotation caused by the axial stabilizing fins, which disturb ("turbulize") the boundary layer flow over the probe and increase the wake turbulence. The net effect is that the flow regime around the probes is fully turbulent. The C_D of the probe is much less affected by variations in Re in fully turbulent flow (Hoerner, 1965), but we should

keep in mind that small changes in C_D over the range of the probe can make significant changes in fall rate.

The buoyancy of a probe also changes due to unspooling of the signal transmission wires. The mass of wire loss is partially replaced by water that is entrained within the probe. To a good approximation the mass loss rate is proportional to the depth. Under extreme operating conditions this simple assumption may be in error (Green, 1981).

Green (1981) found an approximate solution to (2.1) for the special case in which the initial probe speed is equal to the nominal fall rate $[v(z=0) = V_T]$. The mass slowly decreases and the drag coefficient slowly increases linearly with depth; for the model the changes are assumed to be linear:

$$M = M_0(1 + Az),\ C_D = C_0(1 - Bz),$$

$$z < 0;\ A,\ B, \ll 1$$

The approximate solution for fall rate as a function of depth is:

$$v \approx - V_T\,[1 + (A + B)z]^{1/2} \qquad (2.2)$$

and

$$z \approx - V_T t + (A + B)\,V_T t^2 \qquad (2.3)$$

These approximate solutions clearly show the dependence of v and z on the variable parameters. To first order in t the probe depth is determined by the nominal terminal speed $[V_T = (2M_0 g'/\rho_w C_0 S)^{1/2}]$. The coefficient of the quadratic term is also proportional to the relative changes in drag coefficient and mass (A + B). The empirical equations for probe depth as a function of time are typically quadratic, so we can compare the analytical results with empirical fits of observations.

Generally the effects of initial transient adjustment of the speed on the predicted depth cannot be obtained except by numerical integration of (2.1). Results of a numerical integration of with $v(z = 0) = -15\ m \cdot s^{-1}$ are displayed in Fig. 1. The variations of A and B are less than the nominal values for an XCP (Table 4), but the shape of the curve is the same.

3. EXPERIMENTAL ANALYSIS

Present expendables, such as XBT's and XCP's, measure temperature - not depth or pressure. The probe depths can be inferred from equations giving fall time as a function of depth (pressure), but a more accurate method is based on a mapping of the temperature profile of an expendable on a temperature-pressure record obtained with a CTD in spatial and temporal proximity with the probe drop. In

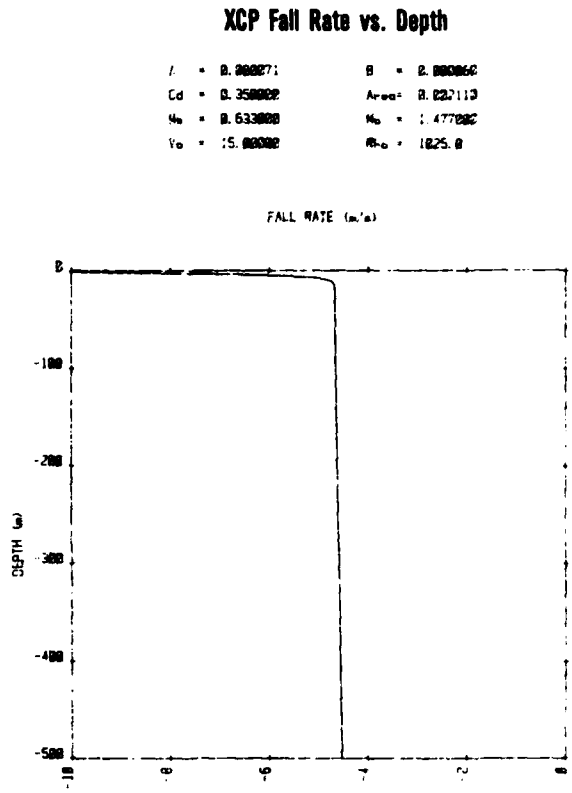


FIGURE 1

the experimental results presented here the CTD was lowered to 350 dbar; at this time the XCP was launched. The XCP passed the CTD before it reached 750 m depth. This method provided near-simultaneity.

The use of concurrent CTD casts, would, if required, make the profiler unsuitable for large scale rapid surveys. Thus, it is important to determine whether a suitable equation for pressure (or depth) as a function of time of drop can be obtained from a relatively small number of intercomparisons with concurrent CTD casts.

We are certainly not the first group to attempt this. Smart (private communication) has performed a similar study by comparing fall times with pressures, referencing both to specific features in the temperature profile. We have chosen to use as much of the temperature profile as possible with the object of reducing random errors.

The data set used for this study was obtained during the deployment cruise of the Acoustically Tracked Oceanographic Mooring (ATOM '79) project (Saunders, Green and Bergin, 1980). Concurrent CTD and XCP casts were made in a region of strong thermal and velocity

gradients in the central Gulf of Mexico in December, 1979. The thermal regime was characterized by a very uniform mixed layer underlain by a strong thermocline, rich in fine structure but belonging to a single water mass.

For each cast, an equation of pressure as a function of the time of drop was determined by fitting the 'observed' pressures and times to a function of the form.

$$p(t,N) = \sum_{n=0}^{N} Q_n t^n \qquad (3.1)$$

The fit is formally obtained by fitting the 'observed' pressures $P_0(t_0)$ to $p(t_0,N)$ where t_0 is the observed time and P_0 is the observed pressure from the CTD cast. These are related by the equations:

$$P_0 = P_0[T_0(CTD)]$$

$$t_0 = t_0[T_0(XCP)]$$

$$T_0(CTD) = T_0(XCP) + T_0(offset)$$

It is easily understood, that this procedure will work only if $T_0(CTD)$ and $T_0(XCP)$ are monotonic functions of P_0 and t_0, respectively. We were fortunate that there were large sections of the temperature profiles in all the casts analyzed where this requirement was satisfied. The mixed layer provided a convenient thermostad wherein the constant offsets between the XCP's and the CTD could be determined. A typical plot of interpolated pressure versus time, with both quantities referenced to the same temperature profile is presented in Figure 2.

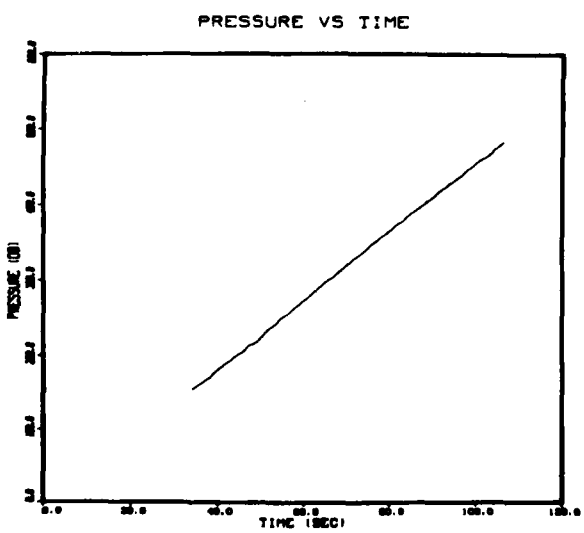


FIGURE 2

The results for the first order and second order fits are summarized in Tables 1 and 2 respectively. For both orders of fit, it is obvious that the initial offsets varied quite widely, while the fall rates appeared to vary in a narrow range.

This range may be associated to the 95% confidence limits computed using the Student's t-distribution with 24 degrees of freedom. These limits are denoted as $\mu_{+.05}$ and $\mu_{-.05}$ and have been computed for all the fitted coefficients. The nominal value for the linear term (using the quadratic formula), 4.544 dbar·s^{-1} is smaller than the lower confidence limit in Table 2, while the constant and quadratic terms, 3.1 dbar and -6.75×10^{-4} dbar·s^{-2}, respectively, fall within the corresponding confidence limits and may be accepted.

Some values for the quadratically fitted coefficients, together with the associated parameters from the model equations are given in Table 3. For the original model, the constant term must be rejected at the 95% level, while the linear and quadratic terms cannot be rejected. After some experimentation with the model, it was found that an initial velocity of about 15 m s^{-1} produced acceptable coefficients.

