

PREFACE

Today, the CTD (conductivity temperature and depth profiling system) observation, from which we can get vertical profile of water temperature and salinity in the ocean, is an important means of marine observation to get precise and fundamental oceanographic information not only for physical oceanography but marine biology and ocean chemistry. However, the sensor of the CTD system is so sensitive that the data obtained is variable according to the condition and the treatment of the sensor and after-processing and quality control of the data is difficult. But we do not have the established method yet and it's often said about fear against reliability of the data obtained.

Realizing these facts, for the purpose of establishing the method to obtain precise data efficiently, "CTD data examination committee" headed by Dr. Kawabe, Oceanographic Institution, Tokyo University, was organized and there they examine the method of obtaining and correcting of the CTD data and also check the effectiveness of that method at actual observation.

We expect disseminating their result widely for observer use will promote the accuracy and reliability of the CTD data obtained in this country and, further more, will contribute to the international global environment project, so we decided to publish it as a JODC Manual/Guide.

We hope this Manual would be used as a guideline for CTD data processing by each observer.

Osamu YAMADA
Director
Japan Oceanographic Data Center
March 1993

刊行にあたって

今日では、海洋における水温・塩分の鉛直分布を得ることのできるCTD（電気伝導度・水温・圧力計）観測は、海洋物理のみならず、海洋生物・化学にとっても海洋の基礎情報を高精度に得るために欠くべからざる基本的な海洋観測手段となっている。特に、各種の地球環境研究計画で要求される高精度のデータは、熟練したCTDシステムの運用により初めて可能である。しかしながら、CTDシステムのセンサーは極めて繊細であるため、得られるデータは時に個々のセンサー及び操作方法により変動する。このことがデータの後処理及び品質管理を困難にしており、確立された方法を求める声を随所で耳にするところである。

このため、東京大学海洋研究所の川辺博士を座長とした「CTDデータ校正検討会」が編成され、CTDデータの取得・校正方法について検討が重ねられ、併せて、実際の観測で本手法の有効性が検証されている。

この成果を広く配布し、観測者の使用に供することにより、我が国で得られるCTDデータの精度・信頼度が高まり、延いては、国際的な地球環境計画への貢献が期待されることから、このたびJODCマニュアル・ガイドとして刊行することとした。

このマニュアルが、我が国のCTDデータ取得・処理の標準として用いられ、我が国の地球環境研究への貢献が進展するなら、望外の喜びである。

1993年3月

日本海洋データセンター

山 田 修

著者の序

近年、気候変動をはじめとする地球環境問題への社会的関心が高まるにつれ、海洋の役割とそれを研究することの重要性が、ますます強く認識されるようになった。こうした関心に応えるためにも、海洋観測の質の向上、特に最も基礎的で重要なデータである水温と塩分の測定技術の向上が求められている。水温と塩分の情報は、水塊の比較や海水混合の見積り、地衡流速の計算、生物環境の評価をする上で極めて重要である。測定には、1世紀もの間転倒温度計とナンセン採水器が使われ、規準深度での測定が行われてきたが、CTD (Conductivity Temperature Depth profiler)の出現により、鉛直方向に連続なデータが取られるようになった。

CTDはこの10年間に我が国でも広く使われるようになり、現在では官庁・大学のほぼ全ての観測船で使われている。データの較正は各機関が独自に行っており、基本的な考え方は同じでも、具体的な方法は異なっている。しかしこれは、各機関がそれぞれの実情にあった最善の方法をとっている、ということでは必ずしもない。採用している方法が精度の点で最も良い方法かどうかについては、誰もが不安を抱いており、また忙しい業務の中で、できるだけ手間のかからない較正方法を求めていることも事実である。精度の良いCTDデータを効率よく得る方法が、求められているのである。そうした事情を踏まえ、ポイントとなる問題点と現時点で最も良いと思われるCTDデータの取得・較正方法についてまとめておこう、というのが、本冊子のねらいである。

基本構想の作成に当たり、気象庁・海上保安庁水路部・水産庁中央水産研究所・伯東株式会社・東京大学海洋研究所の有志による5回の会合を開き、そこでの議論を参考にして一応の較正手順を決め、それを基に作成した較正用ソフトを白鳳丸KH-91-5次航海(WOCE Hydrographic Programmeの一環として実施した東経165度線でのCTD観測)で使用し、修正に修正を重ねてできた最終結果が、ここで紹介するものである。

本冊子は、より良い方法を求めて、時とともに改訂すべきものである。追加・改訂に関する意見や情報、あるいは感想をお寄せいただければ幸いである。

会合に参加し貴重な情報を提供していただいた上記機関の方々と、計算機プログラムの作成にご協力いただいた青木繁明、上原克人両氏に感謝いたします。また、本冊子の印刷を快く引き受けてくださった日本海洋データセンターと、印刷用原稿の作成にご尽力いただいたデータセンターの谷伸、馬場典夫両氏に心からの謝意を表します。

1992年12月

東京大学海洋研究所

川 辺 正 樹

水産庁遠洋水産研究所

川 崎 清

目 次

	ページ
1. 緒言 - CTD観測の概要	1
2. CTDデータの処理手順	2
3. 水温・圧力センサーの陸上較正	4
4. CTD連続データの取得	9
5. CTD連続データの処理	
5-1. ノイズの除去	10
5-2. 3つのセンサーで異なる応答速度の補正	10
6. 採水データの処理	16
7. 電気伝導度(塩分)の較正	
7-1. 採水の値に合わすべきは、電気伝導度か塩分か?	17
7-2. 電気伝導度データの較正方法	19
8. おわりに	22
参考文献	23
附録 (CTDデータ較正のための計算機プログラム)	
1. 圧力・電気伝導度データにかけるデジタル・フィルターの係数FPを 求めるプログラム	附録 1-1
2. CTDの連続データを較正し、1db間隔のデータセットにするプログラム	附録 2-1
3. 採水時のCTDデータおよび採水による塩分・溶存酸素値のデータセット を作成するプログラム	附録 3-1
4. 電気伝導度センサーの補正式を導くプログラム	附録 4-1

1. 緒言 — CTD観測の概要

CTDは、圧力・水温・電気伝導度の3つ、あるいは酸素を加えた4つのセンサーにより、これらの値を鉛直方向に連続的に測定する測器である。水温・塩分・(酸素)をはじめ、密度や地衡流速の鉛直プロファイルを明らかにし、これらの鉛直微分や等値面の深さの正確な評価を可能にする。ナンセン採水器による離散的なデータに比べてはるかに多くの正確な情報を与えてくれるので、力学研究や数値モデルでの計算を含め、これまでのデータよりもはるかに広い範囲で利用されるであろう。

しかし、そのためには十分なデータの質を確保しなければならない。例えば、北太平洋深層の水温・塩分を議論するには、少なくとも 0.05°C 、 0.005psu の精度が必要である。 0.001psu の塩分誤差は、それぞれ、圧力 2dbar 、水温 0.001°C 、又は電気伝導度 0.001mS/cm の誤差で生じるので、圧力と水温については、それ自体の議論のためよりは、むしろ塩分の精度を上げるために高い精度が求められる。ちなみに国際プロジェクトWOCE(World Ocean Circulation Experiment; 世界海洋循環実験)では、圧力 3dbar 、水温 0.002°C 、塩分 0.002psu の精度が要求されている。データに求められる精度は何を研究するかによって異なるので、ある定められた値以上の精度がなければ価値がないということにはならない。しかし、どのような研究にも耐えられるデータを取ろうと思えば、この要求精度を目標とすべきであり、そのためにはCTDデータの取得・較正に対し細心の注意を払わなければならない。CTDは優れた測器に違いないが、センサーを海中に降ろすだけで精度の高いデータが得られるわけではない。きちんとしたデータの取得と較正を必要とする。

CTDの較正は、観測航海の前に行なう陸上でのセンサー較正と観測時に採水して得たデータにCTDの値を合わせるデータ較正の2つの方法で行う。圧力と水温のセンサーには前者の方法をとり、センサーの安定性の悪い電気伝導度については後者の方法をとる。CTD観測では、通常、センサー降下時に圧力・水温・電気伝導度の連続データをとり、センサーを上げる際に適当な深さで止めて採水する。採水と同時にその深さでの圧力・水温・電気伝導度のセンサー・データを計算機に取り込む。採水した水は、適当な時間をおいた後船上で化学分析し、その塩分値を電気伝導度センサーの較正に用いる。

2. CTDデータの処理手順

現時点で最も良いと思われる、CTDおよび採水データの処理手順を表1に示す。表1では、圧力と水温の較正式が、陸上でのセンサー較正によって既に得られているものとしている。また機種としては、Neil Brown Mark 3を念頭に置いている。表1Aは、CTD降下時に取得する連続データの補正とデータセットの編集手順を示し、表1Bは採水時のCTDデータ(P, T, C_{CTD})と採水データ(S_{sample} , C_{sample})から、 C_{CTD} の補正式を求める手順を示している。

圧力と水温の較正済みデータは、A1からA3の処理で得られる。A4以降は電気伝導度データを較正するための

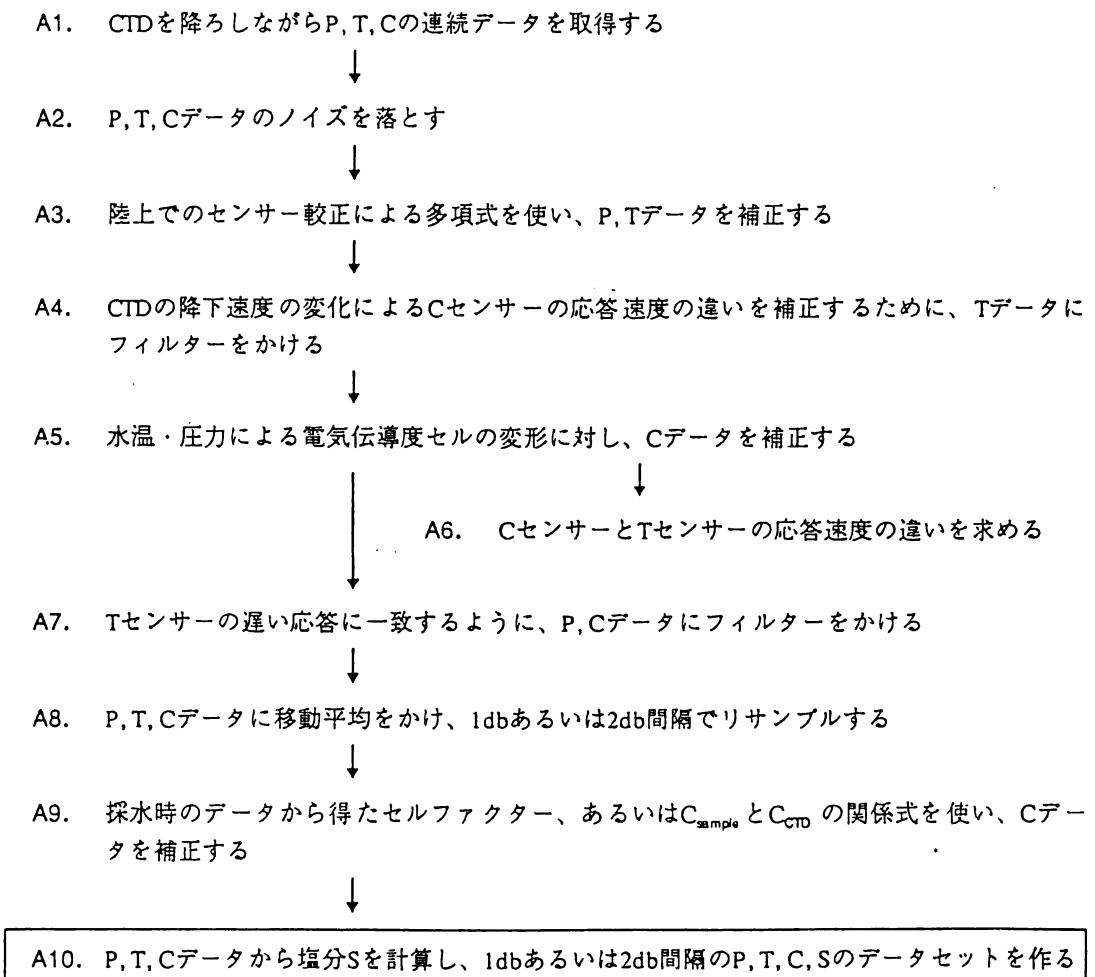


表1A. CTD降下時に取得する圧力(P), 水温(T)及び電気伝導度(C)のデータに補正等の処理を施し、圧力・水温・塩分(S)のデータセットを作成するまでの作業の流れ

作業である。A8では移動平均をかけて細かな変化を落とし、1dbあるいは2db間隔でリサンプルする。この段階でデータ数は大幅に減る(WOCEの資料では2db間隔に入っているデータを平均して、2db毎のデータセットにしている)。A9では、表1Bの一連の作業を通して得られたセルフファクター、あるいは C_{CTD} の補正式を用いて、 C_{CTD} の補正をする。こうして得られたP, T, Cの値から塩分を計算し、1dbないし2db間隔での圧力・水温・塩分のデータセットを作成する(WOCEでは2db間隔を採用している)。

表1を結論するまでに行った検討や表1の主要なプロセスについて、順を追って以下に説明する。

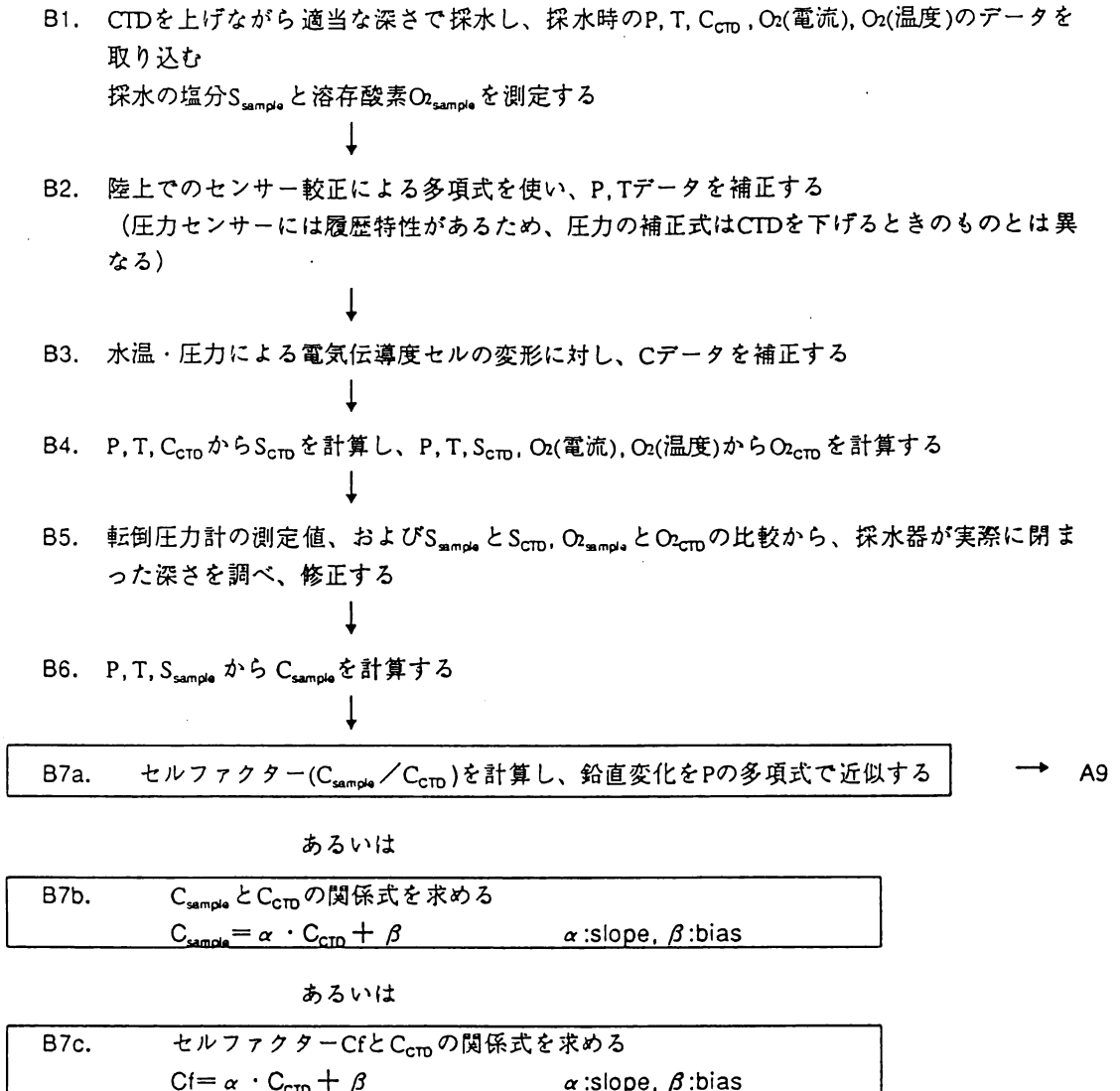


表1B. 採水時にとるCTDデータ(P, T, C_{CTD})と採水から測定した塩分 S_{sample} を使い、 C_{CTD} の補正式を導く作業の流れ

3. 水温・圧力センサーの陸上較正 (プロセスA3・B2で使用する補正式の導出)

水温・圧力センサーの誤差は、個々のセンサーで全く異なるため、1つ1つについて観測航海前に誤差の測定をしておく必要がある。航海後にも較正を行なってチェックする方がよいのは当然であるが、余程のことがない限り、一度得た較正曲線は数か月(少なくとも3か月程度)は有効である。ただし、一時的にしろセンサーを取り外した場合には較正が必要になる。例えば、水中局のセンサー部分にあるOリングを交換したならば、直ちに水温センサーの較正を行なわなければならない。

水温センサーについては、実験室に水の三重点(0.01℃)を作って標準白金抵抗温度計(Standard Platinum Resistance Thermometer, SPRT)を較正し、CTDの水温センサーとSPRTの示す値の差を水温の多項式で近似して補正する。白鳳丸KH-91-5次航海で使った水温センサーの誤差と水温の2次式による誤差曲線を図1に示す。誤差曲線はほとんど直線で、水温が低いほど誤差が大きく、0℃で0.025℃に達した。実際の海の中層・深層に対応する温度帯で誤差が大きく、較正が極めて重要であることを示している。

図1には全データから求めた2次曲線を示してあるが、実際には9.1℃を境に高温側と低温側それぞれで補正式を求めて使った。使用した補正值 ΔT の式は、

$$\Delta T = 0.1219 \times 10^{-4} \times T^2 - 0.8031 \times 10^{-3} \times T + 0.02519 \quad (T < 9.1^\circ\text{C})$$

$$\Delta T = 0.1796 \times 10^{-4} \times T^2 - 0.1276 \times 10^{-2} \times T + 0.02901 \quad (T > 9.1^\circ\text{C})$$

である。

圧力センサーについては、デッドウェイトテスターで測定した誤差曲線を、加圧時と減圧時のそれぞれに対して圧力の多項式で近似して補正する。誤差はセンサーにかかる圧力の履歴によって異なるので、圧力が0から次第に増えていく加圧時(CTDの降下時に対応)はよいが、減圧時(CTDの上昇時)の誤差曲線はセンサーにかけた最大圧力(CTDを下ろした最大深度)によって変わる。現実の観測では測点によってCTDを降ろす深さが違うので、採水時の圧力データに対する補正式は測点ごとに異なる。

この問題への対処方法はセンサーごとにより異なる。図2に示した例では、6000dbまで圧力をかけてから減圧しても3000dbまではほとんど加圧時の誤差をなぞっている。従って、3000db以深までセンサーを降ろす場合については、上昇時の圧力補正として6000dbまで加圧したときの減圧時補正式を使えば良い。このセンサーについては2000db以浅でセンサーを上げてしまう場合のために、2000dbまで加圧したときの減圧時補正式と1000dbまで加圧したときの式を用意しておけば十分であろう。しかし、SCOR Working Group(1988)に示されている例のように、たちの悪い振る舞いをするセンサーもある(図3)。特にCTD7では加圧時と減圧時で誤差が全く異なっている。このようなセンサーについては、最大圧力をいろいろと変えて幾つもの減圧時補正式を用意する必要がある。

白鳳丸KH-91-5次航海で使用した圧力センサーの誤差特性は、これらのいずれとも異なる(図4)。圧力0での補正

値は、-2.7db であり(測定時CTDは空中で3db前後の値を示していたが、この補正をすることで海面での圧力が0になる)、圧力をかけていくと、補正値は表層で急激に変化し、550db付近で負から正になる。それ以深では正の値をとり、変化が小さくなる。2000db~4000dbでは1.7db ほどではほぼ一定である。使用したセンサーは、表層では実際よりも大きな値を示し、中層・深層では小さな値を示すわけである。

減圧していき再び圧力が0になると、補正値も約-2.7db に戻る。しかし、それに至る補正値の変化は最大圧力がいくらかによって異なり、一定の曲線をなぞるというわけではない。そこで、最大圧力が2000db, 3000db, 4000db, 6000dbの場合の減圧時補正式を圧力の6次式で表し、任意の最大圧力についてはこれらの式を使って内挿することで補正値を求めた(図5)。使用した式は、

$$\Delta P = a_0 + a_1 \cdot P + a_2 \cdot P^2 + a_3 \cdot P^3 + a_4 \cdot P^4 + a_5 \cdot P^5 + a_6 \cdot P^6$$

であり、P(db)はCTDの圧力値、 ΔP (db)はPに加えるべき補正値である。増圧時補正式の係数は、

$$\begin{aligned} a_0 &= -2.697, & a_1 &= 0.6996 \times 10^{-2}, & a_2 &= -0.4554 \times 10^{-5}, & a_3 &= 0.1541 \times 10^{-8}, \\ a_4 &= -0.2892 \times 10^{-12}, & a_5 &= 0.2876 \times 10^{-16}, & a_6 &= -0.1165 \times 10^{-20} \end{aligned}$$

減圧時補正式の係数は、最大圧力が2000dbの場合、

$$\begin{aligned} a_0 &= -2.600, & a_1 &= 0.2662 \times 10^{-2}, & a_2 &= 0.2230 \times 10^{-5}, & a_3 &= -0.5678 \times 10^{-8}, \\ a_4 &= 0.3960 \times 10^{-11}, & a_5 &= -0.1043 \times 10^{-14}, & a_6 &= 0.8498 \times 10^{-19} \end{aligned}$$

最大圧力が3000dbの場合、

$$\begin{aligned} a_0 &= -2.652, & a_1 &= 0.3991 \times 10^{-2}, & a_2 &= -0.3852 \times 10^{-5}, & a_3 &= 0.2494 \times 10^{-8}, \\ a_4 &= -0.1068 \times 10^{-11}, & a_5 &= 0.2835 \times 10^{-15}, & a_6 &= -0.3119 \times 10^{-19} \end{aligned}$$

最大圧力が4000dbの場合、

$$\begin{aligned} a_0 &= -2.751, & a_1 &= 0.4673 \times 10^{-2}, & a_2 &= -0.5400 \times 10^{-5}, & a_3 &= 0.3635 \times 10^{-8}, \\ a_4 &= -0.1373 \times 10^{-11}, & a_5 &= 0.2696 \times 10^{-15}, & a_6 &= -0.2071 \times 10^{-19} \end{aligned}$$

最大圧力が6000dbの場合、

$$\begin{aligned} a_0 &= -2.496, & a_1 &= 0.3283 \times 10^{-2}, & a_2 &= -0.1717 \times 10^{-5}, & a_3 &= 0.4343 \times 10^{-8}, \\ a_4 &= -0.5423 \times 10^{-13}, & a_5 &= 0.3947 \times 10^{-17}, & a_6 &= -0.1492 \times 10^{-21} \end{aligned}$$

であった。

減圧時較正曲線の最大圧力による違いについては、どの程度の差まで気にするかによって対処が異なる。例えば図4の場合、補正值について0.5dbの違いを気にしないならば、最大圧力から1000dbほどの部分を除くと、最大圧力によらずほぼ共通の変化をしているといえる。そうした見方に立てば、任意の最大圧力に対する減圧時較正曲線は、基本的には最大圧力6000dbの較正曲線に一致し、各測点での最大深度から1000dbほどの距離の間を適当に補間すればよいということになる。ウッズホール海洋研究所ではこうした方法をとっている。この場合、6000dbまでの実験結果のみを使うことになる。

いずれにしても、センサーによって特性が全く異なるので、とにかくまず6000dbまでの加圧および減圧実験を行ない、加圧時と減圧時の誤差のずれを調べ、さらに適当な圧力までの加圧・減圧実験を行って、最大圧力による減圧時誤差の違いをみる必要がある。そうすれば、減圧時の補正式についてどの程度神経を使うべきかが、おおよそ判断できる。

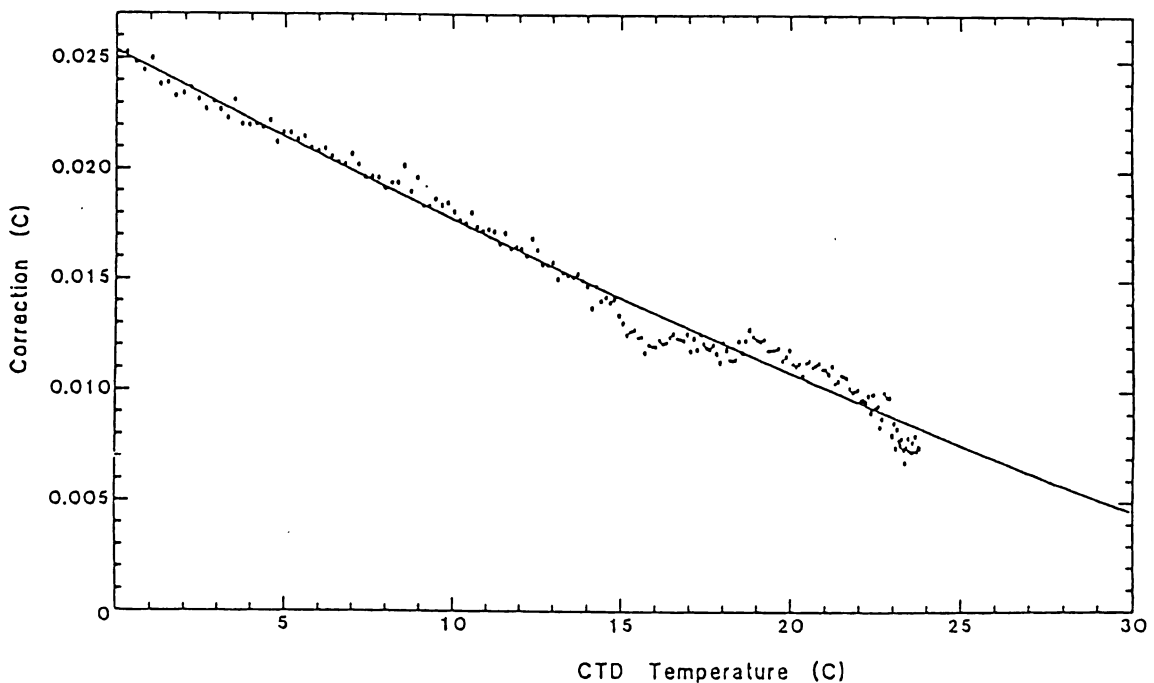


図1. 白鳳丸KH-91-5次航海で使用したCTD水温センサーの補正值。実線は、水温の2次式で合わせた較正曲線。

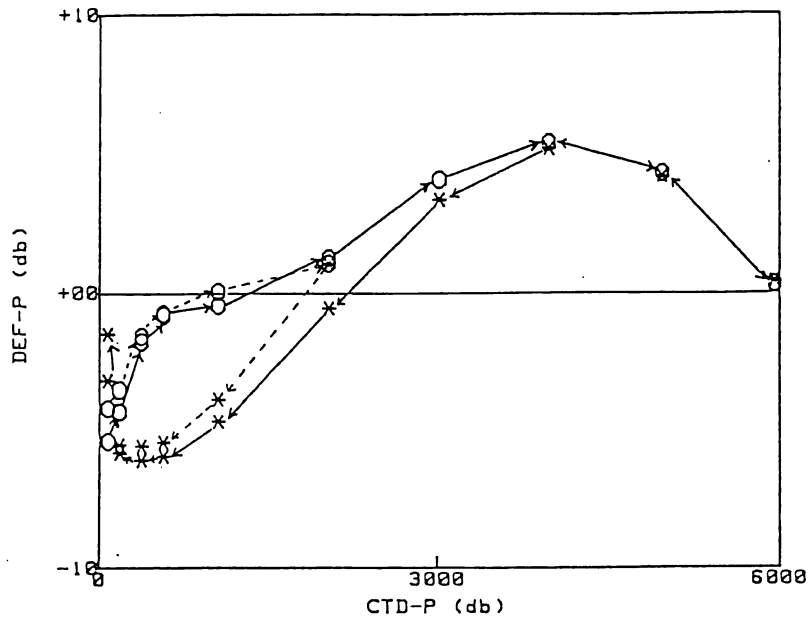


図2. CTD圧力センサーの較正実験の例。横軸はCTDで測定した圧力、縦軸はデッドウェイトテスターの値とCTDの値の差。実線の矢印は6000 dbまで加圧した場合の偏差の変化を示し、点線は2000 dbまでの場合を示す。
(株)白東、中野健一氏（現在(株)沿海生物研究所）提供。

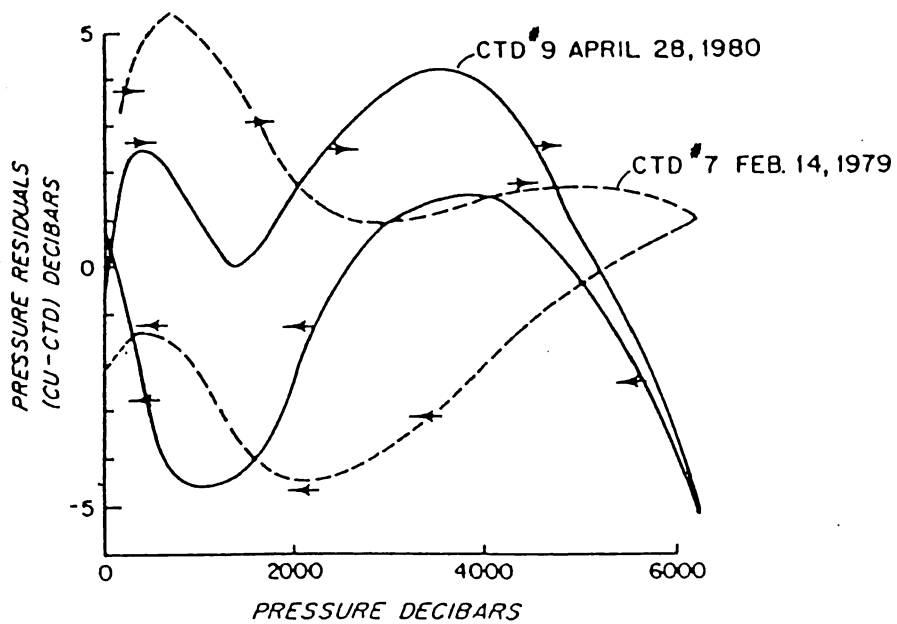


図3. CTD圧力センサーの較正曲線の例。座標軸は図2と同様。右向き矢印は加圧を、左向き矢印は減圧を示す。(SCOR Working Group, 1988)

