AN OBJECTIVE METHOD TO ANALYZE OCEANIC VARICLINES AND THEIR STATISTICAL RELATION TO SURFACE WINDS

JOSÉ MANUEL PINTO BASTOS SALDANHA

United States
Naval Postgraduate School

THESIS

AN OBJECTIVE METHOD TO ANALYZE OCEANIC
VARICLINES AND THEIR STATISTICAL
RELATION TO SURFACE WINDS

by

José Manuel Pinto Bastos Saldanha

Thesis Advisors: Dale Leipper/Noel Boston

September 1971

An Objective Method to Analyze Oceanic Variclines
and
Their Statistical Relation to Surface Winds

by

Jose Manuel Pinto Bastos Saldanha
Lieutenant, Portuguese Navy

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN OCEANOGRAPHY

from the

NAVAL POSTGRADUATE SCHOOL
September 1971

ABSTRACT

Oceanographic and meteorologic data, namely expendable or mechanical bathythermograph traces and wind reports have been systematically collected for years and every day more observations are made.

To handle such a large amount of information, it is advantageous to set up a computerized model which is: (1) automatic, (2) performs an objective analysis of XBT/BT traces to identify the most significant thermal features, such as thermoclines and inversions, (3) restricts the human participation to an initial stage dealing with organization of data available and selection of proper values for the parameters of the model, and (4) preserves the initial information.

A FORTRAN IV program takes into consideration the above requirements to: (1) perform objective analyses of XBT/BT digital traces, (2) process wind information, (3) study the statistical characteristics of the original and processed data, and (4) plot the generated fields.

The digital model was tested and its application may be extended to the study of any environmental variable continuously distributed along a vertical or horizontal axis.

TABLE OF CONTENTS

LIST OF FIGURES

TABLE OF SYMBOLS AND ABBREVIATIONS

BT	Mechanical Bathythermograph
D	Time variable
e	2.718281 ...
k	Any positive integer
L	Second length variable for tri-dimensional distribution
m	Median of a Gaussian distribution
n	Any positive integer
π	3.14159 ...
S	Length variable for two-dimensional distribution
S(b)	Value of the length variable at location b of a given distribution
S(V)	Length variable as a function of the environmental variable
S_{bI}	Value of the length variable at bottom of an inversion
S_{bV}	Value of the length variable at bottom of a varicline
S_{cV}	Value of the length variable at center of a varicline
S_{eI}	Value of the length variable at point of symmetry of an inversion
S_{tI}	Value of the length variable at top of an inversion
S_{tV}	Value of the length variable at top of a varicline
ΔS_I	Range (length) of an inversion
ΔS_V	Range (length) of a varicline
σ	Standard deviation

T	Temperature variable (environmental variable)
T_n	Nth value of temperature read off a trace
ΔT_T	Thermal magnitude of a thermocline
V	Environmental variable
V_n	Nth value of V read off a trace
$V(V_n)$	Nth value of V read off a trace
$V(a)$	Value of V at location a of a given distribution
ΔV	Magnitude of the mesh increment
ΔV_I	Variable magnitude of an inversion
ΔV_V	Variable magnitude of a varicline
XBT	Expendable Bathythermograph
Z	Depth (length variable) increases downwards from the sea surface
$Z(T)$	Depth as a function of temperature
Z_{bT}	Depth of the bottom of a thermocline
Z_{cT}	Depth of the center of a thermocline
Z_{tT}	Depth of the top of a thermocline

ACKNOWLEDGEMENT

This research was conducted under the advice of Dr. Dale F. Leipper, Chairman of the Department of Oceanography, Dr. Taivo Laevastu of the Fleet Numerical Weather Central, Dr. Noel Boston of the Department of Oceanography, and Dr. Kenneth Davidson of the Department of Meteorology.

The project required utilization of the Computer Facility of the Naval Postgraduate School and environmental data from the Fleet Numerical Weather Central in Monterey, whose staffs were particularly helpful and dedicated.

I would like to express my sincere thanks to my advisors, faculty and members of the above facilities, classmates and students, to everyone who really helped me to develop human and environmental understanding.

I. OBJECTIVE OF THE THESIS

The study of oceanographic phenomena occurring in an area west of the Strait of Gibraltar from Cabo Finisterra (in Europe) to Canary Islands (in Africa) (Figure 1) based on climatological information, constituted the initial purpose of this thesis.

Later, the availability of oceanographic and meteorologic data from Fleet Numerical Weather Central (in Monterey), broadened the descriptive nature of the above intention giving opportunity to investigate the variations of the oceanic thermal structure taking place in the upper layers.

The large amount of accessible data, namely expendable bathythermograph traces and surface uncorrected geostrophic wind fields, were processed using a digital model and presented in a form simple to analyze. Initially, the model was set up to study the distribution of temperature with depth; soon, it became evident that it could be applied to other studies dealing with the continuous vertical distribution of any environmental parameter (sound velocity, salinity, light extinction, etc.).

The lack of information about air-sea temperature differences made impossible the transformation of the uncorrected geostrophic wind into "friction velocity" and cut short the last part of the study, which was to establish the statistical interrelation between the corrected wind information and the thermal features given by the objective analysis of traces.

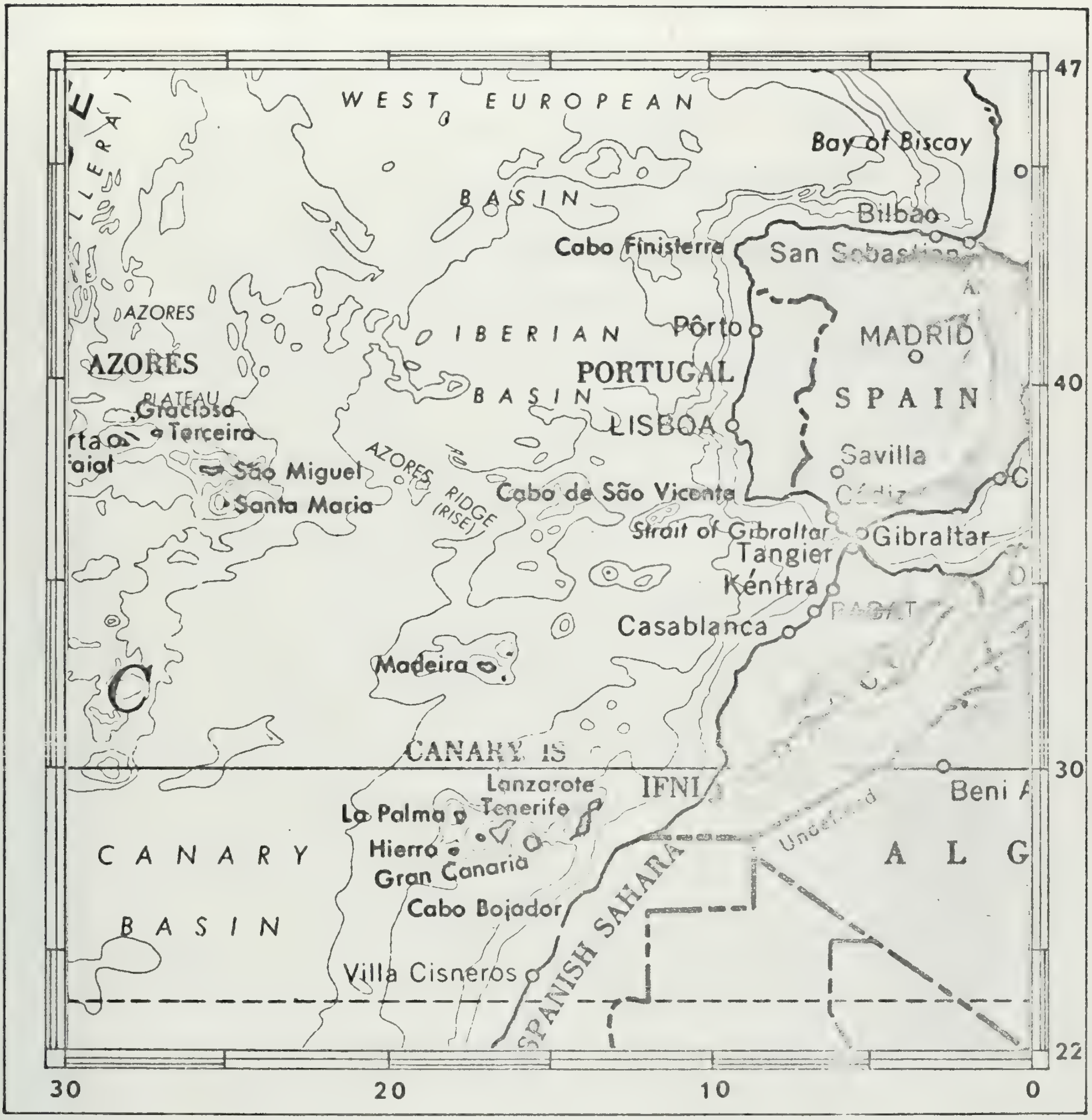


Figure 1. Eastern North Atlantic Ocean

The new objective of the thesis then is defined in two parts:

a. To develop a digital model to analyze the continuous vertical distribution of any environmental parameter, to process the wind data and to present the generated information in a tractable form to reveal geophysical phenomena and their statistical connections.

b. To test the capability of the computer program by means of data available.

II. OBJECTIVE ANALYSIS OF DIGITAL TRACES

A. REVIEW OF PRESENT IDEAS

1. Introduction

The mechanical bathythermograph, BT, first introduced by Rossby and Montgmory [1935] and developed by Spilhaus [1938] provided one of the first successful attempts to obtain a continuous graphical record of an oceanic variable, namely the distribution of temperature with depth. The expendable bathythermograph, XBT, introduced by Francis and Campbell [1965] increased the continuous thermal record to a maximum depth of 457 meters. More sophisticated devices with greater depth ranges have been used to record continuously the vertical distribution of salinity, temperature and sound velocity as discussed by Magruder [1970]. Recently, Tucker and Shepard [1971] presented results from continuous registration of light transmission versus depth.

Continuous horizontal distribution of current speed, temperature, bathymetry and gravity are among the other variables that have been measured with sensors which are carried or towed in one direction at the sea surface or along other horizontal reference planes below or above the sea surface.

It is important to note that although these distributions are usually labelled as temperature (or other variable) versus depth (or other horizontal or vertical axis), the independent variable is the depth or the

successive positions where the sensor measures the oceanographic variable. A significant characteristic of a continuous distribution of a given variable along a length axis compared to time distributions is the absence of regularity or law of repetition, which means that distinct phenomena or different intensities of the same phenomenon are taking place along the sensed axis and makes impossible the application of time series analysis techniques.

The small cost of some equipment, namely BTs, their reliability and easy maintenance, together with a recording operation that does not require great ability from an observer led to widespread use; an immediate consequence was enormous accumulation of BT traces.

