WHP Operations and Methods – October 1993

CTD Oxygen Calibration Procedure

Robert C. Millard, Jr.
Woods Hole Oceanographic Institution
Woods Hole, Massachusetts 02543 U.S.A.

1. Overview

Many conductivity-temperature-depth (CTD) devices have a polarographic dissolved oxygen sensor integrated into the system. The purpose of this report is to provide procedures that can be used to convert the CTD oxygen sensor measurements into oxygen profiles using the documented physics of the sensor and *in situ* oxygen data from the discrete samples.

2. Introduction

The polarographic oxygen sensor first described by Kanwisher (1959) has been used in oceanography for thirty years. The most commonly used oxygen sensor is manufactured by Beckman (now Sensor Medics) and a thorough discussion of this sensor's behavior is given by Greene *et al.* (1970). The Beckman polarographic dissolved oxygen (DO) probe uses a gold cathode and silver anode electrode pair immersed in a KCl electrolyte through which current flows in proportion to the oxygen consumed. The oxygen diffuses through a protective teflon membrane to reach the electrodes in quantities proportional to the surrounding oxygen. The teflon keeps the electrodes from fouling and reduces the probes velocity sensitivity. The rate of diffusion of oxygen through the teflon membrane is both pressure and temperature sensitive. The membrane temperature sensitivity requires a temperature measurement inside the polarographic oxygen sensor. While collecting cruise data, keeping the oxygen sensor membrane damp and clean (*i.e.*, free of oil contamination) between stations is essential for good, stable oxygen measurements.

The following algorithm for converting the polarographic oxygen sensor oxygen current and probe temperature measurements to oxygen (Oxm) is described by Owens and Millard (1985).

$$Oxm = \left[A \cdot \left(Oc + B \cdot \frac{dOc}{dt}\right) + C\right] \cdot Oxsat(T,S) e^{D \cdot [T + E(To - T)] + F \cdot P} \qquad (2.1)$$

Where, Oc is the oxygen current measurement;
P and T are CTD pressure ($dbar$) and temperature (°C);
To is the oxygen sensor temperature (°C);
S is salinity computed on the 1978 practical salinity scale;
A is the oxygen current slope, B the oxygen sensor lag in seconds;
and C is the oxygen current bias.

The three parameters $[D, E, and\ F]$ appearing in the exponential are the teflon membrane temperature sensitivity adjustment and the adjustment for the hydrostatic pressure

effects on the activity of oxygen. The pressure effect predicted from the idea gas equation and data of Enns, *et al.* (1964) is given as F =0.000141, (Al Bradshaw, personal communication) *versus* 0.00015 given in the table in Section 4. The analysis suggests a cross-term involving pressure and temperature in the exponent, but the predicted effect is small (less than a 0.4% change in the resultant oxygen which is nearly constant with pressure). $Oxsat(T, S)$ is the oxygen saturation value which uses the algorithm of Weiss (1970) in units of ml/l for the examples shown. The adoption of the Benson and Krause (1984) oxygen saturation formula is recommended by the Joint Panel of Oceanographic Tables and Standards (UNESCO, 1986) as it incorporates improved oxygen solubility measurements.

3. The Calibration Data

Calibration data are required in order to determine the coefficients [A–F] of the algorithm shown in equation (2.1). Calibration in the laboratory has not been successful at yielding useful field calibration parameters because of the lack of stability of the DO sensor and perhaps also the lack of inclusion of pressure dependence in the laboratory calibration. Instead, *in situ* oxygen samples are collected during CTD profiles using a Rosette multisampler or other water sample collection methods. These analytical results from these samples are used for post-measurement calibration.

The water sample oxygens are first merged with the appropriate CTD observations to be calibrated. It has been found that calibrations derived from the up-profile CTD data collected when stopped to fire the Rosette do not apply well to the down-profile CTD data because the instrument is stopped and the DO probe is flow sensitive at small lowering rates [< 0.4 m s^{-1}]. In general, down-profile CTD data are merged with corresponding up-profile water samples, either at common potential temperature levels or more commonly at corresponding pressure levels. After water sample data entry, the water sample oxygen data are quality controlled using property/property plots, such as potential temperature versus oxygen or by comparing with the oxygen saturation values near the surface. After correcting erroneous data entries, any other suspect oxygen data are flagged (see WHPO 90-1).

The units of CTD oxygen obtained using this oxygen fitting procedure are determined by the units of the input water sample oxygens and oxygen saturation values. To conform to the WOCE recommended oxygen units, one need only provide water sample and oxygen saturation values in $\mu mol/kg$ as input data to program OXFITMR as illustrated by the sample data of section 5.

The calibration of oxygen sensor data is weakly dependent on the calibration of CTD pressure, temperature and conductivity (salinity) to the following extent. A maximum error of 1% in CTD oxygen results from a temperature error of 0.25°C (in either temperature sensor used in the oxygen algorithm), or a pressure error of 50 decibars, or a salinity error of 1 psu. It is recommended that the calibration of pressure, temperature and conductivity (salinity) be carried out prior to the oxygen calibration step.

4. Determining Coefficients of the Algorithm

Use of equation (2.1) requires a non-linear least-squares regression technique in order to determine the best fit coefficients of the model for oxygen sensor behavior to the water sample observations. The program OXFITMR described in the Appendix uses *Numerical Recipes* (Press *et al.*, 1986) Fortran routines MRQMIN, MRQCOF, GAUSSJ, and COVSRT, which are protected by copyright, to perform non-linear least squares regression using Levenberg-Marquardt method. The *Numerical Recipies* Fortran routines, MRQMIN and MRQCOF, have both been slightly modified for use in program OXFITMR. The changes to these routines are given in the Appendix. The purchase of a copy of the book, *Numerical Recipies*, entitles the owner to a machine-readable copy of these routines. A Fortran subroutine [FOXY] describes the oxygen model with the derivatives of the model with respect to the six coefficients.

The program reads the data for a group of stations which are selected as described below. The time rate of change of oxygen current $\frac{dOc}{dt}$ has been found to be adequately determined using a least squares estimate over 10 to 15 second intervals. Normally all of the oxygen data for a given oxygen probe and cruise are initially fit as a single group. The data are edited to remove spurious points, *i.e.*, values less than a threshold oxygen value (OXMIN) or greater than a factor larger than unity (default is 1.2) times the saturation value. The data removed by editing are recorded in an output file of rejected observations. The routine varies the six (or fewer) parameters of the model in such a way as to produce the minimum sum of squares in the difference between the calibration oxygens and the computed values. Individual differences between the calibration oxygens and the computed values (residuals) are then compared with standard deviation σ of the residuals and any residual exceeding an edit factor of 2.8σ is removed and stored in the reject file before refitting the data. The edit factor has a default value of 2.8 but can be changed so as to minimize rejecting valid data while still eliminating erroneous values. A factor of 2.8 will have a 0.5% chance of rejecting a valid oxygen value for a normally distributed set of residuals. The iterative fitting process is continued until none of the data fail the edit criteria and the best fit to the oxygen probe model coefficients is then determined. The oxygen residuals of the final fit are stored together with station number and other measured variables in an output file. By plotting the oxygen residuals versus station, the correct station groupings for further refinements of fitting are obtained. A sample plot of oxygen fit residuals versus station number is shown in Figure 1. The average oxygen residual for each station is indicated with a solid triangle, while the standard deviation is given as a solid square. Note the shift in oxygen calibration which occurs at station 130. The anomalous stations 114 and 124 were traced to bad water sample data. For a well behaved DO probe, station groupings of between 10 and 30 stations are typical. The calibration coefficients [*A*–*F*] are stored, together with a histogram of the final fit residuals, in a histogram file. The example in Figure 2 illustrates two histograms of residuals for a fit to stations beyond station 130, as shown in Figure 1. The standard deviation σ and the histogram of residuals are an indication of the goodness of fit.