It is highly unlikely that the probes ever entered the water at such high speeds, however. The probable cause for the large variation in the constant term is the method for establishing the zero time point (Sanford, private communication). This variation would, for small offsets, affect only the constant term in the pressure (depth)/time equation to a measurable degree.

When the empirical fits were made, the observed pressure was differentiated with respect to time and this derivative was plotted as a function of pressure. A typical example is shown in Figure 3. The large variation in the pressure-time derivative indicated to the possibility that the probe was precessing as it fell, causing an effectively changing drag coefficient. This was incorporated into the model as a drag coefficient, varying sinusoidally with depth. The magnitude of the variation of the drag coefficient turned out to be much too great if it were to account for the same order of the drop rate as was observed. This cursory test does not lay to rest the question of gyroscopic effects on probe dynamics, since it is conceivable that gyroscopic precession coupled with the surrounding flow could cause temperature anomalies and velocity errors. This is a subject for future work. We therefore advance the tentative hypothesis that the large apparent variations in the drop speed are artifacts of the XCP temperature trace and arise from the restricted flow past the XCP thermistor.

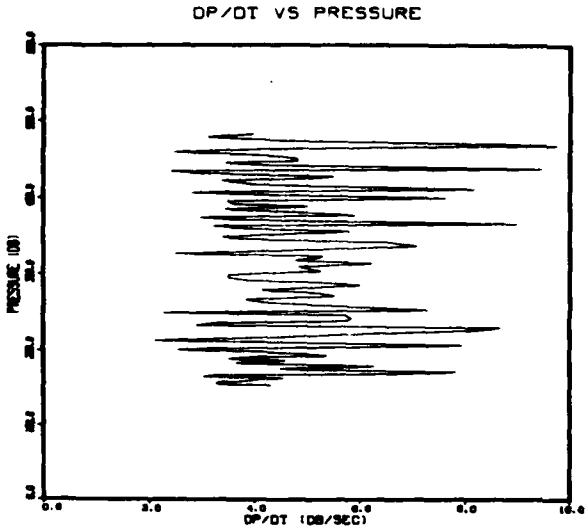


FIGURE 3

4. RECOMMENDATIONS

In view of the variations observed in the pressure time derivatives with depth and the large variation in the initial pressure offsets we make the following recommendations:

a. a signal should be made available to inform the processor of the initial time of launch and surface impact;

b. the thermistor should be mounted in a position (perhaps near the tail) where ventilation would be maximum. This would provide for better intercomparison with concurrent XBT or CTD drops;

c. the initial probe mass (M_0) should be controlled during manufacture to within a 1% tolerance so that variations in fall speed due to mass variations can be reduced;

d. gyroscopic effects on probe dynamics and measurements should be analyzed, if practicable;

e. develop a new body shape nearer to ideal streamlined form with greater diameter to increase current sensitivity and better thermistor ventilation;

f. a pressure signal should be multiplexed and sent up the signal wire.

5. REFERENCES

1. Green, A.W., 1981: Bulk Dynamics of the XBT (T-7), Submitted to Deep Sea Research.

2. Heinmiller, R.H., C.C. Ebbesmeyer, B.A. Taft, D.B. Olson and O.P. Nikitin, 1981: Intercomparisons of XBT and CTD isotherm depths in the Northwestern Atlantic, Submitted to Deep Sea Research.

3. Hoerner, S.F., 1965: Fluid Dynamic Drag, Hoerner Dynamics, Bricktown, New Jersey, U.S.A. 450 pp.

4. Saunders, K.D., A.W. Green, M.T. Bergin, 1981: A Comprehensive Graphical Representation of Data Obtained in the Acoustically Tracked Oceanographic Mooring (ATOM) Experiments, NORDA Technical Note 85, 621 pp.

TABLE 1. LINEAR FIT TO PRESSURE VS. TIME

XCP Cast	Q_0	Q_1	σ_p
4	5.65	4.63	0.40
8	9.60	4.63	1.43
9	7.18	4.54	2.27
11	11.88	4.61	2.23
12	15.03	4.51	2.97
13	6.44	4.57	0.87
14	12.77	4.42	2.80
15	4.66	4.58	0.59
16	38.10	4.66	1.14
17	4.19	4.50	1.22
18	4.52	4.51	1.25
20	5.02	4.53	1.62
21	-0.60	4.85	0.76
22	8.99	4.44	1.26
23	6.73	4.55	1.00
25	10.57	4.47	2.29
26	8.04	4.53	1.30
29	3.84	4.63	0.40
31	9.79	4.55	1.71
32	-4.28	4.58	1.66
34	11.27	4.69	2.81
35	8.03	4.64	1.79
36	12.29	4.63	2.46
37	5.00	4.57	0.89
38	7.99	4.61	1.55
μ	8.504	4.576	
σ	7.464	0.0883	
$t_{.05}\sigma/\sqrt{n-1}$	3.145	0.0372	
$\mu_{+.05}$	11.649	4.613	
$\mu_{-.05}$	5.359	4.539	

TABLE 2. QUADRATIC FIT TO XCP PRESSURE VS. TIME

XCP Cast	Q_0	Q_1	Q_2	σ_p
4	5.65	4.63	-0.141×10^{-5}	0.40
8	7.29	4.71	-0.653×10^{-3}	1.40
9	1.55	4.70	-0.891×10^{-3}	1.93
11	11.26	4.63	-0.128×10^{-3}	2.22
12	3.85	4.87	-0.235×10^{-2}	2.02
13	3.93	4.65	-0.526×10^{-3}	0.70
14	3.16	4.69	-0.150×10^{-2}	1.79
15	2.19	4.68	-0.110×10^{-2}	0.57
16	32.46	4.85	-0.124×10^{-2}	0.62
17	-0.22	4.62	-0.714×10^{-3}	1.24
18	-1.18	4.70	-0.134×10^{-2}	1.09
20	-1.00	4.69	-0.837×10^{-3}	1.20
21	-0.94	4.87	-0.228×10^{-3}	0.76
22	2.85	4.60	-0.849×10^{-3}	0.51
23	1.67	4.70	-0.883×10^{-3}	0.63
25	6.30	4.58	-0.649×10^{-3}	2.17
26	6.20	4.52	-0.313×10^{-3}	1.26
29	6.38	4.68	$+0.124 \times 10^{-2}$	0.38
31	4.95	4.68	-0.758×10^{-3}	1.36
32	-12.44	4.85	-0.194×10^{-2}	1.47
34	1.21	4.96	-0.148×10^{-2}	1.68
35	4.94	4.74	-0.638×10^{-3}	1.71
36	3.55	4.88	-0.144×10^{-2}	1.56
37	3.28	4.61	-0.399×10^{-3}	0.85
38	2.03	4.79	-0.117×10^{-3}	1.14
μ	3.956	4.708	-8.315×10^{-4}	
σ	7.323	0.115	7.042×10^{-4}	
$t_{.05}\sigma/\sqrt{n-1}$	3.085	0.0485	2.967×10^{-4}	
$\mu_{+.05}$	7.041	4.76	-5.35×10^{-4}	
$\mu_{-.05}$	0.871	4.66	-1.13×10^{-3}	

TABLE 3

$A(10^{-5})$	$B(10^{-5})$	C_D	v(z=0)	Q_0	Q_1	$Q_2(10^{-3})$
7.06	6.00	0.35	15	3.6625	4.7007	-0.979
	10.00			3.6772	4.6994	-1.169
	20.00			3.7283	4.6943	-1.613
	30.00			3.7963	4.6877	-2.017
	6.00		0	-2.353	4.7034	-0.979
			5	0.622	4.7022	-0.979
			10	2.387	4.7014	-0.979
		0.05		4.085	12.450	-6.863
		0.10		5.249	8.797	-3.425
		0.15		4.946	7.181	-2.282
		0.20		4.554	6.218	-1.711
		0.25		4.207	5.562	-1.370
		0.30		3.912	5.077	-1.141

TABLE 4. PHYSICAL CHARACTERISTICS OF XBT (T-7) AND XCP

	XBT(T-7)	XCP
Initial Probe Mass	0.740 kg	1.480 kg
Water Displacement	0.161 kg	0.650 kg
Initial Bulk Drag Coefficient	0.125 (Hoerner, 1965:	0.35 torpedo)
Relative Mass Change ($m^{-1} \times 10^{-5}$)	14.1	7.17
Relative Change of C_D ($m^{-1} \times 10^{-4}$)	0.6	1.13
Nominal Fall Speed ($m\ s^{-1}$)	6.5	4.7
Probe Length	0.215 m	0. m
Probe Cross Section Area ($m^2 \times 10^{-3}$)	2.11	2.