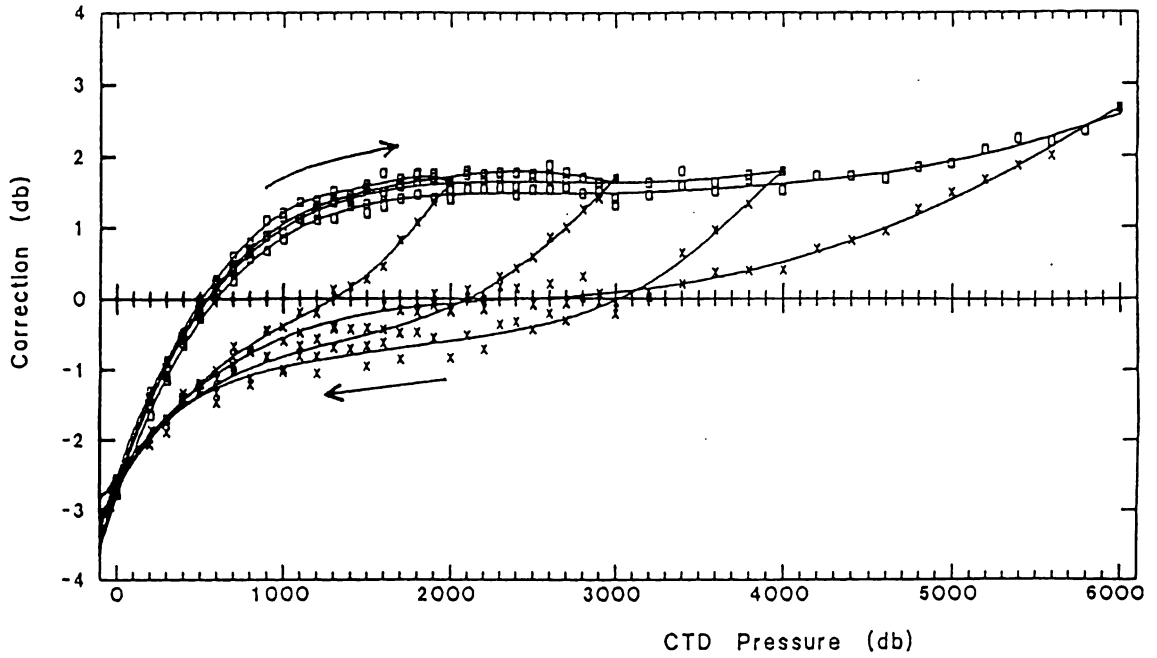


図4. 白鳳丸KH-91-5次航海で使用したCTD圧力センサーの補正値。
 実線は、実験値をCTD圧力値の6次式で合わせた較正曲線。

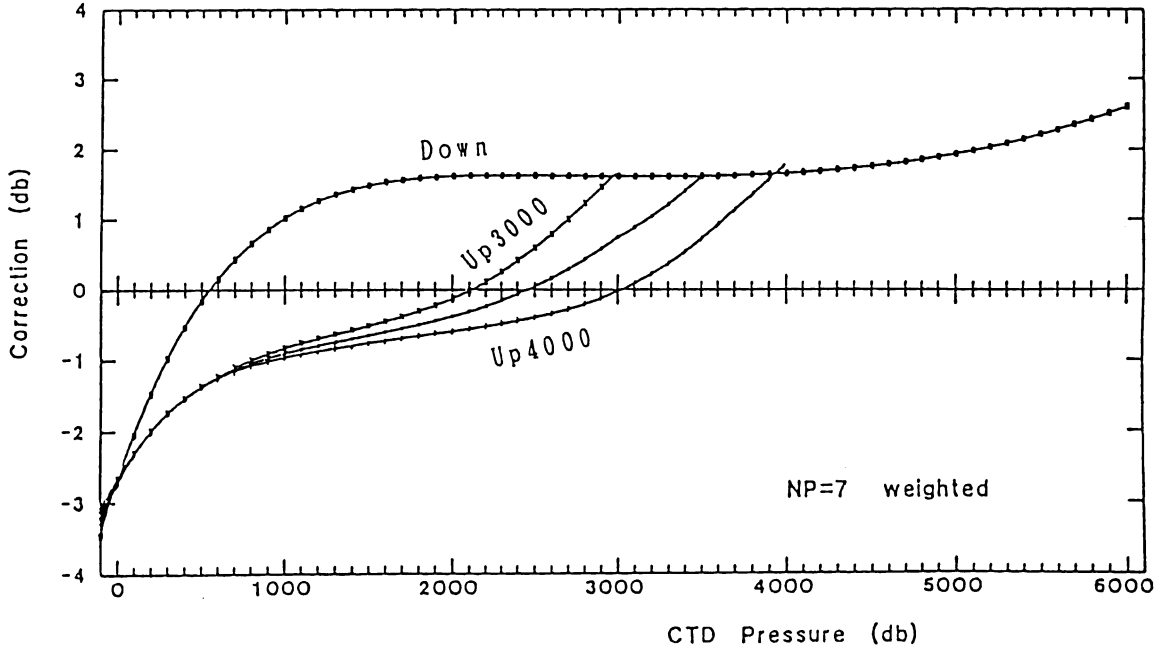


図5. 採水時のCTD圧力データの較正曲線を内挿により求める例。3500 dbまでセンサーを下ろした場合の較正曲線を、加圧時較正曲線(Down)と最大圧力3000 dbおよび4000 dbの減圧時較正曲線(Up3000とUp4000)から求めた。

4. CTD連続データの取得 (プロセスA1)

CTDを降ろしながらセンサーの連続データを取るときは、トラブルがない限り、水中局から送られてくる圧力・水温・電気伝導度のデータを取り込むだけである。Neil Brown Mark 3 CTD の場合、データの取り込みは次のように行われる。センサーから送られてくる信号は水中局でデジタル化され、FSK(Frequency Shift Keyer) 信号に直されてから、ケーブルおよびウィンチのスリッピングを経由してCTD船上局に送られる。FSK信号はCTD船上局でデジタル信号に復調され、GPIBというインターフェイスを通して、計算機に入力される。

データをどのように取り込むかは、後のデータ処理に関連して重要なポイントである。データは等間隔の時系列として処理されるので、水中局から船上局に送られてくる全てのデータ(Neil Brown Mark 3 では毎秒32個)を取得するのが理想である。しかし、船上局と計算機のインターフェイスとしてGPIBを用いているため、船上局は計算機からのデータ転送命令を受けて、初めてその時点での最新データを計算機に送る。そのため、水中局のデータ転送間隔(約32msec)と同じかそれよりも短い間隔で、計算機が命令を送らないと水中局から送られてきたデータを取り損ねることになる。速い計算機と言語を使って十分短い間隔でデータ転送命令を送り、重複したデータを削除することですべてのデータを漏れなく取り込むことができる。

水中局からはデジタル化したデータが送られてくるので、水中局・船上局が正常に作動している限り、データの抜けが生じることは考えにくい。しかし実際には、スリッピングでのノイズ(ケーブルを伝わってきた信号がうまく船上局に吸い取られずにケーブルの端で一部反射することもあるようである)等により、デジタルデータのビット落ちやバイトシフトの生じることがあり、これがひどくなると正常な一組のデータと認識されず、抜け落ちてしまうことがある。データが抜けたかどうかは、前のデータ組との時間差から推定するしか方法がなく、そうしたチェックをしようと思えばデータの取り込みと一緒に行わざるを得ない。しかし、現在の計算機の能力ではデータの抜けをチェックしながら全データを取り込むことは困難であり、どちらかを選択することになる。データの抜けは減多に起きることではないので、まずは全データの取り込みを優先すべきであろう。そして、水中局とケーブルの結合部やスリッピング周辺の整備につとめ、ノイズが発生しないように注意することが肝要である。

5. CTD連続データの処理

5-1. ノイズの除去 (プロセスA2)

CTDの降下時に測定する連続データについては、初めにノイズを除去しなければならない。ノイズには、電気伝導度セルの中を浮遊物が通過することによるものとケーブルの傷とかスリッピングの接触不良等によるもの(伝送ノイズ)がある。前者は電気伝導度値のみに含まれるが、後者は圧力・水温・電気伝導度全ての値に含まれる。伝送時に生じるノイズには、特定のビットが反転している場合とバイト単位でシフトが生じている場合がある。ビットの反転では、高位ビットの場合には前後の値と比較して修正が可能であるが、低位ビットでは検出すら不可能な場合もある。バイト単位でのシフトの場合はC, T, Pの高位バイト、フラグバイトを前後のデータと比較することにより修正のできる場合がある。

修正不能の伝送ノイズや浮遊物によるノイズは以下の方法で判定し、より良い値に置き換えるのがよい。浮遊物が伝導度セルを通過し伝導度が低下して生じるノイズは、プランクトンや懸濁粒子の多い表面近くで起きることが多く、粒子の特に多い海域を除くと、1~3個続けて生じる場合が多い。そこで、2個前と2個後のデータと比較することにする。すなわち、i番目のデータ C_i を(i-2)番目と(i+2)番目のデータと比べ、

$$C_i > C_{TH} + \max(C_{i-2}, C_{i+2})$$

あるいは、

$$C_i < -C_{TH} + \min(C_{i-2}, C_{i+2})$$

のときに C_i をノイズとみなし、 $(C_{i-2} + C_{i+2}) \times 0.5$ で置き換える。ここで、 C_{TH} は閾値であり、例えば $C_{TH} = 0.02 \text{ mS/cm}$ とすればよい。

5-2. 3つのセンサーで異なる応答速度の補正 (プロセスA4・A6・A7)

塩分を算出するには、センサー間の応答速度の違いを補正しないとイケない。圧力・電気伝導度センサーに比べ、水温センサーの応答は遅い。水温値を求める上では無視できる程度の遅れであるが、3つのセンサーの値から塩分を求める場合には無視できない。海水の電気伝導度はほとんど水温で決まり塩分の寄与はわずかであるために、水温センサーの応答の遅れを十分補正しない限り、得られる塩分は真の値からずれ、鉛直変化の大きな部分では非現実的なスパイク構造を生じてしまう。

塩分は圧力にはそれほど敏感でないため、実質的には水温と電気伝導度の応答の差が問題となる。特に水温センサーの応答の遅さが問題となるが、電気伝導度センサーについても、その応答がCTD水中局の降下速度によっ

て変わるという問題がある。電気伝導度の測定はセラミック製の一辺5mm程度の小さな四角い筒を用いているため、特に低速で降ろしたときには筒中の水の通りがよくなく、わずかではあるが応答時間が変化するのである。応答時間とCTD水中局の降下速度の関係は、Giles and McDougall (1986)に示されているので、デジタル・フィルターを使って補正することができる。あとは水温センサーの応答の遅さを補正すればよい。

その補正は、あらかじめ水温センサーの応答速度がわかっているならば容易である。それには次の2つの方法がある。1つは、白金抵抗体の水温センサー以外にサーミスタを装備し、サーミスタの応答の微分を白金抵抗体の応答に加えることで、水温の応答を電気伝導度と同程度に速くする方法である。サーミスタは、精度と安定性では白金抵抗体に劣るものの応答は速く、それによる測定水温をFast Response Temperature(あるいは単に "Fast T")と呼んでいる。もう1つの方法は、電気伝導度の時系列データにデジタルフィルターをかけて、実質的に応答速度を水温なみに遅らせるものである。電気伝導度データを遅らせるのではなく水温データを速めることも可能であるが、応答を速めることはデータの高周波成分を高めることであり、すなわちノイズ成分をも高めてしまっ、かえってまずい結果を招く。

サーミスタによる方法は簡便で良い。しかし、実際には白金抵抗体の時定数は個々の製品によって大きくばらついているので、時定数の公称値に合わせたサーミスタではうまく補正できないのが普通である。白金抵抗温度計とサーミスタの時定数の組合せがたまたまうまくいった場合はいいが、一般的には満足のいく解決にならない。

そのため、デジタル・フィルターを使うことになる。センサーの値Tが真値 T^* との差に比例して変化するとすると、センサーの応答は、

$$dT/dt = (T^* - T) / \tau \tag{5.1}$$

と書ける。ここで、 τ はセンサーの応答時間である。真値をステップ関数($t < t_0$ で0, $t \geq t_0$ で $T^*(t_0)$ [=const])とすると、 $t \geq t_0$ でのTについて

$$T(t, t_0) = T^*(t_0) \cdot [1 - \exp \{-(t-t_0) / \tau\}] \tag{5.2}$$

という解を得る。(5.2)式を(5.1)式に代入すると、

$$dT/dt = \tau^{-1} \cdot T^*(t_0) \cdot \exp \{-(t-t_0) / \tau\} \tag{5.3}$$

となる。一般に真値の時系列は、いろいろな t_0 でのステップ関数の和で表せるので、 $T^*(t_0)$ を時刻 t_0 での真値の変化分とし、時刻 t までの真値に対する応答を積分する。すなわち、(5.3)式に $\int_{-\infty}^t dt_0$ を施し、 t_0 から $\xi = t_0 - t$ に変数変換すると、

$$T(t) = \tau^{-1} \cdot \int_{-\infty}^0 T^*(t+\xi) \cdot \exp(\xi / \tau) d\xi \tag{5.4}$$

となる。これを変形すると、

$$T(t) = \tau^{-1} \cdot \int_{-\infty}^{-\Delta} T^*(t+\xi) \cdot \exp(\xi/\tau) d\xi + \tau^{-1} \cdot \int_{-\Delta}^0 T^*(t+\xi) \cdot \exp(\xi/\tau) d\xi \quad (5.5)$$

を得る。Δはデータのサンプリング間隔である。(5.5)式の右辺は、

$$\begin{aligned} \text{第1項} &= \exp(-\Delta/\tau) \cdot \tau^{-1} \cdot \int_{-\infty}^0 T^*(t-\Delta+\eta) \cdot \exp(\eta/\tau) d\eta \\ &\quad (\text{ここで、} \eta = \xi + \Delta) \\ &= \exp(-\Delta/\tau) \cdot T(t-\Delta), \end{aligned}$$

$$\begin{aligned} \text{第2項} &\doteq \tau^{-1} \cdot T^*(t-\Delta/2) \cdot \int_{-\Delta}^0 \exp(\xi/\tau) d\xi \\ &= \{1 - \exp(-\Delta/\tau)\} \cdot T^*(t-\Delta/2) \end{aligned}$$

と書き直せるので、結局

$$T(t) = \exp(-\Delta/\tau) \cdot T(t-\Delta) + \{1 - \exp(-\Delta/\tau)\} \cdot T^*(t-\Delta/2) \quad (5.6)$$

となる。

電気伝導度センサーの応答時間を τ_c とすると、センサーの値 C は(5.6)式と同様に、

$$C(t) = \exp(-\Delta/\tau_c) \cdot C(t-\Delta) + \{1 - \exp(-\Delta/\tau_c)\} \cdot C^*(t-\Delta/2) \quad (5.7)$$

となる。今求めたいのは、水温センサーと同じ応答時間 τ_T の時系列 C' であり、それについては、

$$C'(t) = \exp(-\Delta/\tau_T) \cdot C'(t-\Delta) + \{1 - \exp(-\Delta/\tau_T)\} \cdot C^*(t-\Delta/2) \quad (5.8)$$

が成り立つ。(5.7)式と(5.8)式から C^* を消去することで、

$$\begin{aligned} C'(t) &= \exp(-\Delta/\tau_T) \cdot C'(t-\Delta) \\ &\quad + \{1 - \exp(-\Delta/\tau_T)\} / \{1 - \exp(-\Delta/\tau_c)\} \cdot \{C(t) - \exp(-\Delta/\tau_c) \cdot C(t-\Delta)\} \end{aligned} \quad (5.9)$$

を得る。電気伝導度センサーの応答は速いので、簡単のために、 τ_c を0とすると、(5.9)式は

$$C'(t) = \exp(-\Delta/\tau_T) \cdot C'(t-\Delta) + \{1 - \exp(-\Delta/\tau_T)\} \cdot C(t) \quad (5.10)$$

となる。こうして、 τ_T 、あるいは $\exp(-\Delta/\tau_T)$ [=FP] さえわかれば、(5.10)式を使って、観測で得られる電気伝導度データ $C(t)$ から応答時間 τ_T のデータ $C'(t)$ を計算することができる。

水温センサーの応答時間 τ_T を見積もるには、少しずつ異なる τ_T の値に対して塩分を計算し、塩分スパイクを

最小にする r_T を探して、その値を応答時間とみなせばよい。実際にはFPの値を仮定して塩分を計算し、1秒(32データ)程度離れたところの塩分値との差の二乗和を最小にする値を求める。さらに、各測点で得られた値をレグあるいは航海ごとに平均し、較正に使用する。ただし、この1秒という間隔は任意である。塩分スパイクを表現する程度に大きく、2点間に海洋構造を含まない程度に小さくとればよい。(白鳳丸KH-91-5次航海のデータで試したところ、0.5秒の間隔で塩分差をとっても1秒間隔の場合とほとんど変わらず、レグの平均値は全く同じであった。)

さて、塩分差の二乗和はどの層について計算すればいいのであろうか。海面から海底までの全てのデータを使うのであればいけないのか、あるいはある特定の層のデータを使うのがいいのか。あらかじめ調べておく必要がある。白鳳丸KH-91-5次航海のレグ1における27回のCTD測定データを使って調べた。使うデータの範囲を圧力で指定し、浅い方の端をPmin、深い方をPmaxとする。まずPmaxを2000dbに固定し、Pminをいろいろ変えて、FPを求めた(表2)。測点C07では、Pmin=1dbの時にFP=0.76という極端に小さな値になった。このような測点が時々あるので、表面にごく近いところは除外した方がよい。Pminを5dbから大きくしていくと、FPが徐々に小さくなる傾向はあるが、大きな変化はない。Pminをどの値にとっても構わないようであるが、水温と電気伝導度の鉛直勾配が大きいのは表層であり、センサーの応答速度の違いが問題になるのはこの部分なので、なるべく浅い方を含めた方がよいであろう。そこで、Pminとして5dbを選ぶことにした。

次にPminを5dbに固定して、Pmaxを変えてみた(表3)。測点C07では、Pmax=6000dbの時に明らかにおかしな値になり、塩分差の二乗和も異常に大きい。これは、おそらく5000dbと6000dbの間にノイズが入っているのであろう。これを除くと、ほとんど同じ値になっている。特にC08とC09では、どの場合も同じである。C07では、Pmaxが200dbと500dbの時にFP=0.90、Pmaxが1000db以上ではFP=0.89になって、僅かながら変化している。このような例はしばしばみられるが、Pmaxが1500db以上になるとほとんど変化しない。これは、少なくとも1500dbまで測定しておけば、FPを求める上では問題ないことを意味している。Pmaxが2000db以上では全く変化しないので、ここではPmax=2000dbにすることにした。もちろんPmaxをもっと深くとっても構わないが、塩分の計算には時間がかかるので、同じ結果が得られるならばPmaxは小さい方がよい。

結局、5db~2000dbの範囲で塩分差の二乗和を計算し、それを最小にするFPを求めればよい、ということになる。このような方法で、白鳳丸KH-91-5次航海の各測点についてFPを求めた結果が、表4である。C16、C23、C28、C29、C30では明らかにおかしな値になっているが、塩分差の二乗和の異常な値から、除ききれなかったノイズのせいであることがわかる。これらの測点を除いて平均をとるとFP=0.90であり、レグ1(C01~C30)とレグ2(C31~C68)のそれぞれの平均も同じ値であった。

データのサンプリング間隔 Δ は、 $\Delta=1\text{秒}/32\text{データ}=31.25\text{mSec}$ なので、FP=0.90は $r_T=297\text{mSec}$ に当たる。水温センサーの応答時間は公称175mSecであるが、実際には110~300mSecの幅があるとされている。これからすると、白鳳丸航海で使用した水温センサーは、一般に言われている幅のうち、最も応答の遅い部類の製品ということになる。

C07					
MIN PR=	1.0	MAX PR=	2000.0	FP=	0.76 del-S= 0.6353535E+01
MIN PR=	5.0	MAX PR=	2000.0	FP=	0.89 del-S= 0.8832964E+00
MIN PR=	10.0	MAX PR=	2000.0	FP=	0.89 del-S= 0.8755649E+00
MIN PR=	20.0	MAX PR=	2000.0	FP=	0.89 del-S= 0.8750797E+00
MIN PR=	50.0	MAX PR=	2000.0	FP=	0.88 del-S= 0.1983034E+00
MIN PR=	100.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1723885E+00
MIN PR=	200.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1668736E+00
MIN PR=	300.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1639438E+00
MIN PR=	500.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1172122E+00
MIN PR=	1000.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1459527E-00
C08					
MIN PR=	1.0	MAX PR=	2000.0	FP=	0.88 del-S= 0.1404221E+01
MIN PR=	5.0	MAX PR=	2000.0	FP=	0.88 del-S= 0.1400842E+01
MIN PR=	10.0	MAX PR=	2000.0	FP=	0.88 del-S= 0.1397167E+01
MIN PR=	20.0	MAX PR=	2000.0	FP=	0.88 del-S= 0.1388318E+01
MIN PR=	50.0	MAX PR=	2000.0	FP=	0.89 del-S= 0.4902246E+00
MIN PR=	100.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1263979E+00
MIN PR=	200.0	MAX PR=	2000.0	FP=	0.86 del-S= 0.1118115E+00
MIN PR=	300.0	MAX PR=	2000.0	FP=	0.86 del-S= 0.9881705E-01
MIN PR=	500.0	MAX PR=	2000.0	FP=	0.86 del-S= 0.5774903E-01
MIN PR=	1000.0	MAX PR=	2000.0	FP=	0.85 del-S= 0.1371067E-01
C09					
MIN PR=	1.0	MAX PR=	2000.0	FP=	0.90 del-S= 0.5685879E+00
MIN PR=	5.0	MAX PR=	2000.0	FP=	0.90 del-S= 0.5680366E+00
MIN PR=	10.0	MAX PR=	2000.0	FP=	0.90 del-S= 0.5659928E+00
MIN PR=	20.0	MAX PR=	2000.0	FP=	0.90 del-S= 0.5604314E+00
MIN PR=	50.0	MAX PR=	2000.0	FP=	0.88 del-S= 0.2027767E+00
MIN PR=	100.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1508043E+00
MIN PR=	200.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1311287E+00
MIN PR=	300.0	MAX PR=	2000.0	FP=	0.86 del-S= 0.1208628E+00
MIN PR=	500.0	MAX PR=	2000.0	FP=	0.86 del-S= 0.8819979E-01
MIN PR=	1000.0	MAX PR=	2000.0	FP=	0.87 del-S= 0.1931789E-01

表2. Pmin, Pmax 間のデータから計算した塩分差の二乗和の最小値(del-S)とFP
 $[=exp(-\Delta/\tau_T)]$ の値。Pmax を 2000 db に固定した場合について、白
 鳳丸KH-91-5次航海の3測点の結果を示している。

C07					
MIN PR=	5.0	MAX PR=	200.00	FP=	0.90 del-S= 0.7036924E+00
MIN PR=	5.0	MAX PR=	500.00	FP=	0.90 del-S= 0.7568198E+00
MIN PR=	5.0	MAX PR=	1000.00	FP=	0.89 del-S= 0.8683640E+00
MIN PR=	5.0	MAX PR=	1500.00	FP=	0.89 del-S= 0.8784102E+00
MIN PR=	5.0	MAX PR=	2000.00	FP=	0.89 del-S= 0.8832964E+00
MIN PR=	5.0	MAX PR=	2500.00	FP=	0.89 del-S= 0.8874831E+00
MIN PR=	5.0	MAX PR=	3000.00	FP=	0.89 del-S= 0.8915793E+00
MIN PR=	5.0	MAX PR=	4000.00	FP=	0.89 del-S= 0.8993164E+00
MIN PR=	5.0	MAX PR=	5000.00	FP=	0.89 del-S= 0.9075936E+00
MIN PR=	5.0	MAX PR=	6000.00	FP=	0.95 del-S= 0.1780020E+02
C08					
MIN PR=	5.0	MAX PR=	200.00	FP=	0.88 del-S= 0.1285783E+01
MIN PR=	5.0	MAX PR=	500.00	FP=	0.88 del-S= 0.1340508E+01
MIN PR=	5.0	MAX PR=	1000.00	FP=	0.88 del-S= 0.1386612E+01
MIN PR=	5.0	MAX PR=	1500.00	FP=	0.88 del-S= 0.1396598E+01
MIN PR=	5.0	MAX PR=	2000.00	FP=	0.88 del-S= 0.1400842E+01
MIN PR=	5.0	MAX PR=	2500.00	FP=	0.88 del-S= 0.1403667E+01
MIN PR=	5.0	MAX PR=	3000.00	FP=	0.88 del-S= 0.1406201E+01
MIN PR=	5.0	MAX PR=	4000.00	FP=	0.88 del-S= 0.1411727E+01
MIN PR=	5.0	MAX PR=	5000.00	FP=	0.88 del-S= 0.1416591E+01
MIN PR=	5.0	MAX PR=	6000.00	FP=	0.88 del-S= 0.1420849E+01
C09					
MIN PR=	5.0	MAX PR=	200.00	FP=	0.90 del-S= 0.4140588E+00
MIN PR=	5.0	MAX PR=	500.00	FP=	0.90 del-S= 0.4591068E+00
MIN PR=	5.0	MAX PR=	1000.00	FP=	0.90 del-S= 0.5476407E+00
MIN PR=	5.0	MAX PR=	1500.00	FP=	0.90 del-S= 0.5586916E+00
MIN PR=	5.0	MAX PR=	2000.00	FP=	0.90 del-S= 0.5680366E+00
MIN PR=	5.0	MAX PR=	2500.00	FP=	0.90 del-S= 0.5711558E+00
MIN PR=	5.0	MAX PR=	3000.00	FP=	0.90 del-S= 0.5737075E+00
MIN PR=	5.0	MAX PR=	4000.00	FP=	0.90 del-S= 0.5822106E+00
MIN PR=	5.0	MAX PR=	5000.00	FP=	0.90 del-S= 0.5865946E+00
MIN PR=	5.0	MAX PR=	6000.00	FP=	0.90 del-S= 0.5908298E+00

表3. 表2と同様。ただし、Pmin を 5 db に固定した場合

C1	22.6	db	- 2000.0	db	FP= 0.89	del-S= 0.5960767E+01
C2	5.0	db	- 2000.0	db	FP= 0.89	del-S= 0.4025132E+01
C3	5.0	db	- 2000.0	db	FP= 0.83	del-S= 0.3324894E+01
C4	5.0	db	- 2000.0	db	FP= 0.88	del-S= 0.1993105E+01
C5	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.5914285E+01
C6	5.0	db	- 2000.0	db	FP= 0.88	del-S= 0.2600421E+01
C7	5.0	db	- 2000.0	db	FP= 0.89	del-S= 0.8836036E+00
C8	5.0	db	- 2000.0	db	FP= 0.88	del-S= 0.1399989E+01
C9	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.5681666E+00
C14	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.1928816E+01
C15	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.6860859E+00
C16	5.0	db	- 2000.0	db	FP= 0.99	del-S= 0.2979433E+02
C17	51.3	db	- 2000.0	db	FP= 0.90	del-S= 0.4668695E+01
C18	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.2161882E+01
C17S	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.2529355E+01
C18S	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.2803788E+01
C19	5.0	db	- 2000.0	db	FP= 0.94	del-S= 0.4349595E+01
C20	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.1730413E+01
C21	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.1212246E+01
C22	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.2040421E+01
C23	5.0	db	- 2000.0	db	FP= 0.96	del-S= 0.6058657E+03
C24	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.3582348E+01
C25	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.1240594E+01
C26	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.1217467E+01
C27	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.7680951E+00
C28	15.2	db	- 2000.0	db	FP= 0.97	del-S= 0.2411823E+02
C29	5.0	db	- 2000.0	db	FP= 0.99	del-S= 0.1871693E+03
C30	5.0	db	- 2000.0	db	FP= 0.99	del-S= 0.6683770E+03
C31	5.0	db	- 2000.0	db	FP= 0.93	del-S= 0.2787151E+01
C32	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.2787112E+02
C33A	5.0	db	- 2000.0	db	FP= 0.93	del-S= 0.4738757E+01
C33	23.8	db	- 2000.0	db	FP= 0.92	del-S= 0.3116409E+01
C34	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.5470300E+01
C35	5.0	db	- 2000.0	db	FP= 0.93	del-S= 0.6805353E+01
C36	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.2912254E+01
C37	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.6495940E+01
C38	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.4075874E+01
C39	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.7350249E+01
C42	5.0	db	- 2000.0	db	FP= 0.93	del-S= 0.3636537E+01
C43	5.0	db	- 1179.8	db	FP= 0.88	del-S= 0.2315461E+01
C44	5.0	db	- 2000.0	db	FP= 0.94	del-S= 0.4685982E+01
C45	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.3087481E+01
C46	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.4254436E+01
C47	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.3465541E+01
C48	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.3114460E+01
C49	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.4832692E+01
C50	5.0	db	- 2000.0	db	FP= 0.88	del-S= 0.2912918E+01
C51	5.0	db	- 2000.0	db	FP= 0.89	del-S= 0.2592515E+01
C52	5.0	db	- 2000.0	db	FP= 0.89	del-S= 0.3105742E+01
C53	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.2447948E+01
C54	5.0	db	- 2000.0	db	FP= 0.87	del-S= 0.1565851E+01
C55	5.0	db	- 2000.0	db	FP= 0.88	del-S= 0.5483459E+01
C56	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.3714524E+01
C57	5.0	db	- 2000.0	db	FP= 0.92	del-S= 0.3404859E+01
C58	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.2693446E+01
C59	5.0	db	- 2000.0	db	FP= 0.89	del-S= 0.2809527E+01
C60	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.4217770E+01
C61	5.0	db	- 2000.0	db	FP= 0.90	del-S= 0.5552881E+01
C62	5.0	db	- 2000.0	db	FP= 0.87	del-S= 0.4977493E+01
C63	5.0	db	- 1653.3	db	FP= 0.89	del-S= 0.6397893E+01
C64	5.0	db	- 1840.5	db	FP= 0.88	del-S= 0.1942797E+01
C65	5.0	db	- 2000.0	db	FP= 0.89	del-S= 0.2074058E+01
C66	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.1684935E+01
C67	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.1636438E+01
C68	5.0	db	- 2000.0	db	FP= 0.91	del-S= 0.1327995E+01

表4. 5 dbと2000 db間のデータから計算したFP。白鳳丸KH-91-5次航海の結果

6. 採水データの処理

CTDを上げる際に採水した水を塩分計と酸素滴定装置にかけ、各採水層での塩分・溶存酸素の濃度を測定する。採水時にとったCTDの圧力・水温・電気伝導度データは、ノイズを落とした後の平均値あるいは中央値をもって、その採水層での値とする。採水時には、CTDを止めてから数秒から10秒程度そのままにしておき、センサーの値を落ち着かせてからデータを取り込む。そのため、センサーによる応答速度の違いはここでは問題にならない。水温・圧力値には陸上で得たセンサー補正式(圧力には減圧時の式)を適用し補正する。

ニスキン採水器が、船上からの指令により毎回1つずつ確実に閉まっていれば、CTDの水温・圧力値と採水の塩分値から採水の電気伝導度を計算し、それに合うようにCTDの電気伝導度を補正すればよい。しかし実際には、必ずしも指令通りに採水器が閉まるわけではない。1回の指令に対し2本まとめて閉まることもあれば、1本も閉まらないこともある。こうした事態が起きると、CTDの測定と採水の深度がずれてしまう。電気伝導度の補正を行うには、まずこの深度のずれをチェックし、修正しなければならない。

そのためには、できるだけ多くの採水器に転倒圧力計を付けることである。圧力計の示す値から、実際に採水器の閉まった深さを知ることができる。全ての採水器にセットできればこの問題は解決するが、ロゼット・サンブラーの構造上それはできない。また、転倒圧力計は高価であり、実際に用意できる本数には限界がある。圧力計のついていない採水器については、CTDの値から大ざっぱに計算した塩分と酸素を採水の塩分・酸素と比較することで実際の採水深度を判断する以外にない。この場合、CTDから計算した値は全くいい加減なものであるが、鉛直プロファイルを採水値と比較する上では問題ない。白鳳丸KH-91-5次航海では、24本のニスキン採水器のうち4本に転倒圧力計と水温計をつけ、残りについては塩分・酸素の鉛直分布から採水深度を推定した。1つ1つの測点について手作業で行わざるを得ない上に測点によっては判断が難しく、多くの時間を要した。その時々サンブラー・システムの調子によって異なるので一概には言えないが、ロゼット・サンブラーの構造を根本的に変えない限り一般には避けられない問題と思われる。

こうして採水深度の修正をした後、CTDの水温・圧力値と採水の塩分値から採水の電気伝導度を計算する。それに合うように、CTDの電気伝導度センサーの補正式を求める。

7. 電気伝導度(塩分)の較正

7-1. 採水の値に合わすべきは、電気伝導度か塩分か？

電気伝導度あるいは塩分に関しCTDの値を採水値に合わす場合、電気伝導度と塩分のどちらで合わすのがよいのか、またそれらの差と比のどちらを使うのがよいのか、という疑問に対し、若干の検討を行った。1989年に気象庁凌風丸によってとられた6測点のデータを使って、以下の手順で検討した。

- (1) センサーと採水による電気伝導度(C_{CTD} と C_{sample})と塩分(S_{CTD} と S_{sample})から、電気伝導度の差 $\Delta C(=C_{CTD} - C_{sample})$ と塩分の差 $\Delta S(=S_{CTD} - S_{sample})$ 、伝導度の比 $Cf(=C_{sample}/C_{CTD})$ と塩分の比 $Sf(=S_{sample}/S_{CTD})$ を計算する。 Cf をセル・ファクターと呼んでいる。
- (2) 測点ごとに、 ΔC 、 ΔS 、 Cf 、 Sf のそれぞれを、最小自乗法で圧力の3次関数に展開する。
- (3) 3次式から計算した塩分値と S_{sample} との差の2乗平均を比較する。

このようにして、4つのうち較正に最も適した量を選ぶことにした。

塩分の鉛直勾配の大きなところでは ΔC 等の値が大きくばらつく。これは、採水とセンサーの測定が厳密には同じ場所で行なわれていないことに起因すると思われる。ロゼット採水器がセンサーの1mほど上方につけられているうえに、センサー値を計算機に取り込んでから採水するまでに若干の時間があり、その間にもCTDの深さが変わり、採水とセンサー測定の深度にわずかながら違いが生じる。表5では、全データを使用して圧力の3次式を求めた場合と塩分の鉛直勾配が $0.003 \text{ psu db}^{-1}$ 以上の所を除いて求めた場合の残差を比較した。(ここでの塩分の鉛直勾配は鉛直方向となりあう2つの採水点の値から計算したもので、正確な値ではない。)