A normal distribution indicates that the fit describes all of the oxygen variation except the measurement uncertainty. When the distribution of residuals is not normal, a plot of the residuals *versus* pressure or alternately temperature can be helpful in deciding

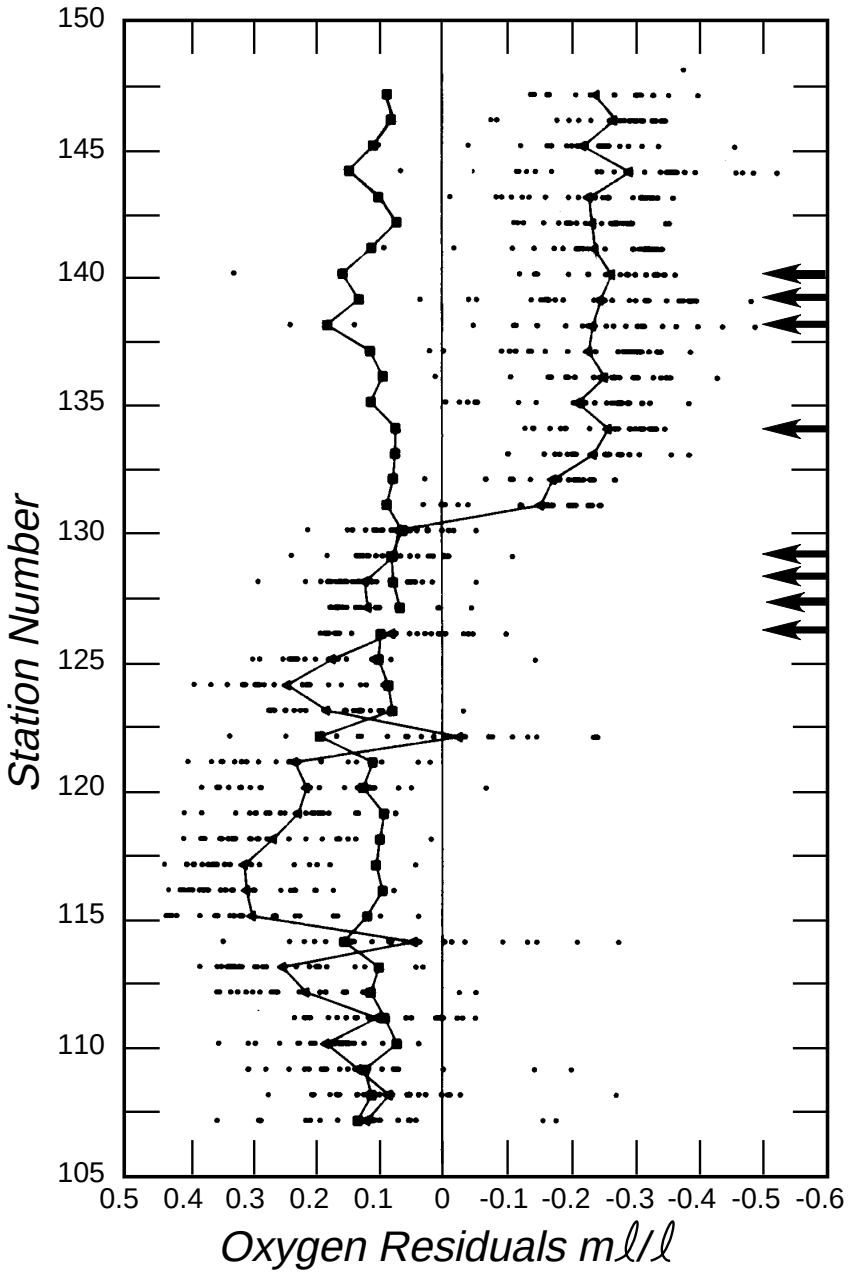

Figure 1: The oxygen residuals *versus* station number for a least squares fit to all of the oxygen observations of *Atlantis II* Cruise 109, Leg 2 between July 16 and August 4, 1981. Note the large shift in oxygen residuals at station 130. The station averaged oxygen residual is indicated by a triangle and the standard deviation by squares.

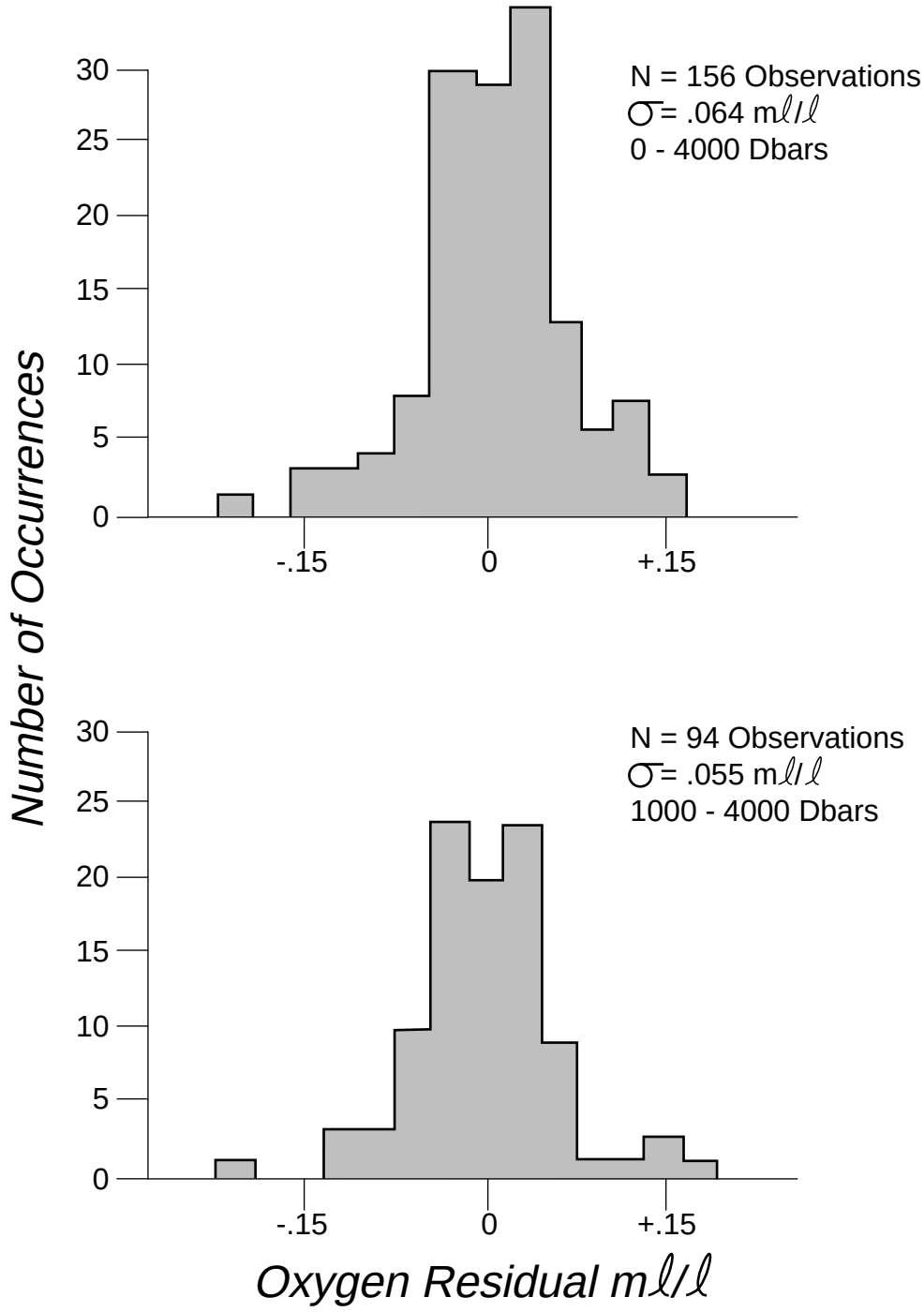

Figure 2: Histograms of final calibrated oxygen residuals for a fit to stations 131 and beyond of *Atlantis II* Cruise 109, Leg 2 of Figure 1: Upper histogram includes all pressures while lower histogram is for pressure greater than 1000 decibars.

the nature of the problem with the fitting procedure. Figure 3 illustrates a well behaved set of residuals plotted *versus* pressure.

Some of the algorithm coefficients have a limited range of values which are reasonable. For example, the oxygen sensor lag parameter (B) should be restricted to positive values with a value in the range of seconds, not hundreds of seconds. The oxygen temperature difference parameter (E) should be bounded between 0 and 1.0 to represent a weighted membrane temperature. The following values for the various algorithm coefficients have been found to be typical.

Parameter		Typical Value
A	=	0.0015 /$\mu A$for a deck unit reading in air of roughly Oc ~1.1
B	=	5.0 seconds
C	=	-.01 (non-dimensional)
D	=	-.035 /°C
E	=	.75 (non-dimensional)
F	=	0.00015 /dbar

The file of oxygen data rejected by the fitting procedure can be helpful to the hydrography personnel for further quality control of the water sample data. The oxygen residuals are differences defined as water sample oxygen minus the DO sensor value using the algorithm, so large differences indicate only that one or the other value is possibly erroneous. A large negative oxygen residual indicates that water sample oxygen is low or that the DO sensor value is high.