Recent development in automatic processing techniques of bathythermographic data is due to LaFond's [1951] pioneering efforts to establish a standardized method for processing BT traces and in predicting the need for automatic technology. Sauer [1964] described the Bathythermograph Card Processor which automatically transposes the BT record to an aperture card. Sauer and Hope [1967] gave a detailed account of the "BT Digitizer" operation which produces semiautomatic digitization of BT aperture cards. Dale and Stevens [1970] presented the Fleet Numerical Weather Central technique to digitize the XBT traces by the definition of the linear portions of the curve.

The analysis of a continuous record, BT or XBT, requires an acute visual sense, experience and superior

understanding of oceanographic processes. The researcher looks for characteristic geometric features of the trace; usually he follows the evolution of one feature throughout a set of records to assure consistency of reasoning. The human system is slow, inaccurate for subtle variations, reactive to routine procedures and subjective.

Efforts towards an automatic analysis of BT traces have been described among others by Boston [1966] who identified the most significant features in the thermal structure of oceanic upper layers and derived mathematical schemes to define the depth of the mixed layer, single and multiple thermoclines, transients and inversions. The mathematical schemes are based on: (1) an assumption of Gaussian or Normal distribution of temperature as a function of depth (Figure 2), and (2) a finite difference method. Grosfils [1968] translated into a computer program the schemes above outlined and Denham [1969] applied the same principles to a digital analysis of internal waves. Boston initially developed and applied his method while examining nearshore internal waves.

2. Evaluation of Boston's Concepts

The first part of this thesis generalizes some of Boston's [1966] concepts and their presentation. An evaluation will outline and hlep to understand the purpose of the present research.

a. Significant Features in the Thermal Structure

Boston defined the significant features of the thermal structure obtainable with a BT within the upper 300 meters of the ocean as

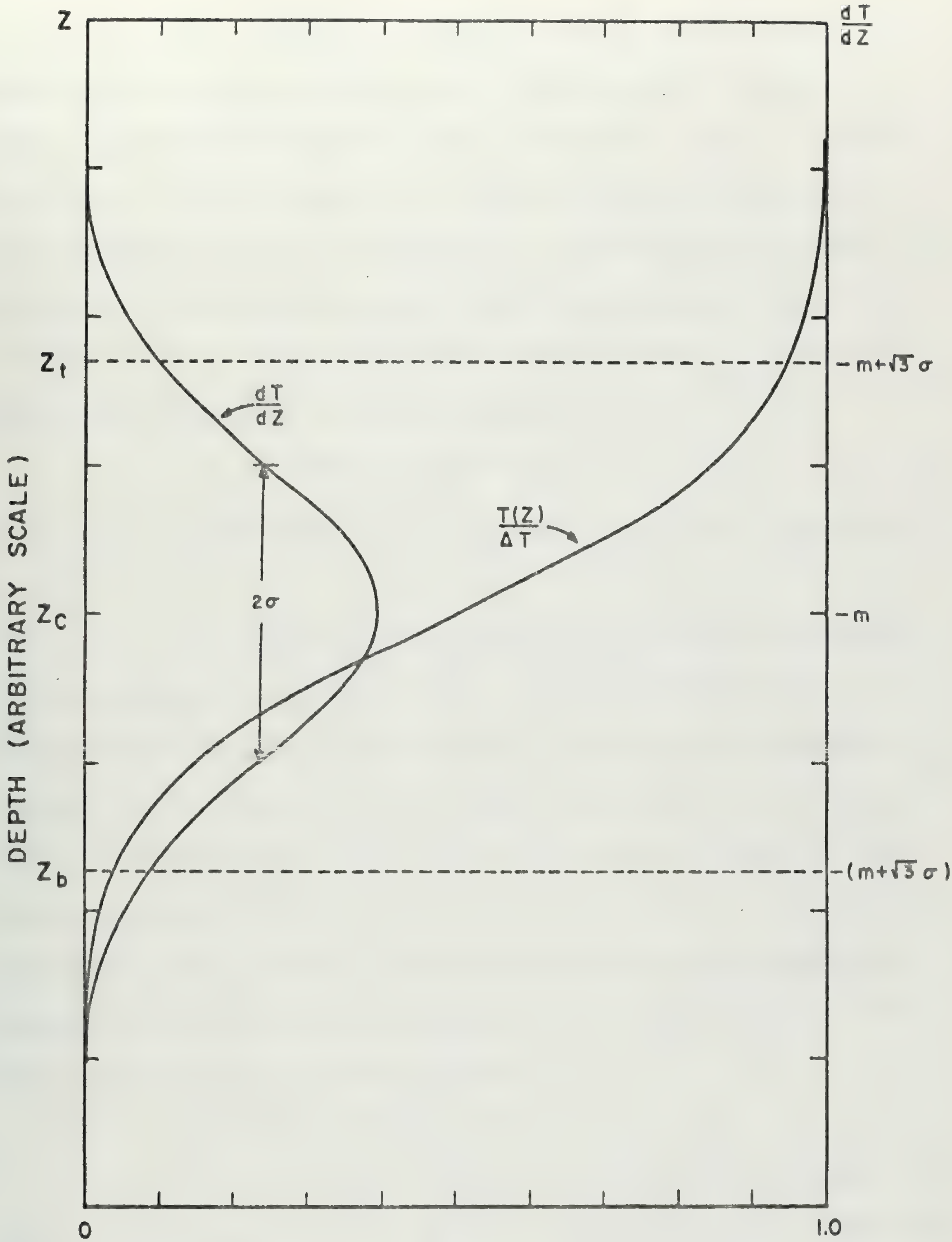


Figure 2. The Gaussian Thermocline and its Frequency Function [Boston, 1966]

(1) Primary features or those characteristics common to most oceanic regions. The thermal structure is simply approximated by an upper isothermal layer, a thermocline and a lower region of nearly constant temperature. Three parameters describe this type of structure, namely (a) the depth of the mixed layer (upper isothermal), Z_{tT}, (b) the depth of the "bottom" of the thermocline, Z_{bT}, and (c) the thermal magnitude of the thermocline is

$$\Delta T_T = T_{tT} - T_{bT} \tag{1}$$

if T_{tT} is the temperature at Z_{tT} and T_{bT} is the temperature at Z_{bT}.

(2) Secondary features are variations occurring in the primary structure at any point and classified as (a) transients or thermal fluctuations less than one degree Celsius and localized at the upper isothermal layer, (b) multiple thermoclines which are the result of heating and mixing conditions and (c) inversions or a type of structure occurring below the main thermocline where the temperature decreases with depth to a minimum, then increases and sometimes regains the decreasing trend.

b. Analytical Techniques

The analytical methods developed by Boston are based on (1) a Gaussian or Normal distribution of temperature with depth, and (2) a finite difference approach.

(1) The Gaussian distribution is described by the well-known frequency function, Figure 2,

$$\frac{dT(Z)}{dZ} = \frac{1}{\sqrt{2\sigma\pi}} e^{-\frac{(Z-m)^2}{2\sigma^2}} \qquad (2)$$

where σ and m are constants. The function (2) has (a) symmetry about $Z = m$, (b) points of inflection at $Z = m \pm \sigma$ and (c) maximum rate of change at $Z = m \pm \sigma\sqrt{3}$. Application of the Gaussian distribution requires that the central moments of odd order be zero.

(2) The finite difference scheme implies the substitution of analog trace $Z(T_n) = f(T_n)$ by a discrete set of successive positions $(T_n, Z(T_n))$ obtained through a constant increment, ΔT, of the independent variable, T_n. The first finite difference of $Z(T_n)$ will be defined as

$$\Delta Z(T_n) = Z(T_{n+1}) - Z(T_n) \qquad (3)$$

and the second finite difference as

$$\Delta^2 Z(T_n) = \Delta Z(T_{n+1}) - \Delta Z(T_n) \qquad (4)$$

The value of the increment ΔT will vary according to the desired accuracy of the scheme.

c. Application of the Analytical Techniques to the Thermal Structure

The above concepts may be applied to the analysis of the thermal structure. It is assumed that the BT trace is replaced by a discrete set of ordered points (T,Z) from the surface and taken at equal thermal increment, ΔT.

(1) The Gaussian Thermocline. In accordance with the Normal or Gaussian distribution of temperature with

depth, $T\left(\frac{Z-m}{\sigma}\right)$, (a) the center of the thermocline is located at

$$Z_{cT} = -m \qquad (5)$$

(b) the top and the bottom of the thermocline are considered to be at the points of the maximum rate of change of the slope of the $T\left(\frac{Z-m}{\sigma}\right)$ curve or at

$$Z_{tT} = -m + \sigma\sqrt{3} \qquad (6)$$

$$Z_{bT} = -m - \sigma\sqrt{3} \qquad (7)$$

(2) The Non-Gaussian Thermocline. The analysis of the non-Gaussian thermocline makes use of the variation in sign of the second finite difference scheme, equation (4), to find the point of inflection of the (T,Z) curve. This point identifies the center of the thermocline, Z_{cT}. The top of the thermocline, Z_{tT}, is determined using the technique outlined in (1) and generating an artificial Gaussian distribution around Z_{cT} by reflection of the upper part of the trace ($Z \leq Z_c T$). The bottom of the thermocline, Z_{bT}, is obtained in the same way but making use of the lower part of the (T,Z) curve, ($Z \geq Z_{cT}$). Figure 3 shows the successive steps of this analysis.

(3) Transients are recognized by the sign changes of the second finite difference, $\Delta^2 Z(T_n)$. A sign change from positive to negative signifies the end of one transient and the beginning of the next one.

(4) Multiple Thermoclines. An examination of the sign changes of the second finite difference, as made in