APPENDIX A

LIST OF SYMBOLS

A	$(M_0 - M)/(M_0\ 700\ m)$, the relative change in probe mass
B	$[C_0 - C_D(700\ m)]/C_D\ 700\ m$
C_0	Initial drag coefficient of the probe after impact
C_D	The probe drag coefficient
g	Gravity acceleration 9.8 $m\ s^{-1}$
g'	$g(1 - M_w/M_0)$
H	Probe range
M	Probe mass
M_0	Probe mass at impact
M_w	Mass of displaced water
N	Order of the least squares polynomial fit
p	Pressure
P_0	Observed CTD pressure
S	Cross section area of probe normal to axis of figure
Q_n	n^{th} coefficient of least squares polynomial
T	Temperature
T_0	Observed temperature
T_0(CTD)	Observed CTD temperatures
T(offset)	Computed temperature offset
T_0(XCP)	Observed XCP temperature
t	Time
v	Vertical speed
V_T	$(2M_0 g'/\rho_w C_0 S)^{1/2}$, the nominal fall speed
z	Depth, z 0
μ	Mean value
σ	Standard deviation

III. Summary Program Documentation

PROGRAM: XCP-FALL-RATE

PURPOSE: To compute emprical rates of XCP's based on simultaneous XCP and CTD temperature data

MACHINE: UNIVAC 1108

LANGUAGE: FORTRAN V

AUTHOR: Kim David Saunders

FILE LOCATIONS: Absolute, Relocatable and symbolic
Elements - CODE331*WORKFILE.MAIN/XCP

INPUT: Unit 8 - input FEB File - XCP data

Unit 9 - input FEB File - CTD data

Unit 5 -

line 1: ISEGX, ISEGY

line 2: PLIMX (1), PLIMX (2)

line 3: PLIMX (3)

line 4: ANS

line 5: IDEGREE

ISEGX - seg. no. for XCP cast on unit 8

ISEGC - seg. no. for CTD cast on unit 9

PLIMX (1) - upper pressure limit for computing temperature offset

PLIMX (2) - lower pressure limit for computing temperature offset

PLIMX (3) - starting pressure for T/P intercomparison

ANS - yes (for XCP temp. smoothing) no (for no smoothing)

IDEGREE - degree of fitting polynomial

OUTPUT: Unit 6 - Pressure limits (repeat of input)

- Short FEB file headers
- Averages of XCP and CTD temperatures
- Temperature offset
- NT, N2, T1, T2
- Coefficients of best fit polynomial
- Standard deviation of fit

Unit 20 - Listings of pressures, temperatures and fall times

Unit 25 - DISSPLA Plots

NT - Number of temperature points in input for interpolation

N2 - Number of points to be interpolated

T1 - Starting temperature

T2 - Ending temperature

SUBROUTINES CALLED: INITLZ

XCPRDR

CTDRDR

TMPOFS

TMPCOR

DPTCOM

PTINTP

TTINTP

FITTER

PLOTER

INTRPL

<<DISPLA Package subroutines>>

ADDITIONAL INFORMATION:

1. The XCP FEB File must have the form:
   4 variables: P, T, U, V

2. The CTD FEB Fi , must have the form:
   8 variables, P, T, (other 6 are not used)

3. The maximum length of a segment is 1000 cycles

4. Each drop is stored in a separate segment, and the data should be interpolated to 1 decibar pressure surfaces, and low pass filtered with a cutoff of about 8 decibars. The filtering and interpolation may be done via ZFILT and ZINTERP. Both of these are standard FEB File utilities.

IV. Runstream to Create and Run the Program

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).MAKER/XCP-CALIB
      1       @USE W.CODE331*WORKFILE
      2       @ASG.UP  XCP-PR-2.F///2000
      3       @FREE XCP-PR-2
      4       @ASG.AX XCP-PR-2
      5       @BRKPT PRINTS/XCP-PR-2
      6       @HDG.P TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331
      7       @PRT.S W.MAKER/XCP-CALIB
      8       @PRT.S W.MAP/XCP-CALIB
      9       @PRT.S W.RUN-1/XCP-CALIB
     10       @ASG.T 20.F///2000
     11       @ASG.UP 25.F///2000
     12       @FREE 25
     13       @ASG.AX 25.
     14       @FRS 25.
     15       @FRS 20.
     16       @ASG.AX W.
     17       @FOR.S W.MAIN/XCP-CALIB
     18       @FOR.S W.INITL2/XCP-CALIB
     19       @FOR.S W.XCPRDR/XCP-CALIB
     20       @FOR.S W.CTDRDR/XCP-CALIB
     21       @FOR.S W.TMPOFS/XCP-CALIB
     22       @FOR.S W.TMPCOR/XCP-CALIB
     23       @FOR.S W.DPTCOM/XCP-CALIB
     24       @FOR.S W.PTINTP/XCP-CALIB
     25       @FOR.S W.TTINTP/XCP-CALIB
     26       @FOR.S W.FITTER/XCP-CALIB
     27       @FOR.S W.PLOTER/XCP-CALIB
     28       @FOR.S KDSFILE.INTRPL/AKIMA
     29       @ADD.PL W.MAP/XCP-CALIB
     30       @PACK W.
     31       @PREP W.
     32       @BRKPT PRINTS
     33       @SYM.U XCP-PR-2..P4
     34       @FREE XCP-PR-2

@PRT.S W.MAP/XCP-CALIB
```

V. Map Runstream

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).MAP/XCP-CALIB
      1       @MAP.IS  .W.XQT/XCP-CALIB
      2       IN W.MAIN/XCP-CALIB..INITL2/XCP-CALIB..XCPRDR/XCP-CALIB
      3       IN W.CTDRDR/XCP-CALIB
      4       IN W.TMPOFS/XCP-CALIB..TMPCOR/XCP-CALIB..DPTCOM/XCP-CALIB
      5       IN W.PTINTP/XCP-CALIB..TTINTP/XCP-CALIB..FITTER/XCP-CALIB
      6       IN W.PLOTER/XCP-CALIB.CODE331*KDSFILE.2READ
      7       IN CODE331*KDSFILE.INTRPL/AKIMA
      8       LIB DISSPLA*LIBR-D.

@PRT.S W.RUN-1/XCP-CALIB
```

VI. Listing of a Typical Program Runstream with an Example of the Output to Unit 6

```
TEST RUN : XCP/CTC CALIBRATION PROGRAM - W D SAUNDERS - X4733 -NOPDA331

COUE331.WORKFILE(1).RUN-1/XCP-CALIB
     1        @FREE 25
     2        @ASG,UP XCP-25,F///2000
     3        @FREE XCP-25
     4        @USE 25,XCP-25
     5        @ASG,AX 25
     6        @ASG,TVJ IN1,U9V,5937
     7        @ASG,T 9,F///2000
     8        @ASG,T 8,F///2000
     9        @MOVE IN1,0
    10        @COPY,G IN1,9
    11        @FREE,S IN1
    12        @ASG,TVJ IN2,U9V,0639
    13        @COPY IN2,8
    14        @FREE IN2
    15        @XQT W.XQT/XCP-CALIB
    16        9,1
    17        50,70
    18        120
    19        YES
    20        1
    21        @FREE 25


@ASG,T 20,F///2000
FACILITY WARNING  100000000000


@ASG,UP XCP-25,F///2000
FACILITY REJECTED 440000400000


@FREE XCP-25
FACILITY WARNING  100000000000


@ASG,AX XCP-25.
READY


@ERS XCP-25.
FURPUR27R38 E33 RL73R1 04/10/81 15:18:57
END ERS.


@USE 25,XCP-25
READY


@ERS 20.
FUNPUR27R38 E33 RL73R1 04/10/81 15:19:00
END ERS.


@ASG,AX W.
FACILITY WARNING  102000000000
```