2つの場合で使用データの変わらないSta.7390を除いた全ての測点で、塩分勾配の大きな所を除いた場合の方が残差が小さい。残差の減り方は測点によって全く異なり、Sta.7405のように使用データ数が9つも違うのに残差はほとんど変わらない場合もあれば、Sta.7445のように2つしか違わないにもかかわらず残差が半減する場合もある。何れにしても、塩分極小層の上部や下部といった塩分勾配の大きな所での採水値は、塩分較正に不適當な場合が多い。この議論では、採水値から計算した塩分勾配が 0.003 db^{-1} よりも大きい所はあらかじめ除くことにする。ただし、1000db以深での残差は、塩分勾配の大きな所を除いても除かなくても結果は大して変わらない(表6)。3次という高次の式で合わせているために、1000db以浅での違いに引きずられずにすんでいる。

ΔC 、 ΔS 、 Cf 、 Sf を使ったそれぞれの場合を比較してみると、残差が最小になる測点のもっとも多いのは、 Cf を使った場合である(表7)。6測点中4点で最小である。次に多いのは ΔC を使った場合であるが、この場合には残差が最大になる測点も3点と多い。4つの方法の中での残差の最小値を基準にし、最小値との差を6測点分たした値を表9に示した。圧倒的にセル・ファクター Cf の成績がよい。1000db以深についても同じことがいえる(表8)。 ΔC や ΔS 、 Sf よりも、セル・ファクター Cf を使って較正する方がはるかによい、と結論される。

SCOR Working Group(1988)も、セル・ファクターの使用を薦めている。その方法は、鉛直方向にも、またいく

つもの測点についてもまとめて平均をとり、その値を全ての深さ、全ての測点で使うというものである。しかしセル・ファクターは鉛直方向に変化するので、平均値でいいとは思えない。WOCEの資料では、 C_{sample} に合うように求めた C_{CTD} の1次式を補正に使用している。

測点番号	7389	7390	7405	7410	7415	7445
ケース1	18 (20)	10.35 (20)	8 (24)	7.8 (19)	3.42 (18)	15.69 (18)
ケース2	16.33 (18)	10.35 (20)	7.87 (15)	5.27 (15)	2.92 (12)	7.88 (16)
ケース3	46.39	18.9	9	20.53	4.58	16.81

表5. 各測点においてセル・ファクター Cf を圧力の3次式に展開し、その3次式を使って計算した塩分値と採水による塩分値の差を1000倍して2乗平均をとった値。値が小さいほど較正がうまくいったことを示す指標。括弧内の数字は使用したデータ数を示す。

ケース1：観測した全部のデータを使った場合。

ケース2：塩分の鉛直勾配が 0.003 db^{-1} 未満におけるデータのみを使った場合。

ケース3：ケース2で得た3次式の鉛直平均値をセル・ファクターとして使った場合。

測点番号	7389	7390	7405	7410	7415	7445
ケース1	5.25 (4)	7.71 (7)	1 (8)	3.5 (8)	4.33 (6)	0 (2)
ケース2	5.25 (4)	7.71 (7)	1.13 (8)	3.5 (8)	3.83 (6)	1 (2)
ケース3	1.5	6.71	2.5	12.38	5.17	42.5

表6. 表5で得た値のうち1000 db以深の値のみで計算した残差。

測点番号	7389	7390	7405	7410	7415	7445
ΔC	17.2	<u>9.4</u>	<u>10.7</u>	<u>7.9</u>	<u>2.9</u>	<u>8.8</u>
ΔS	17.8	10.3	<u>7.3</u>	6.1	3.8	8.6
Cf	<u>16.3</u>	10.4	7.9	<u>5.3</u>	<u>2.9</u>	<u>7.9</u>
Sf	<u>18.3</u>	<u>10.6</u>	8.5	5.9	<u>3.9</u>	<u>8.8</u>

表7. ΔC , ΔS , Cf , Sf を使って、表5のケース2と同様に求めた残差。下線は4つのうちの最大値を、波下線は最小値を示す。

測点番号	7389	7390	7405	7410	7415	7445
△C	<u>9.5</u>	<u>9.9</u>	<u>1.8</u>	<u>3.4</u>	<u>3.8</u>	<u>2.5</u>
△S	<u>5.3</u>	8.6	<u>0.8</u>	<u>3.9</u>	5.3	J
Cf	<u>5.3</u>	<u>7.7</u>	1.1	3.5	<u>3.8</u>	J
Sf	<u>5.3</u>	8.6	0.9	3.5	<u>5.7</u>	J

表8. 表7で得た値のうち1000 db以深の値のみで計算した残差。下線および波線の意味は表7と同じ。

△C	7.9	[8.9]
△S	4.7	[2.9]
Cf	1.5	[0.5]
Sf	6.9	[3.0]

表9. 各測点において、4つの方法(△C, △S, Cf, Sf)による残差からそれらの最小値を引き、その差を6測点分たした値。
[]内は1000 db以深のデータによる場合。

7-2. 電気伝導度データの較正方法 (プロセスB7)

白鳳丸KH-91-5次航海レグ1の採水データと採水時CTDデータから計算したセルフファクター(=C_{sample}/C_{CTD})を図6aに示した。鉛直方向にセルフファクターが大きく変化することと、大きくはずれた値の多いことが特徴である。図中の曲線は、セルフファクターの鉛直変化を最小自乗法により圧力の5次式で表した近似曲線である。全データで求めた図6aの曲線は、はずれたデータの影響で大きく波うっているが、この曲線からのデータのずれのRMS(root mean square)を求め、その4倍以上近似曲線からはずれたデータを除くと図6bのようになって、1500db以深での変化が小さくなる。さらに、同じようにして曲線からのずれのRMSを計算し、その3倍以上はずれているデータを除外して近似曲線を求め(図6c)、さらにRMSの2倍以上はずれているデータを除いて最終的な近似曲線を求めた(図6d)。前節では、塩分が鉛直方向に急激に変化する深度のデータは較正に向きであることを指摘したが、ずれの大きなデータを漸次落としていくことで不適当なデータを自動的に落としている。

レグ2についても同様の鉛直変化が得られた。すなわち、セルフファクターは表面で1に近く、1500dbほどまで大きく増加し、2000db以深ではほとんど一定であった。しかし値自体は、レグ2の方が全深度で小さく、2000db以深での値は、レグ1の約1.00049に対し、レグ2では約1.00037であった。このように、鉛直方向の変化の仕方は変わらなくても、値自体は測定を重ねるにつれ、あるいは数日中断することによって大きく変わる。

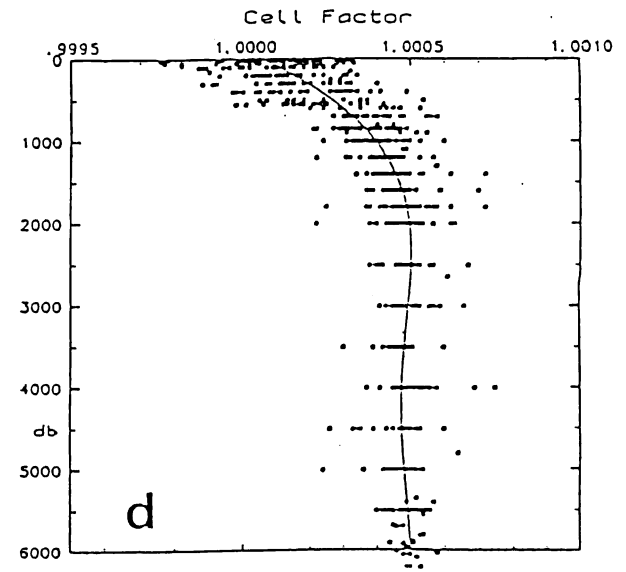
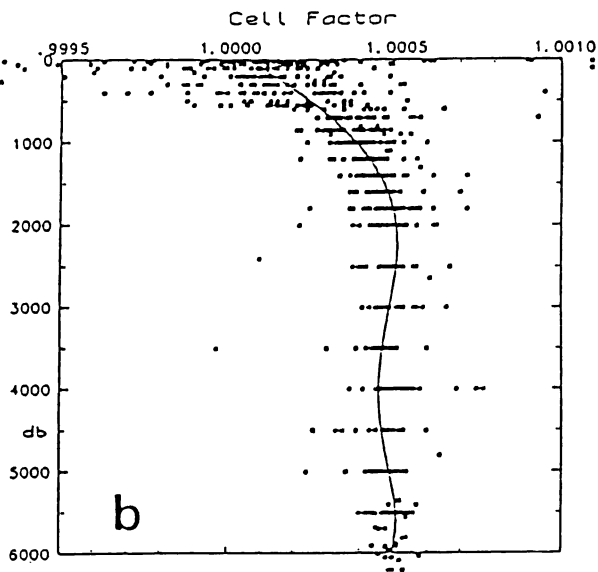
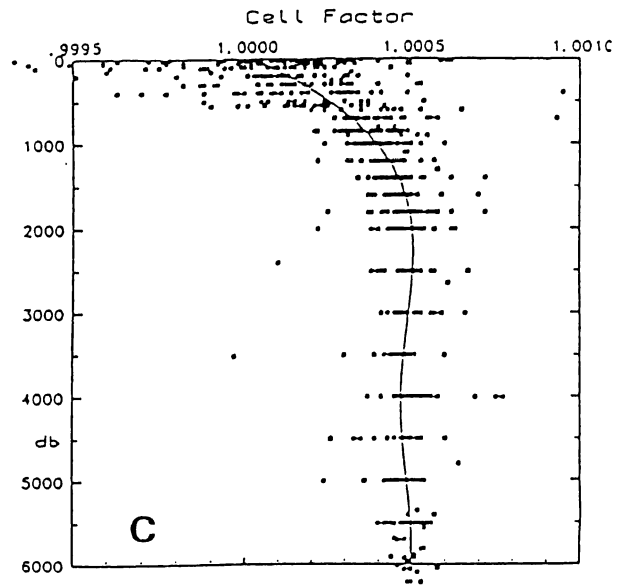
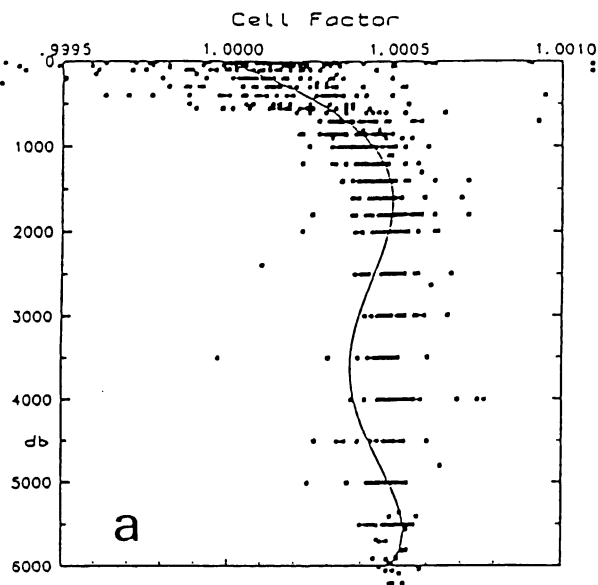


図6. 白鳳丸KH-91-5次航海レグ1でのセルファクターの鉛直分布。
 実線は、圧力の5次式で合わせた近似曲線。a~dについては本文参照。

セルファクターCfの鉛直変化を表す圧力P(db)の5次式

$$Cf = b_0 + b_1P + b_2P^2 + b_3P^3 + b_4P^4 + b_5P^5$$

の各係数は、レグ1では、

$$\begin{aligned} b_0 &= 1.000054, & b_1 &= 0.5069215 \times 10^{-6}, & b_2 &= -0.1927917 \times 10^{-9}, \\ b_3 &= 0.2664620 \times 10^{-13}, & b_4 &= -0.4903089 \times 10^{-18}, & b_5 &= -0.9984428 \times 10^{-22} \end{aligned}$$

であり、レグ2では、

$$\begin{aligned}
 b_0 &= 0.9999086, & b_1 &= 0.6578022 \times 10^{-6}, & b_2 &= -0.3865528 \times 10^{-9}, \\
 b_3 &= 0.1120873 \times 10^{-12}, & b_4 &= -0.1563393 \times 10^{-16}, & b_5 &= 0.8308362 \times 10^{-21}
 \end{aligned}$$

であった。

このように、白鳳丸KH-91-5次航海でのCTD較正には、セルフファクターを圧力の5次式で近似する方法を使ったが、WOCEの資料には、レポートの例として、 C_{sample} に合うように決めた C_{CTD} の1次式による較正が示されている。そこで、白鳳丸KH-91-5次航海のデータを使い、セルフファクターCfと圧力P、 C_{sample} と C_{CTD} 、Cfと C_{CTD} の関係式による3種類の較正方法を比べた。

表10に、それぞれの方法で得られた S_{CTD} と S_{sample} との差を示した。塩分データに高い精度が要求される2000db以深については、Cf-Pの5次式の場合が最も小さな差を示している。それが、この方法を白鳳丸データの較正に用いた理由であるが、他の方法に比べて圧倒的に良いというわけではない。 $C_{\text{sample}} - C_{\text{CTD}}$ も、Cf- C_{CTD} も悪くない結果を出しており、2000db以浅については、これらを使った方がCf-Pの方法よりも残差の小さい場合もある。また、 C_{sample} (or Cf) - C_{CTD} の方法では、1次式や3次式よりも2次式を使う方がよいことを、結果は示している。しかし、最終的な結論を得るには、もっと観測例を増やして検討する必要がある。

	Cf-P N=5	$C_{\text{sample}} - C_{\text{CTD}}$			Cf- C_{CTD}		
		N=1	N=2	N=3	N=1	N=2	N=3
Leg 1							
P<2000db	4.43 (432)	3.93 (414)	3.72 (420)	3.84 (424)	4.53 (424)	3.94 (430)	3.98 (432)
P>2000db	2.41 (258)	2.97 (260)	2.73 (259)	2.73 (259)	2.79 (258)	2.55 (257)	2.48 (256)
Total	3.80 (690)	3.59 (674)	3.38 (679)	3.46 (683)	3.97 (682)	3.49 (687)	3.50 (688)
Leg 2							
P<2000db	3.74 (455)	4.09 (452)	3.56 (446)	3.77 (453)	4.57 (458)	3.69 (454)	3.91 (458)
P>2000db	2.22 (177)	2.96 (178)	2.56 (178)	2.56 (178)	3.17 (178)	2.32 (177)	2.32 (177)
Total	3.39 (632)	3.81 (630)	3.30 (624)	3.47 (631)	4.22 (636)	3.36 (631)	3.54 (635)

表10 各方法で較正した C_{CTD} より求めた塩分 S_{CTD} と採水の塩分 S_{sample} との差のRoot Mean Square($\times 10^{-3}$ psu)。カッコ内はデータ数、Cfはセルフファクターを示す。1行目のY-X(Cf-P等)の表示は、YをXの多項式 $Y = \sum_{n=0}^N a_n \cdot X^n$ で近似し、その式を使って C_{CTD} を補正することを意味する。いずれも図6のように大きくずれた測定値を落としていき、最終的に補正式の導出に用いたデータ(図6dの段階のデータ)を使用した。

8. おわりに

ここでまとめたCTD校正プロセスの中で、センサー値のノイズの除去とセンサー値に施すデジタル・フィルターの係数FPの決定には、複雑な判断が要求される。しかしそれを人間の目で行おうとすれば、大変な時間と手間がかかり、作業は滞ってしまう。本冊子の方法のように、これらを計算機で機械的に処理することが、CTDデータの迅速な校正を可能にする鍵である。

本冊子では、サーミスタによる水温値、いわゆる "Fast T" は使わないことにしたが、現実には "Fast T" の付いたCTDを使っているところも多いであろう。"Fast T" は、塩分変化の大きな比較的浅い層においては塩分スパイクを小さくする働きはあるが、塩分変化の小さな深層においては、安定性の悪さ故にかえって擾乱源になることもあり得る。また "Fast T" が付いている場合には、(5.10)式を変えないといけない可能性もある。このような理由から、"Fast T" を使わない方がスッキリして問題が少ないと考えたのであるが、本当にその方がいいデータを作ることになるのかどうかを確かめたわけではない。"Fast T" の付いたCTDと付いていないCTDを同時に降ろして判断するのが最も良いのであるが、こうした試みは今のところなされていない。

CTDの塩分や溶存酸素の校正を正確に行うには、採った水の塩分と溶存酸素を精度良く測定することが必要である。これらの校正では、採水の値を真値とみなしてそれに合わせるので、当然のことながら、塩検値と滴定値の正確さが前提である。採水ビンに正しく対応した採水器からきちんとした方法で採水しているかどうか、採った水の処置や塩分計と酸素滴定装置の管理や操作をきちんと行っているかどうか、を常に確認する必要がある。また、採水器の水漏れがひどいといった状況が見られた時は、その旨の記述をきちんと残しておくべきである。ちなみに、WOCEでは、データだけでなく採水器の状態についても決められた数値によるフラッグで示すことになっており、水漏れの場合は3、ふたが閉じなかった場合は9というフラッグを立てる取り決めになっている。

酸素センサーは、今のところ十分な安定性を持っているとは言えない。まだまだ改良が必要である。しかし、それでもやはりその機能は重要であり、できるだけ酸素センサーの付いたCTDを使うべきである。理由は2つある。1つは、校正をしてもなお値に問題が残るとしても、酸素の大体の鉛直プロファイルがわかることである。酸素の鉛直プロファイルは、海水混合や海洋大循環を調べる上で、貴重な情報である。ノイズで使えない部分を除けば、採水値に合わず努力を各測点について行えば、値についても何とかなる可能性もある。2つめの理由は、採水時の酸素センサー値があれば、6節で述べた採水深度の割り出しがより正確にでき、また採水による酸素値の品質の判断もやりやすくなることである。CTD・採水データセットの作成において、酸素センサーの値が大きな役割を果たすことも見逃してはならない。

国際度量衡委員会は、これまで100℃としてきた水の沸点を 99.974℃とする水温の新しい定義を採用することにし、1990年からの実施を勧告した。新しい水温スケールITS-90 (International Temperature Scale of 1990)は、これまでのIPTS-68 (International Practical Temperature Scale of 1968) に代わるものとして今後使われることになる。しかし、現時点ではITS-90に基づく塩分や密度の計算式が提出されていないので、しばらくはIPTS-68を使って計算し、最後に水温値をITS-90に変換するのがよいであろう。IPTS-68での水温値を0.99976倍すれば、

おおよそITS-90に基づく値になる。新しい計算式が確立した時点で、水温センサーの較正式をITS-90に基づいて求めるように変更することになる。

なお、CTD観測・データ処理については、以下に挙げた参考文献に丁寧に書いてあるので、本冊子とともに参考にしていただきたい。

参考文献

- Giles, A.B. and T.J. McDougall (1986) : Two methods for the reduction of salinity spiking of CTDs. *Deep-Sea Res.*, 33, 1253-1274.
- 深沢理郎(1988) : CTDデータ更正について。「深層海水循環過程の解明」ニュースレター, No.4.
- 長崎海洋気象台海洋課(1989) : 新長風丸のCTDシステムについて。測候時報, 56, 3, 91-114.
- SCOR Working Group (1988) : The acquisition, calibration, and analysis of CTD data. *UNESCO technical papers in marine science*, 54.

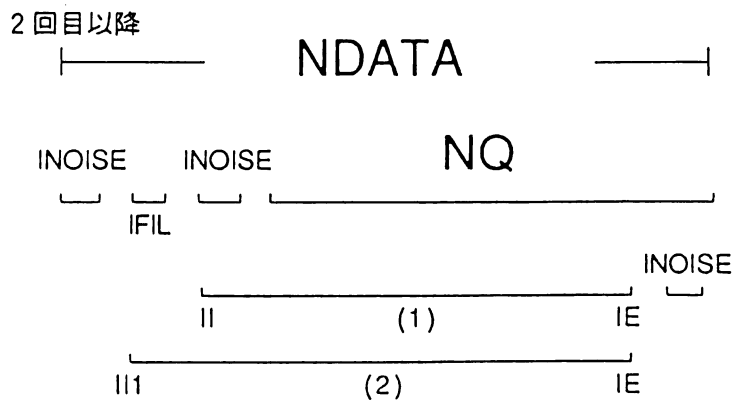
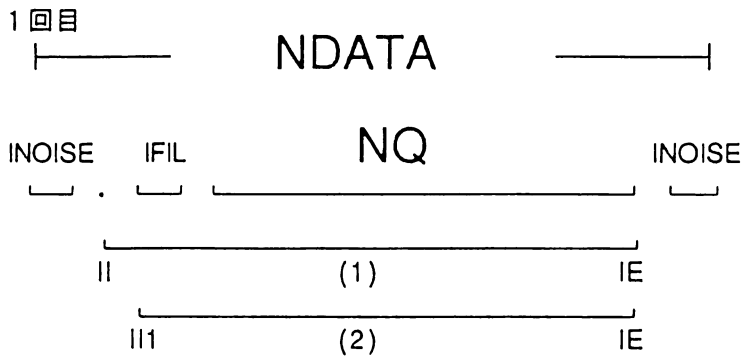
附録 (CTDデータ較正のための計算機プログラム)

プログラム例を最後に載せておく。

* このマニュアル・ガイドに掲載しているプログラムリストは、フロッピーディスクにより提供可能です。
詳細については、日本海洋データセンターまでご連絡下さい。

1. 圧力・電気伝導度データにかけるデジタル・フィルターの係数FPを求めるプログラム (TCRESPON.FOR) [A2→A3→A4→A5→A6]

生データは一度に処理するには膨大すぎるので、NQ個ずつ読んで処理するようにした。そのため、プログラムが多少複雑になっている。各サブルーチンでの処理に使うデータ数やデータの範囲を図示しておく。



- (1) は、SUB ROUTINE NOISE, TCCORRECT, PCORDOWN, TEMPFILT, CELLCORR
(2) は、SUB ROUTINE FILTCOEFF で計算に使う範囲。

```

C*****
C Program to calculate the difference in response *
C between T sensor and P, C sensors (April, 1992) *
C*****
C Files
C INPUT FT30F001 C01.DATA (LRECL=29)
C INPUT FT31F001 C02.DATA (LRECL=29)
C OUTPUT FT50F001 FPC01C02.DATA (LRECL=80)
*
C NSTN: Number of station processed here
C PTH, TTH, CTH: Threshold values used in SUB. NOISE
C PMIN, PMAX: Min. and max. pressures of the data used here
C If IL>1, values of PMIN and PMAX must be defined in the
C increasing order
C FP: Lag of response of T sensor behind P, C sensors
C Seek the best value of FP among NDS (number) values
C with an interval of DFP from the minimum FPS
PARAMETER (NQ=20000,IP=NQ+80)
REAL*8 P(IP),T(IP),C(IP),S(IP)
PARAMETER (NSTN=2,IL=1,NDS=20)
REAL*8 DS(IL,NDS),PA(NDS),CA(NDS)
REAL*8 PTH/0.1D+1/,TTH/0.1D+0/,CTH /0.2D-1/
REAL*8 FP,FPS,DFP
DIMENSION PMIN(IL),PMAX(IL),ISTOP(IL)
DIMENSION PRMIN(IL),PRMAX(IL)
DATA PMIN/5./
DATA PMAX/2000./
CHARACTER*12 FRM10/'(F5.1,2F6.3)'/
CHARACTER*1 DUM
CHARACTER*5 STA
*
C NSEC: Observed data number per second
C INOISE: Used in SUB. NOISE
C IUP, IDN: Used in SUB. TEMPFLT
C IFIL: Used in SUB. FILTCOEF
COMMON /OBSDAT/ NSEC
COMMON /NOISIE/ INOISE
COMMON /TFILTR/ IUP,IDN
COMMON /FILCO/ IFIL
COMMON /COMFPS/ FPS,DFP
*
NSEC=32
INOISE=2
IUP=NSEC/2
IDN=NSEC/2
IFIL=NSEC
FPS=0.80D0
DFP=0.1D-1
*
DO 50 NF=1,NSTN
IDF=NF+29
DO 3 I=1,IL
PRMIN(I)=PMIN(I)
PRMAX(I)=PMAX(I)
3 CONTINUE
C Find the beginning of valid data observed in the sea
C Determine IINIT. The IINIT'th data is the first of
C valid data series measured in the sea
IINIT=0
JC=0
J=0
*
LOOP BEGIN (READ RAW DATA)
READ(IDF,'(A1)') DUM
5 READ(IDF,FRM10) PD,TD,CD
TD=TD*0.5
JC=JC+1
IF(PD.GT.20.0.AND.PD.LT.25.5.AND.
* TD.GT. 5.0.AND.TD.LT.35.0.AND.
* CD.GT.30.0.AND.CD.LT.70.0) GO TO 1
IF(CD.LT.40.0) THEN
J=J+1
IF(J.GE.3) IINIT=JC
ELSE
IF(CD.LT.70.0) J=0.
END IF
GO TO 5
*
1 REWIND IDF
WRITE(6,*) 'IINIT=',IINIT
*

```

```

DO 10 K=1,IL                                TCR00810
  ISTOP(K)=0                                TCR00820
DO 10 I=1,NDS                                TCR00830
  DS(K,I)=0.0D+0                            TCR00840
10 CONTINUE                                  TCR00850
  II=INOISE + 1                              TCR00860
  I1=II + 1                                  TCR00870
  IPROC=INOISE*2+IFIL+1                     TCR00880
*                                             TCR00890
  READ(IDF,'(A5)') STA                       TCR00900
  IF(IINIT.GE.1) THEN                         TCR00910
    DO 15 I=1,IINIT                           TCR00920
      READ(IDF,'(A1)') DUM                    TCR00930
15 CONTINUE                                  TCR00940
  END IF                                      TCR00950
*                                             TCR00960
DO 20 I=1,IPROC                              TCR00970
  READ(IDF,FRM10) P(I),T(I),C(I)            TCR00980
  T(I)=T(I)*.5D+0                            TCR00990
20 CONTINUE                                  TCR01000
  DO 2 I=1,NDS                                TCR01010
    PA(I)=P(II)                               TCR01020
    CA(I)=C(II)                               TCR01030
  2 CONTINUE                                  TCR01040
*                                             TCR01050
  JEND=0                                      TCR01060
  NUM=0                                       TCR01070
25 NI=0                                       TCR01080
  NUM=NUM + 1                                 TCR01090
  DO 30 I=1,NQ                                TCR01100
    IS=I + IPROC                              TCR01110
    READ(IDF,FRM10,END=35) P(IS),T(IS),C(IS) TCR01120
    T(IS)=T(IS)*.5D+0                        TCR01130
    NI=I                                       TCR01140
30 CONTINUE                                  TCR01150
  GO TO 40                                     TCR01160
35 JEND=1                                      TCR01170
  IF(NI.EQ.0) GO TO 900                       TCR01180
40 CONTINUE                                  TCR01190
  NDATA=IPROC + NI                           TCR01200
  IE=NDATA - INOISE                          TCR01210
C Process A2                                 TCR01220
  WRITE(6,*) 'A2: CHECKING NOISE ',NDATA     TCR01230
  CALL NOISE(P,IP,PTH,IFLGP,II,IE)           TCR01240
  CALL NOISE(T,IP,TTH,IFLGT,II,IE)          TCR01250
  CALL NOISE(C,IP,CTH,IFLGC,II,IE)          TCR01260
  WRITE(6,*) 'NOISE CHECK COMPLETED',IFLGP,IFLGT,IFLGC TCR01270
C Process A3                                 TCR01280
  WRITE(6,*) 'A3: T AND P CORRECTION'        TCR01290
  CALL TCORRECT(T,IP,II,IE)                  TCR01300
  CALL PCORDOWN(P,IP,II,IE)                  TCR01310
C Process A4                                 TCR01320
  WRITE(6,*) 'A4: FILTERING TEMP DATA'      TCR01330
  CALL TEMPFILT(T,P,IP,II,IE)                TCR01340
C Process A5                                 TCR01350
  WRITE(6,*) 'A5: CORRECT C FOR CELL DEFORMATION' TCR01360
  CALL CELLCORR(C,P,T,IP,II,IE)             TCR01370
*                                             TCR01380
  IF(P(IE).LT.PMIN(1)) GO TO 55              TCR01390
  CALL FILTCOEF(P,C,T,S,PA,CA,IP,DS,ISTOP,  TCR01400
    PMIN,PMAX,PRMIN,IL,NDS,NUM,I1,IE)      TCR01410
  DO 45 K=1,IL                                TCR01420
    IF(ISTOP(K).EQ.0) GO TO 55               TCR01430
45 CONTINUE                                  TCR01440
  GO TO 800                                    TCR01450
55 CONTINUE                                  TCR01460
*                                             TCR01470
  IPROC=INOISE*2 + IFIL                      TCR01480
  DO 60 I=1,IPROC                              TCR01490
    IS=NDATA - IPROC + I                     TCR01500
    P(I)=P(IS)                               TCR01510
    T(I)=T(IS)                               TCR01520
    C(I)=C(IS)                               TCR01530
60 CONTINUE                                  TCR01540
  II=INOISE+IFIL+1                           TCR01550
  I1=INOISE + 1                              TCR01560
  IF(JEND.EQ.0) GO TO 25                     TCR01570
*                                             TCR01580
900 CONTINUE                                  TCR01590
  DO 65 K=1,IL                                TCR01600

```

```

        IF(ISTOP(K).EQ.0) THEN
            PRMAX(K)=P(NDATA-INOISE)
            KEND=K
            GO TO 700
        END IF
65    CONTINUE
800    KEND=IL
C    Process A6
C    Search for the smallest DS(I) and determine adequate FP
    WRITE(*,*) 'A6: SEARCHING FOR THE BEST LAG OF P,C FILTER.'
700    DO 70 K=1,KEND
        DSMIN=DS(K,1)
        NMIN=1
        DO 75 I=2,NDS
            IF(DS(K,I).LT.DSMIN) THEN
                DSMIN=DS(K,I)
                NMIN=I
            END IF
75    CONTINUE
        FP=FPS + DFP*DFLOAT(NMIN-1)
*
C    Create final dataset
    WRITE(50,100) STA,PRMIN(K),PRMAX(K),FP,DSMIN,IFIL
100    FORMAT(A5,F6.1,' db - ',F6.1,' db',5X,'FP=',F5.2,
*        4X,'del-S=',E15.7,4X,'IFIL=',I2)
70    CONTINUE
50    CONTINUE
*
1000 STOP
    END
*
C*****
    SUBROUTINE NOISE(X,IP,XTH,IFLG,IINI,IEND)
C    Remove noise of raw conductivity data
C    Criterion: Distinguish as noise when
C    X(I) > XTH + MAX(X(I-N),X(I+N))
C    X(I) < -XTH + MIN(X(I-N),X(I+N))
C    --> Revise to X(I) = (X(I-N)+X(I+N))/2
C    The number of revised data will be set in IFLG
*
    REAL*8    X(IP),XTH,XMAX,XMIN
    COMMON /NOISIE/ N
    IFLG=0
*
    DO 10 I=IINI,IEND
        XMAX=MAX(X(I-N),X(I+N))
        XMIN=MIN(X(I-N),X(I+N))
        IF(X(I).GT.XMAX+XTH .OR. X(I).LT.XMIN-XTH) THEN
            X(I) = (X(I-N)+X(I+N))*0.5D0
            IFLG = IFLG+1
        ENDIF
10    CONTINUE
    RETURN
    END
*
C*****
    SUBROUTINE TCORRECT(T,IP,IINI,IEND)
C    Correct temperature data using the quadratic polynomial
C    obtained in a laboratory
C    DT: Correction, DT = A * T**2 + B * T + C
*
    REAL*8    T(IP),DT,A(2),B(2),C(2)
C-----FOR THE CTD WITHOUT O2 SENSOR
C    BELOW "TEMP", ABOVE "TEMP"
C    DATA A/-0.100093D-4, 0.677039D-6/
C    DATA B/-0.566937D-3, -0.721866D-3/
C    DATA C/ 0.166095D-1, 0.170540D-1/
C    DATA TEMP/10.0/
C-----FOR THE CTD EQUIPPED WITH O2 SENSOR
C    BELOW "TEMP", ABOVE "TEMP"
C    DATA A/ 0.121937D-4, 0.179550D-4/
C    DATA B/-0.803083D-3, -0.127571D-2/
C    DATA C/ 0.251885D-1, 0.290126D-1/
C    DATA TEMP/ 9.1/
*
    DO 10 I=IINI,IEND
        K=1
        IF(T(I).LT.TEMP) K=2
        DT = A(K)*T(I)**2 + B(K)*T(I) + C(K)
        T(I) = T(I) + DT