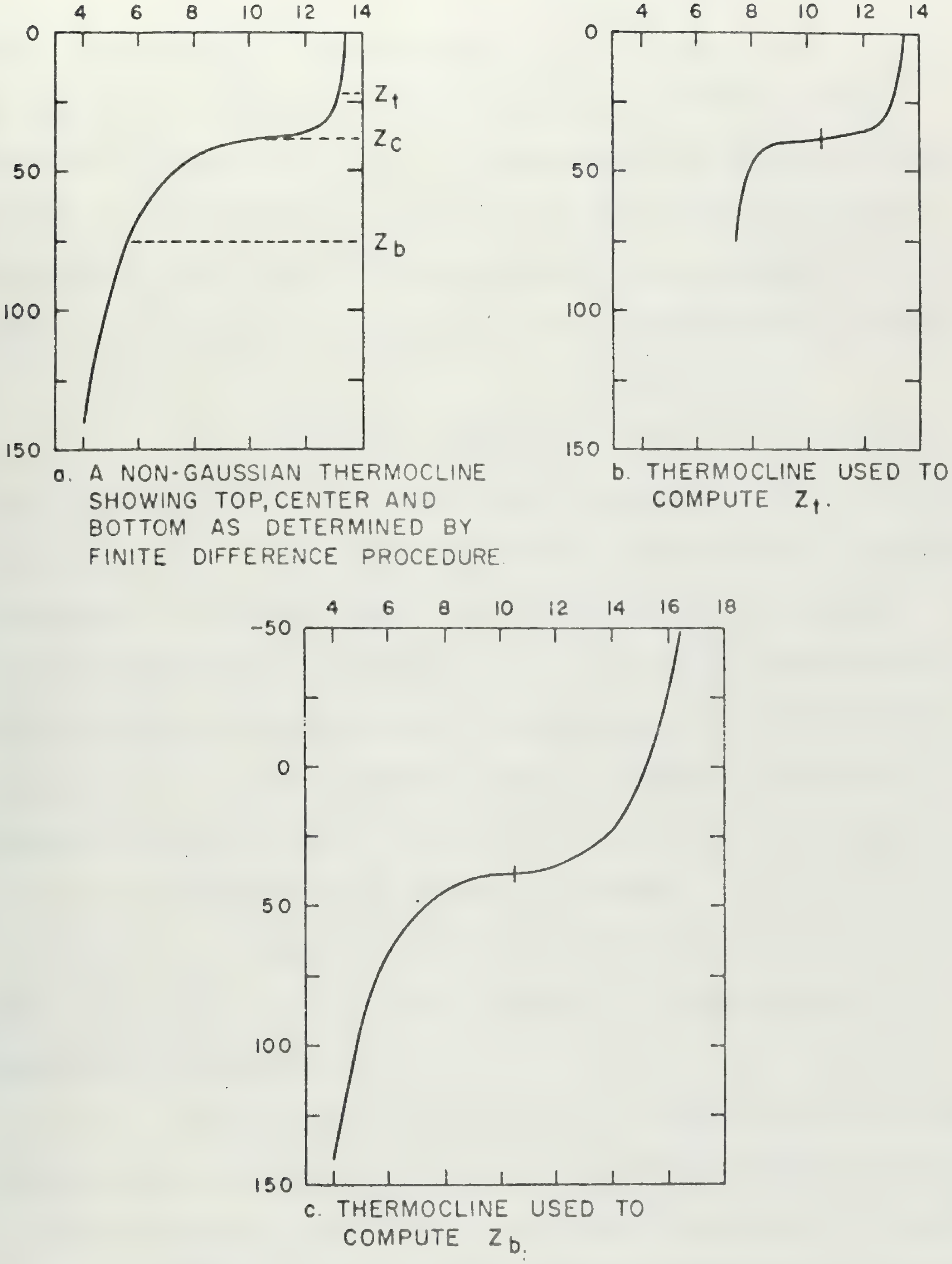


Figure 3. A Non-Gaussian Thermocline Showing Top, Center, and Bottom as Determined by Finite Difference Procedure [Boston, 1966]

(2), will reveal the points of inflection of the (T,Z) curve. These correspond to the centers of the thermoclines. The top and the bottom of each thermocline are found applying the method explained in (2). The separation between thermoclines is determined by the location of successive points where the first finite difference attains a maximum value. Examples of multiple thermoclines are illustrated in Figure 4.

(5) Inversions. The sign change of the first finite difference of the (T,Z) curve indicates the existence of an inversion. The upper and lower limits of any inversion delineate a continuum where the representative trace is double-valued (same temperature at two depths) but outside the region the distribution of temperature with depth is single-valued (one temperature and one depth). The main features investigated by this method are shown in Figure 5.

d. Remarks about Boston's Concepts

(1) The problem of automatic processing of bathythermograph data is well stated and represents a remarkable contribution.

(2) The objective digital analysis of the thermal structure requires application of several methods, each one is independent from the others and each describes a distinct thermal feature. More general objective schemes to analyze other features in the thermal structure would be even more useful and might be extended to the objective study of other oceanographic variables with vertical or horizontal axial distributions.

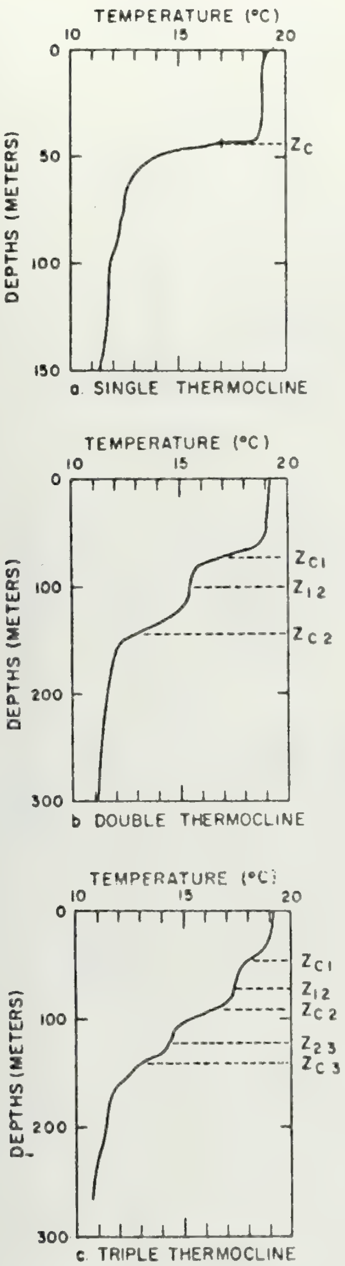


TABLE IV. Multiple thermocline analysis.

Table IV(a).

n	T_n	$z(T_n)$	$\Delta z(T_n)$	$\Delta^2 z(T_n)$	
1	19	5	39		
2	18	44	0	-39	
3	17	44	1	1	z_c
4	16	45	1	0	
5	15	46	4	3	
6	14	50	8	4	
7	13	58	32	24	
8	12	90			

Table IV(b).

n	T_n	$z(T_n)$	$\Delta z(T_n)$	$\Delta^2 z(T_n)$	
1	19	40	28		
2	18	68	4	-24	
3	17	72	8	-4	z_{c1}
4	16	80	40	32	z_{12}
5	15	120	14	-26	
6	14	134	10	-4	
7	13	144	10	0	z_{c2}
8	12	154			

Table IV(c).

n	T_n	$z(T_n)$	$\Delta z(T_n)$	$\Delta^2 z(T_n)$	
1	19.0	28	10		
2	18.5	38	6	-4	
3	18.0	44	12	-6	z_{c1}
4	17.5	56	30	18	z_{12}
5	17.0	86	4	-26	
6	16.5	90	4	0	z_{c2}
7	16.0	94	4	0	
8	15.5	98	4	0	
9	15.0	102	10	6	
10	14.5	112	18	8	z_{23}
11	14.0	130	6	-12	
12	13.5	136	4	-2	
13	13.0	140	10	-6	z_{c3}
14	12.5	150	10	0	
15	12.0	160	20	10	
16	11.5	180	50	30	
17	11.0	230			

Figure 4. Thermal Structure with One, Two, and Three Thermoclines [Boston, 1966]

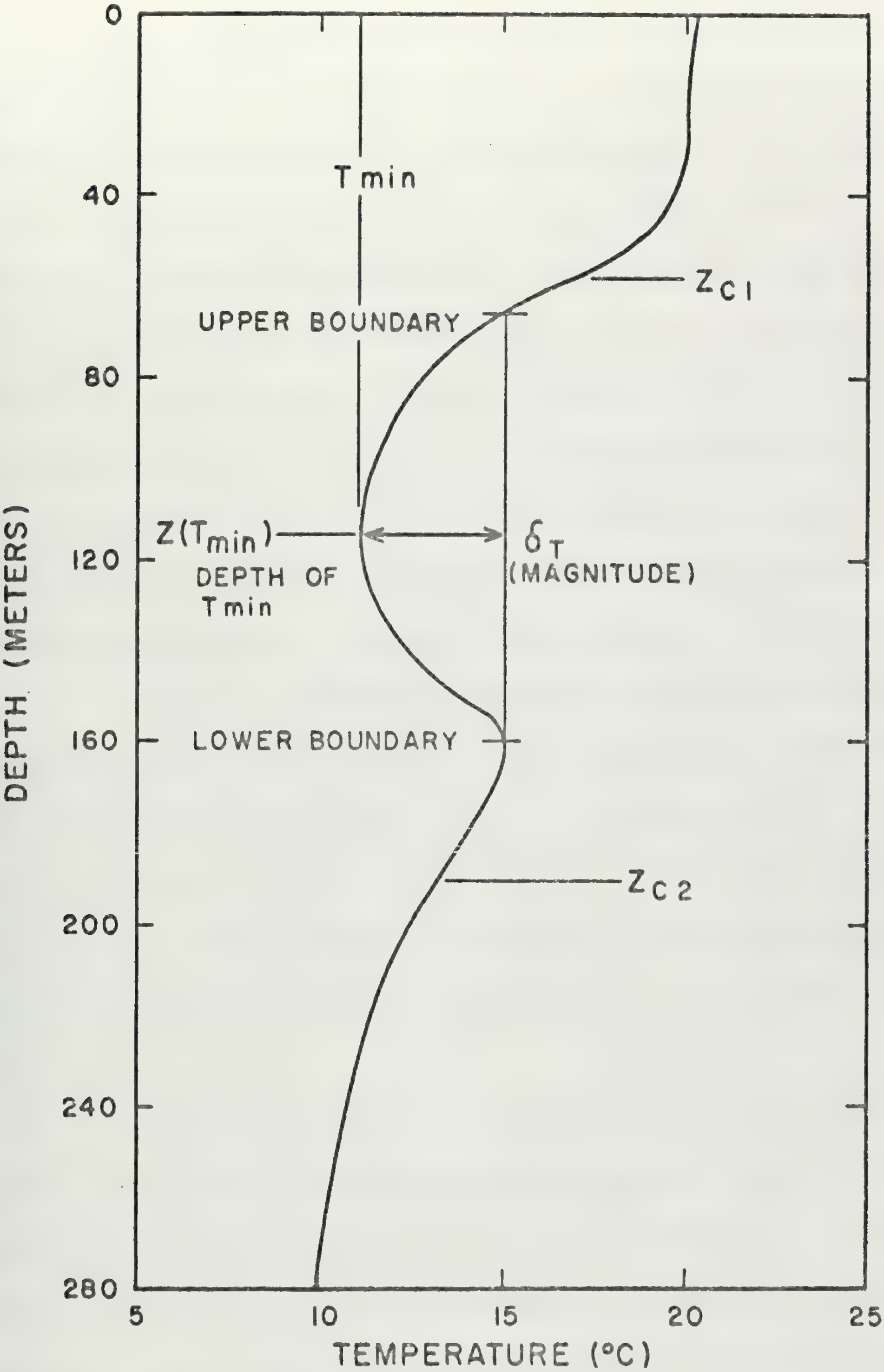


FIGURE 5. Inversion Illustrating Main Features [Boston, 1966]

(3) The Gaussian or normal distribution of temperature with depth constitutes an interesting investigation. Its application to the definition of a thermocline is restricted depending upon whether or not a condition of symmetry around the first moment does exist. If this provision is not fulfilled there is not any thermal normal distribution with depth. However, in this case of non-Gaussian thermocline the top and the bottom of the thermocline are determined by means of an artificial reflected construction of normal distribution around the center which distorts slightly the initial information.

(4) The finite difference technique explores conveniently the behavior of first and second order schemes which give a familiar feeling for fluctuations of slope and points of inflection of the original trace. The development and study of higher order schemes may indicate interesting variations associated with significant features of any curve.