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - D SAUNDERS - X4733 -NORDA331

@ASG,TVJ IN1,U9V,5937
READY


@ASG,T 9,F///2000
FACILITY WARNING  100000000000


@ASG,T 8,F///2000
FACILITY WARNING  100000000000


@MOVE IN1,0
FORPUR27R38 E33 RL73R1 04/10/81 15:19:07


@COPY,G IN1,9
CODE331 *OUTF1(0) COPIED ON 06/11/80 AT 16:22:53 STUDS REFL-5937*
    29 BLOCKS COPIED.
EOF ENCOUNTERED ON INPUT TAPE


@FREE,S IN1
READY


@ASG,TVJ IN2,U9V,0639
READY


@COPY IN2,8
FORPUR27R38 E33 RL73R1 04/10/81 15:20:52
    838 BLOCKS COPIED.
EOF ENCOUNTERED ON INPUT TAPE


@FREE IN2
READY


@XQT *,XQT/XCP-CALIB
 ENTER THE INPUT SEGMENTS FOR THE XCP AND CTD DATA IN FREE FORMAT


 ENTER THE UPPER AND LOWER PRESSURE LIMITS FOR COMPUTING THE TEMPERATURE OFFSET


 PLIMX(1) =          50.000000
 PLIMX(2) =          70.000000


 ENTER STARTING PRESSURE FOR T/P INTERCOMPARISON


 RD    R  9V-SP       9  C     9      P70    4  15   15   40
 DO YOU WANT SMOOTHING OF XCP TEMPERATURES ??
```

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NUFDA331

RU    9   70380     1  100039      491    F  40  40 100
TXAV     =        25.411
TCAV     =        25.473
T OFFSET=         -.51666-01



*****************************************
NT =    377
N2 =    128
T1 =          24.500000
T2 =          11.700000

ENTER THE DEGREE OF FITTING POLYNOMIAL

        I                A(I)

       1                5.239845
       2                4.574780
STANDARD DEVIATION OF PRESSURE =          .91963314

BEGIN OF DISSPLA PLOT GENERATION.



PLOT NO.   1 WITH THE TITLE
     DP/DT VS PRESSURES
HAS BEEN COMPLETED.

PLOT ID. READS
PLOT 1     15.28.45   FRI 10 APR, 1981    JOB=EUKDS    , NAVOCEA  DISSPLA VER  8.0

DATA FOR PLOT

NO. OF CURVES DRAWN  1
-----------------------------
HORIZ. AXIS LENGTH   7.0 INS.
VERT. AXIS LENGTH    9.0 INS.
-----------------------------
 ORIZ. ORIGIN   .0000    VERT.  ORIGIN   .0000
-----------------------------
HORIZ. AXIS LINEAR
STEP SIZE   .14290-01 UNITS/INCH
-----------------------------
```

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - R D SAUNDERS - X8733 -NORDA331

VERT.  AXIS LINEAR
STEP SIZE    .6667+02 UNITS/INCH
-----------------------------

    ..................................
    . LOCATION OF CURRENT PHYSICAL ORIGIN .
    . X=   1.00       Y=    .55      INCHES .
    .    FROM LOWER LEFT CORNER OF PAGE    .
    ..................................


TEST RUN : XCP/CTD CALIBRATION PROGRAM - R D SAUNDERS - X8733 -NORDA331

                    PLOTTING COMMENCING
                    ...................

 ..... DISSPLA VERSION 8.0 .....
 NO. OF FIRST PLOT  2 ...
```

```
PLOT NO.    2 WITH THE TITLE
     DP/DT VS PRESSURE - FILTERED$
HAS BEEN COMPLETED.

PLOT ID. READS
PLOT 2      15.29.04    FRI 10 APR, 1981    JOB=EUNDS    , NAVOCEA  DISSPLA VER  8.0


DATA FOR PLOT

NO. OF CURVES DRAWN  1
 ----------------------------
HORIZ. AXIS LENGTH    7.0 INS.
VERT. AXIS LENGTH     9.0 INS.
----------------------------
HORIZ. ORIGIN    .0000     VERT.  ORIGIN    .0000
----------------------------
HORIZ. AXIS LINEAR
STEP SIZE    .1429+01 UNITS/INCH
----------------------------
VERT.  AXIS LINEAR
STEP SIZE    .6667+02 UNITS/INCH
----------------------------

    ..................................
    . LOCATION OF CURRENT PHYSICAL ORIGIN .
    . X=   1.00       Y=    .55      INCHES .
    .    FROM LOWER LEFT CORNER OF PAGE    .
    ..................................


END DISSPLA --      2830 VECTORS GENERATED IN      2 PLOT FRAMES.


BEOF
BEOF IGNORED - IN CONTROL MODE
```

VII. Program Listings

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).MAIN/XCP-CALIB
      1                 CALL INITL2        @ INITIALIZE PROGRAM
      2                 CALL XCPRDR        @ READ XCP DATA FROM FER FILE
      3                 CALL CTDRDR        @ READ CTD DATA FROM FER FILE ON UNIT 9
      4                 CALL TMPOFS        @ CALCULATE TEMPERATURE OFFSET
      5                 CALL TMPCOR        @ CORRECT XCP TEMPERATURES
      6                 CALL DPTCOM        @ SELECT COMMON DEPTH INTERVAL
      7                 CALL PTINTP        @ INTERPOLATE PRESSURE ON TEMPERATURE
      8                 CALL TTINTP        @ INTERPOLATE TIME ON TEMPERATURE
      9                 CALL FITTER        @ FIT THE PRESSURE/TIME CURVE
     10                CALL PLOTER        @ PLOT THE PRESSURE/TIME CURVE
     11                STOP
     12                END

@PRT.S W.INITL2/XCP-CALIB
```

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X8733 -NORDA331

CODE331*WORKFILE(1).INITL7/XCP-CALIB
      SUBROUTINE INITL7
C
C-----------------------------------------------------------------
C
C  PURPOSE : TO INITIALIZE THE XCP CALIBRATION PROGRAM
C
C-----------------------------------------------------------------
C
      COMMON /DIAGS/ IDIAG(10)
      COMMON /RDATA/ VR(10000)
      COMMON /RDOCI/ IDOCR(100)
      COMMON /RDOCF/ FDOCR(100)
      COMMON /RDOCA/ ADOCR(100)
      COMMON /RHDR/ IRHDR(100)
      COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
      COMMON /BLOCK2/ PLIMX(10),PLIMC(10),ISEGC,ISEGX,TOFSET
      COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
      REAL BUFFER(10,1000)
      EQUIVALENCE (VR(1),BUFFER(1,1))
      WRITE(6,100)
      READ(5,200) ISEGX,ISEGC
100   FORMAT(' ENTER THE INPUT SEGMENTS FOR THE ',
     1 'XCP AND CTD DATA IN FREE FORMAT'//)
200   FORMAT()
2000  WRITE(6,101)
      READ(5,200) PLIMX(1),PLIMX(2)
      WRITE(6,103) PLIMX(1),PLIMX(2)
103   FORMAT(' PLIMX(1) = ',G20.8/' PLIMX(2) = ',G20.8//)
         IF ( PLIMX(1).LT.PLIMX(2)) GOTO 1000
            IF ( PLIMX(1) .EQ. PLIMX(2) ) GOTO 2000
            P = PLIMX(1)
            PLIMX(1)=PLIMX(2)
            PLIMX(2)= P
1000     CONTINUE
      PLIMC(1) = PLIMX(1)
      PLIMC(2) = PLIMX(2)
101   FORMAT(' ENTER THE UPPER AND LOWER PRESSURE LIMITS FOR ',
     1 'COMPUTING THE TEMPERATURE OFFSET'//)
      WRITE(6,102)
      READ(5,200) PLIMX(3)
      PLIMC(3)=PLIMX(3)
102   FORMAT(' ENTER STARTING PRESSURE FOR T/P INTERCOMPARISON'//)
      RETURN
      END

@PRT.S W.XCPROR/XCP-CALIB
```

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).XCPRDR/XCP-CALIB