```

TCR01610
TCR01620
TCR01630
TCR01640
TCR01650
TCR01660
TCR01670
TCR01680
TCR01690
TCR01700
TCR01710
TCR01720
TCR01730
TCR01740
TCR01750
TCR01760
TCR01770
TCR01780
TCR01790
TCR01800
TCR01810
TCR01820
TCR01830
TCR01840
TCR01850
TCR01860
TCR01870
TCR01880
TCR01890
TCR01900
TCR01910
TCR01920
TCR01930
TCR01940
TCR01950
TCR01960
TCR01970
TCR01980
TCR01990
TCR02000
TCR02010
TCR02020
TCR02030
TCR02040
TCR02050
TCR02060
TCR02070
TCR02080
TCR02090
TCR02100
TCR02110
TCR02120
TCR02130
TCR02140
TCR02150
TCR02160
TCR02170
TCR02180
TCR02190
TCR02200
TCR02210
TCR02220
TCR02230
TCR02240
TCR02250
TCR02260
TCR02270
TCR02280
TCR02290
TCR02300
TCR02310
TCR02320
TCR02330
TCR02340
TCR02350
TCR02360
TCR02370
TCR02380
TCR02390
TCR02400

```

10 CONTINUE
RETURN
END
*
C*****
SUBROUTINE PCORDOWN(P,IP,IINI,IEND)
C Correct pressure data using the polynomial
C obtained in a laboratory
C DP: Correction, DP = CDN(1) + CDN(2)* P + CDN(3)*P**2
C +CDN(4)* P**3 + CDN(5)* P**4 + CDN(6)*P**5
C +CDN(7)* P**6
C REAL*8 P(IP),CDN(7),DP
C SELECT ONE OF THE FOLLOWINGS
C 1. FOR CTD WITHOUT O2 SENSOR
C DATA CDN/ 0.356999D+01, -0.681617D-02, -0.127216D-05,
C * 0.331650D-08, -0.118204D-11, 0.170710D-15,
C * -0.905423D-20/
C 2. FOR CTD EQUIPPED WITH O2 SENSOR
C DATA CDN/ -0.269719D+01, 0.699556D-02, -0.455414D-05,
C * 0.154059D-08, -0.289204D-12, 0.287633D-16,
C * -0.116484D-20/
*
DO 10 I=IINI,IEND
DP = CDN(1)+ P(I)*(CDN(2)+ P(I)*(CDN(3)+
+ P(I)*(CDN(4)+ P(I)*(CDN(5)+ P(I)*(CDN(6)+
+ P(I)* CDN(7) )))))
P(I) = P(I) + DP
10 CONTINUE
RETURN
END
*
C*****
SUBROUTINE TEMPFILT(T,P,IP,II,IE)
C Delay T response using digital filter for change of
C C response due to speed change of CTD fish
*
REAL*8 T(IP),P(IP),TINT,SPD,SP
DIMENSION IFT(16)
COMMON /OBSDAT/ NSEC
COMMON /TFILTR/ IUP,IDN
*
C IFT for CTD Neil Brown Mark III:
C DATA IFT/97,84,74,65,59,53,48,44,40,36,33,
C * 30,26,23,20,18/
*
C Estimate downward speed of CTD fish
C using P(I) between I-IUP and I+IDN
IINI=II + IUP
IEND=IE - IDN
TINT=DFLOAT(IUP+IDN)/DFLOAT(NSEC)
DO 10 I=IINI,IEND
C db/s*100 ---> cm/s
SPD=(P(I+IDN)-P(I-IUP))/TINT*0.1D+3
SP=DABS(SPD)
IS10=INT(SP)/10
ISP=IS10*10
IF (ISP.GE.150) THEN
IFT1=IFT(16)
IS=16
ELSE
IS=IS10+1
IFT1=(IFT(IS+1)-IFT(IS))*(SP-ISP)/10.+IFT(IS)
END IF
IFT2=100-IFT1
T(I)=(DFLOAT(IFT1)*T(I-1)+DFLOAT(IFT2)*T(I))/0.1D+3
10 CONTINUE
*
DO 1 I=II,IINI-1
TINT=DFLOAT(I+IDN-II)/DFLOAT(NSEC)
SPD=(P(I+IDN)-P(II))/TINT*0.1D+3
SP=DABS(SPD)
IS10=INT(SP)/10
ISP=IS10*10
IF (ISP.GE.150) THEN
IFT1=IFT(16)
ELSE
IS=IS10+1
IFT1=(IFT(IS+1)-IFT(IS))*(SP-ISP)/10.+IFT(IS)
END IF
IFT2=100-IFT1
TCR02410
TCR02420
TCR02430
TCR02440
TCR02450
TCR02460
TCR02470
TCR02480
TCR02490
TCR02500
TCR02510
TCR02520
TCR02530
TCR02540
TCR02550
TCR02560
TCR02570
TCR02580
TCR02590
TCR02600
TCR02610
TCR02620
TCR02630
TCR02640
TCR02650
TCR02660
TCR02670
TCR02680
TCR02690
TCR02700
TCR02710
TCR02720
TCR02730
TCR02740
TCR02750
TCR02760
TCR02770
TCR02780
TCR02790
TCR02800
TCR02810
TCR02820
TCR02830
TCR02840
TCR02850
TCR02860
TCR02870
TCR02880
TCR02890
TCR02900
TCR02910
TCR02920
TCR02930
TCR02940
TCR02950
TCR02960
TCR02970
TCR02980
TCR02990
TCR03000
TCR03010
TCR03020
TCR03030
TCR03040
TCR03050
TCR03060
TCR03070
TCR03080
TCR03090
TCR03100
TCR03110
TCR03120
TCR03130
TCR03140
TCR03150
TCR03160
TCR03170
TCR03180
TCR03190
TCR03200

```



```

      T(I)=(DFLOAT(IFT1)*T(I-1)+DFLOAT(IFT2)*T(I))/0.1D+3
1  CONTINUE
      *
      DO 2 I=IEND+1,IE
      TINT=DFLOAT(IE-I+IUP)/DFLOAT(NSEC)
      SPD=(P(IE)-P(I-IUP))/TINT*0.1D+3
      SP=DABS(SPD)
      IS10=INT(SP)/10
      ISP=IS10*10
      IF(ISP.GE.150) THEN
          IFT1=IFT(16)
      ELSE
          IS=IS10+1
          IFT1=(IFT(IS+1)-IFT(IS))*(SP-ISP)/10.+IFT(IS)
      END IF
      IFT2=100-IFT1
      T(I)=(DFLOAT(IFT1)*T(I-1)+DFLOAT(IFT2)*T(I))/0.1D+3
2  CONTINUE
      RETURN
      END
      *
      C*****
      SUBROUTINE CELLCORR(C,P,T,IP,IINI,IEND)
      C Conductivity correction for the cell deformation due to T and P
      REAL*8      C(IP),P(IP),T(IP)
      REAL*8      ALPHA,BETA,T0,P0
      *
      ALPHA=-6.5D-6
      BETA = 1.5D-8
      T0 = 1.5D+1
      P0 = 0.0D+0
      *
      DO 10 I=IINI,IEND
      C(I)=C(I)*(1.0D+0 + ALPHA*(T(I)-T0) + BETA*(P(I)-P0))
10  CONTINUE
      RETURN
      END
      *
      C*****
      SUBROUTINE FILTCOEF(P,C,T,S,PA,CA,IP,DS,ISTOP,
      *                   PMIN,PMAX,PRMIN,IL,NDS,NUM,I1,IEND)
      C Compute lag coefficient used in SUB. PCFILTER
      *
      REAL*8      P(IP),C(IP),T(IP),S(IP),PA(NDS),CA(NDS)
      REAL*8      DS(IL,NDS),FP1,FP2,FPS,DFP
      REAL*8      PW(1),CW(1),TW(1),SW(1),SSUM
      COMMON /FILCO/ IFIL
      COMMON /COMFPS/ FPS,DFP
      DIMENSION PMIN(IL),PMAX(IL),ISTOP(IL),PRMIN(IL)
      *
      IINI=I1 + IFIL
      DO 20 N=1,NDS
      FP1=FPS+DFLOAT(N-1)*DFP
      FP2=1.0D+0 - FP1
      PW(1)=PA(N)
      CW(1)=CA(N)
      DO 10 J=I1,IEND
      PW(1)=FP1*PW(1) + FP2*P(J)
      CW(1)=FP1*CW(1) + FP2*C(J)
      TW(1)=T(J)
      CALL SALCOMP(PW,TW,CW,SW,1,1,1)
      S(J)=SW(1)
      IF(J.EQ.IEND-IFIL) THEN
          PA(N)=PW(1)
          CA(N)=CW(1)
      END IF
10  CONTINUE
      SSUM=0.0D0
      DO 70 I=IINI,IEND
      SSUM=SSUM + (S(I)-S(I-IFIL))*(S(I)-S(I-IFIL))
70  CONTINUE
      DO 40 K=1,IL
      IF(P(IINI).GT.PMAX(K)) GO TO 40
      IF(P(IEND).LT.PMIN(K)) GO TO 40
      IF(NUM.EQ.1) THEN
          IF(P(IINI).GT.PRMIN(K)) PRMIN(K)=P(IINI)
      END IF
      IF(P(IINI).GT.PMIN(K).AND.P(IEND).LT.PMAX(K)) THEN
          DS(K,N)=DS(K,N) + SSUM
      ELSE

```

TCR03210
TCR03220
TCR03230
TCR03240
TCR03250
TCR03260
TCR03270
TCR03280
TCR03290
TCR03300
TCR03310
TCR03320
TCR03330
TCR03340
TCR03350
TCR03360
TCR03370
TCR03380
TCR03390
TCR03400
TCR03410
TCR03420
TCR03430
TCR03440
TCR03450
TCR03460
TCR03470
TCR03480
TCR03490
TCR03500
TCR03510
TCR03520
TCR03530
TCR03540
TCR03550
TCR03560
TCR03570
TCR03580
TCR03590
TCR03600
TCR03610
TCR03620
TCR03630
TCR03640
TCR03650
TCR03660
TCR03670
TCR03680
TCR03690
TCR03700
TCR03710
TCR03720
TCR03730
TCR03740
TCR03750
TCR03760
TCR03770
TCR03780
TCR03790
TCR03800
TCR03810
TCR03820
TCR03830
TCR03840
TCR03850
TCR03860
TCR03870
TCR03880
TCR03890
TCR03900
TCR03910
TCR03920
TCR03930
TCR03940
TCR03950
TCR03960
TCR03970
TCR03980
TCR03990
TCR04000

```

60          DO 30 I=IINI, IEND                                TCR04010
              IF(P(I).LT.PMIN(K)) GO TO 30                    TCR04020
              IF(P(I).GT.PMAX(K)) THEN                        TCR04030
                  ISTOP(K)=1                                  TCR04040
                  GO TO 40                                     TCR04050
              END IF                                          TCR04060
              DS(K,N)=DS(K,N) + (S(I)-S(I-IFIL))*(S(I)-S(I-IFIL)) TCR04070
30          CONTINUE                                          TCR04080
              END IF                                          TCR04090
40          CONTINUE                                          TCR04100
20          CONTINUE                                          TCR04110
              RETURN                                          TCR04120
              END                                              TCR04130
*                                                            TCR04140
C*****                                                    TCR04150
          SUBROUTINE SALCOMP(P,T,C,S,IP,IINI,IEND)           TCR04160
C          Compute salinity from P, T and C                  TCR04170
*                                                            TCR04180
              REAL*8    P(IP),T(IP),C(IP),S(IP),SAL78,C35150,CND TCR04190
              EXTERNAL  SAL78                                  TCR04200
              PARAMETER (C35150=0.42914D+2)                  TCR04210
*                                                            TCR04220
              DO 10 I=IINI,IEND                                TCR04230
                  CND=C(I)/C35150                            TCR04240
                  S(I)=SAL78(CND,T(I),P(I),0)                TCR04250
10          CONTINUE                                          TCR04260
              RETURN                                          TCR04270
              END                                              TCR04280
*                                                            TCR04290
C*****                                                    TCR04300
C          SEPT. 28 1983                                       TCR04310
C          ADD TF(S,P) FREEZING PT.                            TCR04320
C          WHOI CTD GROUP DISK FILE SPEC=BLUE::CTDA:<CTDEV.PRSW>PHYPROPSW.FOR TCR04330
*                                                            TCR04340
C          TITLE: ALGORITHMS FOR OCEANOGRAPHIC COMPUTATIONS TCR04350
C          N FOFONOFF & R MILLARD                             TCR04360
*                                                            TCR04370
C          SAL78 FCN ***** MAR 28 1983 *****              TCR04380
              FUNCTION SAL78(CND,T,P,M)                       TCR04390
              IMPLICIT REAL*8 (A-H,O-Z)                       TCR04400
*                                                            TCR04410
C          THE CONDUCTIVITY RATIO (CND) = 1.0000000 FOR SALINITY = 35 PSS-78 TCR04420
C          TEMPERATURE = 15.0 DEG. CELSIUS , AND ATMOSPHERIC PRESSURE. TCR04430
*                                                            TCR04440
C          FUNCTION TO CONVERT CONDUCTIVITY RATIO TO SALINITY (M = 0) TCR04450
C          SALINITY TO CONDUCTIVITY RATIO (M = 1,CND BECOMES INPUT SALINITY) TCR04460
*                                                            TCR04470
C          REFERENCES: ALSO LOCATED IN UNESCO REPORT # 37 1981 TCR04480
C          PRACTICAL SALINITY SCALE 1978: E.L. LEWIS IEEE OCEAN ENG. JAN. 1980 TCR04490
*                                                            TCR04500
C          UNITS:                                              TCR04510
C          PRESSURE      P      DECIBARS                       TCR04520
C          TEMPERATURE  T      DEG CELSIUS (IPTS-68)          TCR04530
C          CONDUCTIVITY CND    RATIO (M=0)                    TCR04540
C          SALINITY     SAL78   (PSS-78) (M=0)                TCR04550
C          CHECKVALUES:                                        TCR04560
C          SAL78=1.888091 :CND= 40.0000,T=40 DEG C,P= 10000 DECIBARS: M= 1 TCR04570
C          SAL78=40.00000 :CND=1.888091,T=40 DEG C,P=10000 DECIBARS: M=0 TCR04580
*                                                            TCR04590
C          SAL78 RATIO: RETURNS ZERO FOR CONDUCTIVITY RATIO: < 0.0005 TCR04600
C          SAL78: RETURNS ZERO FOR SALINITY: < 0.02          TCR04610
*                                                            TCR04620
C          INTERNAL FUNCTIONS                                  TCR04630
*                                                            TCR04640
C          PRACTICAL SALINITY SCALE 1978 DEFINITION WITH TEMPERATURE CORRECTION TCR04650
C          XT=T-15.0 : XR=SQRT(RT)                             TCR04660
              SAL(XR,XT) =                                     TCR04670
              * (((((0.27081D+1*XR-0.70261D+1)*XR+0.140941D+2)*XR+0.253851D+2)*XR TCR04680
              X -0.1692D0)* XR+0.0080D0                       TCR04690
              X +(XT/(1.0D0+0.0162D0*XT))* ((((-0.144D-1*XR+ TCR04700
              X 0.636D-1)*XR-0.375D-1)*XR-0.66D-2)*XR-0.56D-2)*XR+0.5D-3) TCR04710
C          DSAL(XR,XT) FUNCTION FOR DERIVATIVE OF SAL(XR,XT) WITH XR. TCR04720
              DSAL(XR,XT) =                                     TCR04730
              * (((((0.135405D+2*XR-0.281044D+2)*XR+0.422823D+2)*XR+0.507702D+2)*XRTCR04740
              X -0.1692D0)+(XT/(1.0D0+0.162D-1*XT))* ((((-0.720D-1*XR+0.2544D0)*XRTCR04750
              X -0.1125D0)*XR-0.132D-1)*XR-0.56D-2)          TCR04760
C          FUNCTION RT35 : C(35,T,0)/C(35,15,0) VARIATION WITH TEMPERATURE TCR04770
C          WITH TEMPERATURE.                                  TCR04780
              RT35(XT) = (((1.0031D-9*XT-6.9698D-7)*XT+1.104259D-4)*XT TCR04790
              X + 2.00564D-2)*XT + 0.6766097D0              TCR04800

```

```

C POLYNOMIALS OF RP: C(S,T,P)/C(S,T,0) VARIATION WITH PRESSURE          TCR04810
C C(XP) POLYNOMIAL CORRESPONDS TO A1-A3 CONSTANTS: LEWIS 1980          TCR04820
  C(XP) = ((3.989D-15*XP-6.370D-10)*XP+2.070D-5)*XP                    TCR04830
  B(XT) = (4.464D-4*XT+3.426D-2)*XT + 1.0D0                          TCR04840
C A(XT) POLYNOMIAL CORRESPONDS TO B3 AND B4 CONSTANTS: LEWIS 1980    TCR04850
  A(XT) = -3.107D-3*XT + 0.4215D0                                     TCR04860
*                                                                           TCR04870
C ZERO SALINITY/CONDUCTIVITY TRAP                                     TCR04880
  SAL78=0.0D0                                                         TCR04890
  IF((M.EQ.0).AND.(CND.LE.5D-4)) RETURN                               TCR04900
  IF((M.EQ.1).AND.(CND.LE.0.2D-1)) RETURN                             TCR04910
*                                                                           TCR04920
  DT = T - 0.15D+2                                                    TCR04930
C SELECT BRANCH FOR SALINITY (M=0) OR CONDUCTIVITY (M=1)            TCR04940
  IF(M.EQ.1) GO TO 10                                                  TCR04950
*                                                                           TCR04960
C CONVERT CONDUCTIVITY TO SALINITY                                    TCR04970
  R = CND                                                             TCR04980
  RT = R/(RT35(T)*(1.0D0 + C(P)/(B(T) + A(T)*R)))                    TCR04990
  RT = DSQRT(DABS(RT))                                                TCR05000
  SAL78 = SAL(RT,DT)                                                  TCR05010
  RETURN                                                              TCR05020
C ***** END OF CONDUCTIVITY TO SALINITY SECTION *****          TCR05030
*                                                                           TCR05040
C INVERT SALINITY TO CONDUCTIVITY BY THE                             TCR05050
C NEWTON-RAPHSON ITERATIVE METHOD.                                    TCR05060
*                                                                           TCR05070
C FIRST APPROXIMATION                                               TCR05080
  10 RT = DSQRT(CND/0.35D+2)                                           TCR05090
  SI = SAL(RT,DT)                                                     TCR05100
  N = 0                                                                TCR05110
C ITERATION LOOP BEGINS HERE WITH A MAXIMUM OF 10 CYCLES          TCR05120
  15 RT = RT + (CND - SI)/DSAL(RT,DT)                                   TCR05130
  SI = SAL(RT,DT)                                                     TCR05140
  N = N + 1                                                            TCR05150
  DELS = DABS(SI - CND)                                               TCR05160
  IF((DELS.GT.1.0D-4).AND.(N.LT.10))GO TO 15                         TCR05170
C *****END OF ITERATION LOOP *****                               TCR05180
*                                                                           TCR05190
C COMPUTE CONDUCTIVITY RATIO                                         TCR05200
  RTT = RT35(T)*RT*RT                                                 TCR05210
  AT = A(T)                                                            TCR05220
  BT = B(T)                                                            TCR05230
  CP = C(P)                                                            TCR05240
  CP = RTT*(CP + BT)                                                  TCR05250
  BT = BT - RTT*AT                                                    TCR05260
*                                                                           TCR05270
C SOLVE QUADRATIC EQUATION FOR R: R=RT35*RT*(1+C/AR+B)            TCR05280
  R = DSQRT(DABS(BT*BT + 0.4D+1*AT*CP)) - BT                          TCR05290
C CONDUCTIVITY RETURN                                              TCR05300
  SAL78 = 0.5D0*R/AT                                                  TCR05310
  RETURN                                                              TCR05320
  END                                                                  TCR05330

```

[注1]

生データセットには通常の単位による数字で入力しなかったため、TCRESPON.FORとCTDCALIB.FORでは、データを読んだからさらに、2で割ったり（水温と酸素電流値、レグ2の酸素温度値）、128を掛けたり（レグ1の酸素温度値）している。

[注2]

CTDCALIB.FORでの水温は、IPTS-68によるものであり、ITS-90への変換はしていない。

[注3]

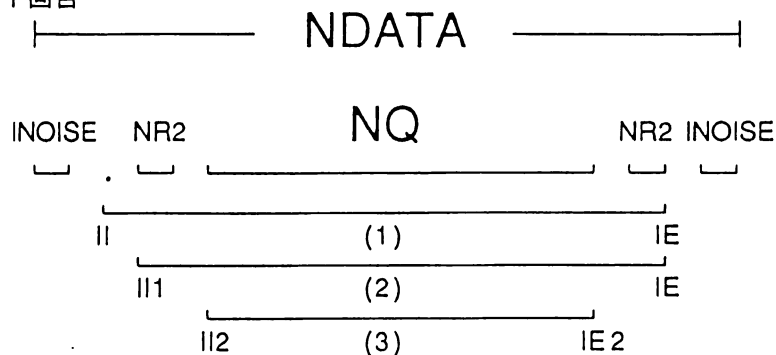
KH-91-5次航海では、予備として酸素センサーの付いていないCTDを持っていった。その圧力・水温センサーの補正係数もプログラムに入れてあるが、実際には使っていない。

2. CTDの連続データを較正し、1db間隔のデータセットにするプログラム
(CTDCALIB.FOR) [A2→A3→A4→A5→A7→A8→A9→A10]

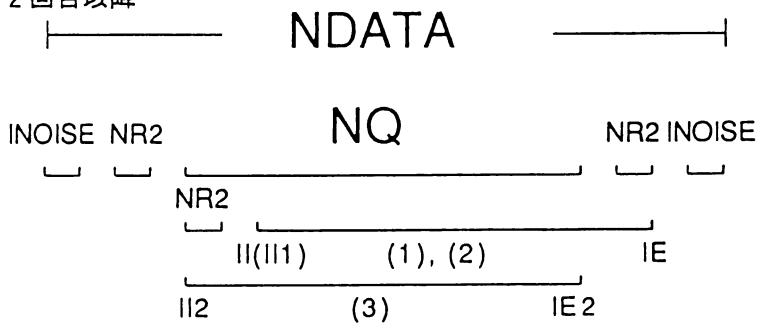
CTDの較正データとともに、酸素データの較正に用いるデータセット(CO1DO2.DATA)を作成する。酸素データは、1秒に1個しかとられていないので、これをひろいでいる。

TCRESPON.FORと同様に、NQ個ずつ読んで処理するようにした。各サブルーチンでの処理に使うデータ数やデータの範囲は以下の通りである。なお、最終的に得られるデータの個数は、LDATAである。

1回目



2回目以降



- (1) は、SUBROUTINE NOISE, TCORRECT, PCORDOWN, TEMPFILT, CELLCORR
- (2) は、SUBROUTINE PCFILTER
- (3) は、SUBROUTINE RESAMPLE で計算に使う範囲。

```

C*****
C Program to calibrate CTD data and to create a dataset *
C of P, T, C, S with 1-db interval (June, 1992) *
C*****
C Files
C INPUT FT10F001 C01.DATA (LRECL=29)
C OUTPUT FT20F001 C01CALIB.DATA (LRECL=40)
C OUTPUT FT30F001 C01DO2.DATA (LRECL=50)
*
PARAMETER (NQ=20000,IP=NQ+80,IR=6500)
REAL*8 P(IP),T(IP),C(IP),OC(IP),OT(IP)
REAL*8 PR(IR),TR(IR),CR(IR),SR(IR)
REAL*8 PRA,TRA,CRA
REAL*8 PP/0.0D+0/
REAL*8 PTH/0.1D+1/,TTH/0.1D+0/,CTH /0.2D-1/
REAL*8 FP/0.90D0/
C CO: Coefficients of (NP-1)-th order polynomial of P for cell factor
C NP=4 for cubic Eq., NP=3 for quadratic Eq.
PARAMETER (NP=6)
REAL*8 CO(NP)
*
C For leg 1 (KH-91-5)
C PARAMETER (LEG=1)
C DATA CO/ 0.1000054D+1, 0.5069215D-6, -0.1927917D-9,
C * 0.2654620D-13,-0.4903089D-18,-0.9984428D-22/
C For leg 2 (KH-91-5)
PARAMETER (LEG=2)
DATA CO/ 0.9999086D+0, 0.6578022D-6, -0.3865528D-9,
* 0.1120873D-12,-0.1563393D-16, 0.8308362D-21/
*
CHARACTER*12 FRM10/'(F5.1,4F6.3)'/
CHARACTER*12 FRM20/'(F8.1,3F8.4)'/
CHARACTER*1 DUM
CHARACTER*5 STA
*
C NSEC: Observed data number per second
C INOISE: Used in SUB. NOISE
C IUP, IDN: Used in SUB. TEMPFILT
C to estimate the falling speed of CTD fish
C NRUN: Running-mean interval
C INTDEP: Data interval (db) in the final dataset
C NRUN, INTDEP: Used in SUB. RESAMPLE
COMMON /OBSDAT/ NSEC
COMMON /NOISIE/ INOISE
COMMON /TFILTR/ IUP,IDN
COMMON /RESAMP/ NRUN,NR2,INTDEP
NSEC=32
NRUN=NSEC
NR2=NRUN/2
NRUN=NR2*2
INOISE=2
IUP=NSEC/2
IDN=NSEC/2
INTDEP=1
*
C Find the beginning of valid data observed in the sea
IINIT=0
JC=0
J=0
*
LOOP BEGIN (READ RAW DATA)
READ(10,'(A1)') DUM
5 READ(10,FRM10) PD,TD,CD,OCD,OTD
TD=TD*0.5
JC=JC+1
IF(PD.GT.20.0.AND.PD.LT.25.5.AND.
* TD.GT. 5.0.AND.TD.LT.35.0.AND.
* CD.GT.30.0.AND.CD.LT.70.0) GO TO 1
IF(CD.LT.40.0) THEN
J=J+1
IF(J.GE.3) IINIT=JC
ELSE
IF(CD.LT.70.0) J=0
END IF
GO TO 5
*
END OF LOOP
1 REWIND 10
WRITE(6,*) 'IINIT=',IINIT
*
PRA=0.0D+0
TRA=0.0D+0

```

```

CRA=0.0D+0
NDR=0
II=INOISE + 1
II1=II + 1
II2=II1 + NR2
IPROC=INOISE*2+NRUN+1
*
READ(10, '(A5)') STA
WRITE(30, '(A5)') STA
IF(IINIT.GE.1) THEN
  DO 15 I=1, IINIT
    READ(10, '(A1)') DUM
15 CONTINUE
END IF
*
DO 20 I=1, IPROC
  READ(10, FRM10) P(I), T(I), C(I), OC(I), OT(I)
  T(I)=T(I)*0.5D+0
  OC(I)=OC(I)*0.5D+0
  IF(LEG.EQ.1) THEN
    OT(I)=OT(I)*0.128D+3
  ELSE
    OT(I)=OT(I)*0.5D+0
  END IF
20 CONTINUE
JEND=0
NUM=0
25 NI=0
  NUM=NUM + 1
  DO 30 I=1, NQ
    IS=I + IPROC
    READ(10, FRM10, END=35) P(IS), T(IS), C(IS), OC(IS), OT(IS)
    T(IS)=T(IS)*0.5D+0
    OC(IS)=OC(IS)*0.5D+0
    IF(LEG.EQ.1) THEN
      OT(IS)=OT(IS)*0.128D+3
    ELSE
      OT(IS)=OT(IS)*0.5D+0
    END IF
    NI=I
30 CONTINUE
GO TO 40
35 JEND=1
IF(NI.EQ.0) GO TO 900
40 CONTINUE
NDATA=IPROC + NI
IE=NDATA - INOISE
IE2=IE-NR2
C Process A2
WRITE(6, *) 'A2: CHECKING NOISE ', NDATA
CALL NOISE(P, IP, PTH, IFLGP, II, IE)
CALL NOISE(T, IP, TTH, IFLGT, II, IE)
CALL NOISE(C, IP, CTH, IFLGC, II, IE)
WRITE(6, *) 'NOISE CHECK COMPLETED', IFLGP, IFLGT, IFLGC
C Process A3
C WRITE(6, *) 'A3: T AND P CORRECTION'
CALL TCORRECT(T, IP, II, IE)
CALL PCORDOWN(P, IP, II, IE)
C Process A4
C WRITE(6, *) 'A4: FILTERING TEMP DATA'
CALL TEMPFILT(T, P, IP, II, IE)
C Process A5
C WRITE(6, *) 'A5: CORRECT C FOR CELL DEFORMATION'
CALL CELLCORR(C, P, T, IP, II, IE)
C Process A7
C WRITE(6, *) 'A7: FILTERING P, C DATA'
CALL PCFILTER(P, C, IP, FP, II, IE)
C Process A8
C WRITE(*, *) 'A8: RUNNING MEAN AND RESAMPLING'
CALL RESAMPLE(P, T, C, IP, PR, TR, CR, IR,
* PRA, TRA, CRA, NDR, PP, II2, IE2)
*
C Make a file for DO2 correction
C WRITE(6, *) 'MAKING A FILE FOR DO2 CORRECTION'
CALL DOFILE(P, T, C, OC, OT, IP, NUM, NN, II2, IE2, 30, CO, NP)
*
IPROC=INOISE*2 + NRUN
DO 60 I=1, IPROC
  IS=NDATA - IPROC + I
  P(I)=P(IS)

```

```

CTD00810
CTD00820
CTD00830
CTD00840
CTD00850
CTD00860
CTD00870
CTD00880
CTD00890
CTD00900
CTD00910
CTD00920
CTD00930
CTD00940
CTD00950
CTD00960
CTD00970
CTD00980
CTD00990
CTD01000
CTD01010
CTD01020
CTD01030
CTD01040
CTD01050
CTD01060
CTD01070
CTD01080
CTD01090
CTD01100
CTD01110
CTD01120
CTD01130
CTD01140
CTD01150
CTD01160
CTD01170
CTD01180
CTD01190
CTD01200
CTD01210
CTD01220
CTD01230
CTD01240
CTD01250
CTD01260
CTD01270
CTD01280
CTD01290
CTD01300
CTD01310
CTD01320
CTD01330
CTD01340
CTD01350
CTD01360
CTD01370
CTD01380
CTD01390
CTD01400
CTD01410
CTD01420
CTD01430
CTD01440
CTD01450
CTD01460
CTD01470
CTD01480
CTD01490
CTD01500
CTD01510
CTD01520
CTD01530
CTD01540
CTD01550
CTD01560
CTD01570
CTD01580
CTD01590
CTD01600

```

```

        T(I)=T(IS)
        C(I)=C(IS)
        OC(I)=OC(IS)
        OT(I)=OT(IS)
50 CONTINUE
    II=INOISE+NRUN+1
    II1=II
    II2=II1 - NR2
    IF(JEND.EQ.0) GO TO 25
*
900 CONTINUE
    LDATA=NDR-1
*
C Process A9
C WRITE(*,*) 'A9: CORRECTING C DATA'
C CALL CCORRECT(CR, PR, IR, CO, NP, 1, LDATA)
*
C Process A10
C WRITE(*,*) 'A10: COMPUTING SALINITY'
C CALL SALCOMP(PR, TR, CR, SR, IR, 1, LDATA)
*
C Create final dataset
C WRITE(*,*) 'CREATING THE FINAL DATASET'
C WRITE(20, '(A5)') STA
C DO 90 I=1, LDATA
C   WRITE(20, FRM20) PR(I), TR(I), CR(I), SR(I)
90 CONTINUE
*
1000 STOP
    END
*
C *****
C SUBROUTINE NOISE(X, IP, XTH, IFLG, IINI, IEND)
C Remove noise of raw conductivity data
C Criterion: Distinguish as noise when
C X(I) > XTH + MAX(X(I-N), X(I+N))
C X(I) < -XTH + MIN(X(I-N), X(I+N))
C --> Revise to X(I) = (X(I-N)+X(I+N))/2
C The number of revised data will be set in IFLG
*
    REAL*8 X(IP), XTH, XMAX, XMIN
    COMMON /NOISIE/ N
    IFLG=0
*
    DO 10 I=IINI, IEND
        XMAX=MAX(X(I-N), X(I+N))
        XMIN=MIN(X(I-N), X(I+N))
        IF(X(I).GT.XMAX+XTH .OR. X(I).LT.XMIN-XTH) THEN
            X(I) = (X(I-N)+X(I+N))*0.5D0
            IFLG = IFLG+1
        ENDIF
10 CONTINUE
    RETURN
    END
*
C *****
C SUBROUTINE TCORRECT(T, IP, IINI, IEND)
C Correct temperature data using the quadratic polynomial
C obtained in a laboratory
C DT: Correction, DT = A * T**2 + B * T + C
*
    REAL*8 T(IP), DT, A(2), B(2), C(2)
C-----FOR THE CTD WITHOUT O2 SENSOR
C BELOW "TEMP", ABOVE "TEMP"
C DATA A/-0.100093D-4, 0.677039D-6/
C DATA B/-0.566937D-3, -0.721866D-3/
C DATA C/ 0.166095D-1, 0.170540D-1/
C DATA TEMP/10.0/
C-----FOR THE CTD EQUIPPED WITH O2 SENSOR
C BELOW "TEMP", ABOVE "TEMP"
C DATA A/ 0.121937D-4, 0.179550D-4/
C DATA B/-0.803083D-3, -0.127571D-2/
C DATA C/ 0.251885D-1, 0.290126D-1/
C DATA TEMP/ 9.1/
*
    DO 10 I=IINI, IEND
        K=1
        IF(T(I).LT.TEMP) K=2
        DT = A(K)*T(I)**2 + B(K)*T(I) + C(K)
        T(I) = T(I) + DT