(5) The digitization of the analog trace by the selection of depths at uniform temperature intervals appears to be advantageous because the data is in a form where it may readily be processed by the finite difference technique. The thermal increment should be chosen as small as possible but compatible with the accuracy of the sensor.

B. AN OBJECTIVE MODEL

The primary purpose of this chapter is the presentation and development of general concepts which may be utilized

in the objective analysis of any given two-dimensional trace.

1. Significant Features in a Two-dimensional Trace

A two-dimensional trace presents the distribution of any oceanographic variable, V, with depth or along a horizontal axis, S, as shown in Figure 6. A pecularity of the trace consists in the continuous increase in S with fluctuations in V. Such curves (V,S) show three general characteristics, namely: (1) trend, (2) varicline, and (3) inversion.

a. Trend

The trend indicates the general direction of development of the curve (V,S) in Figure 6: (1) between the beginning of the record, a, and g, the variable V is decreasing at different rates to the left of the figure; (2) g - h, the variable V is increasing and the trace is directed to the right; and (3) the final position of the curve between h and i again shows a decrease of V.

b. Varicline

Varicline was derived from the composition of the words: relative "variation" of the slope or "inclination"; it identifies portions of the curve where the slope increases relative to the neighbor regions. In Figure 6, parts b - c and d - e of the curve are variclines. Considering varicline d - e, the limiting parameter values are (1) the top, b,

$$S_{tV} = S(d) \tag{8}$$

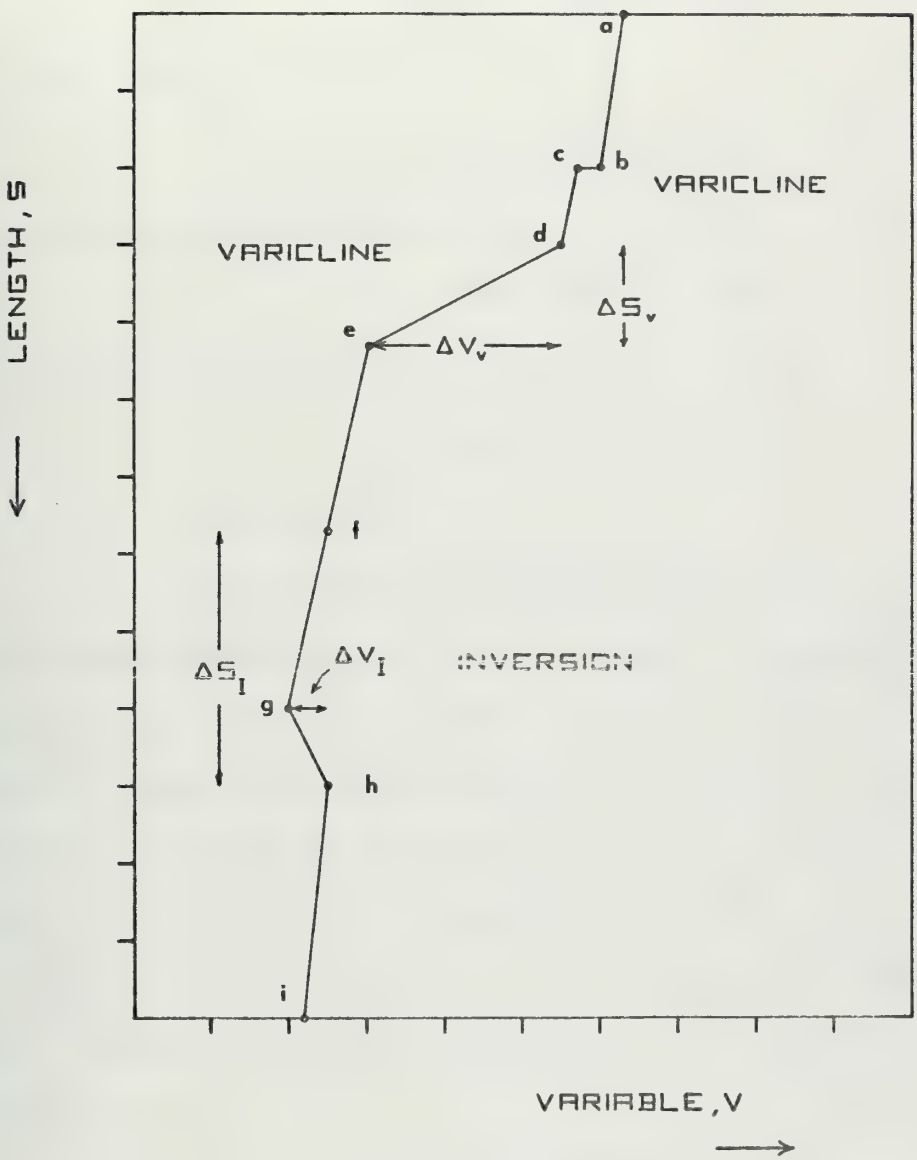


Figure 6. A Two-dimensional Trace

(2) the bottom, e,

$$S_{bV} = S(e) \qquad (9)$$

(3) the range, ΔS_V,

$$\Delta S_V = S(e) - S(d) \qquad (10)$$

(4) the center,

$$S_{cV} = S(d) + \Delta S/2 \qquad (11)$$

(5) the variable magnitude, ΔV_V,

$$\Delta V_V = V(e) - V(d) \qquad (12)$$

and (6) the gradient,

$$G_V = \Delta V_V / \Delta S_V \qquad (13)$$

c. Inversion

An inversion identifies a segment of the curve (V,S) where there exists an approximately symmetrical distribution of the oceanographic variable, V with length S around a point, namely location g in Figure 6. The point of symmetry is found by examination of the variation of the trend: it is a relative maximum or minimum of V. Referring to Figure 6 again, the point g constitutes a relative minimum. The most important parameters of an inversion are (1) the top, f,

$$S_{tI} = S(f) \qquad (14)$$

(2) the bottom, h,

$$S_{bI} = S(h) \qquad (15)$$

(3) the location of the extreme value of V in the inversion, g,

$$S_{eI} = S(g) \qquad (16)$$

(4) the variable magnitude, ΔV_I

$$\Delta V_I = |V(g) - V(f)| \quad (17)$$

$$= |V(g) - V(h)| \quad (18)$$

and (5) the range, ΔS_I,

$$\Delta S_I = S(h) - S(f) \quad (19)$$

2. The Finite Difference Technique

The finite difference technique investigates the variation of a two-dimensional distribution. The original analog record is previously digitized according to semi-automatic procedures. The digital trace consitutes an ordered and discrete set of positions.

a. The Preparation of the Mesh

The finite difference technique requires an organized two-dimensional distribution or mesh such that the values of one variable V, chosen as independent, are read at constant interval ΔV together with the correspondent values of the other variable S(V), dependent, Figure 7. The magnitude of the increment ΔV is directly related to the accuracy of the record and it is unrealistic to select a small but incompatible value. The digital trace may present the structure as a satisfactory mesh with values at constant intervals; if not, it must be altered accordingly. Assuming that the digital trace has a linear variation, the mesh points will be obtained by a simple linear interpolation. However, the selected constant interval may introduce an alteration to the original trace and ΔV must be small enough as shown in Figure 8.