```
                SUBROUTINE XCPRDR
      C
      C
      C-----------------------------------------------------------------
      C                                                                 I
      C PURPOSE : TO READ IN XCP DATA FROM LOGICAL UNIT 6 INTO THE      I
      C           MAIN BUFFER ARRAY AND TRANSFER THE DATA TO THE        I
      C           PRESSURE AND TEMPRATURE ARRAYS. TO COMPUTE THE XCP    I
      C           TIME ARRAY FROM THE PRESSURE ARRAY. USING THE         I
      C           INVERSE OF TOM SANFORD'S ALGORITHM.                   I
      C                                                                 I
      C-----------------------------------------------------------------
      C
      C
                COMMON /DIAGS/ IDIAG(10)
                COMMON /RDATA/ VR(10000)
                COMMON /RDOCI/ IDOCR(100)
                COMMON /RDOCF/ FDOCR(100)
                COMMON /RDOCA/ ADOCR(100)
                COMMON /RHDR/ IRHDR(100)
              COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
              COMMON /BLOCK2/ PLIMX(10),PLIMC(10),ISEGC,ISEGX,TOFSET
              COMMON /BLOCK3/ TEMP(1000),P(1000),TINF(1000)
                COMMON /BLOCK7/ YES,ANS
              REAL BUFFER(4,1000),A,B,C,TXX(1000)
                INTEGER YES,ANS
              EQUIVALENCE (VR(1),BUFFER(1,1)),(VR(1001),TXX(1))
                YES = 6HYES
                IDIAG(1) = 1
                IDIAG(2) = 1
                IDIAG(3) = 1000
                IDIAG(4) = 1000
                IDIAG(5) = 10
                IDIAG(6) = 40
                IDIAG(7) = 40
                IDIAG(8) = 100
                IDIAG(9) = 1
                IDIAG(10)= 1
                DATA C,B,A/3.1,4.544,-0.0006749/
      C
      C
      C-----------------------------------------------------------------
      C                 CLEAR THE BUFFER
      C-----------------------------------------------------------------
      C
      C
                      DO 999 I = 1,10000
                       VR(I) = 0
      999             CONTINUE
                DO 998 I = 1,1000
                      PX(I) = 0
                      TX(I) = 0
                      TIMEX(I)=0
      998       CONTINUE
      C
      C
```

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

```
C-----------------------------------------------------------------
C              LOAD IN THE XCP DATA
C-----------------------------------------------------------------
C
C
      CALL ZRFAD(A,IF,ISFGX)
      DO 1000 I = 1,1000
         IPX = BUFFER(1,I)
         IF(IPX .GT. 1000 .OR. IPX .LT. 1 ) GOTO 1000
         PX(IPX) = BUFFER(1,I)
         TX(IPX) = BUFFER(2,I)
         TIME(IPX)=(-B+SQRT(B**2-4*A*(C-IPX)))/(2*A)
1000  CONTINUE
C
C
C-----------------------------------------------------------------
C
C SMOOTH XCP TEMPERATURES IF DESIRED
C
C-----------------------------------------------------------------
C
C
      WRITE(6,100)
      READ(5,101) ANS
100   FORMAT(' DO YOU WANT SMOOTHING OF XCP TEMPERATURES ??'//)
101   FORMAT(1A6)
      IF(ANS .NE. YES) RETURN
      DO 1001 I = 3,998
      TXX(I) = .25*(TX(I+1)+TX(I-1))+0.2*(TX(I+2)+TX(I-2))+0.3*TX(I)
1001  CONTINUE
      DO 1002 I = 3,998
      TX(I)=TXX(I)/1.2
1002  CONTINUE
      RETURN
      END
```

@PRT.S W.CTDRDR/XCP-CALIB

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).CTDROR/XCP-CALIP

```
      SUBROUTINE CTDROR
C
C
C-----------------------------------------------------------------------
C                                                                       I
C PURPOSE : TO READ IN CTD DATA FROM LOGICAL UNIT 8 INTO THE            I
C           MAIN BUFFER ARRAY AND TRANSFER THE DATA TO THE              I
C           PRESSURE AND TEMPRATURE ARRAYS. TO COMPUTE THE XCP          I
C           TIME ARRAY FROM THE PRESSURE ARRAY, USING THE               I
C           INVERSE OF TOM SANFORD'S ALGORITHM.                         I
C                                                                       I
C-----------------------------------------------------------------------
C
C
      COMMON /DIAGS/ IDIAG(10)
      COMMON /RDATA/ VR(10000)
      COMMON /RDOCI/ IDOCR(100)
      COMMON /RDOCF/ FDOCR(100)
      COMMON /RDOCA/ ADOCR(100)
      COMMON /RHDR/ IRHDR(100)
      COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
      COMMON /BLOCK2/ PLIMX(10),PLIMC(10),ISEGC,ISEGX,TOFSET
      COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
      COMMON /BLOCK7/YES,ANS
      INTEGER YES,ANS
      REAL BUFFER(8,1000),TCX(1000)
      REAL BUFIN(10,1000)
      EQUIVALENCE(VR(1),BUFIN(1,1))
      EQUIVALENCE (VR(1),BUFFER(1,1)),(VR(1001),TCX(1))
      IDIAG(1) = 1
      IDIAG(2) = 1
      IDIAG(3) = 1000
      IDIAG(5) = 10
      IDIAG(6) = 100
      IDIAG(7) = 100
      IDIAG(8) = 100
      IDIAG(9) = 1
      IDIAG(10) = 1
C
C
C-----------------------------------------------------------------------
C CLEAR THE BUFFER          ***************************
C
C
      DO 999 I = 1,10000
       VR(I) = 0
999   CONTINUE
      DO 998 I = 1,1000
       P(I) = 0
       T(I) = 0
998   CONTINUE
C
C
C-----------------------------------------------------------------------
C LOAD IN THE CTD DATA          ****************************
C
```

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

```
      C
      C
                CALL ZREAD(9,IF,ISFGC)
                IF( IRHDR(1).NE.8 .AND. IRHDR(1).NE.10) GOTO 9999
                DO 1000 I = 1,1000
                 IF(IRHDR(1) .NE.8) GOTO 2008
                   IPC = BUFFER(1,I) + .4999
                   IF(IPC.LT.1 .OR. IPC.GT.1000) GOTO  1000
                   PC(IPC) = BUFFER(1,I)
                   TC(IPC) = BUFFER(2,I)
      2008      CONTINUE
                IF(IRHDR(1).NE.10) GOTO 2010
                   IPC = BUF10(1,I) + .4999
                   IF(IPC.LT.1 .OR. IPC.GT.1000) GOTO  1000
                   PC(IPC) = BUF10(1,I)
                   TC(IPC)=BUF10(2,I)
      2010      CONTINUE
      1000      CONTINUE
                IF(ANS .NE. YES) RETURN
                DO 1001 I = 3,998
                TCX(I)=.25*(TC(I+1)+TC(I-1))+0.2*(TC(I+2)+TC(I-2))+0.3*TC(I)
      1001      CONTINUE
                DO 1002 I = 3,998
                TC(I)=TCX(I)/1.2
      1002      CONTINUE
                RETURN
      9999          WRITE(6,10)
      10            FORMAT(' ERROR IN FFR FILE STRUCTURE - TERMINATING')
                    STOP
                END
```

@PRT,S W,TMPOFS/XCP-CALIB

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X0733 -NORDA331

CODE331*WORKFILE(1).TMPOFS/XCP-CALIB
      1                 SUBROUTINE TMPOFS
      2       C
      3       C
      4       C-----------------------------------------------------------------------
      5       C
      6       C PURPOSE  : THIS SUBROUTINE COMPUTES THE AVERAGE OFFSET TEMPERATURE    *
      7       C            OVER A GIVEN ASSUMED DEPTH RANGE ( OBTAINED IN INIT(2)     *
      8       C            FOR CORRECTING THE XCP TEMPERATURES.                       *
      9       C                    TOFSET = AVERAGE( TCTD - TXCP)                     *
     10       C
     11       C-----------------------------------------------------------------------
     12       C
     13       C
     14                 COMMON /DIAGS/ IDIAG(10)
     15                 COMMON /RDATA/ VR(10000)
     16                 COMMON /RDOCT/ IDOCP(100)
     17                 COMMON /RDOFF/ FDOCR(100)
     18                 COMMON /RDOCA/ ADOCR(100)
     19                 COMMON /RHDR/ IRHDR(100)
     20              COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
     21              COMMON /BLOCK2/ PLIMX(10),PLIMC(10),ISEGF,ISEGL,TOFSET
     22              COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
     23              REAL BUFFER(10,100)
     24              EQUIVALENCE (VR(1),BUFFER(1,1))
     25                 REAL TXAV,TCAV
     26       C
     27       C
     28                 TXAV = 0
     29                 TCAV = 0
     30                 N = 0
     31                 I1 = PLIMX(1)
     32                 I2 = PLIMX(2)
     33                 DO 1000 I = I1,I2
     34                    TXAV = TXAV + TX(I)
     35                    TCAV = TCAV + TC(I)
     36                    N = N + 1
     37       1000      CONTINUE
     38                 TXAV = TXAV/N
     39                 TCAV = TCAV/N
     40                 TOFSET=TXAV-TCAV
     41                 WRITE(6,100) TXAV,TCAV,TOFSET
     42       100       FORMAT(' TXAV     =',G16.5/' TCAV    =',G16.5/
     43            1    1 ' OFFSET=',G16.5///)
     44                      RETURN
     45                      END

@PRT,S W.TMPCOR/XCP-CALIB
```