```

```

10 CONTINUE
RETURN
END
*
C*****
SUBROUTINE PCORDOWN(P,IP,IINI,IEND)
C Correct pressure data using the polynomial
C obtained in a laboratory
C DP: Correction, DP = CDN(1) + CDN(2)* P + CDN(3)*P**2
C +CDN(4)* P**3 + CDN(5)* P**4 + CDN(6)*P**5
C +CDN(7)* P**6
C REAL*8 P(IP),CDN(7),DP
C SELECT ONE OF THE FOLLOWINGS
C 1. FOR CTD WITHOUT O2 SENSOR
C DATA CDN/ 0.356999D+01,-0.681617D-02,-0.127216D-05,
C * 0.331650D-08,-0.118204D-11, 0.170710D-15,
C * -0.905423D-20/
C 2. FOR CTD EQUIPPED WITH O2 SENSOR
C DATA CDN/-0.269719D+01, 0.699556D-02,-0.455414D-05,
C * 0.154059D-08,-0.289204D-12, 0.287633D-16,
C * -0.116484D-20/
*
DO 10 I=IINI,IEND
DP = CDN(1)+ P(I)*(CDN(2)+ P(I)*(CDN(3)+
+ P(I)*(CDN(4)+ P(I)*(CDN(5)+ P(I)*(CDN(6)+
+ P(I)* CDN(7) )))))
P(I) = P(I) + DP
10 CONTINUE
RETURN
END
*
C*****
SUBROUTINE TEMPFLT(T,P,IP,II,IE)
C Delay T response using digital filter for change of
C C response due to speed change of CTD fish
*
REAL*8 T(IP),P(IP),TINT,SPD,SP
DIMENSION IFT(16)
COMMON /OBSDAT/ NSEC
COMMON /TFILTR/ IUP,IDN
*
C IFT for CTD Neil Brown Mark III:
C DATA IFT/97,84,74,65,59,53,48,44,40,36,33,
C * 30,26,23,20,18/
*
C Estimate downward speed of CTD fish
C using P(I) between I-IUP and I+IDN
IINI=II + IUP
IEND=IE - IDN
TINT=DFLOAT(IUP+IDN)/DFLOAT(NSEC)
DO 10 I=IINI,IEND
C db/s*100 ---> cm/s
SPD=(P(I+IDN)-P(I-IUP))/TINT*0.1D+3
SP=DABS(SPD)
IS10=INT(SP)/10
ISP=IS10*10
IF (ISP.GE.150) THEN
IFT1=IFT(16)
IS=16
ELSE
IS=IS10+1
IFT1=(IFT(IS+1)-IFT(IS))*(SP-ISP)/10.+IFT(IS)
END IF
IFT2=100-IFT1
T(I)=(DFLOAT(IFT1)*T(I-1)+DFLOAT(IFT2)*T(I))/0.1D+3
10 CONTINUE
*
DO 1 I=II,IINI-1
TINT=DFLOAT(I+IDN-II)/DFLOAT(NSEC)
SPD=(P(I+IDN)-P(II))/TINT*0.1D+3
SP=DABS(SPD)
IS10=INT(SP)/10
ISP=IS10*10
IF (ISP.GE.150) THEN
IFT1=IFT(16)
ELSE
IS=IS10+1
IFT1=(IFT(IS+1)-IFT(IS))*(SP-ISP)/10.+IFT(IS)
END IF
IFT2=100-IFT1

```



```

      T(I)=(DFLOAT(IFT1)*T(I-1)+DFLOAT(IFT2)*T(I))/0.1D+3
1  CONTINUE
*
      DO 2 I=IEND+1,IE
      TINT=DFLOAT(IE-I+IUP)/DFLOAT(NSEC)
      SPD=(P(IE)-P(I-IUP))/TINT*0.1D+3
      SP=DABS(SPD)
      IS10=INT(SP)/10
      ISP=IS10*10
      IF(ISP.GE.150) THEN
        IFT1=IFT(16)
      ELSE
        IS=IS10+1
        IFT1=(IFT(IS+1)-IFT(IS))*(SP-ISP)/10.+IFT(IS)
      END IF
      IFT2=100-IFT1
      T(I)=(DFLOAT(IFT1)*T(I-1)+DFLOAT(IFT2)*T(I))/0.1D+3
2  CONTINUE
      RETURN
      END
*
C*****
      SUBROUTINE CELLCORR(C,P,T,IP,IINI,IEND)
C Conductivity correction for the cell deformation due to T and P
      REAL*8      C(IP),P(IP),T(IP)
      REAL*8      ALPHA,BETA,T0,P0
*
      ALPHA=-6.5D-6
      BETA = 1.5D-8
      T0 = 1.5D+1
      P0 = 0.0D+0
*
      DO 10 I=IINI,IEND
        C(I)=C(I)*(1.0D+0 + ALPHA*(T(I)-T0) + BETA*(P(I)-P0))
10  CONTINUE
      RETURN
      END
*
C*****
      SUBROUTINE PCFILTER(P,C,IP,FP1,IINI,IEND)
C Delay P, C responses using digital filter
*
      REAL*8      P(IP),C(IP),FP1,FP2
*
      FP2=1.0D+0-FP1
      DO 10 I=IINI,IEND
        P(I)=(FP1*P(I-1) + FP2*P(I))
        C(I)=(FP1*C(I-1) + FP2*C(I))
10  CONTINUE
      RETURN
      END
*
C*****
      SUBROUTINE RESAMPLE(P,T,C,IP,PR,TR,CR,IR,
        PRA,TRA,CRA,PRB,TRB,CRB,PP,II,IE)
C Resample P, T and C every N db
*
      REAL*8      P(IP),T(IP),C(IP),PR(IR),TR(IR),CR(IR)
      REAL*8      PRA,TRA,CRA,PRB,TRB,CRB,PP,RATIO
      COMMON /RESAMP/ NRUN,NR2,N
*
C NDACHK=0: NDA IS NOT DEFINED, NDACHK=1: DEFINED
      NDACHK=0
*
      ND=II
      IF(NDR.EQ.0) THEN
        NDA=ND
        NDACHK=1
        CALL RMEAN(P,IP,NDA,NRUN,PRA,II,IE)
        INITP=PRA
        IF(MOD(INITP,N).EQ.0.AND.
          * DABS(PRA-DFLOAT(INITP)).LT.0.1D-7) THEN
          PR(1)=PRA
          CALL RMEAN(T,IP,NDA,NRUN,TRA,II,IE)
          CALL RMEAN(C,IP,NDA,NRUN,CRA,II,IE)
          TR(1)=TRA
          CR(1)=CRA
          NDR=2
          PP=DFLOAT(INITP+N)
          ND=ND+1
        END IF
      END IF

```

CTD03210
 CTD03220
 CTD03230
 CTD03240
 CTD03250
 CTD03260
 CTD03270
 CTD03280
 CTD03290
 CTD03300
 CTD03310
 CTD03320
 CTD03330
 CTD03340
 CTD03350
 CTD03360
 CTD03370
 CTD03380
 CTD03390
 CTD03400
 CTD03410
 CTD03420
 CTD03430
 CTD03440
 CTD03450
 CTD03460
 CTD03470
 CTD03480
 CTD03490
 CTD03500
 CTD03510
 CTD03520
 CTD03530
 CTD03540
 CTD03550
 CTD03560
 CTD03570
 CTD03580
 CTD03590
 CTD03600
 CTD03610
 CTD03620
 CTD03630
 CTD03640
 CTD03650
 CTD03660
 CTD03670
 CTD03680
 CTD03690
 CTD03700
 CTD03710
 CTD03720
 CTD03730
 CTD03740
 CTD03750
 CTD03760
 CTD03770
 CTD03780
 CTD03790
 CTD03800
 CTD03810
 CTD03820
 CTD03830
 CTD03840
 CTD03850
 CTD03860
 CTD03870
 CTD03880
 CTD03890
 CTD03900
 CTD03910
 CTD03920
 CTD03930
 CTD03940
 CTD03950
 CTD03960
 CTD03970
 CTD03980
 CTD03990
 CTD04000

```

        NDA=ND
        CALL RMEAN(P, IP, NDA, NRUN, PRA, II, IE)
    ELSE
        NDR=1
        PP=DFLOAT((INITP/N+1)*N)
    END IF
END IF
*
10 ND=ND+1
   IF (ND.GT.IE) GO TO 20
   NDB=ND
   CALL RMEAN(P, IP, NDB, NRUN, PRB, II, IE)
   IF (DABS(PRB-PP).LT.0.1D-7) THEN
       PR(NDR)=PRB
       CALL RMEAN(T, IP, ND, NRUN, TRB, II, IE)
       CALL RMEAN(C, IP, ND, NRUN, CRB, II, IE)
       TR(NDR)=TRB
       CR(NDR)=CRB
       NDR=NDR+1
       PP=PP+DFLOAT(N)
       ND=ND+1
       IF (ND.GT.IE) THEN
           PRA=PRB
           TRA=TRB
           CRA=CRB
           RETURN
       END IF
       NDA=ND
       NDACHK=1
       CALL RMEAN(P, IP, NDA, NRUN, PRA, II, IE)
       IF (PRA.GT.PP) THEN
           WRITE(6, *) 'PRA>PP', 'PRA=', PRA, 'PP=', PP
           STOP
       END IF
       GO TO 10
   END IF
   IF (PRB.LT.PP) THEN
       IF (PRB.GT.PRA) THEN
           PRA=PRB
           NDA=NDB
           NDACHK=1
       END IF
   ELSE
       RATIO=(PP-PRA)/(PRB-PRA)
       PR(NDR)=PP
       IF (NDACHK.NE.0) THEN
           CALL RMEAN(T, IP, NDA, NRUN, TRA, II, IE)
           CALL RMEAN(C, IP, NDA, NRUN, CRA, II, IE)
       END IF
       CALL RMEAN(T, IP, NDB, NRUN, TRB, II, IE)
       CALL RMEAN(C, IP, NDB, NRUN, CRB, II, IE)
       TR(NDR)=TRA+(TRB-TRA)*RATIO
       CR(NDR)=CRA+(CRB-CRA)*RATIO
       NDR=NDR+1
       PP=PP+DFLOAT(N)
       PRA=PRB
       NDA=NDB
       NDACHK=1
   END IF
   GO TO 10
20 IF (NDACHK.EQ.0) RETURN
   CALL RMEAN(T, IP, NDA, NRUN, TRA, II, IE)
   CALL RMEAN(C, IP, NDA, NRUN, CRA, II, IE)
   RETURN
END
*
C*****
SUBROUTINE RMEAN(X, IP, ND, NRUN, XRUN, II, IE)
C Compute a running mean for the NDth data of X(IP)
C XRUN: Output
C NRUN: Data number used for the calculation of running mean
*
      REAL*8      X(IP), XRUN, TOT
*
C Running mean
      NR2=NRUN/2
      NINI=ND-NR2
      NEND=ND+NR2
      IF (NINI.LT.II-NR2.OR.NEND.GT.IE+NR2) THEN
          WRITE(6, *) 'Invalid NRUN or ND', ' NRUN=', NRUN, ' ND=', ND

```

```

CTD04010
CTD04020
CTD04030
CTD04040
CTD04050
CTD04060
CTD04070
CTD04080
CTD04090
CTD04100
CTD04110
CTD04120
CTD04130
CTD04140
CTD04150
CTD04160
CTD04170
CTD04180
CTD04190
CTD04200
CTD04210
CTD04220
CTD04230
CTD04240
CTD04250
CTD04260
CTD04270
CTD04280
CTD04290
CTD04300
CTD04310
CTD04320
CTD04330
CTD04340
CTD04350
CTD04360
CTD04370
CTD04380
CTD04390
CTD04400
CTD04410
CTD04420
CTD04430
CTD04440
CTD04450
CTD04460
CTD04470
CTD04480
CTD04490
CTD04500
CTD04510
CTD04520
CTD04530
CTD04540
CTD04550
CTD04560
CTD04570
CTD04580
CTD04590
CTD04600
CTD04610
CTD04620
CTD04630
CTD04640
CTD04650
CTD04660
CTD04670
CTD04680
CTD04690
CTD04700
CTD04710
CTD04720
CTD04730
CTD04740
CTD04750
CTD04760
CTD04770
CTD04780
CTD04790
CTD04800

```

```

        STOP
        END IF
        TOT=(X(NINI)+X(NEND))*0.5D0
        DO 10 I=1, NRUN-1
10     TOT=TOT+X(NINI+I)
        XRUN=TOT/DFLOAT(NRUN)
        RETURN
        END
*
C *****
      SUBROUTINE DOFILE(P,T,C,OC,OT,IP,NUM,NN,II,IE,IFILE,CO,NP)
C   Create the file which will be used for the DO2 correction
      REAL*8   P(IP),T(IP),C(IP),OC(IP),OT(IP),PR,TR,CR,S
      REAL*8   SAL78,C35150,CND,CO(NP),CF
      EXTERNAL SAL78
      PARAMETER (C35150=0.42914D+2)
      COMMON /OBSDAT/ NSEC
      COMMON /RESAMP/ NRUN,NR2,INTDEP
*
      IF(NUM.EQ.1) THEN
        NN=II
1       NN=NN+1
        IF(OC(NN).EQ.OC(NN-1).AND.OT(NN).EQ.OT(NN-1)) GO TO 1
        IF(OC(NN).GT.0.2D+1.OR.OT(NN).GT.0.3D+2) GO TO 1
      ELSE
        NN=NN+II-1
      END IF
*
2     CALL RMEAN(P,IP,NN,NRUN,PR,II,IE)
      CALL RMEAN(T,IP,NN,NRUN,TR,II,IE)
      CALL RMEAN(C,IP,NN,NRUN,CR,II,IE)
      CF = CO(1)
      DO 10 J=2,NP
10     CF = CF + CO(J)*PR**(J-1)
      CONTINUE
      CR = CR*CF
      CND=CR/C35150
      S=SAL78(CND,TR,PR,0)
      WRITE(IFILE,100) PR,TR,CR,S,OC(NN),OT(NN)
100   FORMAT(F8.1,3F8.4,2F8.4)
      NN=NN+NSEC
      IF(NN.LE.IE) GO TO 2
      NN=NN-IE
      RETURN
      END
*
C *****
      SUBROUTINE CCORRECT(C,P,IP,CO,NP,IINI,IEND)
C   Multiply observed values of Cctd by the cell factor
*
      REAL*8   C(IP),P(IP),CO(NP),CF
*
      DO 10 I=IINI,IEND
        CF = CO(1)
        DO 20 J=2,NP
20     CF = CF + CO(J)*P(I)**(J-1)
      CONTINUE
      C(I) = C(I)*CF
10    CONTINUE
      RETURN
      END
*
C *****
      SUBROUTINE SALCOMP(P,T,C,S,IP,IINI,IEND)
C   Compute salinity from P, T and C
*
      REAL*8   P(IP),T(IP),C(IP),S(IP),SAL78,C35150,CND
      EXTERNAL SAL78
      PARAMETER (C35150=0.42914D+2)
*
      DO 10 I=IINI,IEND
        CND=C(I)/C35150
        S(I)=SAL78(CND,T(I),P(I),0)
10    CONTINUE
      RETURN
      END
*
C *****
C   SEPT. 23 1983
C   ADD TF(S,P) FREEZING PT.

```

```

C WHOI CTD GROUP DISK FILE SPEC=BLUE::CTDA:<CTDEV.PRSW>PHYPROPSW.FOR      CTD05610
*                                                                              CTD05620
C TITLE: ALGORITHMS FOR OCEANOGRAPHIC COMPUTATIONS                          CTD05630
C N FOFONOFF & R MILLARD                                                    CTD05640
*                                                                              CTD05650
C SAL78 FCN ***** MAR 28 1983 *****                                    CTD05660
  FUNCTION SAL78(CND,T,P,M)                                                  CTD05670
  IMPLICIT REAL*8 (A-H,O-Z)                                                 CTD05680
*                                                                              CTD05690
C THE CONDUCTIVITY RATIO (CND) = 1.0000000 FOR SALINITY = 35 PSS-78        CTD05700
C TEMPERATURE = 15.0 DEG. CELSIUS , AND ATMOSPHERIC PRESSURE.              CTD05710
*                                                                              CTD05720
C                                                                              CTD05730
C FUNCTION TO CONVERT CONDUCTIVITY RATIO TO SALINITY (M = 0)                CTD05740
C SALINITY TO CONDUCTIVITY RATIO (M = 1,CND BECOMES INPUT SALINITY)        CTD05750
*                                                                              CTD05760
C REFERENCES: ALSO LOCATED IN UNESCO REPORT # 37 1981                       CTD05770
C PRACTICAL SALINITY SCALE 1978: E.L. LEWIS IEEE OCEAN ENG. JAN. 1980     CTD05780
*                                                                              CTD05790
C UNITS:                                                                      CTD05800
C PRESSURE P DECIBARS                                                         CTD05810
C TEMPERATURE T DEG CELSIUS (IPTS-68)                                       CTD05820
C CONDUCTIVITY CND RATIO (M=0)                                               CTD05830
C SALINITY SAL78 (PSS-78) (M=0)                                             CTD05840
C CHECKVALUES:                                                                CTD05850
C SAL78=1.888091 :CND= 40.0000,T=40 DEG C,P= 10000 DECIBARS: M= 1CTD05860
C SAL78=40.00000 :CND=1.888091,T=40 DEG C,P=10000 DECIBARS: M=0 CTD05870
*                                                                              CTD05880
C SAL78 RATIO: RETURNS ZERO FOR CONDUCTIVITY RATIO: < 0.0005              CTD05890
C SAL78: RETURNS ZERO FOR SALINITY: < 0.02                                  CTD05900
*                                                                              CTD05910
C INTERNAL FUNCTIONS                                                          CTD05920
*                                                                              CTD05930
C PRACTICAL SALINITY SCALE 1978 DEFINITION WITH TEMPERATURE CORRECTION    CTD05940
C XT=T-15.0 : XR=SQRT(RT)                                                    CTD05950
  SAL(XR,XT) =                                                                CTD05960
  * (((0.27081D+1*XR-0.70261D+1)*XR+0.140941D+2)*XR+0.253851D+2)*XR      CTD05970
  X -0.1692D0)* XR+0.0080D0                                                  CTD05980
  X +(XT/(1.0D0+0.0162D0*XT))*(((((-0.144D-1*XR+                          CTD05990
  X 0.636D-1)*XR-0.375D-1)*XR-0.66D-2)*XR-0.56D-2)*XR+0.5D-3)          CTD06000
C DSAL(XR,XT) FUNCTION FOR DERIVATIVE OF SAL(XR,XT) WITH XR.              CTD06010
  DSAL(XR,XT) =                                                              CTD06020
  * (((0.135405D+2*XR-0.281044D+2)*XR+0.422823D+2)*XR+0.507702D+2)*XR   CTD06030
  X -0.1692D0)+(XT/(1.0D0+0.162D-1*XT))*(((((-0.720D-1*XR+0.2544D0)*XR   CTD06040
  X -0.1125D0)*XR-0.132D-1)*XR-0.56D-2)                                  CTD06050
C FUNCTION RT35 : C(35,T,0)/C(35.15,0) VARIATION WITH TEMPERATURE        CTD06060
C WITH TEMPERATURE.                                                         CTD06070
  RT35(XT) = (((1.0031D-9*XT-6.9698D-7)*XT+1.104259D-4)*XT              CTD06080
  X + 2.00564D-2)*XT + 0.6766097D0                                         CTD06090
C POLYNOMIALS OF RP: C(S,T,P)/C(S,T,0) VARIATION WITH PRESSURE            CTD06100
C C(XP) POLYNOMIAL CORRESPONDS TO A1-A3 CONSTANTS: LEWIS 1980             CTD06110
C C(XP) = ((3.989D-15*XP-6.370D-10)*XP+2.070D-5)*XP                       CTD06120
C B(XT) = (4.464D-4*XT+3.426D-2)*XT + 1.0D0                               CTD06130
C A(XT) POLYNOMIAL CORRESPONDS TO B3 AND B4 CONSTANTS: LEWIS 1980        CTD06140
C A(XT) = -3.107D-3*XT + 0.4215D0                                          CTD06150
*                                                                              CTD06160
C ZERO SALINITY/CONDUCTIVITY TRAP                                           CTD06170
  SAL78=0.0D0                                                                CTD06180
  IF((M.EQ.0).AND.(CND.LE.5D-4)) RETURN                                     CTD06190
  IF((M.EQ.1).AND.(CND.LE.0.2D-1)) RETURN                                  CTD06200
*                                                                              CTD06210
  DT = T - 0.15D+2                                                           CTD06220
C SELECT BRANCH FOR SALINITY (M=0) OR CONDUCTIVITY (M=1)                  CTD06230
  IF(M.EQ.1) GO TO 10                                                        CTD06240
*                                                                              CTD06250
C CONVERT CONDUCTIVITY TO SALINITY                                          CTD06260
  R = CND                                                                     CTD06270
  RT = R/(RT35(T)*(1.0D0 + C(P)/(B(T) + A(T)*R)))                          CTD06280
  RT = DSQRT(DABS(RT))                                                       CTD06290
  SAL78 = SAL(RT,DT)                                                         CTD06300
  RETURN                                                                      CTD06310
C ***** END OF CONDUCTIVITY TO SALINITY SECTION *****                  CTD06320
*                                                                              CTD06330
C INVERT SALINITY TO CONDUCTIVITY BY THE                                    CTD06340
C NEWTON-RAPHSON ITERATIVE METHOD.                                          CTD06350
*                                                                              CTD06360
C FIRST APPROXIMATION                                                       CTD06370
  10 RT = DSQRT(CND/0.35D+2)                                                 CTD06380
  SI = SAL(RT,DT)                                                           CTD06390
  N = 0                                                                      CTD06400

```

*	CTD06410
C ITERATION LOOP BEGINS HERE WITH A MAXIMUM OF 10 CYCLES	CTD06420
15 RT = RT + (CND - SI)/DSAL(RT,DT)	CTD06430
SI = SAL(RT,DT)	CTD06440
N = N + 1	CTD06450
DELS = DABS(SI - CND)	CTD06460
IF((DELS.GT.1.0D-4).AND.(N.LT.10))GO TO 15	CTD06470
C *****END OF ITERATION LOOP *****	CTD06480
*	CTD06490
C COMPUTE CONDUCTIVITY RATIO	CTD06500
RTT = RT35(T)*RT*RT	CTD06510
AT = A(T)	CTD06520
BT = B(T)	CTD06530
CP = C(P)	CTD06540
CP = RTT*(CP + BT)	CTD06550
BT = BT - RTT*AT	CTD06560
*	CTD06570
C SOLVE QUADRATIC EQUATION FOR R: R=RT35*RT*(1+C/AR+B)	CTD06580
*	CTD06590
R = DSQRT(DABS(BT*BT + 0.4D+1*AT*CP)) - BT	CTD06600
C CONDUCTIVITY RETURN	CTD06610
SAL78 = 0.5D0*R/AT	CTD06620
RETURN	CTD06630
END	CTD06640

3. 採水時のCTDデータおよび採水による塩分・溶存酸素値のデータセットを作成するプログラム (CTDROSET.FOR)

IQは採水の数。24層での採水と表面でのバケツ採水、船底取水口からの研究用海水の26サンプルを処理したので、IQ=26になっている。

なお、コメント文中のUMOLとは、 μmol のことである。

プログラムで使用するデータセットと作成するデータセットの例を測点C01について以下に示す。

(1) CTDSTLG1.DATA

WOCEの航海情報用データセット。圧力センサー値の補正に用いる最大圧力値をこのデータセットから読む。

EXPOCODE	WHP-ID	STNNBR	CASTNO	TYPE	DATE	GMT	CODE	LATITUDE	LONGITUDE	CODE	BOTDEP	WHEEL	MAXPRS
49HH915/1	P13	C01	1	ROS	081791	2140	BE	39 59.79N	165 00.29E	GPS	5451		
49HH915/1	P13	C01	1	ROS	081791	2321	BO	39 59.60N	165 01.12E	GPS	5473		5500
49HH915/1	P13	C01	1	ROS	081891	0104	EN	39 59.70N	165 02.09E	GPS	5477		
49HH915/1	P13	C02	1	ROS	081891	0503	BE	39 00.15N	165 00.04E	GPS	5350		
49HH915/1	P13	C02	1	ROS	081891	0637	BO	38 59.77N	165 00.44E	GPS	5372		5401
49HH915/1	P13	C02	1	ROS	081891	0807	EN	38 59.40N	165 01.19E	GPS	5375		
49HH915/1	P13	C03	1	ROS	081891	1201	BE	38 00.04N	164 59.98E	GPS	5441		
49HH915/1	P13	C03	1	ROS	081891	1342	BO	37 59.75N	165 01.49E	GPS	5438		5500
49HH915/1	P13	C03	1	ROS	081891	1524	EN	37 59.65N	165 02.43E	GPS	5417		

(2) CTDROSL1.DATA

採水時のCTDデータ。トリIGGER番号、圧力、水温、電気伝導度、酸素電流値、酸素温度値の順に並んでいる。

C01					
1	5502.8	1.561	32.251	0.0695	1.7
2	4999.3	1.506	32.046	0.0725	1.7
3	4500.6	1.473	31.851	0.0755	1.5
4	4003.2	1.465	31.674	0.0765	1.5
5	3503.5	1.483	31.507	0.0765	1.4
6	3004.5	1.556	31.376	0.0720	1.4
7	2501.6	1.697	31.287	0.0630	1.5
8	2002.9	1.967	31.281	0.0455	1.7
9	1803.2	2.090	31.286	0.0410	1.8
10	1602.7	2.239	31.303	0.0350	2.0
11	1403.1	2.453	31.379	0.0315	2.2
12	1203.1	2.712	31.480	0.0290	2.3
13	1003.1	3.053	31.647	0.0305	2.6
14	852.8	3.390	31.818	0.0355	2.9
15	702.7	3.850	32.088	0.0465	3.2
16	602.7	4.163	32.258	0.0620	3.4
17	503.0	4.518	32.453	0.0870	3.8
18	403.5	4.769	32.514	0.1330	4.1
19	302.6	5.929	33.448	0.2190	4.6
20	203.0	7.228	34.619	0.7735	5.4
21	103.6	8.117	35.281	0.3345	6.3
22	53.0	11.741	38.638	0.3820	7.4
23	23.2	17.087	43.713	0.4290	9.6
24	13.3	17.231	43.874	0.4470	11.4

(3) ROSETLG1.DATA

採水データ。ボトル番号, トリッガー番号, 塩検による電気伝導度比, 2本とった酸素用サンプルについてのビン番号, 滴定に使ったチオ硫酸ナトリウムの量と規定度が並んでいる。

トリッガー番号は、初めはボトル番号と同じであるが、何回目のトリッガーでボトルが開まったのかを判定して(プロセスB5)、トリッガー番号を修正する。測点C01の場合、19回目のトリッガーまでは順調に閉まっていたものの20回目では1本も開まらず、21回目で2本まとめて閉まったと判定した。

STN	BTL	TRI	CNDRTIO	OXYGEN1			OXYGEN2		
C01	1	1	0.99211	8888	1.1875	0.05089	091	1.3325	0.05089
C01	2	2	0.99209	064	1.3175	0.05089	508	1.2950	0.05089
C01	3	3	0.99201	281	1.2950	0.05089	493	1.2675	0.05089
C01	4	4	0.99189	359	1.2675	0.05089	155	1.2450	0.05089
C01	5	5	0.99169	454	1.0100	0.05089	093	1.1450	0.05089
C01	6	6	0.99132	274	1.0075	0.05089	074	1.0450	0.05089
C01	7	7	0.98937	246	0.8175	0.05089	239	0.8225	0.05089
C01	8	8	0.98934	262	0.5450	0.05089	458	0.5400	0.05089
C01	9	9	0.98875	309	0.4950	0.05089	109	0.4950	0.05089
C01	10	10	99.99999	9999	99.9999	0.05089	9999	99.9999	0.05089
C01	11	11	99.99999	9999	99.9999	0.05089	9999	99.9999	0.05089
C01	12	12	99.99999	9999	99.9999	0.05089	9999	99.9999	0.05089
C01	13	13	0.98423	385	0.3450	0.05089	486	0.3525	0.05089
C01	14	14	0.98239	108	0.3450	0.05089	115	0.3775	0.05089
C01	15	15	0.98027	094	0.4425	0.05089	507	0.4575	0.05089
C01	16	16	0.9785	114	0.5625	0.05089	446	0.5375	0.05089
C01	17	17	0.97622	198	0.7650	0.05089	144	0.7850	0.05089
C01	18	18	0.9728	221	1.1475	0.05089	138	1.1150	0.05089
C01	19	19	0.9715	258	1.6850	0.05089	120	1.6850	0.05089
C01	20	21	0.97047	100	2.3075	0.05089	098	2.2275	0.05089
C01	21	21	99.99999	9999	99.9999	0.05089	9999	99.9999	0.05089
C01	22	22	0.97184	071	2.2200	0.05089	491	2.1875	0.05089
C01	23	23	0.97192	503	2.0900	0.05089	088	2.0975	0.05089
C01	24	24	0.97219	350	2.1575	0.05089	361	2.0250	0.05089
C01	25		0.97056	099	2.1000	0.05089	252	2.0925	0.05089
C01	26		0.97108	PORTASAL TEMP=25.010					

(4) O2BINFAC.DATA

酸素ビンの番号とビン係数のデータセット。

1	52.41	2	52.59	3	53.60	4	54.57	5	54.91
6	52.89	7	53.43	8	52.47	9	53.49	10	53.93
11	50.00	12	54.65	13	55.37	14	53.00	15	55.73
16	53.31	17	54.94	18	57.60	19	51.94	20	56.95
21	54.65	22	53.03	23	52.57	24	52.34	25	53.59
26	52.20	27	52.37	28	53.94	29	52.51	30	51.97
31	53.08	32	55.81	33	52.19	34	52.83	35	53.61
36	55.50	37	52.59	38	53.74	39	53.34	40	53.53
41	51.79	42	54.83	43	52.75	44	56.21	45	54.75
46	52.83	47	56.08	48	53.97	49	52.26	50	55.51
51	55.72	52	54.57	53	55.49	54	53.45	55	55.34
56	54.75	57	54.22	58	52.59	59	54.84	60	57.43
61	55.15	62	54.65	63	54.62	64	54.10	65	54.29
66	53.22	67	55.09	68	55.73	69	56.54	70	52.91
71	54.81	72	56.07	73	55.32	74	53.01	75	55.49
76	52.72	77	54.52	78	54.47	79	54.78	80	51.76
81	52.44	82	54.29	83	52.79	84	52.16	85	52.65
86	55.37	87	53.21	88	54.42	89	53.82	90	57.20

計算結果。

[Step 1] 初めはセルフファクターも酸素センサーの補正係数（プログラムでの定数A, C, D, E, F）もわからず、ボトルがどこで閉まったかわからないので、セルフファクターには1、酸素センサー値には適当な補正係数、トリグラー番号はボトル番号と同じ値を使って計算し、そのデータを使ってボトルが実際に閉まった深さを判断し、ROSETLG1.DATAのTRIを修正する。

[Step 2] 修正したROSETLG1.DATAを使って計算し直し、その結果をCFACPOLY.FORを使って、圧力の多項式によるセルフファクターの近似式を算出する（もしくは、CSCCPOLY.FORで C_{cnd} の補正式を導く）。さらに、酸素センサーの補正式を適当な方法で求める。