When attempting to fit very small station groups (less than 5 stations), it is sometimes helpful or necessary to pre-set some of the DO algorithm parameters. The parameters are arranged in the program to allow the most commonly pre-set parameters to be fixed while the others are adjusted. The pre-setting of a parameter may be supported by special constraints of the water sample oxygen data. For example, if the water sample oxygen is zero for portions of the profile, then any reading of the DO probe current must be compensated with an oxygen current bias of opposite sign, thus fixing the oxygen bias value if a zero oxygen value is to be achieved.

5. Details of the Water Sample Oxygen File to Fit

The following water sample data format is used by the oxygen fitting program OXFITMR. The file format is intended only to suggest the information required by the data fitting program. Notice that the oxygen current (Ox cur.) values in the following table and in the examples in the back are the unscaled oxygen current values multiplied by .0014. Program OXFITMR prompts for this scaling value (see page 4 of the appendix).

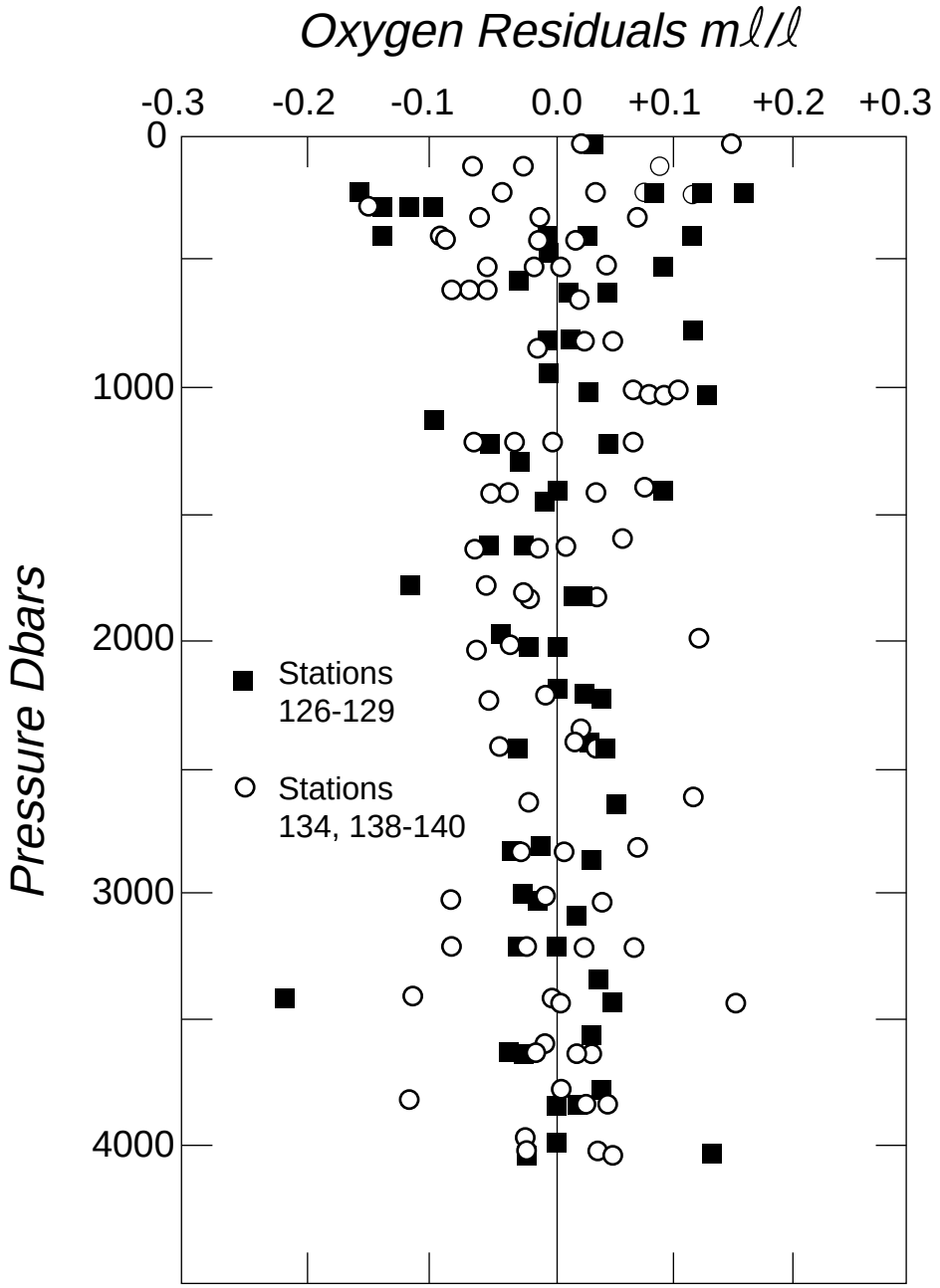

Figure 3: An example of oxygen residuals of final fits *versus* pressure for separate fits to down profiles of *Atlantis II* Cruise 109 Leg 2 stations 126–129 indicated by squares, and stations 134, 138–140 indicated by circles.

Pres. dbars	Temp. °C	Ox Cur. μA	Ox Tmp. °C	Ox-WS μmol/kg *	Ox-Sat μmol/kg *	Doc/Dt μA/s	Descent Rate dbars/s	Station .Bottle #
1.3	30.471	0.886	30.03	197.06	190.10	0.01080	0	55.11
59.1	29.006	1.142	30.34	195.95	192.71	-0.00001	0	55.10
159.0	23.901	0.795	29.96	143.99	203.30	-0.00690	0	55.09

* **Note:** To obtain oxygen calibrations in units of $m\ell/\ell$, specify data columns Ox-WS (water sample oxygen) and Ox-Sat (oxygen saturation) in units of $m\ell/\ell$.

6. Conclusion

A procedure for determining the coefficients of an oxygen algorithm for the polarographic DO oxygen probe has been developed. The procedure for merging the data from the CTD oxygen sensor for comparison with water sample data must be tailored to the particular water sample collection procedure. For example, a water sampler which permits collection of water samples while the CTD is descending at several meters per second, a design specification of the WOCE water sampler, would eliminate the down/up cast merging step created by collecting samples only on the up cast while the instrument is stopped. The oxygen algorithm might benefit from modifications shown in equation (6.1) which incorporate a flow rate dependent oxygen lag term $[G]$ and a station dependent oxygen current bias term $[H]$ as the analysis of oxygen data from one cruise has suggested.

$$\begin{aligned} Oxm &= \left[BA \cdot \left(Oc + \left\{ B + G \cdot \left(\frac{dp}{dt} \right) \right\} \cdot \frac{dOc}{dt} \right) \right. \\ &+ \left. C + H \cdot (STno) \right] \cdot Oxsat(T,S) e^{D \cdot [T + E \cdot (To - T)] + F \cdot P} \end{aligned} \tag{6.1}$$

where dp/dt is the lowering rate in $dbars$ per second, and $STno$ is the station number.

The complete program documentation and Fortran source code listing are given in the Appendix.

7. References

Benson, B. and D. Krause Jr. The concentration and isotopic fractionation of oxygen dissolved in fresh and seawater in equilibrium with the atmosphere (1984). *Limnol. Oceanogr.* **29**,(3), 620–632.

Enns, T., P. F. Scholander, and E. D. Bradshaw 1965. Effects od Hydrostatic Pressure on Gases Dissolved in Water. *The Journal of Physical Chemistry*, **69**(2), 389–392.

Greene, M. W., R. D. Gafford and D. G. Rohrbaugh, 1970. A Continuous-Profiling Deep-Submersible Dissolved Oxygen Monitor. *Mar. Technol. Soc. Six Annual Reprints*, **2**, 1502 pp.

Joyce, T., C. Corry, and M. Stalcup, eds., 1991. *Requirements for WOCE Hydrographic Programme Data Reporting*, WHPO-90-1, Rev. 1, WOCE Report No. 67/91.WHP Office, Woods Hole Oceanographic Institution, Woods Hole, Massachusetts, 70 pp.

Kanwisher, John, 1959. Polarographic Oxygen Electrode. *Limnol. Oceanogr.*, **4**(2), 210–217.