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).TMPCOR/XCP-CALIB
      1                 SUBROUTINE TMPCOR
      2                 COMMON /DIAGS/ IDIAG(10)
      3                 COMMON /RDATA/ VR(10000)
      4                 COMMON /RDOCI/ IDOCR(100)
      5                 COMMON /RDOCF/ FDOCR(100)
      6                 COMMON /RDOCA/ ADOCR(100)
      7                 COMMON /RHDR/ IRHDR(100)
      8              COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
      9              COMMON /BLOCK2/ PLIMX(10),PLIMC(10),ISEGC,ISEGX,TOFSET
     10              COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
     11              REAL BUFFER(10,1000)
     12              EQUIVALENCE (VR(1),BUFFER(1,1))
     13                 DO 1000 I = 1,1000
     14                    IF(TC(I) .EQ. 0 .OR. TX(I) .EQ. 0) GOTO 1000
     15                    TX(I) = TX(I) - TOFSET
     16       1000      CONTINUE
     17                 RETURN
     18                 END

@PRT,S W.OPTCOM/XCP-CALIB


TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).OPTCOM/XCP-CALIB
      1                 SUBROUTINE OPTCOM
      2                 COMMON /DIAGS/ IDIAG(10)
      3                 COMMON /RDATA/ VR(10000)
      4                 COMMON /RDOCI/ IDOCR(100)
      5                 COMMON /RDOCF/ FDOCR(100)
      6                 COMMON /RDOCA/ ADOCR(100)
      7                 COMMON /RHDR/ IRHDR(100)
      8              COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
      9              COMMON /BLOCK2/ PLIMX(10),PLIMC(10),ISEGC,ISEGX,TOFSET
     10              COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
     11              REAL BUFFER(10,1000)
     12              EQUIVALENCE (VR(1),BUFFER(1,1))
     13                 I1 = PLIMX(3) + 0.001
     14                 I2 = I1
     15                 DO 1000 I = I1,1000
     16                 IF( TX(I-1).LE.TX(I) .OR. TC(I-1) .LE. TC(I) ) GOTO 1001
     17                 IF(TX(I) .EQ.0  .OR. TC(I) .EQ. 0) GOTO 1001
     18                 I2 = I2 + 1
     19       1000      CONTINUE
     20       1001       PLIMX(4) = I2-6
     21              RETURN
     22              END

@PRT,S W.PTINTP/XCP-CALIB
```

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).PTINTP/XCP-CALIB

```
                    SUBROUTINE PTINTP
                COMMON /DIAGS/ IDIAG(10)
                COMMON /RDATA/ VR(10000)
                COMMON /RDOCI/ IDOCR(100)
                COMMON /RDOCF/ FDOCR(100)
                COMMON /RDOCA/ ADOCR(100)
                COMMON /RHDR/ IRHDR(100)
            COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
            COMMON /BLOCK2/ PLIMX(10),PLIMC(10),TSFGC,TSFGX,TOFSET
            COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
            COMMON /BLOCK4/TEMP2(1000),P2(1000),TIME2(1000)
            COMMON /BLOCK5/T1,T2,DT,NT,N2
            REAL BUFFER(10,1000)
            EQUIVALENCE (VR(1),BUFFER(1,1))
             DT = 0.1
             I1 = PLIMX(3) + 0.001
             I2 = PLIMX(4) + 0.001
             T1 = TX(I1)
             T2 = TX(I2)
             IF(T1 .GT. TC(I1)) T1 = TC(I1)
             IF(T2 .LT. TC(I2)) T2 = TC(I2)
             IT1 = 10*T1 - 0.5
             IT2 = 10*T2 + 0.5
             T1 = IT1/10.
             T2 = IT2/10.
               DO 1000 I = I1,I2
                  II = I - I1 + 1
                  IJ = I2 - II +1
                  TEMP(II) = TC(IJ)
                  P(II)    = PC(IJ)
      1000      CONTINUE
                N2 = 0
                DO 1001 I = 1,1000
                   N2 = N2 + 1
                   TEMP2(N2) = T2+DT*(I-1)
                   IF(TEMP2(N2) .GE. T1) GOTO 1002
      1001      CONTINUE
      1002      N2 = N2 - 1
                DO 1003 I = 1,1000
                   P2(I) = 0
      1003      CONTINUE
                CALL INTRPL(6,NT,TEMP,P,N2,TEMP2,P2)
              RETURN
              END
```

@PRT.S W.TTINTP/XCP-CALIB

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).TTINTP/XCP-CALIF

```
                SUBROUTINE TTINTP
                COMMON /DIAGS/ IDIAG(10)
                COMMON /RDATA/ VR(10000)
                COMMON /RDOCI/ IDOCR(100)
                COMMON /RDOCF/ FDOCR(100)
                COMMON /RDOCA/ ADOCR(100)
                COMMON /RNDR/ IRNDR(100)
            COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
            COMMON /BLOCK2/ PLIMX(10),PLIMC(10),TSEGC,TSEGX,TOFSET
            COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
            COMMON /BLOCK4/TEMP2(1000),P2(1000),TIME2(1000)
            COMMON /BLOCK5/T1,T2,DT,NT,N2
            REAL BUFFER(10,1000)
            EQUIVALENCE (VR(1),BUFFER(1,1))
                I1 = PLIMX(3) + 0.001
                I2 = PLIMX(4) + 0.001
                DO 1000 I = I1,I2
                   II = I - I1 + 1
                   IJ = I2- I1 + 1
                   TEMP(II) = TX(IJ)
                   TIME(II) = TIMEX(IJ)
      1000      CONTINUE
                CALL INTRPL(6,NT,TEMP,TIME,N2,TEMP2,TIME2)
                WRITE(6,300)NT,N2,T1,T2
      300       FORMAT(//// ***************************************/
     1          ' NT = ',I5/' N2 = ',I5/' T1 = ',G20.8/' T2 = ',G20.8//)
                WRITE(20,199)
               DO 1002 I = 1,1000
                WRITE(20,200)I,PX(I),TX(I),TIMEX(I),PC(I),TC(I)
      1002      CONTINUE
      199       FORMAT(1H1,' DISPLAY OF ARRAYS PX,TX,TIMEX,PC,TC'////)
      200       FORMAT(5X,I5,5G20.8)
      C
      C
                WRITE(20,99)
                DO 1001 I = 1,1000
                   WRITE(20,100) I,TEMP2(I),P2(I),TIME2(I)
      1001      CONTINUE
      99        FORMAT(1H1,' DISPLAY OF ARRAYS TEMP2,P2,TIME2'////)
      100       FORMAT(5X,I5,3G20.8)
                RETURN
                END
```

APRT,S W,FITTER/XCP-CALIB

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).FITTER/XCP-CALIB