[Step 3] 修正したROSETLG1.DATA、電気伝導度と酸素の補正式を使って計算し、最終結果を得る。

ここに示したのは最終結果である。

STN	BO	TR	CTDPRS DBAR	CTDTMP IPTS68	CTDTMP ITS-90	CTDCND MS/CM	CTDSAL PSS-78	CNDTVT MS/CM	SALNTY PSS-78	CTDOXY UMOL/KG	OXYGEN UMOL/KG	CTDOXY ML/L	OXYGEN ML/L	OXYGEN ML/L	
C01	1	1	5505.0	1.5881	1.5877	32.2724	34.6905	32.2716	34.6895	167.5	9999.9	159.9	3.857	99.999	3.681
C01	2	2	5000.9	1.5331	1.5328	32.0667	34.6892	32.0663	34.6887	164.9	157.6	157.2	3.795	3.627	3.620
C01	3	3	4501.7	1.5002	1.4998	31.8711	34.6834	31.8729	34.6856	161.8	154.5	153.3	3.726	3.556	3.530
C01	4	4	4003.9	1.4922	1.4918	31.6937	34.6810	31.6936	34.6809	154.4	147.3	147.1	3.554	3.392	3.386
C01	5	5	3503.8	1.5102	1.5098	31.5266	34.6730	31.5267	34.6730	145.1	139.7	138.0	3.341	3.216	3.177
C01	6	6	3004.6	1.5831	1.5827	31.3957	34.6591	31.3952	34.6585	128.0	121.4	122.5	2.947	2.795	2.819
C01	7	7	2501.6	1.7239	1.7235	31.3066	34.6346	31.2638	34.5818	104.5	98.3	98.8	2.406	2.264	2.274
C01	8	8	2002.8	1.9936	1.9931	31.3002	34.5806	31.3002	34.5806	69.9	66.1	66.7	1.608	1.522	1.536
C01	9	9	1803.0	2.1164	2.1159	31.3048	34.5542	31.3074	34.5574	61.0	56.6	58.0	1.404	1.304	1.335
C01	10	10	1602.5	2.2652	2.2647	31.3212	34.5164	99.9999	99.9999	50.3	9999.9	9999.9	1.158	99.999	99.999
C01	11	11	1402.8	2.4790	2.4784	31.3965	34.4822	99.9999	99.9999	43.7	9999.9	9999.9	1.005	99.999	99.999
C01	12	12	1202.7	2.7377	2.7370	31.4965	34.4319	99.9999	99.9999	38.7	9999.9	9999.9	0.891	99.999	99.999
C01	13	13	1002.5	3.0783	3.0775	31.6623	34.3756	31.6659	34.3799	39.1	43.3	42.3	0.901	0.995	0.973
C01	14	14	852.1	3.4149	3.4141	31.8322	34.3036	31.8356	34.3076	44.1	42.2	44.3	1.016	0.970	1.019
C01	15	15	701.8	3.8744	3.8734	32.1009	34.2215	32.1035	34.2244	55.8	53.5	55.4	1.284	1.232	1.275
C01	16	16	601.6	4.1870	4.1860	32.2699	34.1484	32.2756	34.1550	72.7	65.6	65.6	1.672	1.510	1.508
C01	11	17	501.8	4.5416	4.5405	32.4639	34.0615	32.4674	34.0656	99.4	92.2	93.0	2.287	2.122	2.139
C01	18	18	402.1	4.7923	4.7912	32.5237	33.9256	32.5287	33.9315	148.9	136.4	136.0	3.425	3.137	3.127
C01	19	19	300.9	5.9511	5.9497	33.4563	33.8724	33.4636	33.8806	229.7	206.4	205.2	5.283	4.747	4.718
C01	20	21	101.4	8.1368	8.1349	35.2862	33.8275	35.2981	33.8402	309.8	275.4	274.9	7.123	6.332	6.321
C01	21	21	101.4	8.1368	8.1349	35.2862	33.8275	99.9999	99.9999	309.8	9999.9	9999.9	7.123	99.999	99.999
C01	22	22	50.7	11.7584	11.7556	38.6418	33.8782	38.6579	33.8939	298.1	269.5	269.6	6.849	6.192	6.194
C01	23	23	20.8	17.1020	17.0979	43.7351	33.8963	43.7360	33.8970	264.0	253.0	253.1	6.059	5.806	5.809
C01	24	24	10.8	17.2460	17.2418	43.8759	33.9009	43.8837	33.9076	273.6	252.7	252.0	6.279	5.799	5.783
C01	25	0	0.0	99.9999	99.9999	99.9999	99.9999	99.9999	33.8437	9999.9	9999.9	9999.9	99.999	5.786	5.765
C01	26	0	0.0	99.9999	99.9999	99.9999	99.9999	99.9999	33.8641	9999.9	9999.9	9999.9	99.999	99.999	99.999


```

C*****
C Create the final dataset of bottle-sampling data *
C June, 1992 *
C*****
C Files
C INPUT FT10F001 CTDSTLG1.DATA (LRECL=130)
C INPUT FT20F001 CTDROSL1.DATA (LRECL=80)
C INPUT FT30F001 ROSETLG1.DATA (LRECL=60)
C INPUT FT40F001 O2BINFAC.DATA (LRECL=50)
C OUTPUT FT50F001 CTD RMSL1.DATA (LRECL=100)
*
C IQ: Number of water samples at each station
PARAMETER (IQ=26,JQ=13)
REAL*8 PLD(IQ,JQ),PMAx,SALTMP,CNC1(IQ),CNC2(IQ)
REAL*8 CTD(IQ,5),OCUR(IQ),OTMP(IQ)
INTEGER NBI(IQ),NB2(IQ),NBOT(IQ),NTRI(IQ),KTRI(IQ)
CHARACTER*10 FRM10/'(9IX,F4.0)'/
CHARACTER*25 FRM20/'(I4,F7.1,2F7.3,F7.4,F5.1)'/
CHARACTER*31 FRM30/'(A4,I3,I4,F9.5,2(I4,F7.4,F8.5))'/
CHARACTER*33 FRM50/'(A4,I2,I3,F7.1,6F8.4,3F6.1,3F6.3)'/
CHARACTER*4 STA,STN
CHARACTER*1 DUM
*
C CO: Coefficients of (NP-1)-th order polynomial of P for cell factor
C NP=4 for cubic Eq., NP=3 for quadratic Eq.
PARAMETER (NP=6)
REAL*8 CO(NP),A,C,D,E,F
*
C For leg 1 (KH-91-5)
DATA CO/ 0.1000054D+1, 0.5069215D-6, -0.1927917D-9,
* 0.2664620D-13,-0.4903089D-18,-0.9984428D-22/
DATA A,C,D,E,F/0.383540951D+1,-0.229237D-2,-0.2297186D-1,
* -0.4631460D-1,0.12202D-3/
C For leg 2 (KH-91-5)
DATA CO/ 0.9999086D+0, 0.6578022D-6, -0.3865528D-9,
* 0.1120873D-12,-0.1563393D-16, 0.8308362D-21/
DATA A,C,D,E,F/0.410260346D+1,0.422961D-2,-0.2553547D-1,
* 0.68299419D+0,0.11620D-3/
*
C PLD file
C PLD(I,J): Parameters at each sampling depth
C IQ, JQ : Size of array PLD
C I : Bottle Number
C J=1: Pressure, J=2: Temp.(IPTS-68), J=3: Temp.(ITS-90)
C J=4: Cctd, J=5: Sctd, J=6: Csample, J=7: Ssample
C J=8: O2ctd (UMOL/KG), J=9,10: O2sample (UMOL/KG)
C J=11: O2ctd (ML/L), J=12,13: O2sample (ML/L)
*
C CTD(I,J) file
C I : Trigger Number
C J : 1 Pressure, 2 Temperature, 3 Conductivity
C 4 O2 e.current, 5 O2 temperature
*
DO 60 I=IQ-1,IQ
PLD(I,1) =0.0D+0
PLD(I,2) =0.999999D+2
PLD(I,3) =0.999999D+2
PLD(I,4) =0.999999D+2
PLD(I,8) =0.99999D+4
PLD(I,11)=0.99999D+2
60 CONTINUE
PLD(IQ,9) =0.99999D+4
PLD(IQ,10)=0.99999D+4
PLD(IQ,12)=0.99999D+2
PLD(IQ,13)=0.99999D+2
*
READ(10,'(A1)') DUM
20 READ(10,'(17X,A4)',END=1000) STN
C PMAx: Maximum pressure in the CTD cast
READ(10,FRM10) PMAx
READ(10,'(A1)') DUM
*
READ(20,'(A4)') STA
IF(STA.NE.STN) THEN
WRITE(*,*) 'INCORRECT STATION ORDER IN FILE 20'
GO TO 1000
END IF
DO 10 I=1,IQ
READ(20,FRM20) NTRI(I),(CTD(I,JS),JS=1,5)
10 CONTINUE

```

```

      READ(20,'(A1)') DUM                                CTD00810
*
      READ(30,'(A1)') DUM                                CTD00820
C   CNC1, CNC2: Concentration of Na2S2O3 for oxygen titration CTD00830
  DO 40 I=1,IQ-1
    READ(30,FRM30) STA,NBOT(I),KTRI(I),PLD(I,7),NB1(I), CTD00840
    *   PLD(I,12),CNC1(I),NB2(I),PLD(I,13),CNC2(I) CTD00850
    *   IF(NBOT(I).NE.I) THEN CTD00860
      WRITE(*,*) 'INCORRECT BOTTLE NUMBER IN FILE 30' CTD00870
      GO TO 1000 CTD00880
    END IF CTD00890
  40 CONTINUE CTD00900
    IF(STA.NE.STN) THEN CTD00910
      WRITE(*,*) 'INCORRECT STATION ORDER IN FILE 30' CTD00920
      GO TO 1000 CTD00930
    END IF CTD00940
C   SALTMP: Temperature in the reservoir of salinometer CTD00950
  READ(30,'(4X,I3,I4,F9.5,16X,F6.3)') CTD00960
  *   NBOT(IQ),KTRI(IQ),PLD(IQ,7),SALTMP CTD00970
  READ(30,'(A1)') DUM CTD00980
*
C   Put CTD data taken at the bottle-closed depth into PLD CTD00990
  DO 70 I=1,IQ-2 CTD01000
    K=KTRI(I) CTD01010
    IF(K.EQ.0) THEN CTD01020
      PLD(I,1)=0.999999D+5 CTD01030
      PLD(I,2)=0.999999D+2 CTD01040
      PLD(I,4)=0.999999D+2 CTD01050
      OCUR(I)=0.99999D+1 CTD01060
      OTMP(I)=0.999D+2 CTD01070
    ELSE CTD01080
      PLD(I,1)=CTD(K,1) CTD01090
      PLD(I,2)=CTD(K,2) CTD01100
      PLD(I,4)=CTD(K,3) CTD01110
      OCUR(I)=CTD(K,4) CTD01120
      OTMP(I)=CTD(K,5) CTD01130
    END IF CTD01140
  70 CONTINUE CTD01150
*
C   Process B2 CTD01160
  CALL TCCORRECT(PLD(1,2),IQ,1,IQ-2) CTD01170
  CALL PCORREUP(IQ-2,PLD(1,1),IQ,PMAX) CTD01180
C   Process B3 CTD01190
  CALL CELLCORR(PLD(1,4),PLD(1,1),PLD(1,2),IQ,1,IQ-2) CTD01200
C   Process B4, B6 CTD01210
  CALL CCORRECT(PLD(1,4),PLD(1,1),IQ,CO,NP,1,IQ-2) CTD01220
  CALL CSALCOMP(IQ,PLD,IQ,JQ,SALTMP) CTD01230
  CALL CTDOXYGN(IQ-2,PLD(1,1),PLD(1,2),PLD(1,5) CTD01240
  *   ,PLD(1,11),OCUR,OTMP,IQ,A,C,D,E,F) CTD01250
  CALL SMPLOXGN(IQ-1,NB1,PLD(1,12),CNC1,IQ,40) CTD01260
  CALL SMPLOXGN(IQ-1,NB2,PLD(1,13),CNC2,IQ,40) CTD01270
  CALL OXYUNIT(PLD,IQ,JQ,1,IQ-1) CTD01280
*
C   Temperature unit o IPTS-68 kara ITS-90 ni suru CTD01290
  DO 50 I=1,IQ-2 CTD01300
    IF(PLD(I,2).GT.0.9D+2) THEN CTD01310
      PLD(I,3)=0.999999D+2 CTD01320
    ELSE CTD01330
      PLD(I,3)=PLD(I,2)*0.99976 CTD01340
    END IF CTD01350
  50 CONTINUE CTD01360
*
C   Create final dataset CTD01370
  WRITE(50,100) CTD01380
100 FORMAT('STN BO TR CTDPRS CTDTMP CTDTMP CTDCND CTDSAL ', CTD01390
  *   ' CNDTVT SALNTY CTDOXY OXYGEN CTDOXY OXYGEN') CTD01400
  WRITE(50,200) CTD01410
200 FORMAT(11X,'DBAR IPTS68 ITS-90 MS/CM PSS-78 MS/CM ', CTD01420
  *   ' PSS-78 UMOL/KG UMOL/KG ML/L ML/L ML/L') CTD01430
  DO 30 I=1,IQ CTD01440
    WRITE(50,FRM50) STA,NBOT(I),KTRI(I),(PLD(I,JS),JS=1,JQ) CTD01450
  30 CONTINUE CTD01460
*
  GO TO 20 CTD01470
*
1000 STOP CTD01480
  END CTD01490
*
C ***** CTD01500
  SUBROUTINE TCCORRECT(T,IQ,IINI,IEND) CTD01510
  CTD01520
  CTD01530
  CTD01540
  CTD01550
  CTD01560
  CTD01570
  CTD01580
  CTD01590
  CTD01600

```

```

C Correct temperature data using the quadratic Eq. CTD01610
C obtained in a laboratory CTD01620
C DT: Correction,  $DT = A * T^{*2} + B * T + C$  CTD01630
* CTD01640
REAL*8 T(IQ),DT,A(2),B(2),C(2) CTD01650
*-----FOR THE CTD WITHOUT O2 SENSOR CTD01660
* BELOW "TEMP", ABOVE "TEMP" CTD01670
C DATA A/-0.100093D-4, 0.677039D-6/ CTD01680
C DATA B/-0.566937D-3, -0.721866D-3/ CTD01690
C DATA C/ 0.166095D-1, 0.170540D-1/ CTD01700
C DATA TEMP/10.0/ CTD01710
*-----FOR THE CTD EQUIPPED WITH O2 SENSOR CTD01720
* BELOW "TEMP", ABOVE "TEMP" CTD01730
C DATA A/ 0.121937D-4, 0.179550D-4/ CTD01740
C DATA B/-0.803083D-3, -0.127571D-2/ CTD01750
C DATA C/ 0.251885D-1, 0.290126D-1/ CTD01760
C DATA TEMP/ 9.1/ CTD01770
* CTD01780
DO 10 I=IINI,IEND CTD01790
IF(T(I).GT.0.9D+2) GO TO 10 CTD01800
K=1 CTD01810
IF(T(I).LT.TEMP) K=2 CTD01820
DT = A(K)*T(I)**2 + B(K)*T(I) + C(K) CTD01830
T(I) = T(I) + DT CTD01840
10 CONTINUE CTD01850
RETURN CTD01860
END CTD01870
* CTD01880
C ***** CTD01890
SUBROUTINE PCORREUP(NDAT,P,IQ,PMAX) CTD01900
C Correct pressure data using fitted equations of CTD01910
C order NP-1 obtained from laboratory data CTD01920
C CDN(NP): Coefficients of the fitted equation for increasing pressure CTD01930
C CUP(J,NP): Similar coefficients for pressure decreasing from DEP(J) CTD01940
* CTD01950
PARAMETER (NP=7) CTD01960
REAL*8 P(IQ),PMAX,CDN(NP),CUP(4,NP),DEP(4) CTD01970
REAL*8 C,CO,C1,DELC,DELCO,DC,CA,CB,CAP,CBP,CGRAD,CRATIO CTD01980
C SELECT ONE OF THE FOLLOWINGS (CHANGE SUB.PCORRED2 ALSO) CTD01990
C 1. FOR CTD WITHOUT O2 SENSOR CTD02000
C DATA CDN/ 0.356999D+01,-0.681617D-02,-0.127216D-05, CTD02010
C * 0.331650D-08,-0.118204D-11, 0.170710D-15, CTD02020
C * -0.905423D-20/ CTD02030
C DATA CUP/ 0.241536D+1, 0.241695D+1, 0.268556D+1, 0.331024D+1, CTD02040
C * -0.286692D-2, -0.988938D-2, -0.122474D-1, -0.177953D-1, CTD02050
C * -0.748313D-4, -0.101950D-4, -0.374745D-5, 0.112576D-4, CTD02060
C * 0.282491D-6, 0.199215D-7, 0.122420D-7, -0.221939D-8, CTD02070
C * -0.480765D-9, -0.948127D-11,-0.643834D-11, 0.413944D-13, CTD02080
C * 0.398057D-12, 0.148217D-14, 0.139717D-14, 0.349808D-16, CTD02090
C * -0.127104D-15, 0.937150D-20,-0.111414D-18,-0.303500D-20/CTD02100
C DATA DEP/ 1.0D+3, 2.0D+3, 4.0D+3, 6.0D+3/ CTD02110
C 2. FOR CTD WITH O2 SENSOR EQUIPPED CTD02120
C DATA CDN/-0.269719D+01, 0.699556D-02,-0.455414D-05, CTD02130
C * 0.154059D-08,-0.289204D-12, 0.287633D-16, CTD02140
C * -0.116484D-20/ CTD02150
C DATA CUP/-0.260042D+1, -0.265158D+1, -0.275144D+1, -0.249635D+1, CTD02160
C * 0.266200D-2, 0.399053D-2, 0.467332D-2, 0.328313D-2, CTD02170
C * 0.223005D-5, -0.385196D-5, -0.540009D-5, -0.171657D-5, CTD02180
C * -0.567799D-8, 0.249443D-8, 0.363514D-8, 0.434294D-9, CTD02190
C * 0.395952D-11,-0.106755D-11,-0.137340D-11,-0.542308D-13, CTD02200
C * -0.104276D-14, 0.283531D-15, 0.269568D-15, 0.394729D-17, CTD02210
C * 0.849785D-19,-0.311947D-19,-0.207116D-19,-0.149196D-21/CTD02220
C DATA DEP/ 2.0D+3, 3.0D+3, 4.0D+3, 6.0D+3/ CTD02230
* CTD02240
C Correction value at P=PMAX (C0) CTD02250
C0=CDN(1) CTD02260
DO 1 K=2,NP CTD02270
1 C0=C0 + CDN(K)*PMAX**(K-1) CTD02280
* CTD02290
IF(PMAX.LE.DEP(1)) THEN CTD02300
C1=CUP(1,1) CTD02310
DO 2 K=2,NP CTD02320
2 C1=C1 + CUP(1,K)*PMAX**(K-1) CTD02330
DELC=C0 - C1 CTD02340
DELCO=(CDN(1) - CUP(1,1))*(1.0D0-PMAX/DEP(1)) CTD02350
DO 4 I=1,NDAT CTD02360
IF(P(I).GT.0.9D+5) GO TO 4 CTD02370
DC=DELCO + (DELC-DELCO)*P(I)/PMAX CTD02380
C=CUP(1,1) CTD02390
DO 3 K=2,NP CTD02400

```

```

3          C=C + CUP(1,K)*P(I)**(K-1)          CTD02410
          C=C + DC                             CTD02420
          P(I)=P(I) + C                       CTD02430
4    CONTINUE                                CTD02440
      RETURN                                  CTD02450
END IF                                        CTD02460
*
IF(PMAX.LE.DEF(2)) THEN                    CTD02470
  JA=1                                       CTD02480
  JB=2                                       CTD02490
ELSE                                         CTD02500
  IF(PMAX.LE.DEF(3)) THEN                  CTD02510
    JA=2                                     CTD02520
    JB=3                                     CTD02530
  ELSE                                       CTD02540
    JA=3                                     CTD02550
    JB=4                                     CTD02560
  END IF                                     CTD02570
END IF                                     CTD02580
END IF                                     CTD02590
C1=CUP(JB,1)                               CTD02600
DO 10 K=2,NP                               CTD02610
10  C1=C1 + CUP(JB,K)*PMAX**(K-1)         CTD02620
    CA=CUP(JA,1)                           CTD02630
    DO 11 K=2,NP                           CTD02640
11  CA=CA + CUP(JA,K)*DEF(JA)**(K-1)     CTD02650
    CB=CUP(JB,1)                           CTD02660
    DO 12 K=2,NP                           CTD02670
12  CB=CB + CUP(JB,K)*DEF(JA)**(K-1)     CTD02680
    CGRAD=(C0 - C1)/PMAX                   CTD02690
    CRATIO=CGRAD*DEF(JA)/(CA - CB)         CTD02700
    DO 13 I=1,NDAT                         CTD02710
      IF(P(I).GT.0.9D+5) GO TO 13         CTD02720
      CBP=CUP(JB,1)                       CTD02730
      DO 14 K=2,NP                         CTD02740
14  CBP=CBP + CUP(JB,K)*P(I)**(K-1)     CTD02750
      IF(P(I).GT.DEF(JA)) THEN            CTD02760
        DC=CGRAD*P(I)                    CTD02770
        C=CBP + DC                       CTD02780
        P(I)=P(I) + C                    CTD02790
      ELSE                                  CTD02800
        CAP=CUP(JA,1)                     CTD02810
        DO 15 K=2,NP                       CTD02820
15  CAP=CAP + CUP(JA,K)*P(I)**(K-1)     CTD02830
        C=CBP + (CAP-CBP)*CRATIO         CTD02840
        P(I)=P(I) + C                    CTD02850
      END IF                               CTD02860
    END IF                                  CTD02870
  END IF                                  CTD02880
END IF                                  CTD02890
END IF                                  CTD02900
*
C *****
SUBROUTINE CELLCORR(C,P,T,IQ,IINI,IEND)
C Conductivity correction for the cell deformation due to T and P
REAL*8      C(IQ),P(IQ),T(IQ)
REAL*8      ALPHA,BETA,T0,P0
*
ALPHA=-6.5D-6
BETA = 1.5D-8
T0 = 1.5D+1
P0 = 0.0D+0
*
DO 10 I=IINI,IEND
  IF(C(I).GT.0.9D+2.OR.P(I).GT.0.9D+5.OR.T(I).GT.0.9D+2) THEN
    C(I)=0.999999D+2
  ELSE
    C(I)=C(I)*(1.0D+0 + ALPHA*(T(I)-T0) + BETA*(P(I)-P0))
  END IF
10 CONTINUE
RETURN
END
*
C *****
SUBROUTINE CCORRECT(C,P,IP,CO,NP,IINI,IEND)
C Multiply observed values of Cctd by the cell factor
*
REAL*8      C(IP),P(IP),CO(NP),CF
*
DO 10 I=IINI,IEND
  IF(C(I).GT.0.9D+2) GO TO 10
  CF = CO(1)

```

```

DO 20 J=2,NP
CF = CF + CO(J)*P(I)**(J-1)
20 CONTINUE
C(I) = C(I)*CF
10 CONTINUE
RETURN
END
*
C *****
SUBROUTINE CSALCOMP(NDAT,PLD,IQ,JQ,STMP)
C Compute Sctd (PLD(I,5)) from Cctd,
C Csample (PLD(I,6)) from Ssample
C STMP: TEMPERATURE IN THE RESERVOIR OF PORTASAL
*
REAL*8 PLD(IQ,JQ),SAL78,C35150,CND,SALINO78,STMP
EXTERNAL SAL78,SALINO78
PARAMETER (C35150=0.42914D+2)
*
DO 10 I=1,NDAT
IF(PLD(I,1).GT.0.9D+5.OR.PLD(I,2).GT.0.9D+2.OR.
* PLD(I,4).GT.0.9D+2) THEN
PLD(I,5)=0.999999D+2
ELSE
CND=PLD(I,4)/C35150
PLD(I,5)=SAL78(CND,PLD(I,2),PLD(I,1),0)
END IF
IF(PLD(I,7).GT.0.9D+2) THEN
PLD(I,6)=0.999999D+2
PLD(I,7)=0.999999D+2
ELSE
PLD(I,7)=SALINO78(PLD(I,7),STMP)
IF(PLD(I,1).GT.0.9D+5.OR.PLD(I,2).GT.0.9D+2) THEN
PLD(I,6)=0.999999D+2
ELSE
PLD(I,6)=SAL78(PLD(I,7),PLD(I,2),PLD(I,1),1)
PLD(I,6)=PLD(I,6)*C35150
END IF
END IF
10 CONTINUE
RETURN
END
*
C *****
C SEPT. 28 1983
C ADD TF(S,P) FREEZING PT.
C WHOI CTD GROUP DISK FILE SPEC=BLUE::CTDA:<CTDEV.PRSW>PHYPROPSW.FOR
*
C TITLE: ALGORITHMS FOR OCEANOGRAPHIC COMPUTATIONS
C N FOFONOFF & R MILLARD
*
C SAL78 FCN ***** MAR 28 1983 *****
FUNCTION SAL78(CND,T,P,M)
IMPLICIT REAL*8 (A-H,O-Z)
*
C THE CONDUCTIVITY RATIO (CND) = 1.0000000 FOR SALINITY = 35 PSS-78
C TEMPERATURE = 15.0 DEG. CELSIUS , AND ATMOSPHERIC PRESSURE.
*
C FUNCTION TO CONVERT CONDUCTIVITY RATIO TO SALINITY (M = 0)
C SALINITY TO CONDUCTIVITY RATIO (M = 1,CND BECOMES INPUT SALINITY)
*
C REFERENCES: ALSO LOCATED IN UNESCO REPORT # 37 1981
C PRACTICAL SALINITY SCALE 1978: E.L. LEWIS IEEE OCEAN ENG. JAN. 1980
*
C UNITS:
C PRESSURE P DECIBARS
C TEMPERATURE T DEG CELSIUS (IPTS-68)
C CONDUCTIVITY CND RATIO (M=0)
C SALINITY SAL78 (PSS-78) (M=0)
C CHECKVALUES:
C SAL78=1.888091 :CND= 40.0000,T=40 DEG C,P= 10000 DECIBARS: M=1
C SAL78=40.00000 :CND=1.888091,T=40 DEG C,P=10000 DECIBARS: M=0
*
C SAL78 RATIO: RETURNS ZERO FOR CONDUCTIVITY RATIO: < 0.0005
C SAL78: RETURNS ZERO FOR SALINITY: < 0.02
*
C INTERNAL FUNCTIONS
*
C PRACTICAL SALINITY SCALE 1978 DEFINITION WITH TEMPERATURE CORRECTION
C XT=T-15.0 : XR=SQRT(RT)
SAL(XR,XT) =

```

```

* (((0.27081D+1*XR-0.70261D+1)*XR+0.140941D+2)*XR+0.253851D+2)*XR CTD04010
X -0.1692D0)* XR+0.0080D0 CTD04020
X +(XT/(1.0D0+0.0162D0*XT))*(((((-0.144D-1*XR+ CTD04030
X 0.636D-1)*XR-0.375D-1)*XR-0.66D-2)*XR-0.56D-2)*XR+0.5D-3) CTD04040
C DSAL(XR,XT) FUNCTION FOR DERIVATIVE OF SAL(XR,XT) WITH XR. CTD04050
DSAL(XR,XT) = CTD04060
*((((0.135405D+2*XR-0.281044D+2)*XR+0.422823D+2)*XR+0.507702D+2)*XRCTD04070
X -0.1692D0)+(XT/(1.0D0+0.162D-1*XT))*(((((-0.720D-1*XR+0.2544D0)*XRCTD04080
X -0.1125D0)*XR-0.132D-1)*XR-0.56D-2) CTD04090
C FUNCTION RT35 : C(35,T,0)/C(35,15,0) VARIATION WITH TEMPERATURE CTD04100
C WITH TEMPERATURE. CTD04110
RT35(XT) = (((1.0031D-9*XT-6.9698D-7)*XT+1.104259D-4)*XT CTD04120
X + 2.00564D-2)*XT + 0.6766097D0 CTD04130
C POLYNOMIALS OF RP: C(S,T,P)/C(S,T,0) VARIATION WITH PRESSURE CTD04140
C C(XP) POLYNOMIAL CORRESPONDS TO A1-A3 CONSTANTS: LEWIS 1980 CTD04150
C(XP) = ((3.989D-15*XP-6.370D-10)*XP+2.070D-5)*XP CTD04160
B(XT) = (4.464D-4*XT+3.426D-2)*XT + 1.0D0 CTD04170
C A(XT) POLYNOMIAL CORRESPONDS TO B3 AND B4 CONSTANTS: LEWIS 1980 CTD04180
A(XT) = -3.107D-3*XT + 0.4215D0 CTD04190
* CTD04200
C ZERO SALINITY/CONDUCTIVITY TRAP CTD04210
SAL78=0.0D0 CTD04220
IF((M.EQ.0).AND.(CND.LE.5D-4)) RETURN CTD04230
IF((M.EQ.1).AND.(CND.LE.0.2D-1)) RETURN CTD04240
* CTD04250
DT = T - 0.15D+2 CTD04260
C SELECT BRANCH FOR SALINITY (M=0) OR CONDUCTIVITY (M=1) CTD04270
IF(M.EQ.1) GO TO 10 CTD04280
* CTD04290
C CONVERT CONDUCTIVITY TO SALINITY CTD04300
R = CND CTD04310
RT = R/(RT35(T)*(1.0D0 + C(P)/(B(T) + A(T)*R))) CTD04320
RT = DSQRT(DABS(RT)) CTD04330
SAL78 = SAL(RT,DT) CTD04340
RETURN CTD04350
C ***** END OF CONDUCTIVITY TO SALINITY SECTION ***** CTD04360
* CTD04370
C INVERT SALINITY TO CONDUCTIVITY BY THE CTD04380
C NEWTON-RAPHSON ITERATIVE METHOD. CTD04390
* CTD04400
C FIRST APPROXIMATION CTD04410
10 RT = DSQRT(CND/0.35D+2) CTD04420
SI = SAL(RT,DT) CTD04430
N = 0 CTD04440
* CTD04450
C ITERATION LOOP BEGINS HERE WITH A MAXIMUM OF 10 CYCLES CTD04460
15 RT = RT + (CND - SI)/DSAL(RT,DT) CTD04470
SI = SAL(RT,DT) CTD04480
N = N + 1 CTD04490
DELS = DABS(SI - CND) CTD04500
IF((DELS.GT.1.0D-4).AND.(N.LT.10))GO TO 15 CTD04510
C *****END OF ITERATION LOOP ***** CTD04520
* CTD04530
C COMPUTE CONDUCTIVITY RATIO CTD04540
RTT = RT35(T)*RT*RT CTD04550
AT = A(T) CTD04560
BT = B(T) CTD04570
CP = C(P) CTD04580
CP = RTT*(CP + BT) CTD04590
BT = BT - RTT*AT CTD04600
* CTD04610
C SOLVE QUADRATIC EQUATION FOR R: R=RT35*RT*(1+C/AR+B) CTD04620
R = DSQRT(DABS(BT*BT + 0.4D+1*AT*CP)) - BT CTD04630
C CONDUCTIVITY RETURN CTD04640
SAL78 = 0.5D0*R/AT CTD04650
RETURN CTD04660
END CTD04670
* CTD04680
C ..... CTD04690
FUNCTION SALINO78(CND,T) CTD04700
IMPLICIT REAL*8 (A-H,O-Z) CTD04710
* CTD04720
C CALCULATE SALINITY FROM SALINOMETER OUTPUT FOR SAMPLING WATER CTD04730
C UNITS: CTD04740
C PRESSURE P = 0 DECIBARS CTD04750
C TEMPERATURE T DEG CELSIUS (IPTS-68) CTD04760
C CONDUCTIVITY CND RATIO CTD04770
C SALINITY SAL78 (PSS-78) CTD04780
* CTD04790
C INTERNAL FUNCTIONS CTD04800