Millard, R. C., 1982. CTD calibration and data processing techniques at WHOI using the 1978 practical salinity scale. *Proceedings of the International STD Conference and Workshop, La Jolla, California, 8–11 February 1982; Marine Technology Society*, 19 pp.

Owens, B. and R. Millard, 1985. A New Algorithm for CTD Oxygen Calibration. *J. Phys. Oceanogr.*, **15**, 621–631.

Press, W., B. Flannery, S. Teukolsky, and W. Vetterling, 1986. *Numerical Recipes: The Art of Scientific Computing*, Cambridge Press, 818 pp.

Unesco, 1986. Progress on Oceanographic tables and standards 1983–1986: Work and recommendations of Unesco/SCOR/ICES/IAPSO Joint Panel. *Unesco Technical Papers in Marine Science, No. 50*, 59 pp.

Weiss R. F., 1970. The solubility of nitrogen, oxygen, and argon in water and seawater. *Deep Sea Res.*, **17**, 721–735.

APPENDIX

Program: OXFITMR
Computer: VAX/PC
Language: FORTRAN 77 (Ryan-McFarland 2.42 on PC)

LINK OXFITMR,FOXY,MRQMIN,MRQCOF,GAUSSJ,COVSRT

Input: Input data file has extension of *.OXY see below for record data structure.

Output:
Three Output data files are created with the input file name (*) and given extensions of *.HIS; *.REJ; and *.RES. Sample file contents are given in the output file description. The oxygen fit parameters and a histogram of water sample minus final fit CTD oxygen algortihm residuals are found in the histogram *.HIS file while the rejected observations are stored in the *.REJ file. The observations and residuals used in the final fit are stored in the *.RES file.

Special requirements:
Numerical Recipes (reference 1) Fortran routines [MRQMIN,MRQCOF,GAUSSJ,COVSRT] to perform non-linear least squares regression using Levenberg-Marquardt method. Function FOXY is a user supplied function with the Oxygen algorithm model after Owens and Millard (1985).

Purpose:
To fit for the CTD oxygen algorithm parameters as described in Owens & Millard (1985). CTD oxygen sensor variables are fit to water sample oxygen data to determine up to 6 parameters A-F of the oxygen algorithm (see below). For water sample data collected using the common practice of stopping the instrument during the up-profile, the data file used for fitting is composed of down-profile CTD variables merged by pressure to up-profile water sample oxygens. An edit is performed on the input oxygen data aganist both limits and the oxygen algorithm of each pass.

Fitting Procedure

An iterative fitting procedure is used for determining oxygen algorithm (model) coefficients. The water sample derived values are differenced from the CTD/O2 sensor modelled values adjusting the coefficients of the model to minimize the sum of squares of this difference. Individual differences between water sample values and the sensor model are checked against an edit criteria involving the product of the square root of the variance and a constant factor (2.8). An initial edit of the input data is done against values less than OXMIN or greater than the oxygen saturation value times a factor OXFACT. The edit factor is chosen so as to mimimize rejecting valid data while still eliminating erroneous values. A factor of 2.8 will have a .5 percent chance of rejecting a good data for a normally distributed process. After rejecting differences exceeding the data editing criteria, the remaining data are refit again in the same fashion. This continues until no further observations are rejected.

Oxygen sensor modelling

This is contained in subprogram FOXY.

The oxygen model has the following form (Owens & Millard, 1985)

$$Oxm = \left[A \cdot \left(Oc + B \cdot \frac{dOc}{dt}\right) + C\right] \cdot Oxsat(T,S)e^{D\cdot[T+E(To-T)]+F\cdot P} \qquad (7.2)$$

Where, A is the slope; B is the lag; and C is bias for oxygen current
D is TCOR; E is WT; and F is PCOR (Owens & Millard, 1985 - reference 2)
$Oxsat(T,S)$ is the oxygen saturation value.

The oxygen model requires a non-linear regression method to minimize the residual variance with respect to the six parameters A through F.

The Numerical Recipes routine MRQMIN will minimize the oxygen model with respect to the first NPAR parameters. Sometimes it is necessary to pre-set fixed values of some oxygen algorithm parameters. For example, when only a few water sample comparisons are available (few degrees of freedom) or a parameter is constrainted to a fixed value such as the oxygen current bias when the water sample oxygen goes to zero. To pre-set parameters, the order of evaluation of the parameters can be rearranged using the LISTA() array, so that pre-set oxygen model parameters can be moved beyond the first NPAR values and thus remain unchanged. The default order of evaluation of oxygen model parameters is:
A = OC slope; D = TCOR; F = PCOR; E = WT; C = OC bias and B = OC lag.

So, if NPAR is changed to 4, pre-set values for OC bias and lag would be used for fit.

$\sigma^2 = \frac{1}{N}\ \Sigma\ (Oxw - Oxm)^2$; Residual variance where Oxw is the water sample oxygen values.

The residuals are edited and refit as discussed previously according the criteria:

$|\ (Oxw - Oxm)\ |>$ Edit_Factor .

æ

Operating Procedures

The following section describes the program operating parameters and includes a sample data file with program prompts and outputs for an MS-DOS personal computer in order to facilitate installing OXFITMR on other systems.

Program OXFITMR runtime Parameters

OXFACT=1.2: The factor OXFACT multiplies the oxygen saturation value to form a data edit criteria for rejecting input data greater than this product.

OXMIN=0.0: The factor OXMIN forms a data edit criteria for rejecting input data less this value.

EDIT FACT=2.8: The factor EDIT FACT multiplys the standard deviation of each model description of the data to form a edit criteria for rejecting data before forming the next model description.

HIST BIN=.025: The parameter HISO BIN is the bin size for forming the histogram of residuals.

OCSLOPE=.0014: The parameter OCSLOPE is the nominal factor multiplying the oxygen current before fitting for the model parameters.

NPAR=6: NPAR is the number of model parameters to be determined.

PAR. # : Each oxygen algorithm parameter is assigned a number as follows;
1: (C) BIAS 2: (A) SLOPE 3: (F) PCOR 4: (D) TCOR 5: (E) WT 6: (B) LAG

NEW FIT ORDER: The order of the above PAR. # determines their order of evaluation by the fitting routine. Only the first NPAR parameters of the model are evaluated while the remaining parameters maintain the initial value.

VALUE: Each parameter (PAR. #) has a corresponding default VALUE which can be modified.

	1: BIAS	2:SLOPE	3:PCOR	4:TCOR	5: WT	6: LAG
Defaults	0.000	0.1400E-02	0.1500E-03	-0.3600E-01	0.7500E+00	0.8000E+01

Note: The following sample data are provided to illustrate the fitting procedure and provide check data for program OXFITMR. The oxygen values given in the example have units of $m\ell/\ell$. The recommended units for final WOCE data are $\mu m/kg$ rather than $m\ell/\ell$.

```
Sample Run output to the screen; ~  ~  bracket user responds

OXFITMR

Enter name of .OXY file: ~PRC5042~
Enter oxfact & a minimum acceptable oxygen value: [  1.20  0.00]: ~1.2,0.0/~
Enter std dev factor to identify reasonable data [ 2.80]
      bin size for histogram [0.02500]
      and value for ocslope [0.1400E-02]
: ~2.8,.025,.0014~
   press.   tempt.    oc     ot     ox(ws) oxsat   DOC/DT    DP/DT     sta.
 ================================================================================
 Number of parameters to fit? [ 6]: ~6/~
          1: Bias   2:Slope      3:Pcor     4:Tcor      5: Wt       6: Lag
 OX Parms: 0.000  0.1400E-02  0.1500E-03 -0.3600E-01 0.7500E+00  0.8000E+01
 order of evaluation:    2   4   3   5   1   6
Enter new fit order: [2 3 5 6 1 4]: ~/~
 order of evaluation:    2   4   3   5   1   6
 Is this order OK? [Y]/N: ~Y~
For each parameter: enter parameter #, value
   use 99/ to exit this loop...
 param #, value? ~99/~
  1 Min/Max Sta:  42.-  51.  1 StdDev:  0.1656E+00 #Obs:   77  dOx: 0.539
  2 Min/Max Sta:  42.-  51.  1 StdDev:  0.9273E-01 #Obs:   75  dOx: 0.312
  3 Min/Max Sta:  42.-  51.  1 StdDev:  0.8260E-01 #Obs:   74  dOx: 0.250
  4 Min/Max Sta:  42.-  51.  1 StdDev:  0.8092E-01 #Obs:   74  dOx: 0.227
 OX Parms:   0.015  0.1514E-02  0.1395E-03 -0.3508E-01  0.4095E+00  0.3172E+01


Test Data File

$ TYPE PRC5042.OXY

 press.   tempt.    oc     ot     ox(ws) oxsat   DOC/DT    DP/DT   sta.
================================================================================
  19.0   29.643   2.476   26.74    4.45    4.40   0.00444    0     42.1
 253.0   11.468   0.626   26.26    2.10    6.12  -0.01486    0     42.2
 507.0    7.820   0.294   17.03    9.99    6.62   0.00006    0     42.3
 757.0    5.974   0.305   10.62    1.59    6.93  -0.00006    0     42.4
1007.0    4.587   0.274    7.14    2.06    7.17  -0.00017    0     42.5
1257.0    3.548   0.263    5.06    2.20    7.35   0.00003    0     42.6
1504.9    2.977   0.262    3.70    2.39    7.47  -0.00008    0     42.7
1755.1    2.521   0.259    2.87    2.59    7.55   0.00004    0     42.8
2008.9    2.188   0.262    2.30    2.73    7.61  -0.00002    0     42.9
```