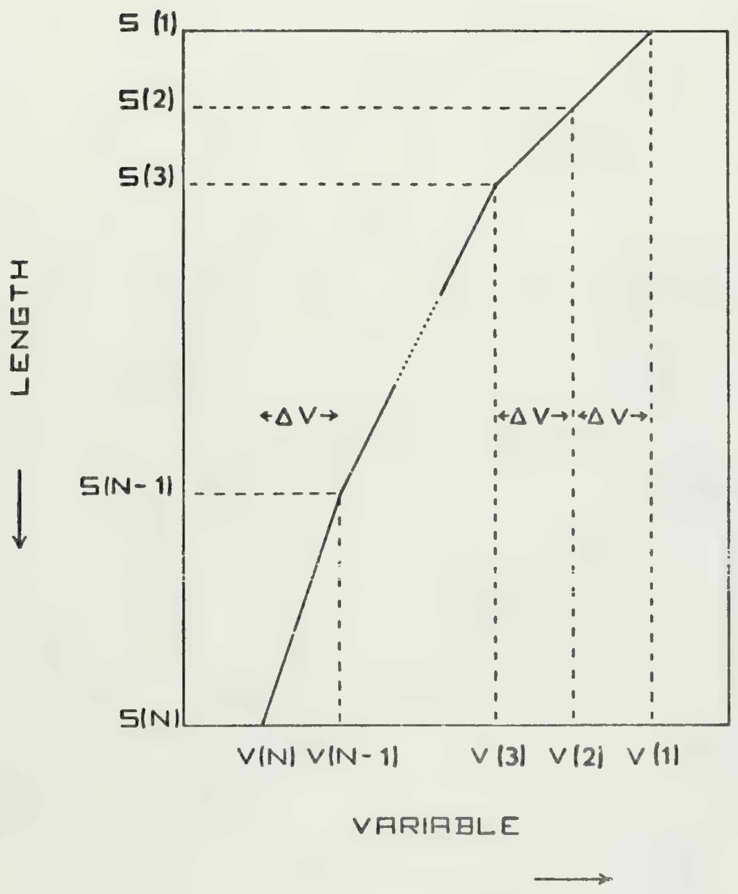


Figure 7. The Mesh

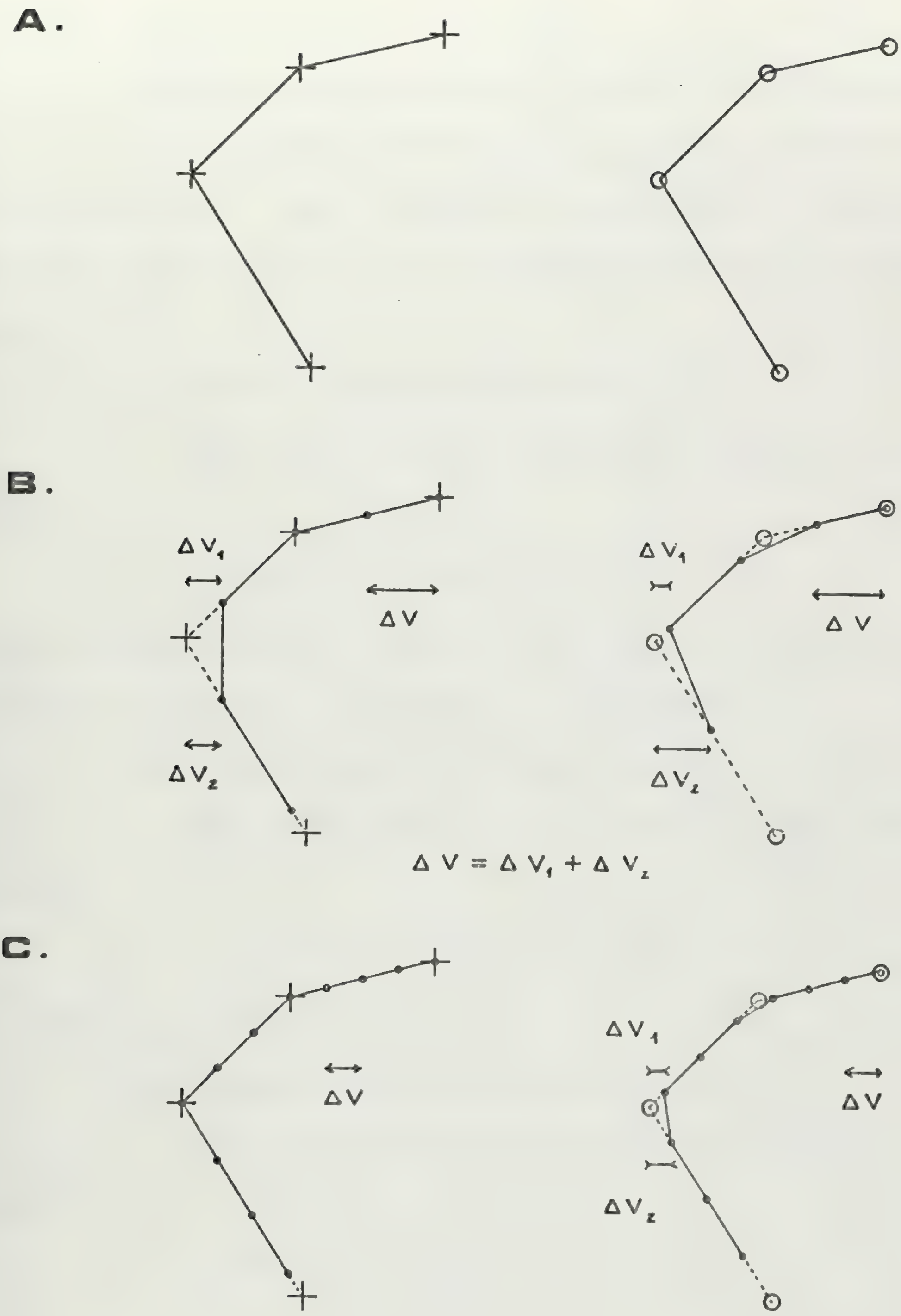


Figure 8. The Linear Interpolation of a Digital Trace: A. Digital Trace, B. Mesh (increment V = 2 arbitrary units), C. Mesh (increment V = 1 arbitrary unit)

b. Finite Difference Schemes

A mesh consists of an ordered set of points at fixed intervals of variable and two consecutive positions define one linear segment. Each segment is specified by (a) the length component and (b) the variable component as illustrated in Figure 9A.

(1) First Finite Difference of V

The first finite difference of $V(V_n)$ is defined as

$$\Delta V(V_n) = V(V_{n+1}) - V(V_n) \tag{20}$$

and may be positive (pointing to the right) or negative (directed to the left), as shown in Figure 9C.

(2) First Finite Difference of S

The first finite difference of $S(V_n)$ is stated as

$$\Delta S(V_n) = S(V_{n+1}) - S(V_n) \tag{21}$$

and is always positive, Figure 9B.

(3) Second Finite Difference of S

The second finite difference of $S(V_n)$ is described as

$$\Delta^2 S(V_n) = \Delta S(V_{n+1}) - \Delta S(V_n) \tag{22a}$$

$$= S(V_{n+2}) - 2S(V_{n+1}) + S(V_n) \tag{22b}$$

and employs three points, n, n+1, n+2, from the mesh. The three points define two linear segments, whose relative arrangement will be related to the negative, positive or null value of $\Delta^2 S(V_n)$, Figure 10.

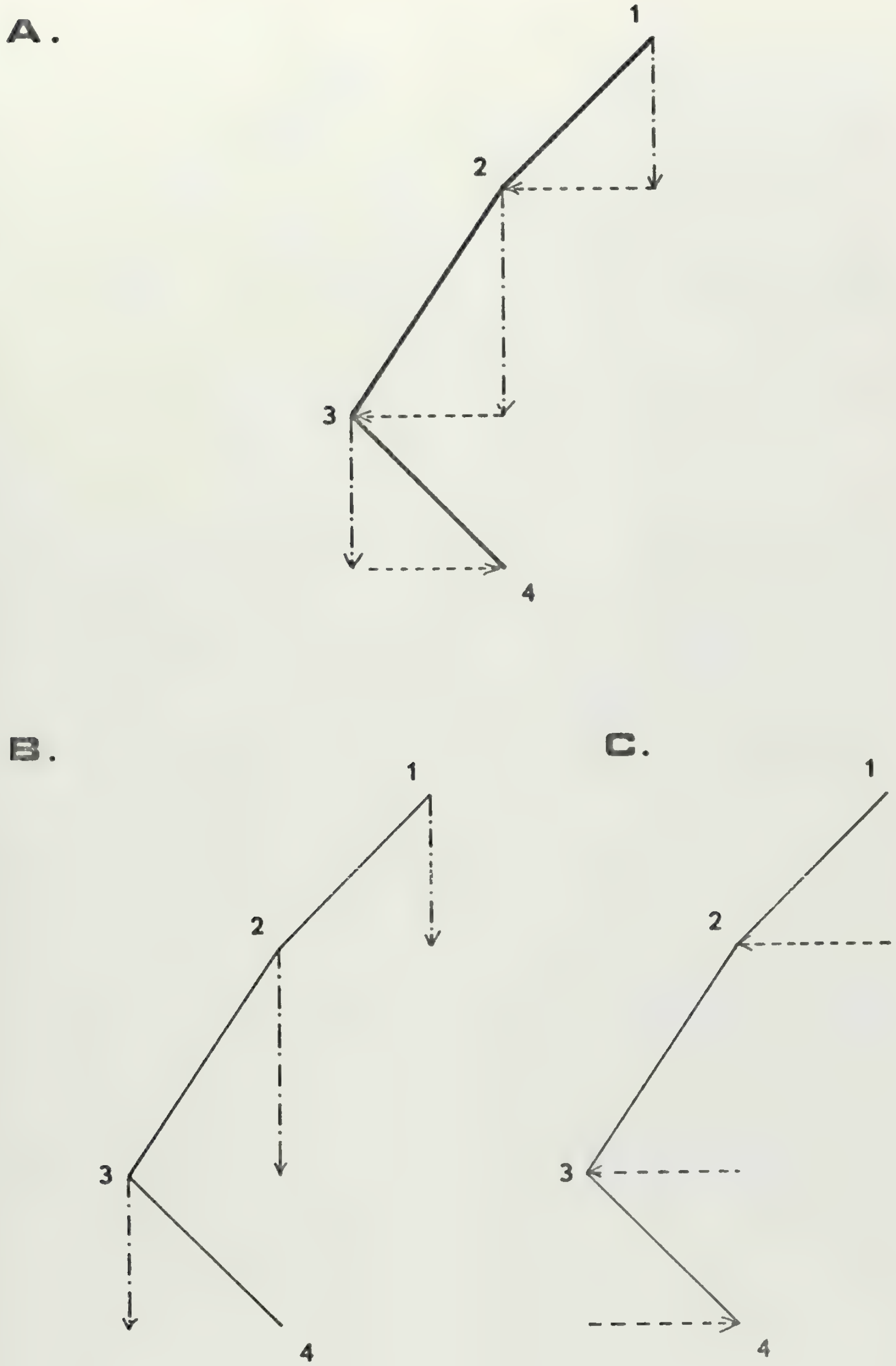
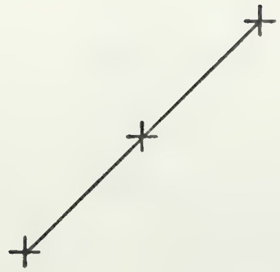


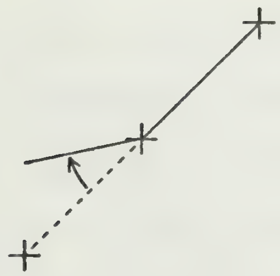
Figure 9. The Mesh and Its Components:
A. Segments of the Mesh
B. Length Components
C. Variable Components

A.



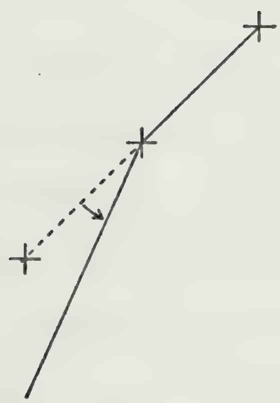
	S(V)	Δ	Δ^2
N	10	15	0
N+1	25		
N+2	40	15	

B.



	S(V)	Δ	Δ^2
N	10	15	-12
N+1	25		
N+2	28	3	

C.



	S(V)	Δ	Δ^2
N	10	15	18
N+1	25		
N+2	58	33	

Figure 10. Schemes of the Second Finite Differences: A. Constant Slope, B. Increasing Slope, C. Decreasing Slope

(4) Third Finite Difference of S

The third finite difference of $S(V_n)$ is specified as

$$\Delta^3 S(V_n) = \Delta^2 S(V_{n+1}) - \Delta^2 S(V_n) \quad (23a)$$

$$= \Delta S(V_{n+2}) - 2\Delta S(V_{n+1}) + \Delta S(V_n) \quad (23b)$$

$$= S(V_{n+3}) - 3S(V_{n+2}) - 3S(V_{n+1}) + S(V_n) \quad (23c)$$

and requires four points, n, n+1, n+2, n+3, from the mesh. The $\Delta^3 S(V_n)$ summarizes the relative disposition between three linear segments, defined by four points. Five basic schemes of the $\Delta^3 S(V_n)$ are presented in Figure 11.

(5) Higher Order Finite Difference of S

Higher order finite difference of S,

$$\Delta^k S(V_n) = \Delta^{k-1} S(V_{n+1}) - \Delta^{k-1} S(V_n) \quad (24)$$

will demand more points from the mesh, namely the number of the order plus one and concurrently the analysis of the relative structure will increase in complexity. The development of the finite difference schemes up to the third order seems to constitute adequate advancement for the operational needs in the present research.

3. Two-dimensional Objective Analysis

The finite difference schemes above introduced, will be applied to the significant features of a two-dimensional trace and illustrated by schematic diagrams.

a. Trend

As explained before, the trend shows the tendency of the independent variable, V, which is investigated by the

A.

	S (V)	Δ	$Δ^2$	$Δ^3$
N	10	10	0	
N+1	20	10		0
N+2	30		0	
N+3	40	10		

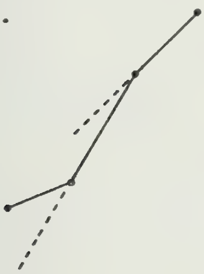
B.

	S (V)	Δ	$Δ^2$	$Δ^3$
N	10	10	-3	
N+1	20	7		-1
N+2	27		-4	
N+3	30	3		

C.

	S (V)	Δ	$Δ^2$	$Δ^3$
N	10	10	2	
N+1	20	12		2
N+2	32		4	
N+3	48	16		

D.



	S (V)	Δ	$Δ^2$	$Δ^3$
N	10	10	7	
N+1	20	17		-20
N+2	37		-13	
N+3	41	4		

E.