```

```

*
C PRACTICAL SALINITY SCALE 1978 DEFINITION WITH TEMPERATURE CORRECTION CTD04810
C XT=T-15.0 : XR=SQRT(RT) CTD04820
  SAL(XR,XT) = CTD04830
  * (((0.27081D+1*XR-0.70261D+1)*XR+0.140941D+2)*XR+0.253851D+2)*XR CTD04840
  X -0.1692D0)* XR+0.0080D0 CTD04850
  X +(XT/(1.0D0+0.0162D0*XT))*(((((-0.144D-1*XR+ CTD04860
  X 0.636D-1)*XR-0.375D-1)*XR-0.66D-2)*XR-0.56D-2)*XR+0.5D-3) CTD04870
  CTD04880
* CTD04890
C ZERO SALINITY/CONDUCTIVITY TRAP CTD04900
  SALINO78=0.0D0 CTD04910
  IF(CND.LE.5D-4) RETURN CTD04920
* CTD04930
  DT = T - 0.15D+2 CTD04940
* CTD04950
C CONVERT CONDUCTIVITY TO SALINITY CTD04960
  RT = CND CTD04970
  RT = DSQRT(DABS(RT)) CTD04980
  SALINO78 = SAL(RT,DT) CTD04990
  RETURN CTD05000
  END CTD05010
* CTD05020
C ***** CTD05030
  SUBROUTINE SMPLOXGN(NDAT,NBS,OX,CNC,IQ,II) CTD05040
C Compute the concentration of dissolved oxygen in water samples CTD05050
  REAL*8 OX(IQ),CNC(IQ),B(5) CTD05060
  INTEGER NBS(IQ),NB(5) CTD05070
* CTD05080
  DO 10 I=1,NDAT CTD05090
  IF(NBS(I).GT.8000.OR.OX(I).GT.0.9D+2) THEN CTD05100
    OX(I)=0.99999D+2 CTD05110
  ELSE CTD05120
30 READ(II,'(5(I4,F6.2))',END=40) (NB(IS),B(IS),IS=1,5) CTD05130
  DO 20 J=1,5 CTD05140
    IF(NB(J).EQ.NBS(I)) THEN CTD05150
      OX(I)=OX(I)*CNC(I)*B(J) CTD05160
      REWIND II CTD05170
      GO TO 10 CTD05180
    END IF CTD05190
20 CONTINUE CTD05200
  GO TO 30 CTD05210
40 OX(I)=0.88888D+2 CTD05220
  REWIND II CTD05230
  END IF CTD05240
10 CONTINUE CTD05250
  RETURN CTD05260
  END CTD05270
* CTD05280
C ***** CTD05290
  SUBROUTINE CTDOXYGN(NDAT,P,T,S,OX,OCUR,OTMP,IQ,A,C,D,E,F) CTD05300
C Precise estimate the concentration of dissolved oxygen CTD05310
C from the O2 sensor data CTD05320
  IMPLICIT REAL*8 (A-H,O-Z) CTD05330
  DIMENSION P(IQ),T(IQ),S(IQ),OX(IQ),OCUR(IQ),OTMP(IQ) CTD05340
* CTD05350
  OSAT(X,SA)=DEXP((( -2.18492D+1*X - 1.734292D+2)*X CTD05360
  * + 2.496339D+2)/X + SA*(( -0.17D-2*X + 1.4259D-2)*X CTD05370
  * - 3.3096D-2) + 1.433483D+2*DLOG(X)) CTD05380
* CTD05390
  DO 10 I=1,NDAT CTD05400
  IF(P(I).GT.0.9D+5.OR.T(I).GT.0.9D+2.OR.S(I).GT.0.9D+2. CTD05410
  * OR.OCUR(I).GT.0.9D+1.OR.OTMP(I).GT.0.9D+2) THEN CTD05420
    OX(I)=0.99999D+2 CTD05430
  ELSE CTD05440
    Q=DEXP(D*(T(I) + E*(OTMP(I) - T(I))) + F*P(I)) CTD05450
    AT=(T(I) + 2.7316D+2)*0.1D-1 CTD05460
    OX(I)=(A*OCUR(I)+C)*Q*OSAT(AT,S(I)) CTD05470
  END IF CTD05480
10 CONTINUE CTD05490
  RETURN CTD05500
  END CTD05510
* CTD05520
C ***** CTD05530
  SUBROUTINE OXYUNIT(PLD,IQ,JQ,IINI,IEND) CTD05540
C Oxygen unit o ml/l kara micromol/kg ni henkan suru CTD05550
  REAL*8 PLD(IQ,JQ),THETA,PT0,SGO,COEF CTD05560
  EXTERNAL THETA CTD05570
* CTD05580
  DO 10 I=IINI,IEND CTD05590
  IF(PLD(I,1).GT.0.9D+5.OR.PLD(I,2).GT.0.9D+2.OR. CTD05600

```

```

*      PLD(I,5).GT.0.9D+2) THEN                                CTD05610
      PLD(I,8) =0.99999D+4                                    CTD05620
      PLD(I,9) =0.99999D+4                                    CTD05630
      PLD(I,10)=0.99999D+4                                    CTD05640
    ELSE                                                        CTD05650
      PT0=THETA(PLD(I,5),PLD(I,2),PLD(I,1),0.0D+0)          CTD05660
      CALL SIGMAN(PLD(I,5),PT0,0.0D+0,SG0)                   CTD05670
      COEF=1.0D+3/0.224D+2/(0.1D+1 + SG0*0.1D-2)           CTD05680
      IF(PLD(I,11).GT.0.8D+2) THEN                            CTD05690
        PLD(I,8)=0.99999D+4                                    CTD05700
      ELSE                                                       CTD05710
        PLD(I,8)=PLD(I,11)*COEF                               CTD05720
      END IF                                                    CTD05730
      IF(PLD(I,12).GT.0.8D+2) THEN                            CTD05740
        PLD(I,9)=0.99999D+4                                    CTD05750
      ELSE                                                       CTD05760
        PLD(I,9)=PLD(I,12)*COEF                               CTD05770
      END IF                                                    CTD05780
      IF(PLD(I,13).GT.0.8D+2) THEN                            CTD05790
        PLD(I,10)=0.99999D+4                                  CTD05800
      ELSE                                                       CTD05810
        PLD(I,10)=PLD(I,13)*COEF                              CTD05820
      END IF                                                    CTD05830
    END IF                                                      CTD05840
  10 CONTINUE                                                  CTD05850
  RETURN                                                         CTD05860
  END                                                            CTD05870
*                                                                CTD05880
C *****                                                       CTD05890
  SUBROUTINE SIGMAN(S,T,P0,SIGMA)                               CTD05900
  IMPLICIT REAL*8 (A-H,K,O-Z)                                  CTD05910
C T=POT. TEMP. AT P0, P0=3000 NO TOKI SIGMA=SIGMA-3          CTD05920
C MODIFIED RCM                                                CTD05930
C *****                                                       CTD05940
C SPECIFIC VOLUME ANOMALY (STERIC ANOMALY) BASED ON 1980 EQUATION CTD05950
C OF STATE FOR SEAWATER AND 1978 PRACTICAL SALINITY SCALE.   CTD05960
C REFERENCES                                                    CTD05970
C MILLERO, ET AL (1980) DEEP-SEA RES.,27A,255-264            CTD05980
C MILLERO AND POISSON 1981,DEEP-SEA RES.,28A PP 625-629.    CTD05990
C BOTH ABOVE REFERENCES ARE ALSO FOUND IN UNESCO REPORT 38 (1981) CTD06000
C UNITS:                                                         CTD06010
C   PRESSURE           P0           DECIBARS                   CTD06020
C   TEMPERATURE       T           DEG CELSIUS (IPSS-68)       CTD06030
C   SALINITY          S           (IPSS-78)                   CTD06040
C   SPEC. VOL. ANO.   SVAN        M**3/KG *1.0E-8             CTD06050
C   DENSITY ANO.     SIGMA        KG/M**3                     CTD06060
C *****                                                       CTD06070
C CHECK VALUE: SVAN=981.3021 E-8 M**3/KG. FOR S = 40 (IPSS-78) , CTD06080
C T = 40 DEG C, P0= 10000 DECIBARS.                            CTD06090
C CHECK VALUE: SIGMA = 59.82037 KG/M**3 FOR S = 40 (IPSS-78) , CTD06100
C T = 40 DEG C, P0= 10000 DECIBARS.                            CTD06110
C *****                                                       CTD06120
C EQUIV                                                         CTD06130
  EQUIVALENCE (E,D,B1),(BW,B,R3),(C,A1,R2)                    CTD06140
  EQUIVALENCE (AW,A,R1),(KW,K0,K)                              CTD06150
C *****                                                       CTD06160
C DATA                                                         CTD06170
  DATA R3500,R4/1.0281063D3,4.8314D-4/                       CTD06180
  DATA DR350/2.8106331D1/                                      CTD06190
C R4 IS REFERED TO AS C IN MILLERO AND POISSON 1981          CTD06200
C CONVERT PRESSURE TO BARS AND TAKE SQUARE ROOT SALINITY.    CTD06210
  P=P0/1.0D1                                                    CTD06220
  SR =DSQRT(DABS(S))                                           CTD06230
C *****                                                       CTD06240
C PURE WATER DENSITY AT ATMOSPHERIC PRESSURE                  CTD06250
C BIGG P.H.,(1967) BR. J. APPLIED PHYSICS 8 PP 521-537.     CTD06260
C                                                               CTD06270
  R1 = (((6.536332D-9*T-1.120083D-6)*T+1.001685D-4)*T        CTD06280
  X-9.095290D-3)*T+6.793952D-2)*T-2.8263737D1                CTD06290
C SEAWATER DENSITY ATM PRESS.                                  CTD06300
C COEFFICIENTS INVOLVING SALINITY                             CTD06310
C R2 = A IN NOTATION OF MILLERO AND POISSON 1981             CTD06320
  R2 = (((5.3875D-9*T-8.2467D-7)*T+7.6438D-5)*T-4.0899D-3)*T CTD06330
  X+8.24493D-1                                                  CTD06340
C R3 = B IN NOTATION OF MILLERO AND POISSON 1981             CTD06350
  R3 = (-1.6546D-6*T+1.0227D-4)*T-5.72466D-3                 CTD06360
C INTERNATIONAL ONE-ATMOSPHERE EQUATION OF STATE OF SEAWATER CTD06370
  SIG = (R4*S + R3*SR + R2)*S + R1                             CTD06380
C SPECIFIC VOLUME AT ATMOSPHERIC PRESSURE                     CTD06390
  V350P = 1.0D0/R3500                                          CTD06400

```



```

SVA = -SIG*V350P/(R3500+SIG)
SIGMA=SIG+DR350
C SCALE SPECIFIC VOL. ANOMALY TO NORMALLY REPORTED UNITS
SVAN=SV*1.0D+8
IF(P.EQ.0.0) RETURN
C *****
C ***** NEW HIGH PRESSURE EQUATION OF STATE FOR SEAWATER *****
C *****
C MILLERO, ET AL , 1980 DSR 27A, PP 255-264
C CONSTANT NOTATION FOLLOWS ARTICLE
C *****
C COÖPUTE COMPRESSION TERMS
E = (9.1697D-10*T+2.0816D-8)*T-9.9348D-7
BW = (5.2787D-8*T-6.12293D-6)*T+3.47718D-5
B = BW + E*S
C
D = 1.91075D-4
C = (-1.6078D-6*T-1.0981D-5)*T+2.2838D-3
AW = ((-5.77905D-7*T+1.16092D-4)*T+1.43713D-3)*T
X-0.1194975D0
A = (D*SR + C)*S + AW
C
B1 = (-5.3009D-4*T+1.6483D-2)*T+7.944D-2
A1 = ((-6.1670D-5*T+1.09987D-2)*T-0.603459D0)*T+5.46746D1
KW = (((-5.155288D-5*T+1.360477D-2)*T-2.327105D0)*T
X+1.484206D2)*T-1.93006D3
K0 = (B1*SR + A1)*S + KW
C EVALUATE PRESSURE POLYNOMIAL
C *****
C K EQUALS THE SECANT BULK MODULUS OF SEAWATER
C DK=K(S,T,P)-K(35,0,P)
C K35=K(35,0,P)
C *****
DK = (B*P + A)*P + K0
K35 = (5.03217D-5*P+3.359406D0)*P+2.158227D4
GAM=P/K35
PK = 1.0D0 - GAM
SVA = SVA*PK + (V350P+SVA)*P*DK/(K35*(K35+DK))
C SCALE SPECIFIC VOL. ANOMALY TO NORMALLY REPORTED UNITS
SVAN=SV*1.0D+8
V350P = V350P*PK
C *****
C COMPUTE DENSITY ANOMALY WITH RESPECT TO 1000.0 KG/M**3
C 1) DR350: DENSITY ANOMALY AT 35 (IPSS-78), 0 DEG. C AND 0 DECIBARS
C 2) DR35P: DENSITY ANOMALY 35 (IPSS-78), 0 DEG. C , PRES. VARIATION
C 3) DVAN : DENSITY ANOMALY VARIATIONS INVOLVING SPECIFIC VOL. ANOMALY
C *****
C CHECK VALUE: SIGMA = 59.82037 KG/M**3 FOR S = 40 (IPSS-78),
C T = 40 DEG C, P0= 10000 DECIBARS.
C *****
DR35P=GAM/V350P
DVAN=SV/(V350P*(V350P+SVA))
SIGMA=DR350+DR35P-DVAN
RETURN
END
*
C *****
C REAL FUNCTION ATG(S,T,P)
IMPLICIT REAL*8 (A-H,O-Z)
C *****
C ADIABATIC TEMPERATURE GRADIENT DEG C PER DECIBAR
C REF: BRYDEN,H.,1973,DEEP-SEA RES.,20,401-408
C UNITS:
C PRESSURE P DECIBARS
C TEMPERATURE T DEG CELSIUS (IPSS-68)
C SALINITY S (IPSS-78)
C ADIABATIC ATG DEG. C/DECIBAR
C CHECKVALUE: ATG=3.255976E-4 C/DBAR FOR S=40 (IPSS-78),
C T=40 DEG C,P0=10000 DECIBARS
DS = S - 3.5D1
ATG = (((-2.1687D-16*T+1.8676D-14)*T-4.6206D-13)*P
X+((2.7759D-12*T-1.1351D-10)*DS+((-5.4481D-14*T
X+8.733D-12)*T-6.7795D-10)*T+1.8741D-8))*P
X+(-4.2393D-8*T+1.8932D-6)*DS
X+((6.6228D-10*T-6.836D-8)*T+8.5258D-6)*T+3.5803D-5
RETURN
END
*
C *****
C REAL FUNCTION THETA(S,TO,P0,PR)

```

CTD06410
CTD06420
CTD06430
CTD06440
CTD06450
CTD06460
CTD06470
CTD06480
CTD06490
CTD06500
CTD06510
CTD06520
CTD06530
CTD06540
CTD06550
CTD06560
CTD06570
CTD06580
CTD06590
CTD06600
CTD06610
CTD06620
CTD06630
CTD06640
CTD06650
CTD06660
CTD06670
CTD06680
CTD06690
CTD06700
CTD06710
CTD06720
CTD06730
CTD06740
CTD06750
CTD06760
CTD06770
CTD06780
CTD06790
CTD06800
CTD06810
CTD06820
CTD06830
CTD06840
CTD06850
CTD06860
CTD06870
CTD06880
CTD06890
CTD06900
CTD06910
CTD06920
CTD06930
CTD06940
CTD06950
CTD06960
CTD06970
CTD06980
CTD06990
CTD07000
CTD07010
CTD07020
CTD07030
CTD07040
CTD07050
CTD07060
CTD07070
CTD07080
CTD07090
CTD07100
CTD07110
CTD07120
CTD07130
CTD07140
CTD07150
CTD07160
CTD07170
CTD07180
CTD07190
CTD07200

```

      IMPLICIT REAL*8 (A-H,O-Z)
C *****
C TO COMPUTE LOCAL POTENTIAL TEMPERATURE AT PR
C USING BRYDEN 1973 POLYNOMIAL FOR ADIABATIC LAPSE RATE
C AND RUNGE-KUTTA 4-TH ORDER INTEGRATION ALGORITHM.
C REF: BRYDEN,H.,1973,DEEP-SEA RES.,20,401-408
C FOFONOFF,N.,1977,DEEP-SEA RES.,24,489-491
C UNITS:
C     PRESSURE      P0      DECIBARS
C     TEMPERATURE  T0      DEG CELSIUS (IPSS-68)
C     SALINITY     S        (IPSS-78)
C     REFERENCE PRS PR      DECIBARS
C     POTENTIAL TMP. THETA  DEG CELSIUS
C CHECKVALUE: THETA= 36.89073 C,S=40 (IPSS-78),T0=40 DEG C,
C P0=10000 DECIBARS,PR=0 DECIBARS
C
C     SET-UP INTERMEDIATE TEMPERATURE AND PRESSURE VARIABLES
C     P=P0
C     T=T0
C *****
      H = PR - P
      XK = H*ATG(S,T,P)
      T = T + 0.5D0*XK
      Q = XK
      P = P + 0.5D0*H
      XK = H*ATG(S,T,P)
      T = T + 0.29289322D0*(XK-Q)
      Q = 0.58578644D0*XK + 0.121320344D0*Q
      XK = H*ATG(S,T,P)
      T = T + 1.707106781D0*(XK-Q)
      Q = 3.414213562D0*XK - 4.121320344D0*Q
      P = P + 0.5D0*H
      XK = H*ATG(S,T,P)
      THETA = T + (XK-2.0D0*Q)/6.0D0
      RETURN
      END

```

```

CTD07210
CTD07220
CTD07230
CTD07240
CTD07250
CTD07260
CTD07270
CTD07280
CTD07290
CTD07300
CTD07310
CTD07320
CTD07330
CTD07340
CTD07350
CTD07360
CTD07370
CTD07380
CTD07390
CTD07400
CTD07410
CTD07420
CTD07430
CTD07440
CTD07450
CTD07460
CTD07470
CTD07480
CTD07490
CTD07500
CTD07510
CTD07520
CTD07530
CTD07540
CTD07550
CTD07560

```

4. 電気伝導度センサーの補正式を導くプログラム

(1) CFACPOLY.FOR [B7a]

セルファクターの鉛直変化を圧力の多項式で近似する。

(2) CSCCPOLY.FOR [B7b]

C_{sample} に合うように、 C_{CTD} の補正値を C_{CTD} の多項式で近似する。

C_{sample} をセルファクターに変えることでB7cにも使える。

```

C***** CFA00010
C Program to compute polynomial of pressure * CFA00020
C fitting the vertical change of cell factor (April, 1992) * CFA00030
C***** CFA00040
C Files CFA00050
C INPUT FT10F001 CTDRMSL1.DATA (LRECL=100) CFA00060
C OUTPUT FT20F001 CFPOLLG1.DATA (LRECL=100) CFA00070
* CFA00080
C NP=4 for cubic Eq., NP=3 for quadratic Eq. CFA00090
PARAMETER (NP=6) CFA00100
C NST: Number of station treated here CFA00110
C NST=34 for leg 1, 38 for leg 2 (KH-91-5) CFA00120
PARAMETER (NST=34, IQ=NST*24) CFA00130
REAL*8 PR(IQ), CF(IQ), CO(NP), ECO(NP) CFA00140
REAL*8 CC(IQ), CS(IQ), CCTD, CSMP CFA00150
REAL*8 TC(IQ), TCTD, SS(IQ), SSMP CFA00160
REAL*8 PRES, CFCUT CFA00170
PARAMETER (ID=3) CFA00180
REAL*8 CFRMS(ID), CRMS(ID), SRMS(ID), AN(ID) CFA00190
REAL*8 POWER1 CFA00200
EXTERNAL POWER1 CFA00210
CHARACTER*29 FRM10/'(A4, 5X, F7.1, 2(F8.4, 8X), 2F8.4)'/ CFA00220
CHARACTER*1 DUM CFA00230
CHARACTER*4 STN(NST) CFA00240
* CFA00250
N=0 CFA00260
DO 50 IT=1, NST CFA00270
STN(IT)=' ' CFA00280
READ(10, '(A1)', END=900) DUM CFA00290
READ(10, '(A1)') DUM CFA00300
DO 30 I=1, 24 CFA00310
READ(10, FRM10) STN(IT), PRES, TCTD, CCTD, CSMP, SSMP CFA00320
IF (PRES.GT.9.D+4.OR.CCTD.GT.9.D+1.OR.CSMP.GT.9.D+1) GO TO 30 CFA00330
N=N+1 CFA00340
PR(N)=PRES CFA00350
CC(N)=CCTD CFA00360
CF(N)=CSMP/CCTD CFA00370
CS(N)=CSMP CFA00380
TC(N)=TCTD CFA00390
SS(N)=SSMP CFA00400
30 CONTINUE CFA00410
READ(10, '(A1)') DUM CFA00420
READ(10, '(A1)') DUM CFA00430
50 CONTINUE CFA00440
* CFA00450
900 CONTINUE CFA00460
IF (N.LE.0) THEN CFA00470
WRITE(6, *) 'No data. Check dataset.' CFA00480
STOP CFA00490
END IF CFA00500
* CFA00510
WRITE(20, 150) STN CFA00520
150 FORMAT('STN:', 10(1X, A4)) CFA00530
WRITE(20, '(/)') CFA00540
* CFA00550
C Calculate the least-square-fit polynomial of P for cell factor CFA00560
C using all data CFA00570
CFCUT=0.0D0 CFA00580
CALL LSQFIT(N, PR, CF, IQ, POWER1, CO, ECO, NP) CFA00590
CALL RMSCFAC(PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, CFRMS, CRMS, SRMS, AN, ID) CFA00600
CALL OUTPUT(N, CO, ECO, NP, CFCUT, CFRMS, CRMS, SRMS, AN, ID) CFA00610
* CFA00620
C Recalculate the polynomial using the data within the range of four CFA00630
C times of RMS of the anomaly from the previous-step polynomial CFA00640
CFCUT=CFRMS(3)*0.4D+1 CFA00650
CALL CFACCUT(PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, CFCUT) CFA00660
CALL LSQFIT(N, PR, CF, IQ, POWER1, CO, ECO, NP) CFA00670
CALL RMSCFAC(PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, CFRMS, CRMS, SRMS, AN, ID) CFA00680
CALL OUTPUT(N, CO, ECO, NP, CFCUT, CFRMS, CRMS, SRMS, AN, ID) CFA00690
* CFA00700
C Recalculate using the data within three times of RMS CFA00710
CFCUT=CFRMS(3)*0.3D+1 CFA00720
CALL CFACCUT(PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, CFCUT) CFA00730
CALL LSQFIT(N, PR, CF, IQ, POWER1, CO, ECO, NP) CFA00740
CALL RMSCFAC(PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, CFRMS, CRMS, SRMS, AN, ID) CFA00750
CALL OUTPUT(N, CO, ECO, NP, CFCUT, CFRMS, CRMS, SRMS, AN, ID) CFA00760
* CFA00770
C Recalculate using the data within twice of RMS CFA00780
CFCUT=CFRMS(3)*0.2D+1 CFA00790
CALL CFACCUT(PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, CFCUT) CFA00800

```

```

CALL LSQFIT(N, PR, CF, IQ, POWER1, CO, ECO, NP)
CALL RMSCFAC (PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, CFRMS, CRMS, SRMS, AN, ID)
CALL OUTPUT(N, CO, ECO, NP, CFCUT, CFRMS, CRMS, SRMS, AN, ID)
*
STOP
END
*
C *****
SUBROUTINE CFACCUT (PR, CF, CC, CS, TC, SS, CO, IQ, NP, N, DCFCUT)
C Remove the cell factors which deviate from the fitted
C polynomial values more than DCFCUT
REAL*8 PR(IQ), CF(IQ), CC(IQ), CS(IQ), CO(NP), TC(IQ), SS(IQ)
REAL*8 CFP, DCF, DCFCUT
I=0
30 I=I+1
IF (I.GT.N) GO TO 31
CFP=CO(I)
DO 35 K=2, NP
CFP=CFP + CO(K)*PR(I)**(K-1)
35 CONTINUE
DCF=DABS (CFP-CF(I))
IF (DCF.GE.DCFCUT) THEN
IF (I.EQ.N) GO TO 41
DO 40 J=I, N-1
PR(J)=PR(J+1)
CF(J)=CF(J+1)
CC(J)=CC(J+1)
CS(J)=CS(J+1)
TC(J)=TC(J+1)
SS(J)=SS(J+1)
40 CONTINUE
41 N=N-1
I=I-1
END IF
GO TO 30
31 CONTINUE
RETURN
END
*
C *****
SUBROUTINE LSQFIT (ND, XD, YD, IQ, FNC, CO, ECO, NO)
C Least square fit of YD to a polynomial of XD
PARAMETER (NOMAX=6)
REAL*8 XD(IQ), YD(IQ)
REAL*8 FNC
EXTERNAL FNC
REAL*8 CO(NO), ECO(NO)
REAL*8 B(NOMAX, NOMAX), C(NOMAX)
REAL*8 BW(NOMAX, NOMAX), CW(NOMAX), WW(NOMAX)
REAL*8 S, X, Y, E, ESQ
INTEGER K, L, I
*
C The purpose of this program is to determine CO(K) such that
C MINIMIZE SUM(I=1, ND) (E(I)**2)
C where erros are given by
C E(I)=DY(I) - SUM(K=1, NO) (CO(K)*FNC(DX(I), K))
C FNC(X, I) is a set of fitting functions. Any linear independent
C functions can be used for the fitting. For example, FNC(X, 1)=1
C and FNC(X, 2)=X when linear fitting is considered.
C Anyway, the user must prepare to declare these functions
C with a double precision variable as the first argument and
C integral variable as the second argument.
*
C Now start procedure
DO 10 K=1, NO
C(K)=0.0D+0
CW(K)=0.0D+0
DO 10 L=1, NO
B(K, L)=0.0D+0
BW(K, L)=0.0D+0
10 CONTINUE
*
C Construct normal equation temporary use CO(K) as FNC(X, K)
DO 20 I= 1, ND
X = XD(I)
Y = YD(I)
DO 30 K=1, NO
CO(K) = FNC(X, K)
30 CONTINUE
DO 40 K=1, NO

```

CFA00810
CFA00820
CFA00830
CFA00840
CFA00850
CFA00860
CFA00870
CFA00880
CFA00890
CFA00900
CFA00910
CFA00920
CFA00930
CFA00940
CFA00950
CFA00960
CFA00970
CFA00980
CFA00990
CFA01000
CFA01010
CFA01020
CFA01030
CFA01040
CFA01050
CFA01060
CFA01070
CFA01080
CFA01090
CFA01100
CFA01110
CFA01120
CFA01130
CFA01140
CFA01150
CFA01160
CFA01170
CFA01180
CFA01190
CFA01200
CFA01210
CFA01220
CFA01230
CFA01240
CFA01250
CFA01260
CFA01270
CFA01280
CFA01290
CFA01300
CFA01310
CFA01320
CFA01330
CFA01340
CFA01350
CFA01360
CFA01370
CFA01380
CFA01390
CFA01400
CFA01410
CFA01420
CFA01430
CFA01440
CFA01450
CFA01460
CFA01470
CFA01480
CFA01490
CFA01500
CFA01510
CFA01520
CFA01530
CFA01540
CFA01550
CFA01560
CFA01570
CFA01580
CFA01590
CFA01600

```

      C(K) = C(K) + Y*CO(K)
      DO 40 L=X,NO
      B(K,L)=B(K,L)+CO(K)*CO(L)
40    CONTINUE
20    CONTINUE
      DO 50 K=1,NO-1
      DO 50 L=K+1,NO
      B(L,K)=B(K,L)
50    CONTINUE
*
      DO 43 J=1,NO
      CW(J)=C(J)
      DO 43 I=1,NO
      BW(I,J)=B(I,J)
43    CONTINUE
      CALL LINLD(BW,NO,NOMAX,CW,-1.0D0,CO,IER)
*
      DO 63 K=1,NO
      CO(K)=CW(K)
63    CONTINUE
*
C Check error bar
      DO 60 K=1,NO
      C(K)=0.
60    CONTINUE
      ESQ=0.
      DO 70 I= 1,ND
      X = XD(I)
      E = YD(I)
      DO 80 K=1,NO
      E = E-FNC(X,K)*CO(K)
80    CONTINUE
      ESQ=ESQ+E*E
70    CONTINUE
      E=DSQRT(ESQ/FLOAT(ND-NO))
      DO 90 I=1,ND
      X = XD(I)
      DO 100 K=1,NO
      C(K) = C(K) + E*FNC(X,K)
100   CONTINUE
90    CONTINUE
*
      DO 102 J=1,NO
      CW(J)=C(J)
      DO 102 I=1,NO
      BW(I,J)=B(I,J)
102   CONTINUE
      CALL LINLD(BW,NO,NOMAX,CW,-1.0D0,WW,IER)
      DO 101 K=1,NO
      ECO(K)=CW(K)
101   RETURN
      END
*
C *****
      FUNCTION POWER1(X,K)
C Function used for polynomial fitting
      REAL*8 POWER1,X
      INTEGER K
      POWER1=X**(K-1)
      RETURN
      END
*
C *****
      SUBROUTINE LINLD(A, N, N1, B, EPS, P, IER)
C Simultaneous equations for real symmetric matrix
C by modified Cholesky method
C Parameters
C (1) A : 2-dim. array containing real symmetric matrix
C (2) N : Order of the matrix to solve
C (3) N1 : Size of array a defined in 'DIMENSION' statement
C (4) B : 1-dim. array containing the right hand side vector
C (5) EPS : Parameter to check singularity of the matrix
C Standard value = 1.0D-14
C (6) P : 1-dim. array containing reciprocals of diagonals of D
C (7) IER : Error code
C Copy right T. OGUNI July 30 1989 Version 1.0
*
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N1,N1), B(N), P(N)
C Check the input data

```

```

      EPS1 = 1.0D-14
      IF ((N1 .LT. N) .OR. (N .LE. 1)) THEN
        WRITE(*,*) '(SUBR. LINLD) INVALID ARGUMENT. N1 < N.', N1, N
        IER = 2
        RETURN
      ENDIF
      IF (EPS .LE. 0.0) EPS = EPS1
      IF (DABS(A(1,1)) .LE. EPS) THEN
        J = 1
        WRITE(*,*) '(SUBR. LINLD) SINGULAR AT STEP # ', J
        IER = 1
        RETURN
      ENDIF
C Initialization
      P(1) = 1.0D0 / A(1,1)
      IER = 0
C Modified Cholesky decomposition
      DO 150 J=2,N
        IF (J .GT. 2) THEN
          DO 120 I=2,J-1
            S = 0.0D0
            DO 110 K=1,I-1
110          S = S + A(K,I) * A(K,J)
              A(I,J) = A(I,J) - S
120          CONTINUE
            ENDIF
            S = 0.0D0
            DO 140 I=1,J-1
              T = A(I,J)
              A(I,J) = P(I) * T
              S = S + A(I,J) * T
140          CONTINUE
              T = A(J,J) - S
              IF (DABS(T) .LE. EPS) THEN
                WRITE(*,*) '(SUBR. LINLD) SINGULAR AT STEP # ', J
                IER = 1
                RETURN
              ENDIF
              P(J) = 1.0D0 / T
              A(J,J) = T
150          CONTINUE
        *
        ENTRY LDSUB(A,B)
C Forward substitution
        DO 200 I=2,N
          S = 0.0D0
          DO 190 K=1,I-1
190          S = S + A(K,I) * B(K)
              B(I) = B(I) - S
200          CONTINUE
          DO 210 I=1,N
210          B(I) = B(I) * P(I)
C Backward substitution
        DO 240 K=2,N
          J = N + 2 - K
          T = B(J)
          IF (T .NE. 0.0) THEN
230          B(I) = B(I) - A(I,J) * T
          ENDIF
240          CONTINUE
          RETURN
        END
        *
C *****
      SUBROUTINE RMSCFAC(P,CF,CC,CS,T,S,CO,IQ,NP,N,
        *
        CFRMS,CRMS,SRMS,AN,ID)
C Compute rms difference of cell factor
C from the fitted polynomial curve
      REAL*8 P(IQ),CF(IQ),CC(IQ),CS(IQ),T(IQ),S(IQ),CO(NP)
      REAL*8 CFRMS(ID),CRMS(ID),SRMS(ID),AN(ID)
      REAL*8 CFA,DVARCF,DVARC,DVARS
      REAL*8 C35150,SAL78,CND,SAL
      PARAMETER (C35150=0.42914D+2)
      EXTERNAL SAL78
        *
      DO 5 K=1,ID
        CFRMS(K) = 0.0D0
        CRMS(K) = 0.0D0
        SRMS(K) = 0.0D0

```

```

      AN(K)      = 0.0D0
5 CONTINUE
*
      I=0
10 I=I+1
      IF(I.GT.N) GO TO 20
      CFA=CO(1)
      DO 30 K=2,NP
        CFA=CFA + CO(K)*P(I)**(K-1)
30 CONTINUE
      DVARCF = (CFA - CF(I))*(CFA - CF(I))
      CFRMS(3)= CFRMS(3) + DVARCF
      DVARC = (CFA*CC(I) - CS(I))*(CFA*CC(I) - CS(I))
      CRMS(3) = CRMS(3) + DVARC
      CND=CFA*CC(I)/C35150
      SAL=SAL78(CND,T(I),P(I),0)
      DVAR = (SAL - S(I))*(SAL - S(I))
      SRMS(3) = SRMS(3) + DVAR
      AN(3) = AN(3) + 0.1D+1
      IF(P(I).LT.0.2D+4) THEN
        CFRMS(1)= CFRMS(1) + DVARCF
        CRMS(1) = CRMS(1) + DVARC
        SRMS(1) = SRMS(1) + DVAR
        AN(1) = AN(1) + 0.1D+1
      ELSE
        CFRMS(2)= CFRMS(2) + DVARCF
        CRMS(2) = CRMS(2) + DVARC
        SRMS(2) = SRMS(2) + DVAR
        AN(2) = AN(2) + 0.1D+1
      END IF
      GO TO 10
*
120 DO 15 K=1,ID
      CFRMS(K) = DSQRT(CFRMS(K)/AN(K))
      CRMS(K) = DSQRT(CRMS(K)/AN(K))
      SRMS(K) = DSQRT(SRMS(K)/AN(K))
15 CONTINUE
      RETURN
      END
*
C *****
C SEPT. 28 1983
C ADD TF(S,P) FREEZING PT.
C WHOI CTD GROUP DISK FILE SPEC=BLUE::CTDA:<CTDEV.PRSW>PHYPROPSW.FOR
*
C TITLE: ALGORITHMS FOR OCEANOGRAPHIC COMPUTATIONS
C N FOFONOFF & R MILLARD
*
C SAL78 FCN ***** MAR 28 1983 *****
      FUNCTION SAL78(CND,T,P,M)
      IMPLICIT REAL*8 (A-H,O-Z)
*
C THE CONDUCTIVITY RATIO (CND) = 1.0000000 FOR SALINITY = 35 PSS-78
C TEMPERATURE = 15.0 DEG. CELSIUS , AND ATMOSPHERIC PRESSURE.
*
C FUNCTION TO CONVERT CONDUCTIVITY RATIO TO SALINITY (M = 0)
C SALINITY TO CONDUCTIVITY RATIO (M = 1,CND BECOMES INPUT SALINITY)
*
C REFERENCES: ALSO LOCATED IN UNESCO REPORT # 37 1981
C PRACTICAL SALINITY SCALE 1978: E.L. LEWIS IEEE OCEAN ENG. JAN. 1980
*
C UNITS:
C PRESSURE P DECIBARS
C TEMPERATURE T DEG CELSIUS (IPTS-68)
C CONDUCTIVITY CND RATIO (M=0)
C SALINITY SAL78 (PSS-78) (M=0)
C CHECKVALUES:
C SAL78=1.888091 :CND= 40.0000,T=40 DEG C,P= 10000 DECIBARS: M= 1
C SAL78=40.00000 :CND=1.888091,T=40 DEG C,P=10000 DECIBARS: M=0
*
C SAL78 RATIO: RETURNS ZERO FOR CONDUCTIVITY RATIO: < 0.0005
C SAL78: RETURNS ZERO FOR SALINITY: < 0.02
*
C INTERNAL FUNCTIONS
*
C PRACTICAL SALINITY SCALE 1978 DEFINITION WITH TEMPERATURE CORRECTION
C XT=T-15.0 : KR=SQRT(RT)
      SAL(XR,XT) =
      * (((0.27031D+1*XR-0.70261D+1)*XR+0.140941D+2)*XR+0.253351D+2)*XR
      X -0.1692D0)* XR+0.0080D0