2511.3	1.782	0.269	1.66	3.14	7.69	-0.00001	0	42.10
2805.1	1.660	0.270	1.40	3.29	7.72	-0.00003	0	42.11
17.0	29.773	2.449	24.10	4.43	4.39	-0.00136	0	43.1
57.0	29.753	2.458	24.48	4.42	4.39	0.00429	0	43.2
104.9	27.605	2.110	25.12	4.13	4.46	-0.01903	0	43.3
507.0	7.541	0.332	15.68	1.93	6.72	0.00140	0	43.4
757.0	5.780	0.293	9.94	1.91	6.98	-0.00043	0	43.5
1003.0	4.462	0.277	6.63	2.15	7.20	-0.00012	0	43.6
1503.0	2.844	0.261	3.44	2.49	7.48	-0.00004	0	43.7
2005.0	2.150	0.261	2.17	2.78	7.62	0.00001	0	43.8
3007.0	1.596	0.266	1.30	3.43	7.73	-0.00011	0	43.9
3502.9	1.569	0.253	1.22	3.47	7.74	-0.00005	0	43.10
3707.1	1.556	0.249	1.19	3.61	7.74	-0.00006	0	43.11
17.0	29.782	2.710	30.73	100.00	4.37	0.00057	0	44.1
59.1	29.697	2.688	30.73	4.51	4.39	-0.00174	0	44.2
109.1	29.006	2.575	30.73	4.44	4.44	-0.01761	0	44.3
743.3	5.827	0.345	7.65	2.28	6.97	-0.00029	0	44.4
1008.9	4.474	0.271	5.48	2.26	7.20	-0.00006	0	44.5
1011.1	4.472	0.271	5.48	2.03	7.20	-0.00006	0	44.6
1505.0	2.970	0.258	3.48	2.30	7.46	-0.00009	0	44.7
2007.0	2.194	0.257	2.30	2.66	7.62	-0.00009	0	44.8
2507.0	1.863	0.258	1.66	2.95	7.68	0.00000	0	44.9
3007.0	1.617	0.264	1.33	3.30	7.73	-0.00006	0	44.10
3406.9	1.571	0.254	1.26	3.42	7.74	-0.00005	0	44.11
17.0	29.446	2.467	26.01	4.31	4.39	0.00116	0	45.1
105.0	28.942	2.380	26.52	4.31	4.43	-0.00704	0	45.2
306.9	9.294	0.421	23.83	2.30	6.42	0.00126	0	45.3
503.0	7.225	0.413	16.21	2.26	6.74	0.00024	0	45.4
753.0	5.598	0.343	10.20	2.15	7.01	-0.00053	0	45.5
1003.0	4.494	0.281	6.91	2.38	7.20	-0.00005	0	45.6
1249.1	3.623	0.279	5.02	2.42	7.35	0.00006	0	45.7
1503.0	3.158	0.262	3.70	2.62	7.45	-0.00011	0	45.8
1753.0	2.680	0.257	2.95	2.71	7.53	0.00009	0	45.9
2009.0	2.299	0.256	2.36	2.84	7.60	-0.00006	0	45.10
14.9	29.484	2.399	24.73	4.49	4.40	-0.00059	0	46.1
55.0	29.532	2.504	24.93	4.48	4.39	-0.00171	0	46.2
108.9	28.991	2.329	25.49	4.31	4.42	-0.01068	0	46.3
504.9	7.836	0.558	15.55	4.28	6.69	-0.00101	0	46.4
757.0	6.017	0.348	9.69	2.40	6.95	-0.00005	0	46.5
1009.0	4.424	0.284	6.54	2.17	7.20	-0.00009	0	46.6
1506.9	3.078	0.258	3.32	2.37	7.44	0.00003	0	46.7
2005.2	2.280	0.256	2.17	2.72	7.60	-0.00003	0	46.8
2395.0	1.950	0.255	1.66	2.76	7.66	0.00001	0	46.9
17.1	29.501	2.437	25.75	4.44	4.42	-0.00323	0	47.1
57.0	29.483	2.464	25.81	4.34	4.39	-0.00672	0	47.2
109.0	29.123	2.354	26.20	4.19	4.43	-0.00711	0	47.3
509.0	7.832	0.512	16.52	3.05	6.68	-0.00218	0	47.4
756.9	5.966	0.348	10.26	2.33	6.94	-0.00094	0	47.5

```
1007.0    4.304   0.286    6.74    2.23    7.22   0.00002    0     47.6
1506.8    3.124   0.264    3.48    2.51    7.45  -0.00007    0     47.7
2005.1    2.319   0.254    2.35    2.77    7.60   0.00004    0     47.8
2503.1    1.823   0.257    1.61    3.07    7.69  -0.00001    0     47.9
2625.0    1.731   0.260    1.43    3.08    7.71   0.00002    0     47.10
  20.9   29.412   2.676   30.73    4.41    4.41  -0.00117    0     48.1
  60.9   29.506   2.634   30.73    4.39    4.39  -0.00548    0     48.2
 509.0    7.699   0.532   17.82    2.82    6.67   0.00014    0     48.3
 759.0    6.322   0.411   10.94    2.60    6.87  -0.00213    0     48.4
1005.0    4.534   0.323    7.22    2.40    7.20  -0.00028    0     48.5
1507.1    3.209   0.265    3.88    2.46    7.44   0.00013    0     48.6
1755.0    2.797   0.262    3.06    2.54    7.50   0.00005    0     48.7
2005.2    2.284   0.257    2.50    2.76    7.60   0.00008    0     48.8
2205.0    2.040   0.264    2.09    2.94    7.65   0.00001    0     48.9
2402.9    1.911   0.259    1.82    3.01    7.67   0.00005    0     48.10
  17.0   29.405   2.527   27.51    4.47    4.41  -0.00547    0     50.1
  56.9   29.186   2.485   27.54    4.50    4.40   0.00089    0     50.2
 109.0   28.790   2.297   27.80    4.27    4.44   0.00309    0     50.3
 507.1    7.787   0.565   16.91    3.03    6.65   0.00040    0     50.4
 757.1    5.500   0.393   10.01    2.56    7.02  -0.00037    0     50.5
1505.0    3.073   0.270    3.47    2.66    7.45  -0.00003    0     50.6
2007.0    2.373   0.259    2.28    2.74    7.58  -0.00001    0     50.7
2500.9    1.840   0.258    1.66   99.99    7.67   0.00007    0     50.8
3033.1    1.615   0.256    1.36    3.12    7.73   0.00009    0     50.9
```

Using an edit factor EDIT FACT equals 2.8 the following data were rejected.