	S (V)	Δ	$Δ^2$	$Δ^3$
N	10	10	-5	
N+1	20	5		12
N+2	25		7	
N+3	37	12		

Figure 11. Schemes of the Third Finite Differences: A. Constant Slope, B. Successive Increase of the Slope, C. Successive Decrease of the Slope, D. Successively Decreasing and Increasing the Slope, E. Successively Increasing and Decreasing the Slope

use of its first finite difference, $\Delta V(V_n)$ expressed in (20). Assuming that V increases to the right, as shown in Figure 6, the analysis of V may be summarized as follows:

(1) Variation

(a) $\Delta V(V_n) < 0$: $V_{n+1} < V_n$, V is decreasing or directed to the left of the graph.

(b) $\Delta V(V_n) = 0$: $V_{n+1} = V_n$, V is constant or parallel to the length axis.

(c) $\Delta V(V_n) > 0$: $V_{n+1} > V_n$, V is increasing or tending to the right.

(2) Extreme values

The change in sign of the first finite difference of V indicates that the trend has changed direction and defines an extreme feature:

(a) $V(V_{n+1}) < 0$ and $V(V_n) > 0$: relative maximum at point n+1.

(b) $V(V_{n+1}) > 0$ and $V(V_n) < 0$: relative minimum at point n+1.

(c) The largest and the least of the V values above found with the values of V at the beginning and end of the trace give the absolute maximum and minimum of the trace.

b. Varicline

The basic concept of the objective varicline is presented in Figure 11E composed of three segments and, although their relative arrangement is unchangeable the middle part can be divided into more than one segment. The analysis of the characteristics,

(1) $\Delta^2 V(V_n) < 0$: top of varicline at point n+1.

(2) $\Delta^2 V(V_{n+1}) > 0$: base or bottom of varicline at point n+2 is indicated in Figure 11.

c. Inversion

The analysis of the trend indicates whether or not there are maxima and minima present in the record. The maxima and minima (excluding the values of V in the beginning and the end of the trace) evidence areas of approximate symmetry of V around the extrema. Referring to Figure 6 the analysis of the trend gave (a) the absolute minimum at g, (b) a relative maximum at h, and (c) the absolute maximum value in V was found at a, the beginning of the trace. It is evident that locations f and h define a region of approximate symmetry of V around the relative minimum g. An inversion is characterized by a portion of the (V,S) curve where the trend changes sign, illustrated by the layer g - h in Figure 6. The limits of the approximate symmetry are established by comparison of the variable magnitudes, ΔV's (17 or 18) determined using the point of symmetry of the extrema in consideration, g, and the adjacent positions defining extreme values in V, namely a and h. The least ΔV is chosen as the variable magnitude of the inversion.

d. Complex type

The complex type of trace combines the basic investigations of trend, variclines and inversions. The method of procedure is outlined in three stages:

(1) Objective analysis of the trend.

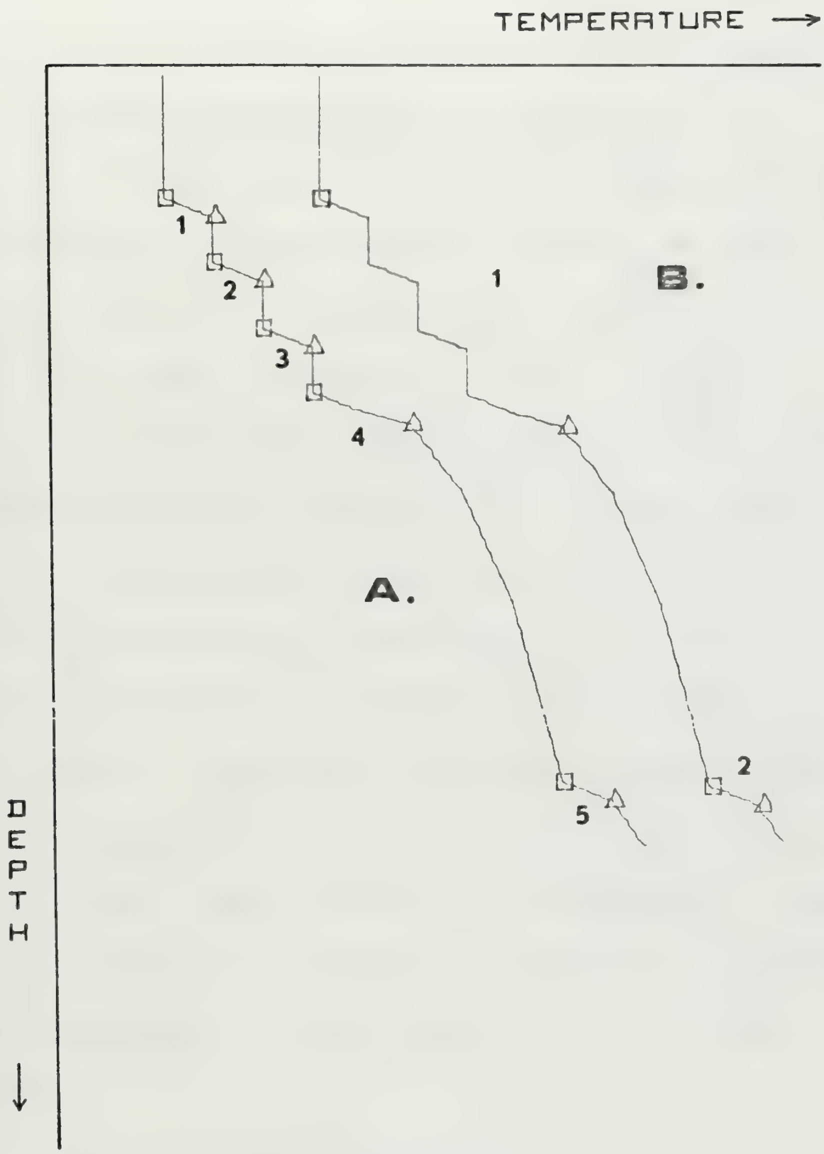


Figure 12. Computer Output Plot of a Digital Trace Showing the Tops (squares) and the Bottoms (triangles) of Analyzed Variclines. (The control of the conditions limiting the bottom of a varicline allows: A. identification of all the variclines present, and B. definition of the significant features)

(2) Objective analysis of inversions.

(3) Objective analysis of variclines, which may be conducted throughout the entire trace or be restricted to those portions outside the symmetrical regions.

4. Tri-dimensional Objective Analysis

The two-dimensional objective analysis studies the distribution of a given variable V along a length axis S. The availability of a collection of traces spread out in time, D, or along a horizontal, L, or vertical, S, direction illustrated in Figure 13, makes possible the investigation of a tri-dimensional distribution of V in (a) length (V,S,L) and in (b) length and time (V,S,D). An objective analysis of this tri-dimensional distribution is possible if the law of variation of V is known or the interval (in time or length) between consecutive traces is small enough to assume a linear interpolation or digitization of the new trace. One of the above requirements is fundamental to generate a mesh and to apply the concepts of the two-dimensional objective analysis to each one of the new traces, shown in Figure 13B.

5. The Problem of Filtering

Usually a trace shows a complicated form with small variations superimposed on a general pattern. The small fluctuations may have physical significance or may simply represent an artificial response of the sensor; however, it may be impossible to distinguish whether a regular feature is valid or invalid. The most common approach

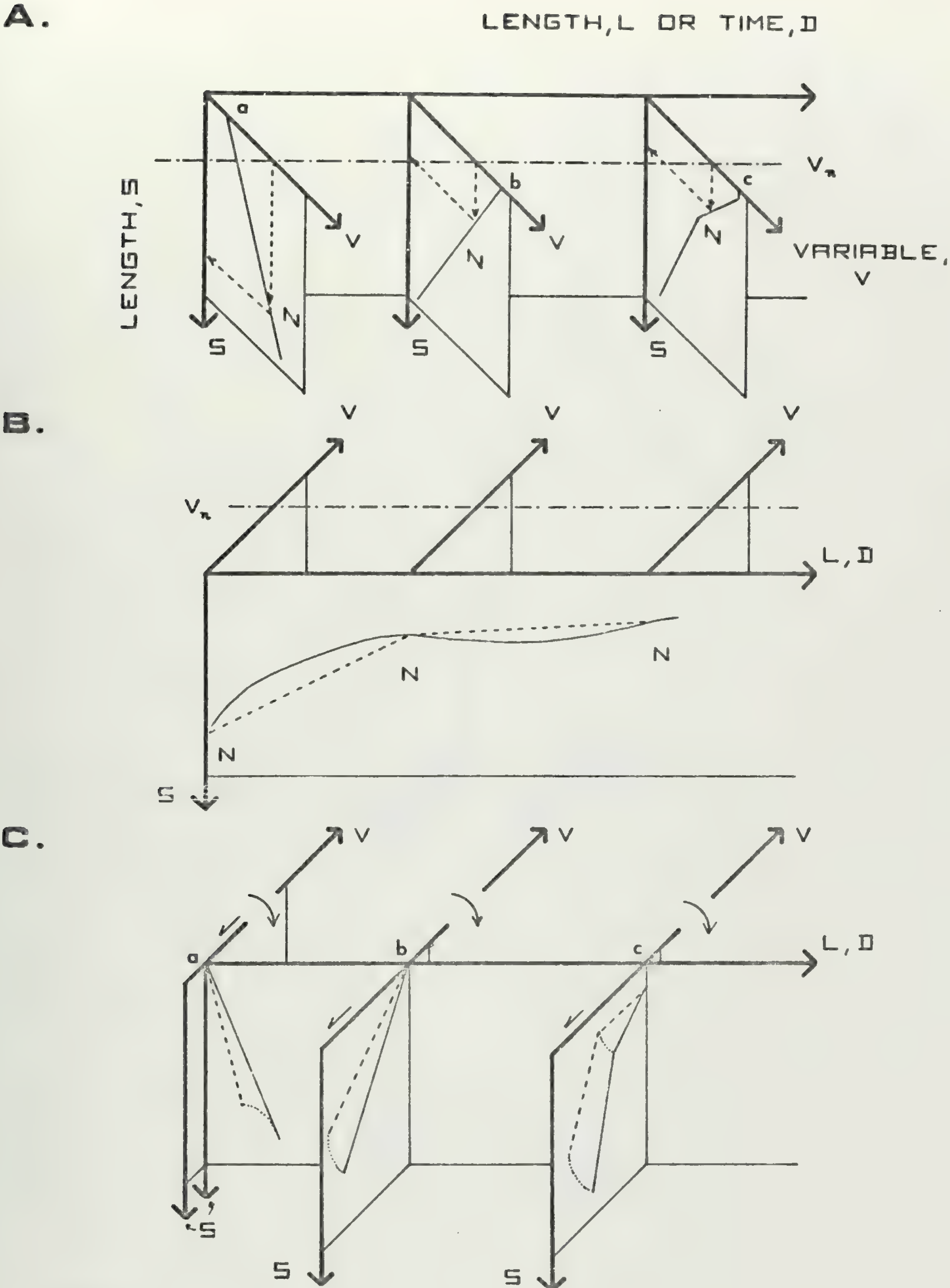
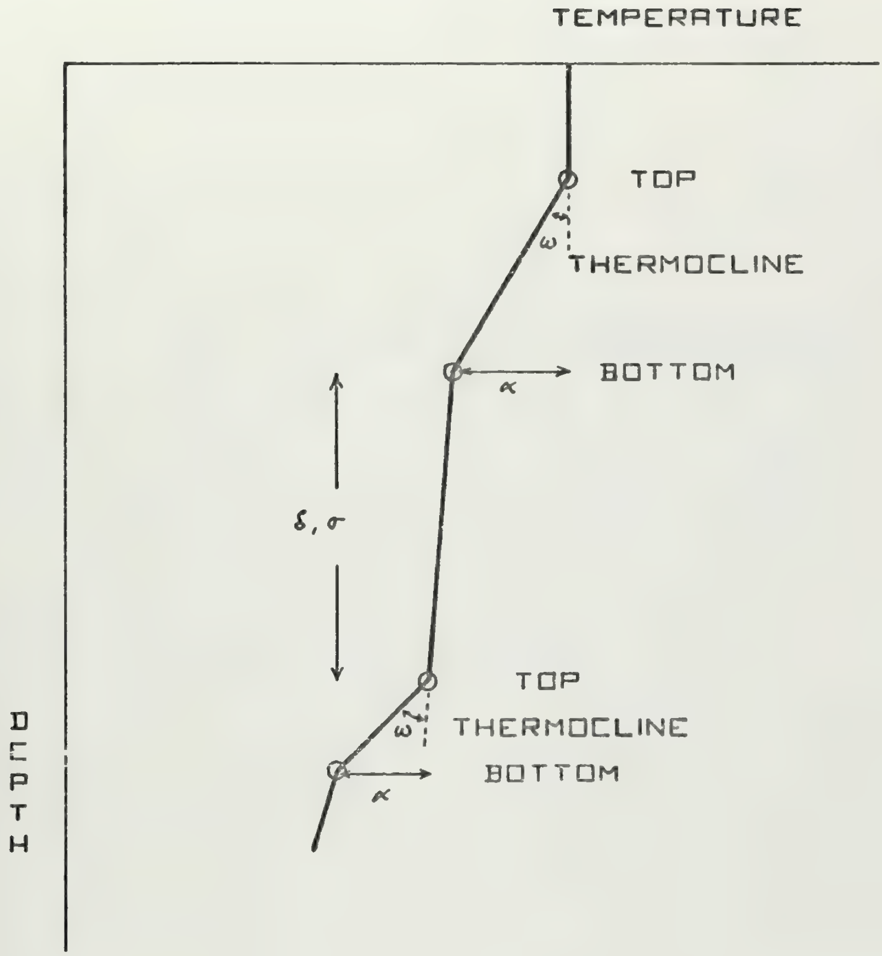