```



```

X +(XT/(1.0D0+0.0162D0*XT))*(((((-0.144D-1*XR+
X 0.636D-1)*XR-0.375D-1)*XR-0.66D-2)*XR-0.56D-2)*XR+0.5D-3) CFA04010
C DSAL(XR,XT) FUNCTION FOR DERIVATIVE OF SAL(XR,XT) WITH XR. CFA04020
DSAL(XR,XT) = CFA04030
*(((0.135405D+2*XR-0.281044D+2)*XR+0.422823D+2)*XR+0.507702D+2)*XRCFA04050
X -0.1692D0)+(XT/(1.0D0+0.162D-1*XT))*(((((-0.720D-1*XR+0.2544D0)*XRCFA04060
X -0.1125D0)*XR-0.132D-1)*XR-0.56D-2) CFA04070
C FUNCTION RT35 : C(35,T,0)/C(35,15,0) VARIATION WITH TEMPERATURE CFA04080
C WITH TEMPERATURE. CFA04090
RT35(XT) = (((1.0031D-9*XT-6.9698D-7)*XT+1.104259D-4)*XT
X + 2.00564D-2)*XT + 0.6766097D0 CFA04100
C POLYNOMIALS OF RP: C(S,T,P)/C(S,T,0) VARIATION WITH PRESSURE CFA04120
C C(XP) POLYNOMIAL CORRESPONDS TO A1-A3 CONSTANTS: LEWIS 1980 CFA04130
C(XP) = ((3.989D-15*XP-6.370D-10)*XP+2.070D-5)*XP CFA04140
B(XT) = (4.464D-4*XT+3.426D-2)*XT + 1.0D0 CFA04150
C A(XT) POLYNOMIAL CORRESPONDS TO B3 AND B4 CONSTANTS: LEWIS 1980 CFA04160
A(XT) = -3.107D-3*XT + 0.4215D0 CFA04170
* CFA04180
C ZERO SALINITY/CONDUCTIVITY TRAP CFA04190
SAL78=0.0D0 CFA04200
IF((M.EQ.0).AND.(CND.LE.5D-4)) RETURN CFA04210
IF((M.EQ.1).AND.(CND.LE.0.2D-1)) RETURN CFA04220
* CFA04230
DT = T - 0.15D+2 CFA04240
C SELECT BRANCH FOR SALINITY (M=0) OR CONDUCTIVITY (M=1) CFA04250
IF(M.EQ.1) GO TO 10 CFA04260
* CFA04270
C CONVERT CONDUCTIVITY TO SALINITY CFA04280
R = CND CFA04290
RT = R/(RT35(T)*(1.0D0 + C(P)/(B(T) + A(T)*R))) CFA04300
RT = DSQRT(DABS(RT)) CFA04310
SAL78 = SAL(RT,DT) CFA04320
RETURN CFA04330
C ***** END OF CONDUCTIVITY TO SALINITY SECTION ***** CFA04340
* CFA04350
C INVERT SALINITY TO CONDUCTIVITY BY THE CFA04360
C NEWTON-RAPHSON ITERATIVE METHOD. CFA04370
* CFA04380
C FIRST APPROXIMATION CFA04390
10 RT = DSQRT(CND/0.35D+2) CFA04400
SI = SAL(RT,DT) CFA04410
N = 0 CFA04420
* CFA04430
C ITERATION LOOP BEGINS HERE WITH A MAXIMUM OF 10 CYCLES CFA04440
15 RT = RT + (CND - SI)/DSAL(RT,DT) CFA04450
SI = SAL(RT,DT) CFA04460
N = N + 1 CFA04470
DELS = DABS(SI - CND) CFA04480
IF((DELS.GT.1.0D-4).AND.(N.LT.10))GO TO 15 CFA04490
C *****END OF ITERATION LOOP ***** CFA04500
* CFA04510
C COMPUTE CONDUCTIVITY RATIO CFA04520
RTT = RT35(T)*RT*RT CFA04530
AT = A(T) CFA04540
BT = B(T) CFA04550
CP = C(P) CFA04560
CP = RTT*(CP + BT) CFA04570
BT = BT - RTT*AT CFA04580
* CFA04590
C SOLVE QUADRATIC EQUATION FOR R: R=RT35*RT*(1+C/AR+B) CFA04600
R = DSQRT(DABS(BT*BT + 0.4D+1*AT*CP)) - BT CFA04610
C CONDUCTIVITY RETURN CFA04620
SAL78 = 0.5D0*R/AT CFA04630
RETURN CFA04640
END CFA04650
* CFA04660
C ***** CFA04670
SUBROUTINE OUTPUT(N,CO,ECO,NP,CFCUT,CFRMS,CRMS,SRMS,AN,ID) CFA04680
REAL*8 CO(NP),ECO(NP),CFCUT CFA04690
REAL*8 CFRMS(ID),CRMS(ID),SRMS(ID),AN(ID) CFA04700
* CFA04710
WRITE(20,100) N,CFCUT, CFA04720
* (AN(IS),CFRMS(IS),CRMS(IS),SRMS(IS),IS=1,ID) CFA04730
100 FORMAT('DATA NUMBER=',I4,5X,'CFCUT=',F8.5/ CFA04740
* 5X,'PR < 2000db',3X,'NUM=',F5.0,3X, CFA04750
* 'RMS ERROR OF CELL FACTOR =',E15.7/ CFA04760
* 31X,'RMS ERROR OF CONDUCTIVITY=',E15.7/ CFA04770
* 31X,'RMS ERROR OF SALINITY =',E15.7/ CFA04780
* 5X,'PR > 2000db',3X,'NUM=',F5.0,3X, CFA04790
* 'RMS ERROR OF CELL FACTOR =',E15.7/ CFA04800

```

```
*      31X,'RMS ERROR OF CONDUCTIVITY=',E15.7/      CFA04810
*      31X,'RMS ERROR OF SALINITY   =',E15.7/      CFA04820
*      5X,'Total      ',3X,'NUM=',F5.0,3X,      CFA04830
*      'RMS ERROR OF CELL FACTOR =',E15.7/      CFA04840
*      31X,'RMS ERROR OF CONDUCTIVITY=',E15.7/      CFA04850
*      31X,'RMS ERROR OF SALINITY   =',E15.7/      CFA04860
WRITE(20,110) CO      CFA04870
110 FORMAT(' CO: ',6E15.7)      CFA04880
WRITE(20,120) ECO      CFA04890
120 FORMAT(' ECO:',6E15.7/)      CFA04900
RETURN      CFA04910
END      CFA04920
```

```

C*****
C Program to compute polynomial of conductivity from CTD *
C for water-sample conductivity (April, 1992) *
C*****
C Files
C INPUT FT10F001 CTD RMSL1.DATA (LRECL=100)
C OUTPUT FT20F001 CSCC1LG1.DATA (LRECL=72)
*
C NP: 2 for linear, 3 for quadratic, 4 for cubic
PARAMETER (NP=2)
C NST: Number of station
C NST=34 for leg 1, 38 for leg 2 (KH-91-5)
PARAMETER (NST=38, IQ=NST*24)
*
REAL*8 CC(IQ),CS(IQ),CO(NP),ECO(NP)
REAL*8 PC(IQ),TC(IQ),SS(IQ),PCTD,TCTD,SSMP
REAL*8 CCTD,CSMP,CCUT
PARAMETER (ID=3)
REAL*8 CRMS(ID),SRMS(ID),AN(ID)
REAL*8 POWER1
EXTERNAL POWER1
CHARACTER*29 FRM10/'(A4,5X,F7.1,2(F8.4,8X),2F8.4)'/
CHARACTER*1 DUM
CHARACTER*4 STN(NST)
*
N=0
DO 50 IT=1,NST
STN(IT)=' '
READ(10,'(A1)',END=900) DUM
READ(10,'(A1)') DUM
DO 30 I=1,24
READ(10,FRM10) STN(IT),PCTD,TCTD,CCTD,CSMP,SSMP
IF(CCTD.GT.0.9D+2.OR.CSMP.GT.0.9D+2) GO TO 30
N=N+1
PC(N)=PCTD
TC(N)=TCTD
CC(N)=CCTD
CS(N)=CSMP
SS(N)=SSMP
30 CONTINUE
READ(10,'(A1)') DUM
READ(10,'(A1)') DUM
50 CONTINUE
*
900 CONTINUE
IF(N.LE.0) THEN
WRITE(6,*) 'No data. Check dataset.'
STOP
END IF
*
WRITE(20,150) STN
150 FORMAT('STN:',10(1X,A4))
WRITE(20,'(//)')
*
C Calculate the least-square-fit polynomial of Cctd for Csample
C using all data
CCUT=0.0D0
CALL LSQFIT(N,CC,CS,IQ,POWER1,CO,ECO,NP)
CALL RMSCOND(CC,CS,PC,TC,SS,CO,IQ,NP,N,CRMS,SRMS,AN,ID)
CALL OUTPUT(N,CO,ECO,NP,CCUT,CRMS,SRMS,AN,ID)
*
C Recalculate the polynomial using the data within the range of four
C times of RMS of the anomaly from the previous-step polynomial
CCUT=CRMS(3)*0.4D+1
CALL CONDCUT(CC,CS,PC,TC,SS,CO,IQ,NP,N,CCUT)
CALL LSQFIT(N,CC,CS,IQ,POWER1,CO,ECO,NP)
CALL RMSCOND(CC,CS,PC,TC,SS,CO,IQ,NP,N,CRMS,SRMS,AN,ID)
CALL OUTPUT(N,CO,ECO,NP,CCUT,CRMS,SRMS,AN,ID)
*
C Recalculate using the data within three times of RMS
CCUT=CRMS(3)*0.3D+1
CALL CONDCUT(CC,CS,PC,TC,SS,CO,IQ,NP,N,CCUT)
CALL LSQFIT(N,CC,CS,IQ,POWER1,CO,ECO,NP)
CALL RMSCOND(CC,CS,PC,TC,SS,CO,IQ,NP,N,CRMS,SRMS,AN,ID)
CALL OUTPUT(N,CO,ECO,NP,CCUT,CRMS,SRMS,AN,ID)
*
C Recalculate using the data within twice of RMS
CCUT=CRMS(3)*0.2D+1
CALL CONDCUT(CC,CS,PC,TC,SS,CO,IQ,NP,N,CCUT)
CALL LSQFIT(N,CC,CS,IQ,POWER1,CO,ECO,NP)

```

```

CALL RMSCOND(CC,CS,PC,TC,SS,CO,IQ,NP,N,CRMS,SRMS,AN,ID)
CALL OUTPUT(N,CO,ECO,NP,CCUT,CRMS,SRMS,AN,ID)
*
STOP
END
*
C *****
SUBROUTINE CONDCUT(X,Y,P,T,S,CO,IQ,NP,N,CUT)
C Remove the conductivity data which deviate
C from the fitted polynomial values more than CUT
REAL*8 X(IQ),Y(IQ),CO(NP),P(IQ),T(IQ),S(IQ)
REAL*8 YA,DEV,CUT
I=0
30 I=I+1
IF(I.GT.N) GO TO 31
YA=CO(1)
DO 35 K=2,NP
YA=YA + CO(K)*X(I)**(K-1)
35 CONTINUE
DEV=DABS(YA - Y(I))
IF(DEV.GE.CUT) THEN
IF(I.EQ.N) GO TO 41
DO 40 J=I,N-1
X(J)=X(J+1)
Y(J)=Y(J+1)
P(J)=P(J+1)
T(J)=T(J+1)
S(J)=S(J+1)
40 CONTINUE
41 N=N-1
I=I-1
END IF
GO TO 30
31 CONTINUE
RETURN
END
*
C *****
SUBROUTINE LSQFIT(ND,XD,YD,IQ,FNC,CO,ECO,NO)
C Least square fit of YD to a polynomial of XD
PARAMETER(NOMAX=6)
REAL*8 XD(IQ),YD(IQ)
REAL*8 FNC
EXTERNAL FNC
REAL*8 CO(NO),ECO(NO)
REAL*8 B(NOMAX,NOMAX),C(NOMAX)
REAL*8 BW(NOMAX,NOMAX),CW(NOMAX),WW(NOMAX)
REAL*8 S,X,Y,E,ESQ
INTEGER K,L,I
*
C The purpose of this program is to determine CO(K) such that
C MINIMIZE SUM(I=1,ND)(E(I)**2)
C where erros are given by
C E(I)=DY(I) - SUM(K=1,NO)(CO(K)*FNC(DX(I),K))
C FNC(X,I) is a set of fitting functions. Any linear independent
C functions can be used for the fitting. For example, FNC(X,1)=1
C and FNC(X,2)=X when linear fitting is considered.
C Anyway, the user must prepare to declare these functions
C with a double precision variable as the first argument and
C integral variable as the second argument.
*
C Now start procedure
DO 10 K=1,NO
C(K)=0.0D+0
CW(K)=0.0D+0
DO 10 L=1,NO
B(K,L)=0.0D+0
BW(K,L)=0.0D+0
10 CONTINUE
*
C Construct normal equation temporary use CO(K) as FNC(X,K)
DO 20 I= 1,ND
X = XD(I)
Y = YD(I)
DO 30 K=1,NO
CO(K) = FNC(X,K)
30 CONTINUE
DO 40 K=1,NO
C(K) = C(K) + Y*CO(K)
DO 40 L=K,NO

```

CSC00810
CSC00820
CSC00830
CSC00840
CSC00850
CSC00860
CSC00870
CSC00880
CSC00890
CSC00900
CSC00910
CSC00920
CSC00930
CSC00940
CSC00950
CSC00960
CSC00970
CSC00980
CSC00990
CSC01000
CSC01010
CSC01020
CSC01030
CSC01040
CSC01050
CSC01060
CSC01070
CSC01080
CSC01090
CSC01100
CSC01110
CSC01120
CSC01130
CSC01140
CSC01150
CSC01160
CSC01170
CSC01180
CSC01190
CSC01200
CSC01210
CSC01220
CSC01230
CSC01240
CSC01250
CSC01260
CSC01270
CSC01280
CSC01290
CSC01300
CSC01310
CSC01320
CSC01330
CSC01340
CSC01350
CSC01360
CSC01370
CSC01380
CSC01390
CSC01400
CSC01410
CSC01420
CSC01430
CSC01440
CSC01450
CSC01460
CSC01470
CSC01480
CSC01490
CSC01500
CSC01510
CSC01520
CSC01530
CSC01540
CSC01550
CSC01560
CSC01570
CSC01580
CSC01590
CSC01600

```

      B(K,L)=B(K,L)+CO(K)*CO(L)
40      CONTINUE
20      CONTINUE
      DO 50 K=1,NO-1
      DO 50 L=K+1,NO
        B(L,K)=B(K,L)
50      CONTINUE
*
      DO 43 J=1,NO
      CW(J)=C(J)
      DO 43 I=1,NO
        BW(I,J)=B(I,J)
43      CONTINUE
      CALL LINLD(BW,NO,NOMAX,CW,-1.0D0,CO,IER)
*
      DO 63 K=1,NO
      CO(K)=CW(K)
63      CONTINUE
*
C Check error bar
      DO 60 K=1,NO
      C(K)=0.
60      CONTINUE
      ESQ=0.
      DO 70 I= 1,ND
      X = XD(I)
      E = YD(I)
      DO 80 K=1,NO
      E = E-FNC(X,K)*CO(K)
80      CONTINUE
      ESQ=ESQ+E*E
70      CONTINUE
      E=DSQRT(ESQ/FLOAT(ND-NO))
      DO 90 I=1,ND
      X = XD(I)
      DO 100 K=1,NO
      C(K) = C(K) + E*FNC(X,K)
100     CONTINUE
90      CONTINUE
*
      DO 102 J=1,NO
      CW(J)=C(J)
      DO 102 I=1,NO
      BW(I,J)=B(I,J)
102     CONTINUE
      CALL LINLD(BW,NO,NOMAX,CW,-1.0D0,WW,IER)
      DO 101 K=1,NO
      ECO(K)=CW(K)
101     RETURN
      END
*
C *****
      FUNCTION POWER1(X,K)
C Function used for polynomial fitting
      REAL*8 POWER1,X
      INTEGER K
      POWER1=X**(K-1)
      RETURN
      END
*
C *****
      SUBROUTINE LINLD(A, N, N1, B, EPS, P, IER)
C Simultaneous equations for real symmetric matrix
C by modified Cholesky method
C Parameters
C (1) A : 2-dim. array containing real symmetric matrix
C (2) N : Order of the matrix to solve
C (3) N1 : Size of array a defined in 'DIMENSION' statement
C (4) B : 1-dim. array containing the right hand side vector
C (5) EPS : Parameter to check singularity of the matrix
C standard value = 1.0D-14
C (6) P : 1-dim. array containing reciprocals of diagonals of D
C (7) IER : Error code
C Copy right T. OGUNI July 30 1989 Version 1.0
*
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(N1,N1), B(N), P(N)
C Check the input data
      EPS1 = 1.0D-14
      IF ((N1 .LT. N) .OR. (N .LE. 1)) THEN

```

```

        WRITE(*,*) '(SUBR. LINLD) INVALID ARGUMENT. N1 < N.', N1, N      CSC02410
        IER = 2                                                         CSC02420
        RETURN                                                         CSC02430
    ENDIF                                                             CSC02440
    IF (EPS .LE. 0.0) EPS = EPS1                                       CSC02450
    IF (DABS(A(1,1)) .LE. EPS) THEN                                     CSC02460
        J = 1                                                         CSC02470
        WRITE(*,*) '(SUBR. LINLD) SINGULAR AT STEP # ', J           CSC02480
        IER = 1                                                         CSC02490
        RETURN                                                         CSC02500
    ENDIF                                                             CSC02510
C Initialization                                                         CSC02520
    P(1) = 1.0D0 / A(1,1)                                             CSC02530
    IER = 0                                                             CSC02540
C Modified Cholesky decomposition                                       CSC02550
    DO 150 J=2,N                                                       CSC02560
        IF (J .GT. 2) THEN                                           CSC02570
            DO 120 I=2,J-1                                           CSC02580
                S = 0.0D0                                           CSC02590
                DO 110 K=1,I-1                                       CSC02600
110                 S = S + A(K,I) * A(K,J)                         CSC02610
                    A(I,J) = A(I,J) - S                             CSC02620
120                CONTINUE                                         CSC02630
            ENDIF                                                     CSC02640
            S = 0.0D0                                                 CSC02650
            DO 140 I=1,J-1                                           CSC02660
                T = A(I,J)                                           CSC02670
                A(I,J) = P(I) * T                                     CSC02680
                S = S + A(I,J) * T                                   CSC02690
140            CONTINUE                                             CSC02700
            T = A(J,J) - S                                           CSC02710
            IF (DABS(T) .LE. EPS) THEN                                CSC02720
                WRITE(*,*) '(SUBR. LINLD) SINGULAR AT STEP # ', J   CSC02730
                IER = 1                                             CSC02740
                RETURN                                             CSC02750
            ENDIF                                                     CSC02760
            P(J) = 1.0D0 / T                                         CSC02770
            A(J,J) = T                                               CSC02780
150 CONTINUE                                                         CSC02790
*
        ENTRY LDSUB(A,B)                                             CSC02800
C Forward substitution                                                 CSC02810
    DO 200 I=2,N                                                       CSC02820
        S = 0.0D0                                                     CSC02830
        DO 190 K=1,I-1                                               CSC02840
190         S = S + A(K,I) * B(K)                                     CSC02850
            B(I) = B(I) - S                                         CSC02860
200 CONTINUE                                                         CSC02870
        DO 210 I=1,N                                                 CSC02880
210         B(I) = B(I) * P(I)                                       CSC02890
C Backward substitution                                               CSC02900
    DO 240 K=2,N                                                       CSC02910
        J = N + 2 - K                                               CSC02920
        T = B(J)                                                     CSC02930
        IF (T .NE. 0.0) THEN                                         CSC02940
            DO 230 I=1,J-1                                           CSC02950
230         B(I) = B(I) - A(I,J) * T                                 CSC02960
            ENDIF                                                     CSC02970
240 CONTINUE                                                         CSC02980
    RETURN                                                         CSC02990
    END                                                             CSC03000
    END                                                             CSC03010
*
C *****                                                             CSC03020
    SUBROUTINE RMSCOND(CC,CS,P,T,S,CO,IQ,NP,N,CRMS,SRMS,AN,ID)         CSC03030
C Compute rms difference of sample conductivity                       CSC03040
C from the fitted polynomial curve                                  CSC03050
    REAL*8 CC(IQ),CS(IQ),CO(NP),P(IQ),T(IQ),S(IQ)                   CSC03060
    REAL*8 CRMS(ID),SRMS(ID),AN(ID)                                 CSC03070
    REAL*8 CCA,DVARC,DVARS                                         CSC03080
    REAL*8 C35150,SAL78,CND,SAL                                    CSC03090
    PARAMETER (C35150=0.42914D+2)                                  CSC03100
    EXTERNAL SAL78                                                 CSC03110
*
    DO 5 K=1,ID                                                       CSC03120
        CRMS(K) = 0.0D0                                             CSC03130
        SRMS(K) = 0.0D0                                             CSC03140
        AN(K) = 0.0D0                                               CSC03150
5 CONTINUE                                                           CSC03160
*
    I=0                                                               CSC03170
                                                                    CSC03180
                                                                    CSC03190
                                                                    CSC03200

```

```

10 I=I+1                                CSC03210
   IF(I.GT.N) GO TO 20                   CSC03220
   CCA=CO(1)                             CSC03230
   DO 30 K=2,NP                          CSC03240
     CCA=CCA + CO(K)*CC(I)**(K-1)       CSC03250
30 CONTINUE                             CSC03260
   DVARC = (CCA - CS(I))*(CCA - CS(I))  CSC03270
   CRMS(3) = CRMS(3) + DVARC            CSC03280
   CND = CCA/C35150                     CSC03290
   SAL = SAL78(CND,T(I),P(I),0)        CSC03300
   DVARS = (SAL - S(I))*(SAL - S(I))   CSC03310
   SRMS(3) = SRMS(3) + DVARS           CSC03320
   AN(3) = AN(3) + 0.1D+1              CSC03330
   IF(P(I).LT.0.2D+4) THEN             CSC03340
     CRMS(1) = CRMS(1) + DVARC         CSC03350
     SRMS(1) = SRMS(1) + DVARS         CSC03360
     AN(1) = AN(1) + 0.1D+1           CSC03370
   ELSE                                  CSC03380
     CRMS(2) = CRMS(2) + DVARC         CSC03390
     SRMS(2) = SRMS(2) + DVARS         CSC03400
     AN(2) = AN(2) + 0.1D+1           CSC03410
   END IF                                CSC03420
   GO TO 10                              CSC03430
*                                         CSC03440
20 DO 15 K=1,ID                          CSC03450
   CRMS(K) = DSQRT(CRMS(K)/AN(K))      CSC03460
   SRMS(K) = DSQRT(SRMS(K)/AN(K))      CSC03470
15 CONTINUE                              CSC03480
   RETURN                                CSC03490
   END                                    CSC03500
*                                         CSC03510
C *****                                CSC03520
C SEPT. 28 1983                          CSC03530
C ADD TF(S,P) FREEZING PT.              CSC03540
C WHOI CTD GROUP DISK FILE SPEC=BLUE: :CTDA:<CTDEV.PRSW>PHYPROPSW.FOR
*                                         CSC03550
C TITLE: ALGORITHMS FOR OCEANOGRAPHIC COMPUTATIONS
C N FOFONOFF & R MILLARD                CSC03560
*                                         CSC03570
C SAL78 FCN ***** MAR 28 1983 *****
   FUNCTION SAL78(CND,T,P,M)             CSC03580
   IMPLICIT REAL*8 (A-H,O-Z)            CSC03590
*                                         CSC03600
C THE CONDUCTIVITY RATIO (CND) = 1.0000000 FOR SALINITY = 35 PSS-78
C TEMPERATURE = 15.0 DEG. CELSIUS , AND ATMOSPHERIC PRESSURE.
*                                         CSC03610
C FUNCTION TO CONVERT CONDUCTIVITY RATIO TO SALINITY (M = 0)
C SALINITY TO CONDUCTIVITY RATIO (M = 1,CND BECOMES INPUT SALINITY)
*                                         CSC03620
C REFERENCES: ALSO LOCATED IN UNESCO REPORT # 37 1981
C PRACTICAL SALINITY SCALE 1978: E.L. LEWIS IEEE OCEAN ENG. JAN. 1980
*                                         CSC03630
C UNITS:                                  CSC03640
C PRESSURE P DECIBARS                    CSC03650
C TEMPERATURE T DEG CELSIUS (IPTS-68)   CSC03660
C CONDUCTIVITY CND RATIO (M=0)          CSC03670
C SALINITY SAL78 (PSS-78) (M=0)        CSC03680
C CHECKVALUES:                           CSC03690
C SAL78=1.888091 :CND= 40.0000,T=40 DEG C,P= 10000 DECIBARS: M= 1CSC03700
C SAL78=40.00000 :CND=1.888091,T=40 DEG C,P=10000 DECIBARS: M=0 CSC03710
*                                         CSC03720
C SAL78 RATIO: RETURNS ZERO FOR CONDUCTIVITY RATIO: < 0.0005
C SAL78: RETURNS ZERO FOR SALINITY: < 0.02
*                                         CSC03730
C INTERNAL FUNCTIONS                      CSC03740
*                                         CSC03750
C PRACTICAL SALINITY SCALE 1978 DEFINITION WITH TEMPERATURE CORRECTION
C XT=T-15.0 : XR=SQRT(RT)                CSC03760
   SAL(XR,XT) =                          CSC03770
   * (((((0.27081D+1*XR-0.70261D+1)*XR+0.140941D+2)*XR+0.253851D+2)*XR
   X -0.1692D0)*XR+0.0080D0              CSC03780
   X +(XT/(1.0D0+0.0162D0*XT))*(((((-0.144D-1*XR+
   X 0.636D-1)*XR-0.375D-1)*XR-0.66D-2)*XR-0.56D-2)*XR-0.5D-3)
C DSAL(XR,XT) FUNCTION FOR DERIVATIVE OF SAL(XR,XT) WITH XR.
   DSAL(XR,XT) =                          CSC03790
   * (((((0.135405D+2*XR-0.281044D+2)*XR+0.422823D+2)*XR+0.507702D+2)*XRCSC03800
   X -0.1692D0)+(XT/(1.0D0+0.162D-1*XT))*(((((-0.720D-1*XR+0.2544D0)*XRCSC03810
   X -0.1125D0)*XR-0.132D-1)*XR-0.56D-2)
C FUNCTION RT35 : C(35,T,0)/C(35,15,0) VARIATION WITH TEMPERATURE
C WITH TEMPERATURE.                      CSC03820
*                                         CSC03830
*                                         CSC03840
*                                         CSC03850
*                                         CSC03860
*                                         CSC03870
*                                         CSC03880
*                                         CSC03890
*                                         CSC03900
*                                         CSC03910
*                                         CSC03920
*                                         CSC03930
*                                         CSC03940
*                                         CSC03950
*                                         CSC03960
*                                         CSC03970
*                                         CSC03980
*                                         CSC03990
*                                         CSC04000

```

```

      RT35(XT) = (((1.0031D-9*XT-6.9698D-7)*XT+1.104259D-4)*XT
      X      + 2.00564D-2)*XT + 0.6756097D0
C POLYNOMIALS OF RP: C(S,T,P)/C(S,T,0) VARIATION WITH PRESSURE
C C(XP) POLYNOMIAL CORRESPONDS TO A1-A3 CONSTANTS: LEWIS 1980
      C(XP) = ((3.989D-15*XP-6.370D-10)*XP+2.070D-5)*XP
      B(XT) = (4.464D-4*XT+3.426D-2)*XT + 1.0D0
C A(XT) POLYNOMIAL CORRESPONDS TO B3 AND B4 CONSTANTS: LEWIS 1980
      A(XT) = -3.107D-3*XT + 0.4215D0
*
C ZERO SALINITY/CONDUCTIVITY TRAP
      SAL78=0.0D0
      IF((M.EQ.0).AND.(CND.LE.5D-4)) RETURN
      IF((M.EQ.1).AND.(CND.LE.0.2D-1)) RETURN
*
      DT = T - 0.15D-2
C SELECT BRANCH FOR SALINITY (M=0) OR CONDUCTIVITY (M=1)
      IF(M.EQ.1) GO TO 10
*
C CONVERT CONDUCTIVITY TO SALINITY
      R = CND
      RT = R/(RT35(T)*(1.0D0 + C(P)/(B(T) + A(T)*R)))
      RT = DSQRT(DABS(RT))
      SAL78 = SAL(RT,DT)
      RETURN
C ***** END OF CONDUCTIVITY TO SALINITY SECTION *****
*
C INVERT SALINITY TO CONDUCTIVITY BY THE
C NEWTON-RAPHSON ITERATIVE METHOD.
*
C FIRST APPROXIMATION
      10 RT = DSQRT(CND/0.35D+2)
      SI = SAL(RT,DT)
      N = 0
*
C ITERATION LOOP BEGINS HERE WITH A MAXIMUM OF 10 CYCLES
      15 RT = RT + (CND - SI)/DSAL(RT,DT)
      SI = SAL(RT,DT)
      N = N + 1
      DELS = DABS(SI - CND)
      IF((DELS.GT.1.0D-4).AND.(N.LT.10))GO TO 15
C *****END OF ITERATION LOOP *****
*
C COMPUTE CONDUCTIVITY RATIO
      RTT = RT35(T)*RT*RT
      AT = A(T)
      BT = B(T)
      CP = C(P)
      CP = RTT*(CP + BT)
      BT = BT - RTT*AT
*
C SOLVE QUADRATIC EQUATION FOR R: R=RT35*RT*(1+C/AR+B)
      R = DSQRT(DABS(BT*BT + 0.4D+1*AT*CP)) - BT
C CONDUCTIVITY RETURN
      SAL78 = 0.5D0*R/AT
      RETURN
      END
*
C *****
      SUBROUTINE OUTPUT(N,CO,ECO,NP,CUT,CRMS,SRMS,AN, ID)
      REAL*8      CO(NP), ECO(NP), CUT
      REAL*8      CRMS(ID), SRMS(ID), AN(ID)
*
      WRITE(20,100) N,CUT, (AN(IS),CRMS(IS),SRMS(IS), IS=1, ID)
100 FORMAT('DATA NUMBER=',I4,5X,'CUT=',F8.5/
*      5X,'PR < 2000db',3X,'NUM=',F5.0,3X,
*      'RMS ERROR OF CONDUCTIVITY=',E15.7/
*      31X,'RMS ERROR OF SALINITY      =',E15.7/
*      5X,'PR > 2000db',3X,'NUM=',F5.0,3X,
*      'RMS ERROR OF CONDUCTIVITY=',E15.7/
*      31X,'RMS ERROR OF SALINITY      =',E15.7/
*      5X,'Total      ',3X,'NUM=',F5.0,3X,
*      'RMS ERROR OF CONDUCTIVITY=',E15.7/
*      31X,'RMS ERROR OF SALINITY      =',E15.7)
      WRITE(20,110) CO
110 FORMAT(' CO: ',4E15.7)
      WRITE(20,120) ECO
120 FORMAT(' ECO:',4E15.7)
      WRITE(20,'(1X)')
      RETURN
      END

```