File = PRC5042.REJ

```
                                  Oxy.
  prs.      temp.      oxy(ws)   (ws-CTD)      DOC/DT       oxsat     sta.
================================================================================
$ TYPE PRC5042.REJ

 507.0      7.820      9.990     0.0000E+00  0.4286E-01    6.620    42.30
 17.00      29.78      100.0     0.0000E+00  0.4071        4.370    44.10
 2501.      1.840      99.99     0.0000E+00  0.5000E-01    7.670    50.80
 504.9      7.836      4.280     0.8850      -.1010E-02    6.690    46.40
 757.0      5.974      1.590     -.4376      -.6000E-04    6.930    42.40
 306.9      9.294      2.300     0.3476      0.1260E-02    6.420    45.30
 509.0      7.832      3.050     0.3023      -.2180E-02    6.680    47.40


File =  PRC5042.HIS

$ TYPE PRC5042.HIS

EDIT FACT=  2.80 HISTO BIN= 0.2500E-01 OCSLOPE= 0.1400E-02
```

```
 1 MIN/MAX STA,#   42.   51.  1 STD DEV.= 0.1927E+00 #OBS=  77  DOX=0.539
         1: BIAS   2:SLOPE      3:PCOR     4:TCOR      5: WT       6: LAG
OX PARMS=  -0.040  0.1808E-02  0.1442E-03 -0.4086E-01  0.1261E+00  0.3225E+01
 2 MIN/MAX STA,#   42.   51.  1 STD DEV.= 0.1115E+00 #OBS=  75  DOX=0.312
         1: BIAS   2:SLOPE      3:PCOR     4:TCOR      5: WT       6: LAG
OX PARMS=  -0.011  0.1645E-02  0.1442E-03 -0.3743E-01  0.3170E+00  0.4269E+01
 3 MIN/MAX STA,#   42.   51.  1 STD DEV.= 0.8916E-01 #OBS=  74  DOX=0.250
         1: BIAS   2:SLOPE      3:PCOR     4:TCOR      5: WT       6: LAG
OX PARMS=   0.001  0.1597E-02  0.1396E-03 -0.3665E-01  0.3834E+00  0.3170E+01
 4 MIN/MAX STA,#   42.   51.  1 STD DEV.= 0.8092E-01 #OBS=  74  DOX=0.227
         1: BIAS   2:SLOPE      3:PCOR     4:TCOR      5: WT       6: LAG
OX PARMS=   0.015  0.1514E-02  0.1395E-03 -0.3508E-01  0.4095E+00  0.3172E+01
```

```
CLASS INT:   0.025000
 -15   0
 -14   0
 -13   0
 -12   0
 -11   0
 -10   0
  -9   0
  -8   0
  -7   0
  -6   3***
  -5   2**
  -4   5*****
  -3   7*******
  -2   9*********
  -1   5*****
   0  12************
   1   5*****
   2  10**********
   3   6******
   4   3***
   5   2**
   6   3***
   7   1*
   8   1*
   9   0
  10   0
  11   0
  12   0
  13   0
  14   0
  15   0
  74    AVERAGE =     0.001249    ST. DEV. =     0.081464
```