Figure 13. Tri-dimensional Analysis: A. The Tri-dimensional Distribution, B. A New Trace, C. The Projection of Two-dimensional Traces Along a Given Axis



FILTERING STEPS	
1	δ or σ, θ = tan ω
2	α or β = tan ω

Figure 14. The Filtering Parameters of Variclines

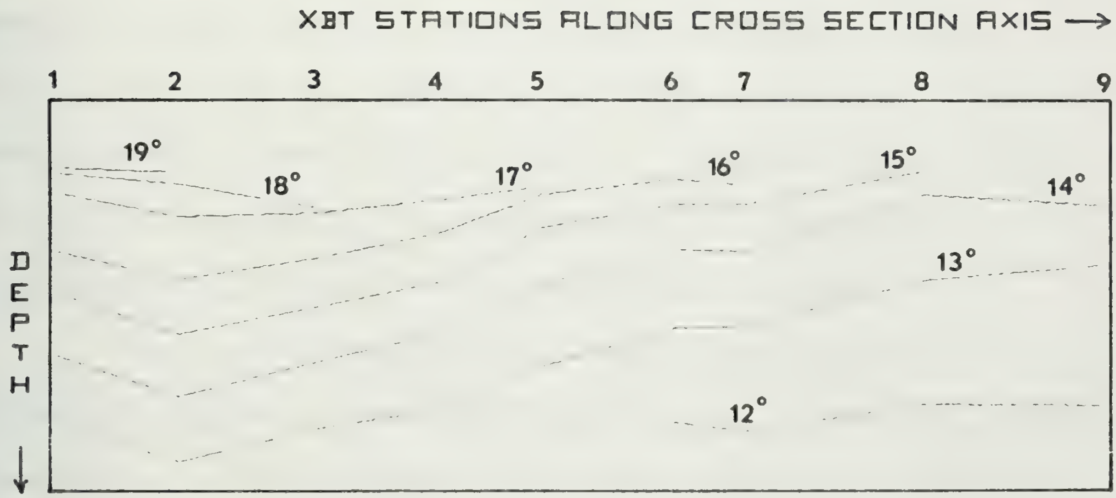


Figure 15. Computer Output Plot of the Linear Field of Isotherms for One Cruise or the Average Linear Field of Isotherms for a Given Number of Cruises

consists in the identification of the general pattern of the trace and this outlines a problem of scale: if the investigator is interested in microstructure, his main efforts are concentrated in the improvement of the accuracy of the sensor. The technique of suppressing the small features has not found definite and complete solution. It reveals a compromise established between the need to avoid small fluctuations and the unfortunate results when those variations were physically meaningful and their filtering has introduced artificial values. The best approach requires a good understanding of the geophysical phenomena and a suitable application of the filtering methods.

In this thesis study, analytical filters such as the binomial expansion and the least square polynomial approximation were used to smooth the generated mesh but they introduced severe modifications which (as presented in Figure 16B) completely deformed the original information, and their further application was avoided.

The discussion of the filtering capability of the finite difference technique constitutes the purpose of the present chapter.

a. The Mesh

The generated mesh depends upon the selection of the constant increment, ΔV: (a) a large value erases the small features but introduces distortion in the digital trace, and (b) a small value (compatible with the accuracy of the sensor) keeps the details and reproduces closely the digital trace, as shown in Figure 8.

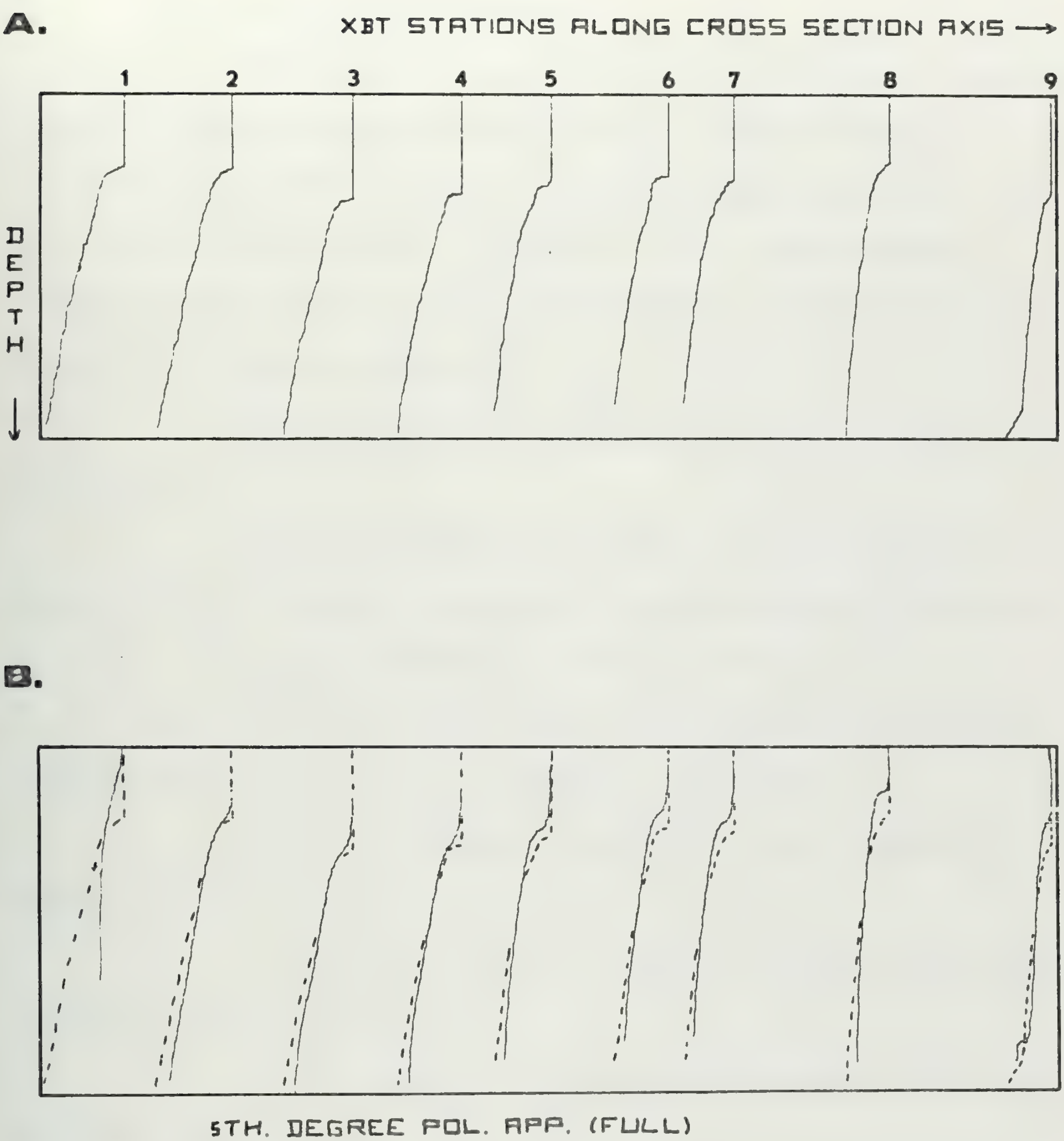


Figure 16. Computer Output Plots of: A. The Interpolated Digital Traces Along a Cross Section for a Given Cruise, and B. The Fifth Order Least Square Polynomial Approximation of the Same Digital Traces

b. Significant Features

Generically, significant features like inversions and variclines may be filtered out by specification of filtering parameters or the minimum values of (a) the defining parameters (8 - 11) and (b) the separation in length between consecutive and identical features (bottom of one and the top of the next).

Specifically, the filtering of variclines may be obtained by:

(1) The conditions limiting the top and the bottom of variclines (especially the bottom) and expressed in II.B.3.b., may be modified and extended to more points of the mesh above and below the feature resulting in its more clear definition, as evidenced in Figure 12.

(2) The filtering parameters illustrated in Figure 14:

(a) ALPHA or the minimum variable magnitude limiting the varicline.

(b) BETA or the minimum gradient defining the varicline.

(c) DELTA or the minimum separation in length between consecutive variclines.

(d) THETA or the minimum gradient selecting the variclines to be merged, using simultaneously the SIGMA parameter.

(e) SIGMA or the maximum separation in length between variclines to be merged and already defined by the THETA parameter.

C. A COMPUTER PROGRAM

The investigation of the vertical distribution of temperature in the upper oceanic layers can provide a digital model which applies the concepts of objective analysis described above.