```
      SUBROUTINE FITTER
      COMMON /DIAGS/ IDIAG(10)
      COMMON /RDATA/ VR(10000)
      COMMON /RDOCI/ IDOCR(100)
      COMMON /RDOCF/ FDOCR(100)
      COMMON /RDOCA/ ADOCR(100)
      COMMON /RHDR/ IRHDR(100)
      COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
      COMMON /BLOCK2/ PLINX(10),PLINC(10),ISEGC,ISEGH,IOFSET
      COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
      COMMON /BLOCK4/TEMP2(1000),P2(1000),TIME2(1000)
      COMMON /BLOCK5/T1,T2,DT,NT,N2
      REAL A(1000),COEF(1000),TTT1(1000),TTT2(1000),TTT3(1000)
      REAL ALPHA(1000),BETA(1000),W(1000)
      EQUIVALENCE (VR(1),ALPHA(1)),(VR(1001),BETA(1)),(VR(2001),
     1 TTT1(1)),
     1 (VR(3001),TTT2(1)),(VR(4001),TTT3(1)),(VR(5001),A(1)),
     1 (VR(6001),COEF(1)),(VR(7001),W(1))
      DO 1000 I= 1,1000
        W(I) =  1.0
 1000 CONTINUE
C
C
 900  WRITE(6,300)
      READ(5,301,END=999) K
        IF(K .EQ. 0 .OR. K .GT. 9) RETURN
      IF(K.LT. 0) K = 1
 300  FORMAT(' ENTER THE DEGREE OF FITTING POLYNOMIAL'//)
 301  FORMAT()
        CALL ORTHLS(TIME2,P2,W,N2,0,0,COEF,ALPHA,BETA,K,TTT1,
     1                  TTT2,TTT3,IND1)
        CALL COEFS(0,COEF,ALPHA,BETA,K,A,TTT1,TTT2,TTT3,IND2)
        K1 = K + 1
        WRITE(6,99)
        DO 1001 I = 1,K1
          WRITE(6,100) I,A(I)
 1001   CONTINUE
        ERROR = 0
        DO 2000 I = 1,N2
          PP = 0
          X = TIME2(I)
          DO 2001 J = 1,K1
            PP = PP + A(J)*X**(J-1)
 2001     CONTINUE
          ERROR = ERROR + (P2(I)-PP)**2
 2000   CONTINUE
        ERROR  = ERROR/N2
        ERROR  = SQRT(ERROR)
        WRITE(6,200) ERROR
 200    FORMAT(' STANDARD DEVIATION OF PRESSURE  = ',G20.8//)
 99     FORMAT(4X,'I',15X,'A(I)'//)
 100    FORMAT(5X,I3,10X,G15.7)
        GO TO 900
 999  RETURN
      END
```

```
TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

@PRT,S W.PLOTER/XCP-CALIB



TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

CODE331*WORKFILE(1).PLOTER/XCP-CALIB
      1                 SUBROUTINE PLOTER
      2                 COMMON /DIAGS/ IDIAG(10)
      3                 COMMON /RDATA/ VR(10000)
      4                 COMMON /RDOCI/ IDOCR(100)
      5                 COMMON /RDOCF/ FDOCR(100)
      6                 COMMON /RDOCA/ ADOCR(100)
      7                 COMMON /RHDR/ IRHDR(100)
      8              COMMON /BLOCK1/ TIMEX(1000),TX(1000),PX(1000),TC(1000),PC(1000)
      9              COMMON /BLOCK2/ PLIMX(10),PLIMC(10),ISEGC,ISEGX,TOFSET
     10              COMMON /BLOCK3/ TEMP(1000),P(1000),TIME(1000)
     11              COMMON /BLOCK4/TEMP2(1000),P2(1000),TIME2(1000)
     12              COMMON /BLOCK5/T1,T2,DT,NT,N2
     13                 REAL W(1000),PW(1000),WW(1000),TT(1000)
     14                 EQUIVALENCE (W(1),V(1)),(PW(1),VR(1001)),(WW(1),VR(2001))
     15                 EQUIVALENCE (TT(1),VR(3001))
     16       C
     17       C
     18       C
     19       C-----------------------------------------------------------
     20       C
     21       C        COMPUTE FALL SPEEDS AND ASSOCIATED PRESSURES
     22       C
     23       C-----------------------------------------------------------
     24       C
     25       C
     26                 DO 1000 I = 2,N2
     27                    J = N2 - I + 2
     28                    W(I) = (P2(J)-P2(J-1))/(TIME2(J)-TIME2(J-1))
     29                    PW(I)= (P2(J)+P2(J-1))/2.
     30                    TT(I) = TIME2(J)
     31       1000      CONTINUE
     32       C
     33       C
     34       C-----------------------------------------------------------
     35       C
     36       C        PLOT THE PRESSURE TIME DERIVATIVE AS A FUNCTION
     37       C                          OF PRESSURE
     38       C
     39       C-----------------------------------------------------------
     40       C
     41       C
     42                 N1 = N2 -1
     43                 CALL COMPRS
     44                 CALL TITLE('DP/DT VS PRESSURE$',100,'DP/DT (DB/SEC)$',100,
     45            1                'PRESSURE (DB)$',100,7,.0,.1)
     46                 CALL GRAF(0.,2.,10.,0.,100.,600.)
     47                 CALL FRAME
     48                 CALL CROSS
     49                 CALL CURVE(W(2),PW(2),N1,0)
     50                 CALL ENDPL(1)
     51                 WRITE(20,500)
     52                 DO 5000 I = 1,N2
     53                   WRITE(20,501) I,W(I),PW(I)
     54       5000      CONTINUE
     55                 DO 6000 I = 2,N2
     56                   WW(I)=0.25*(W(I-1)+W(I+1))+0.5*W(I)
```

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA33)

```
 6000       CONTINUE
            CALL BGNPL(2)
            CALL TITLE('DP/DT VS PRESSURE - FILTERED$',100,
     1      'DP/DT (DB/SEC)$',100,
     1                 'PRESSURE (DB)$',100,7,.9,)
            CALL GRAF(0,.7,.10,.0,.100,.600,)
            CALL FRAME
            CALL CROSS
            CALL CURVE(DW(2),PW(2),N1,0)
            CALL ENDPL(2)
            CALL BGNPL(3)
            CALL TITLE('PRESSURE VS TIME$',100,
     1      'TIME (SEC)$',100,
     1                 'PRESSURE (DB)$',100,7,.9,)
            CALL GRAF(0,.20,.120,.0,.100,.600,)
            CALL FRAME
            CALL CROSS
            CALL CURVE(TT(2),PW(2),N1,0)
            CALL ENDPL(3)
            CALL DONEPL
            WRITE(20,500)
            DO 5010 I = 1,N2
              WRITE(20,501) I,W(I),TT(I),PW(I)
 5010       CONTINUE
 500        FORMAT(1H1,' LISTING OF ARRAYS W(I) , TT(I),  PW(I) '///)
 501        FORMAT(5X,I5,3G20.8)
            RETURN
            END
```

@PRT,S KDSFILE.INTRPL/AKIMA

TEST RUN : XCP/CTD CALIBRATION PROGRAM - K D SAUNDERS - X4733 -NORDA331