æ

```
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C
C        Copyright (c) 1992
C        Woods Hole Oceanographic Institution
C        Woods Hole, Ma. 02543  USA
C
C
C        All rights reserved.
C
C        The authors and their employers make no warranties, expressed or
C        implied, as to the applicability or functionality of the software
C        distributed to you. The software is provided on an "as is" basis.
C        The user has determined to his/her satisfaction that the software
C        is suitable and operable in its present form for his/her purposes,
C        and the user assumes full responsibility for the implimentation
C        and functionality of the program on his/her computer system. This
C        material cannot be distributed or sold without prior written per-
C        mission of the author(s).
C
C * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
C Program OXFITMR
c
c  Fits the CTD oxygen values to the water sample oxygen data to deter-
c  mine up to six parameters of the Oxygen Algorithm  as described by
C  B. Owens & R. Millard (1985) "A New Algorithm for CTD Oxygen Calibra-
C  tion"  Journal of Physical Oceanography Vol. 15 pp. 621-631. For water
c  samples collected during the upcast, the CTD values should be extracted
c  from the downcast profile to avoid the problems associated with stop-
c  ping the instrument.  The program OXEXTRCT extracts the best down pro-
c  file oxygen values by searching for the pressure at which each sample
c  was taken.
c
C
      PARAMETER (NCA=10)
      PARAMETER (MAXDAT=1000)
c
      EXTERNAL foxy
c
      INTEGER*2 istar(60),nz(31)
      INTEGER   lista(NCA)
c
      REAL*4 ista, istmin, istmax
      REAL*4 data(MAXDAT,10), covar(NCA,NCA), alpha(NCA,NCA)
      REAL*4 p(MAXDAT), t(MAXDAT), oc(MAXDAT), os(MAXDAT), ot(MAXDAT)
      REAL*4 dpdt(MAXDAT), ox(MAXDAT), doxc(MAXDAT), stano(MAXDAT)
      REAL*4 oxcals(10)
      REAL*4 parm(10)
```

```
      REAL*4 parmi(10),  o2var
C
      CHARACTER*20 infile, root
      CHARACTER*20 hisfile, rejfile, goodfile
      CHARACTER*80 headers
      CHARACTER*1  reply

C************************
      EQUIVALENCE (data(1,1),p(1)),
     &   (data(1,2),t(1)),
     &   (data(1,3),oc(1)),
     &   (data(1,4),ot(1)),
     &   (data(1,5),ox(1)),
     &   (data(1,6),os(1)),
     &   (data(1,7),doxc(1)),
     &   (data(1,8),dpdt(1)),
     &   (data(1,9),stano(1))
c
C initial guess oygen parameters; bias,slope,pcor,tcor,wt, & tau ...
c
      DATA parmi /0.0,.0014,.00015,-.036,.75,8.0,0.0,0.0,0.0,0.0/
c
      DATA  lista /2, 4, 3, 5, 1, 6, 7, 8, 9, 10/
      DATA  fact /2.8/
      DATA  istcnt /0/
      DATA  oxfact /1.2/
      DATA  satmax /10./
      DATA  oxmin /0./
      DATA   doxed / -1./
      DATA  npar, mpar /6, 6/
      DATA  istmin, istmax /99999, -99999/
      DATA  kin, kout, kerr, klog /5, 6, 0, 33/
      DATA  lugood, lurej, luhis /17, 16, 60/
      DATA  luoxy /22/
      DATA  unit /0.025/
      DATA  alamda /-1./
      DATA  sig /.6/
      DATA  crit /.96/
      DATA  itmax /5/
      DATA  i1 /1/
      DATA  ipass /1/
c
c*****************************************************
      OPEN(UNIT=klog, FILE='oxfitmr.log',
     &   STATUS='unknown', ACCESS='append')
      write(klog,*) '***** OXFITMR *****'
c  initialize these parameters...
c
```

```
      ocslope = parmi(2)
      ma = nca
c
      WRITE(kout,*) 'OXFITMR  '
c
      WRITE(kout,1201)
 1201 FORMAT(/' Enter name of .OXY file: ')
      READ(kin,1202,ERR=904) infile
 1202 FORMAT(a20)
      if (INDEX(infile,'.').EQ.0) then
          infile = infile(1:INDEX(infile,' ')-1) // '.OXY'
      end if
c
      write(klog,1000) infile
 1000 format('     input file: ',a14)
c
      nchars = INDEX(infile, '.') - 1
      root = infile(1:nchars)
c
c  open all input/output files ...
c
      OPEN(UNIT=luoxy, FILE=infile, ERR=900, STATUS='UNKNOWN')
c
      rejfile = root(1:nchars)//'.REJ'
      OPEN(UNIT=lurej, FILE= rejfile,
     &      STATUS='unknown', ERR= 901)
c
      goodfile = root(1:nchars)//'.RES'
      OPEN(UNIT=lugood, FILE= goodfile,
     &      STATUS='unknown', ERR= 902)
c
      hisfile = root(1:nchars)//'.HIS'
      OPEN(UNIT=luhis, FILE= hisfile,
     &      STATUS='unknown', ERR= 903)
c
      write(klog,1204) goodfile, rejfile, hisfile
 1204 format(/' Output files: ',/3(6x,a20))
c
      write(kout,1206)  oxfact, oxmin
 1206 format(' Enter oxfact & a minimum acceptable oxygen value: [',
     &        2f6.2,']: ')
      READ(kin,*,ERR=906) oxfact, oxmin
      write(klog,7000) oxfact, oxmin
 7000 format('     oxfact: ',f6.2,/'      oxmin:',f6.2)
c
      write(kout,1207) fact, unit, ocslope
 1207 format(' Enter std dev factor to identify reasonable data [',
     &  f5.2,']'/'       bin size for histogram [',f7.5,']',
```

```
     & /'       and value for ocslope [',e10.4,']'/' : ')
c
      read(kin,*,ERR=907) fact,unit,ocslope
      write(klog,7001) fact, unit, ocslope
 7001 format('  edit fact: ',f5.2,/ '   bin size: ',f8.5,/
     &       '    ocslope: ',e10.4)
      write(luhis,1208) fact,unit,ocslope
 1208 format(' Edit Fact= ',f5.2,' Histo Bin= ',e10.4,' OcSlope= ',
     &  e10.4)
c
      PARMI(2)=OCSLOPE
      READ(luoxy,'(A80)',END=5,ERR=905) HEADERS
      write(kout, *) HEADERS
      READ(luoxy,'(A80)',END=5,ERR=905) HEADERS
      write(kout,*) HEADERS
      IF(INDEX(HEADERS,'=====') .LE. 0) THEN
         write(kerr,1008) infile
 1008    format(1x, a20,' does not have correct format')
         STOP
      end if
C
c  loop to read all water sample data ...
c
      j = 1
    1 continue
      READ(luoxy,*,END=5,ERR=905) (data(j,k),k=1,9)
c
C  scale oxygen to digitizer units ...
c
      oc(j) = oc(j) / ocslope
      doxc(j) = doxc(j) / ocslope
c
c   find min/max station # for fit
c
      if(istmin.gt.data(j,9)) istmin=data(j,9)
      if(istmax.lt.data(j,9)) istmax=data(j,9)
c
c   find oxygen values larger than oxfact*saturation list/delete
c
      doxsat = os(j) * oxfact
      if (os(j).gt.satmax.or.ox(j).gt.doxsat.or.ox(j).lt.oxmin) then

         write(lurej,1010) p(j),t(j),ox(j),data(j,10),doxc(j),os(j),
     &       data(j,9)
 1010    format(7(1x,g10.4))
         j=j-1
      end if
c
```

```
      if (ista.ne.data(j,9)) then
         istcnt = istcnt+1
         ista = data(j,9)
      end if
c
      j=j+1
      IF (j.LE.MAXDAT)  GO TO 1
c
    5 continue
c     now proceed to do non-linear fit...
c
      itot=j-1
      write(klog,1212) istmin, istmax, itot
 1212 format('  min/max stations: ',f4.0,',',f4.0,'  # of obs: ',i5)
c
      do 7 i=1,10
         parm(i)=parmi(i)
    7 continue
c
      write(kout, 1209) npar
 1209 format('  Number of parameters to fit? [',i2,']: ')
      READ(kin,*,ERR=908) npar
c
      write(kout,1012)
      write(kout,1011) (parmi(i),i=1,mpar)
      write(kout,1099) (lista(i),i=1,mpar)
 1011 FORMAT(' ',' OX Parms:  ',1F6.3,5(' ',E11.4))
 1012 FORMAT('           1: Bias   2:Slope      3:Pcor     4:Tcor'
     &,'      5: Wt       6: Lag')
 1099 format ('  order of evaluation: ',8i4)
c
      j = 99
  122 continue
      write(kout,*)'Enter new fit order: [2 3 5 6 1 4]: '
      READ(kin,*,ERR=909) (lista(i), i=1,mpar)
c
      WRITE(kout,1099) (lista(i),i=1,mpar)
      write(kout,*,ERR=910) ' Is this order OK? [Y]/N: '
      READ(kin,1112) reply
 1112 format(a)
c
      IF(reply.EQ.'N'.OR.reply.EQ.'n') GO TO 122
c
      write(klog,1099) (lista(i),i=1,mpar)
      if (npar.lt.mpar) then
         write(klog,1199) (lista(i),i=npar+1,mpar)
         write(kout,1199) (lista(i),i=npar+1,mpar)
 1199    format (' fixed parameters  =', 8i4)
```

```
      end if
c
      write(kout,1220)
 1220 format(' For each parameter: enter parameter #, value'/
     &       '    use 99/ to exit this loop...')
c
   12 continue
      j = 99
      write(kout,1221)
 1221 format('  param #, value? ')
      read(kin,*,ERR=911) j, val
      IF (j.LE.10.AND.j.GT.0) then
         parm(j) = val
         GO TO 12
      end if
c
      write(klog,*)'parameter values:'
      do 888 i=1,npar
        write(klog,*) i, parm(i)
  888 continue

   77 CONTINUE
      kpass = 0
      alamda = -1.
      do 21 iter=1,itmax
         CALL MRQMIN(data(1,1),ox(1),sig,itot,parm(i1),ma,lista,npar,
     *    covar,alpha,nca,chisq,foxy,alamda)
         kpass = kpass + 1
         chisc = itot * crit
         if (chisq.lt.chisc) GO TO 223
   21 continue
c
c   finish calculation with lamda = 0 ...
c
 223  continue
      alamda=0
      CALL MRQMIN(data(1,1),ox(1),sig,itot,parm,ma,lista,npar,
     *    covar,alpha,nca,chisq,foxy,alamda)
      ox22 = 0.0
      kk = 0
      o2var = 0.0
      do 19 j=1,itot
  19     o2var=data(j,10)**2+o2var
c
      std = SQRT(o2var/itot)
      doxed = std * fact
      o2var = 0.0
      do 20 i =1,itot
```

```
         IF (ABS(data(i,10)).gt.doxed) then
c          log rejected observations to look over later
           write(lurej,1010) p(i),t(i),ox(i),data(i,10),
     &       doxc(i)*ocslope, os(i),data(i,9)
         ELSE
           o2var = data(i,10)**2 + o2var
           kk = kk + 1
           do 1009 j=1,10
 1009         data(kk,j)=data(i,j)
         END IF
   20 continue
c
      if (itot.gt.kk) then
         itot = kk
         sig = SQRT(o2var/itot)
         write(klog,1006) ipass,istmin,istmax,kpass,sig,itot,doxed
         write(kout,1006) ipass,istmin,istmax,kpass,sig,itot,doxed
         write(luhis,1006) ipass,istmin,istmax,kpass,std,itot,doxed
         do 23 i= 1,mpar
           oxcals(i)=parm(i)
   23    continue
         WRITE(luhis,1012)
         WRITE(luhis,1011) (oxcals(i),i=1,mpar)
c  reset coefficients to reason values after each pass thru data
         ipass=ipass+1
         GO TO 77
      end if
c
c  write good data to file...
      write(lugood,1010) (p(i),t(i),ox(i),data(i,10),doxc(i)*ocslope,
     & os(i),data(i,9),i=1,itot)
      CLOSE(lugood)
c
c   write info to the histogram file ...
      do 24 i= 1,mpar
           oxcals(i)=parm(i)
   24 continue
      WRITE(luhis,1006) ipass,istmin,istmax,kpass,std,itot,doxed
      WRITE(luhis,1012)
      WRITE(luhis,1011) (oxcals(i),i=1,6)
 1006 FORMAT(1X,I3,' Min/Max Sta: ',F4.0,'- ',f4.0,I3,' ','StdDev: ',
     & E11.4,' #Obs: ',I4,'  dOx: ',F5.3)
c
c  ...and to the stdout device ...
      WRITE(kout,1006) ipass,istmin,istmax,kpass,std,itot,doxed
      write(kout,1011) (oxcals(i),i=1,mpar)
      WRITE(klog,1006) ipass,istmin,istmax,kpass,std,itot,doxed
      write(klog,7011) (oxcals(i),i=1,mpar)
```

```
 7011 format(' Computed Ox Parameters...'/
     &       '   ********************'/
     &       '      bias: ',f6.3,/
     &       '     slope: ',e11.4,/
     &       '      Pcor: ',e11.4,/
     &       '      Tcor: ',e11.4,/
     &       '        Wt: ',e11.4,/
     &       '       Lag: ',e11.4,/
     &       '   ********************')
      itot = kk
c
C    print histogram of residuals ...
c
      do 2502 k = 1,60
 2502    istar(k) = 2h**
c
      KDP = 0
      R2 = 0.0
      SUM = 0.0
      DO 251 K = 1,31
  251    NZ(K) = 0
      WRITE(luhis,2505)UNIT
 2505 FORMAT(11H CLASS INT:,F11.6)
c
      do 2550 m = 1,itot
         kdp = kdp+1
         zm = data(m,10)
         sum = sum + zm
         r2 = r2 + zm*zm
         if(unit.eq.0) unit=.1
         zm = zm/unit
         k = SIGN(abs(zm)+0.5,zm)
         IF ((IABS(K)-15).GT. 0) k = ISIGN(15,k)
         nz(k+16) = nz(k+16) + 1
 2550 continue
c
      if ( kdp .le. 1 ) then
        write (luhis,*) ' no obs in interval'
        STOP
      end if
      do 256 k = 1,31
         m = k - 16
         inz = nz(k)
  256    write(luhis,2560,ERR=257) m, nz(k), (istar(j),j=1,inz)
 2560    FORMAT(' ',2I4,60(A1))
c
  257 xn = kdp
      sum = sum/xn
```

```
      r2 = SQRT(ABS((r2-xn*sum*sum)/(xn-1.0)))
      WRITE(luhis,2570) kdp,sum,r2
 2570 FORMAT('   ',I4,'    AVERAGE = ',F12.6,'    STD DEV. = ',F12.6,/)
c
      STOP
c*******************************************************
c  error messages...
c
  900 write(kerr,1900) infile
 1900 format(/' Unable to open ',a20)
      STOP
c
  901 write(kerr,1900) rejfile
      STOP
c
  902 write(kerr,1900) goodfile
      STOP
c
  903 write(kerr,1900) hisfile
      STOP
c
  904 write(kerr,*)'Unable to read name of .OXY file.'
      STOP
c
  905 write(kerr,1905) infile
 1905 format(/' Error reading ',a20)
      STOP
c
  906 write(kerr,*)'Unable to parse oxfact, oxmin.'
      STOP 906
c
  907 write(kerr,*)'Unable to parse fact, unit, ocslope'
      STOP 907
c
  908 write(kerr,*)'Unable to parse number of parameters to fit'
      STOP 908
c
  909 write(kerr,*)'Unable to parse fitting order.'
      STOP 909
c
  910 write(kerr,*)'Unable to parse Y/N response.'
      STOP 910
c
  911 write(kerr,*)'Unable to parse parameter value.'
      STOP 911
c
      END
```