```
CODE331*KOSFILE(1).INTRPL/AKIMA
      SUBROUTINE INTRPL(IU,L,X,Y,N,U,V)
C     INTERPOLATION OF A SINGLE VALUED FUNCTION
C     ****************************************************************
C      THIS SUBROUTINE INTERPOLATES, FROM VALUES OF THE FUNCTION
C     GIVEN A ORDINATES OF INPUT DATA POINTS IN THE X-Y PLANE
C     AND FOR A GIVEN SET OF X-VALUES(ABCISSAS). THE VALUES OF
C     A SINGLE VALUED FUNCTION Y=Y(X).
C     ****************************************************************
C
C     THIS ALGORITHM WAS PUBLISHED IN COMM. ACM. 15(10)OCT 1972
C     WRITTEN BY HIROSHI AKIMA,U.S.DEPT OF COMMERCE,OFFICE OF
C     TELECOMMUNICATIONS.
C     INSTITUTE OF TELECOMMUNICATIONS SCIENCES, BOULDER COLO
C
C     ****************************************************************
C
C     THE INPUT PARAMETERS ARE
C                IU = LOGICAL UNIT NUMBER OF STANDARD OUTPUT UNIT
C                L  = NUMBER OF INPUT DATA POINTS
C                X  = ARRAY OF DIMENSION L STORING THE X VALUES
C                     (ABCISSAS) OF THE DATA POINTS IN ASCENDING ORDER
C                Y  = ARRAY OF DIMENSION L STORING THE Y VALUES
C                     (ORDINATES) OF THE INPUT DATA POINTS
C                N  = NUMBER OF POINTS AT WHICH INTERPOLATION OF THE
C                     Y VALUES (ORDINATE) IS DESIRED
C                U  = ARRAY OF DIMENSION N STORING THE X VALUES OF THE
C                     DESIRED POINTS.
C
C     OUTPUT PARAMETERS
C                V  = ARRAY OF DIMENSION N WHERE THE INTERPOLATED Y
C                     VALUES ARE STORED
C     ****************************************************************
C
C     DECLARATION STATEMENTS
C
      DIMENSION X(L),Y(L),U(N),V(N)
      EQUIVALENCE (P0,X3),(Q0,Y3),(Q1,Y3)
      REAL M1,M2,M3,M4,M5
      EQUIVALENCE (HK,DX),(HMN,X2,A1,M1),(HMX,X5,A5,M5),
     1            (J,SW,SA),(Y2,W2,W4,Q2),(Y5,W3,Q3)
C
C     ****************************************************************
C                PRELIMINARY PROCESSING
C     ****************************************************************
10    LD=L
      LM1=LD-1
      LM2=LM1-1
      LP1=LD+1
      ND=N
      IF( LM2  .LT. 0 )    GO TO 90
      IF( ND .LE. 0 )      GO TO 91
      DO 11 I=2,LD
                IF(X(I-1)-X(I))     11,95,96
11    CONTINUE
      IPV = 0
C
```

```
C          ****************************************************************
C                                MAIN DO LOOP
C          ****************************************************************
           DO 80 K = 1,N0
           UK=U(K)
C
C
C                        ROUTINE TO LOCATE DESIRED POINT
C
C
20         IF(LM2 .EQ. 0) GO TO 27
           IF(UK .GE. X(LP)) GO TO 26
           IF(UK .LT. X(1)) GO TO 25
           IMN=2
           IMX = LP
21         I = (IMN+IMX)/2
           IF(UK .GT. X(I)) GO TO 23
22         IMX = I
           GO TO 24
23         IMN = I + 1
24         IF( IMX .GT. IMN) GO TO 21
           I = IMX
           GO TO 30
25         I=1
           GO TO 30
26         I = LP1
           GO TO 30
27         I=2
C
C
C                        CHECK IF I = IPV
C
C
30         IF(I .EQ. IPV) GO TO 70
           IPV = I
C
C
C                        ROUTINES TO PICK UP NECESSARY X AND Y
C                        VALUES AND TO ESTIMATE THEM IF NECESSARY
C
C
40         J=I
           IF(J.EQ.1) J=2
           IF(J.EQ.LP1) J=LP
           X3=X(J-1)
           Y3=Y(J-1)
           X4=X(J)
           Y4=Y(J)
           A3=X4-X3
           M3=(Y4-Y3)/A3
           IF(LM2 .EQ. 0) GO TO 43
           IF(J .EQ. 2) GO TO 41
           X2=X(J-2)
           Y2=Y(J-2)
           A2=X3-X2
           M2=(Y3-Y2)/A2
           IF(J .EQ. LP1) GO TO 42
```

```
41          X5=X(J+1)
            Y5=Y(J+1)
            A4=X5-X4
            M4=(Y5-Y4)/A4
            IF(J .EQ. 2) M2=M3 + M3 - M4
            GO TO 45
42          M4=M3+M3-M2
            GO TO 45
43          M2 = M3
45          IF(J .LE. 3) GO TO 46
            A1=X2-X(J-3)
            M1=(Y2-Y(J-3))/A1
            GO TO 47
46          M1=M2+M2-M3
47          IF(J .GE. LM1) GO TO 48
            A5 = X(J+2) - X5
            M5 = (Y(J+2) - Y5)/A5
            GO TO 50
48          M5=M4+M4-M3
C
C
C                       NUMERICAL DIFFERENTIATION
C
C
50          IF( I .EQ. LP1) GO TO 52
            W2=ABS(M4-M3)
            W3=ABS(M2-M1)
            SW=W2+W3
            IF(SW .NE. 0.0) GO TO 51
            W2=0.5
            W3=0.5
            SW=1.0
51          T3=(W2*M2+W3*M3)/SW
            IF(I .EQ. 1) GO TO 54
52          W3=ABS(M5-M4)
            W4=ABS(M3-M2)
            SW=W3+W4
            IF(SW .NE. 0.0) GO TO 53
            W3 = 0.5
            W4 = 0.5
            SW = 1.0
53          T4=(W3*M3+W4*M4)/SW
            IF(I .NE. LP1) GO TO 60
            T3 = T4
            SA = A2 + A3
            T4 = 0.5*(M4+M5-A2*(A2-A3)*(M2-M3)/(SA*SA))
            X3 = X4
            Y3 = Y4
            A3 = A2
            M3 = M4
            GO TO 60
54          T4=T3
            SA=A3+A4
            T3=0.5*(M1+M2-A4*(A3-A4)*(M3-M4)/(SA*SA))
            X3 = X3 - A4
            Y3 = Y3 - M2*A4
            A3 = A4
```

```
                M3 = M2
C
C
C                       DETERMINATION OF THE COEFFICIENTS
C
C
60              U2=(2.0*(M3-T3)+M3 - T4)/A3
                G3=(-M3-M3+T3+T4)/(A3*A3)
C
C
C                       COMPUTATION OF THE POLYNOMIAL
C
C
70              UX=UK-P0
80              V(K)=D0+DX*(D1+DX*(D2+DX*D3))
                RETURN
C
C
C                       ERROR EXITS
C
C
90              WRITE(IU,2090)
                GO TO 99
91              WRITE(IU,2091)
                GO TO 99
95              WRITE(IU,2095)
                GO TO 97
96              WRITE(IU,2096)
97              WRITE(IU,2097) I,X(I)
99              WRITE(IU,2099) LA,NA
                RETURN
2090            FORMAT(1X,'   *** L = 1 OR LESS'/)
2091            FORMAT(1X,'   *** N = 0 OR LESS'/)
2095            FORMAT(1X,'   *** IDENTICAL X VALUES'/)
2096            FORMAT(1X,'   *** X VALUES OUT OF SEQUENCE'/)
2097            FORMAT(1X,'      I=',I7,10X,'X(I) =',F17.3/)
2099            FORMAT(1X,'      I=',I7,10X,'N =',I7/
     *1X,'****ERROR DETECTED IN ROUTINE INTRPL********'//)
                END
```

BRKPT PRINTS

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NORDA Report 50	2. GOVT ACCESSION NO. AD-A119 090	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Documentation of Software for Estimating Expendable Shear Profiler (XCP) Fall Rates		5. TYPE OF REPORT & PERIOD COVERED Final 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) K.D. Saunders		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Ocean Research and Development Activity NSTL Station, Mississippi 39529		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Ocean Research and Development Activity NSTL Station, Mississippi 39529		12. REPORT DATE June 1982 13. NUMBER OF PAGES 33
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution Unlimited

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

18. SUPPLEMENTARY NOTES

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

CTD	physical oceanography
XCP	instrumentation
fall rate equations	expendables
numerical modeling	ocean engineering
oceanography	

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

Expendable Current Profilers (XCPs) do not transmit a pressure signal from the instrument to the ship. The depth is determined either by an empirical equation or by comparison with a reference oceanic variable. This report provides documentation of a computer program for the determination of a least squares best fitted polynomial for determining the depth as a function of time, using temperature as measured from a CTD at the time of the profiler drop.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ATE

LMED

8