æ

```
      SUBROUTINE FOXY(DATA,PARS,OXF,DYPAR,NUM)
      DIMENSION PARS(10),DYPAR(10)
      DIMENSION DATA(1000,10)
C
C   R. Millard
C   May 1991
C Correct oxygen lag term ( multiple by slope)
C     Following the oxygen algorithm of:
C  B. Owens & R. Millard (1985) "A New Algorthm for CTD Oxygen
C  Calibration"  J. of Physical Oceanography Vol. 15 pp. 621--631.
C
       P=DATA(NUM,1)
       T=DATA(NUM,2)
       OXC=DATA(NUM,3)
       OXT=DATA(NUM,4)
       OXS=DATA(NUM,6)
       DOXC=DATA(NUM,7)
       B=PARS(1)
       S=PARS(2)
       PCOR=PARS(3)
       TCOR=PARS(4)
       WT=PARS(5)
       TAU=PARS(6)
C ***********************************************
C OXYGEN SENSOR ALGORITM.
C OXYGEN SATURATION: OXS
C      OX = BIAS + SLOPE * OXY. CURRENT
      OX = B+S*(OXC + TAU*DOXC)
      OXEA = (TCOR*(T+WT*(OXT-T))+PCOR*P)
      OXA = EXP(OXEA)
      OXB=OX*OXA
C           OXY*OXY. SAT
      OXF  = OXB*OXS
C RETURN DERIVATIVES OF FUNCTION
      DYPAR(1)=OXS*OXA
      DYPAR(2)=(OXC+TAU*DOXC)*DYPAR(1)
      DYPAR(6)=DOXC*DYPAR(1)*S
      DYPAR(4)=((T+WT*(OXT-T)))*OXF
      DYPAR(5)=TCOR*((OXT-T))*OXF
      DYPAR(3)=P*OXF
      RETURN
      END
```

æ

```
This file contains the differences between the orginal Numerical
recipies Fortran routines MRQMIN and MRQCOF and the modified versions
used by the oxygen fitting routine OXFITMR.

$DIFF

_File 1: ,mrqmin.for ' modified for OXFITMR

_File 2: mrqmin.old  ' orginal


************
File PODA:<CTDEV.OXFITMR>MRQMIN.FOR;10
    3         PARAMETER (MMAX=20,MDATA=1000)
    6         DIMENSION X(MDATA,10),Y(MDATA),A(MA),LISTA(MA),
    7        *  COVAR(NCA,NCA),ALPHA(NCA,NCA),ATRY(MMAX),BETA(MMAX),DA(MMAX)
    8         IF(ALAMDA.LT.0.)THEN
    9           IF(SIG.LE.0.0) SIG=1.
   10           KK=MFIT+1
******
File PODA:<CTDEV.OXFITMR>MRQMIN.OLD;1
    3         PARAMETER (MMAX=20)
    4         DIMENSION X(NDATA),Y(NDATA),SIG(NDATA),A(MA),LISTA(MA),
    5        *  COVAR(NCA,NCA),ALPHA(NCA,NCA),ATRY(MMAX),BETA(MMAX),DA(MMAX)
    6         IF(ALAMDA.LT.0.)THEN
    7           KK=MFIT+1
************
************
File PODA:<CTDEV.OXFITMR>MRQMIN.FOR;10
   25           CALL MRQCOF(X,Y,SIG,NDATA,A,MA,LISTA,MFIT,ALPHA,BETA,NCA,
   26        &CHISQ,FUNCS)
   27            JKL=JKL+1
   28           OCHISQ=CHISQ
******
File PODA:<CTDEV.OXFITMR>MRQMIN.OLD;1
   22           CALL MRQCOF(X,Y,SIG,NDATA,A,MA,LISTA,MFIT,ALPHA,BETA,NCA,CHISQ,F

   23        *UNCS)
   24           OCHISQ=CHISQ
************
************
File PODA:<CTDEV.OXFITMR>MRQMIN.FOR;10
   48         CALL MRQCOF(X,Y,SIG,NDATA,ATRY,MA,LISTA,MFIT,COVAR,DA,NCA,
   49        &CHISQ,FUNCS)
   50         IF(CHISQ.LT.OCHISQ)THEN
******
File PODA:<CTDEV.OXFITMR>MRQMIN.OLD;1
   44         CALL MRQCOF(X,Y,SIG,NDATA,ATRY,MA,LISTA,MFIT,COVAR,DA,NCA,CHISQ,FU

   45        *NCS)
   46         IF(CHISQ.LT.OCHISQ)THEN
************
```

```
Number of difference sections found: 3
Number of difference records found: 12


DIFFERENCES /IGNORE=()/MERGED=1-
    PODA:<CTDEV.OXFITMR>MRQMIN.FOR;10-
    PODA:<CTDEV.OXFITMR>MRQMIN.OLD;1

$ diff mrqmin.for mrqmcof.for (MODIFIED) mrqcof.old (ORGINAL)

************
File PODA:<CTDEV.OXFITMR>MRQCOF.FOR;22
    1         SUBROUTINE MRQCOF(DATA,Y,SIG,NDATA,A,MA,LISTA,MFIT,
    2        *ALPHA,BETA,NALP,CHISQ,FOXY)
    3         PARAMETER (MMAX=20,MDATA=1000)
    6         DIMENSION DATA(MDATA,10),Y(MDATA),ALPHA(NALP,NALP),BETA(MA),
    7        *    DYDA(MMAX),LISTA(MFIT),A(MA)
******


File PODA:<CTDEV.OXFITMR>MRQCOF.OLD;1
    1         SUBROUTINE MRQCOF(X,Y,SIG,NDATA,A,MA,LISTA,MFIT,ALPHA,BETA,NALP,
    2        *CHISQ,FUNCS)
    3         PARAMETER (MMAX=20)
    4         DIMENSION X(NDATA),Y(NDATA),SIG(NDATA),ALPHA(NALP,NALP),BETA(MA),
    5        *    DYDA(MMAX),LISTA(MFIT),A(MA)
************
************
File PODA:<CTDEV.OXFITMR>MRQCOF.FOR;22
   16           CALL  FOXY(DATA,A,YMOD,DYDA,I)
   17            SIG2I=1./(SIG*SIG)
   18           DY=Y(I)-YMOD
   19           DATA(I,10)=DY
   20           DATA(I,8)=YMOD
   21           DO 14 J=1,MFIT
******
File PODA:<CTDEV.OXFITMR>MRQCOF.OLD;1
   14           CALL FUNCS(X(I),A,YMOD,DYDA,MA)
   15           SIG2I=1./(SIG(I)*SIG(I))
   16           DY=Y(I)-YMOD
   17           DO 14 J=1,MFIT
************

Number of difference sections found: 2
Number of difference records found: 11

DIFFERENCES /IGNORE=()/MERGED=1-
    PODA:<CTDEV.OXFITMR>MRQCOF.FOR;22-
    PODA:<CTDEV.OXFITMR>MRQCOF.OLD;1
```

æ