Usually the data available is too non-uniformly distributed in time and length to make possible the use of a tri-dimensional objective analysis; however, a two-dimensional type may be utilized to examine each trace and to plot the result of the investigation (of all the traces obtained in a cruise or passage along a cross section) in a way which allows a subjective analysis.

An introduction of the model, its characteristics and operation is summarized in the next presentation.

1. Description

a. The Basic Computer Model

The computer model is organized in two parts: (a) the main program, and (b) the subroutines. The main program presents in a logical way a sequence of events concerning the execution of the job. This arrangement is a compromise between the space available in the computer memory and the least time of execution. The subroutines deal with specific stages of that execution.

The following description outlines the major operations of the model and their implementation must be monitored closely by the use of the flow diagram presented in Appendix B, with the symbols explained in Appendix A.

(1) "START" to "NORDER = 1"

The computer program receives the general information about the number identifying the cross section in study (ICSEC), the number of cruises per cross section (ICRUI), the number (ISO) and the values (TI) of isotherms, the mesh interval (DTW) and the number (NBT) and location (X) of stations for the cross section.

(2) "DO 460" Loop

The computer program processes the complete set of information about the thermal structure and wind field for a given cross section, cruise after cruise:

(a) "ITITLE'S" reads the titles of the plots drawn per cruise.

(b) "DO 250" loop reads the location (ISTA), the number (NOP) and pairs of points (ZO,TO) defining each digital trace in the cruise and processes the information about the depth of each isothermal value (GTVALU or ARTFT1). This scheme applies to all digital traces of a cruise.

(c) "CALL PLOTIT" receives the above generated data for the whole cruise and plots the field of isotherms (TI).

(d) "DO 350" loop reads the location (ISTA), the number (NOP) and pairs of points (ZO,TO) defining each digital trace in the cruise and prepares the mesh for that trace (LINTPS), plots the obtained mesh (PLOTBT) and analyzes it to identify thermoclines (THERMC).

(e) "CALL PLOTFT" receives the above generated data for the whole cruise and plots the field of thermocline isopleths.

(f) "DO 450" loop reads the location (ISTA) and the wind observations (WIND); determines the components of the wind along and perpendicular to the next coastline at the day of XBT/BT drop and one day before (WINDO1), two days before (WINDO2), three days before (WINDO3), five days before (WINDO5) and ten days before (WIND10); plots the wind and its components.

(g) "NPLOT = 3" to "460" completes the drawing of the wind information for a cruise and sets in the process for a new cruise.

(3) "IKONTR" to "IKONTR = 0"

This step controls the execution of the remaining part of the program. If the IKONTR value is zero the job ends.

(4) "IERIOD ..." to "CALL STATCS"

The computer program reads the required information to start the statistical study (STATCS) between any two parameters (KEK = 1, KEK = 2) and to plot the scatter diagram and the least square regression lines in the same graph (ITITL5). The continuation of the statistical study relies on the value of KRK read if different from zero.

(5) "ILONTR" to "ILONTR = 0"

This part controls the execution of the last part of the program. If the ILONTR value is zero, the job ends.

(6) "ITITL6" to "PLTMIT"

The program receives the generated data of the whole cross section and plots the average field of isotherms (PLTMIT) with a printed label (ITITL6).

(7) "ICSEC" to "STORAGE"

The program indicates that the study of the cross section (ICSEC) is completed and ends the job by the storage in a disk of the original and generated information (STORAGE).

b. Other Subroutines Available

Other subroutines are available to improve or to modify the operation of the basic model but keeping its main structure.

(1) Subroutine TREND

The subroutine TREND, Appendixes D, G and M, analyzes the digital trace using the finite difference technique to locate the maxima and the minima.

(2) Subroutine INVERS

The subroutine INVERS, Appendixes E, G and M, determines the inversion parameters for each minimum identified by subroutine TREND.

(3) Subroutine PROPAR

The subroutine PROPAR, Appendixes H and K, computes the friction velocity, roughness parameter and stability length from wind speed and air-sea temperature difference data.

(4) Subroutine ITRATE

The subroutine ITRATE, Appendixes I and L, calculates the friction velocity, inflow angle, roughness parameter and stability length from the large scale synoptic parameters.

(5) Subroutine DRAW

The subroutine DRAW, Appendix J, neutralizes the "IBM subroutine DRAW" avoiding output plots.

2. Input

The form of input to this program consists in data deck.

a. Main Program

The input to the main program is identified by the symbol READ shown in Appendix A. The flow diagram of the main program, Appendix B, gives the ordered sequence of the data deck. The format of individual input is specified in the computer program, Appendix F.

b. Subroutine THERMC

The subroutine THERMC requires the insertion of chosen values for the filtering parameters (ALPHA, BETA, DELTA, SIGMA, THETA) before running the program.

c. Subroutine PROPAR

The subroutine PROPAR requires special data described in Appendix H, which must be read together with the wind information as shown in Appendix K.

d. Subroutine ITRATE

The subroutine ITRATE needs large scale synoptic information pictured in Appendix I, which must be read together with the wind data as presented in Appendix L.

3. Output

The forms of putput from this program are (1) printouts, (2) plots, and (3) storage in disk.

a. Printouts

The printouts are identified by the symbol PRINTOUT, in Appendix A. The flow diagrams show the occurrence of printouts and their formats are specified in the computer programs.

The inclusion of the DUMMY card, Appendix P.a. neutralizes this form of output.

b. Plots

The basic program presents six types of plots which appear according to the execution of the following subroutines:

(1) PLOTIT

The subroutine PLOTIT plots the linear field of isotherms per cruise, as shown in Figure 15.

(2) PLOTBT

The subroutine PLOTBT plots the interpolated digital traces obtained in a given cruise, Figure 16.

(3) PLOTFT

The subroutine PLOTFT plots the linear field of isopleths defining the top, the mean depth and the bottom of the thermoclines analyzed from the interpolated digital traces of a given cruise, Figure 17.

(4) PLTMIT

The subroutine PLTMIT plots the average linear field of isotherms for a specified number of cruises, Figure 15.

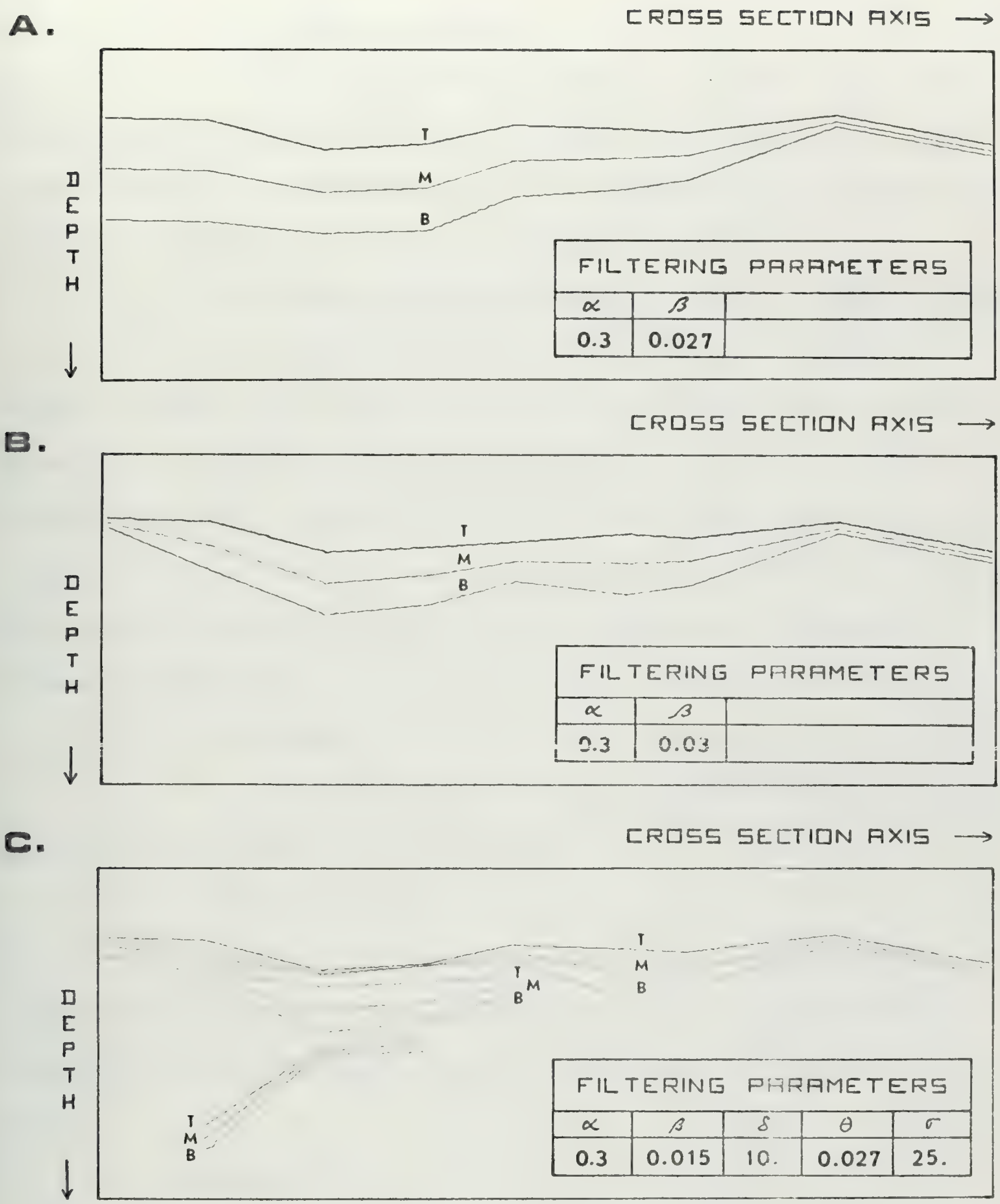


Figure 17. Computer Output Plots of the Linear Field of Isopleths Defining the Top (T), the Mean Depth (M), and the Bottom (B) of the Three Shallowest Thermoclines for Different Values of the Filtering Parameters

(5) RLINE and STATCS

The subroutines RLINE and STATCS plot the least square regression lines (RLINE) and the scatter diagram (STATCS) in the same graph to illustrate the statistical relation between two given variables, Figure 18.

(6) WINDOO, WINDO1, WINDO2, WINDO3, WINDO5, WIND10

The subroutines WINDOO, WINDO1, WINDO2, WINDO3, WINDO5 and WIND10 plot the wind vector and its components (along and perpendicular to the direction of the near coastline) measured at given station and day of XBT observation and at one, two, three, five and ten days before. The drawings are made in the same diagram for a specific cruise, Figure 19.

(7) DRAW

The insertion of the subroutine DRAW, Appendix J, will avoid this type of output.

c. Storage in Disk

The initial and processed data are stored in allocated space of disk depending upon the inclusion of the Job Control Language instructions shown in Appendix O.b, c,d.

4. Operation

a. Characteristics of the Basic Program

(1) Source Language

The computer program is written in FORTRAN IV (H) and the calculations are performed in real single precision arithmetic.

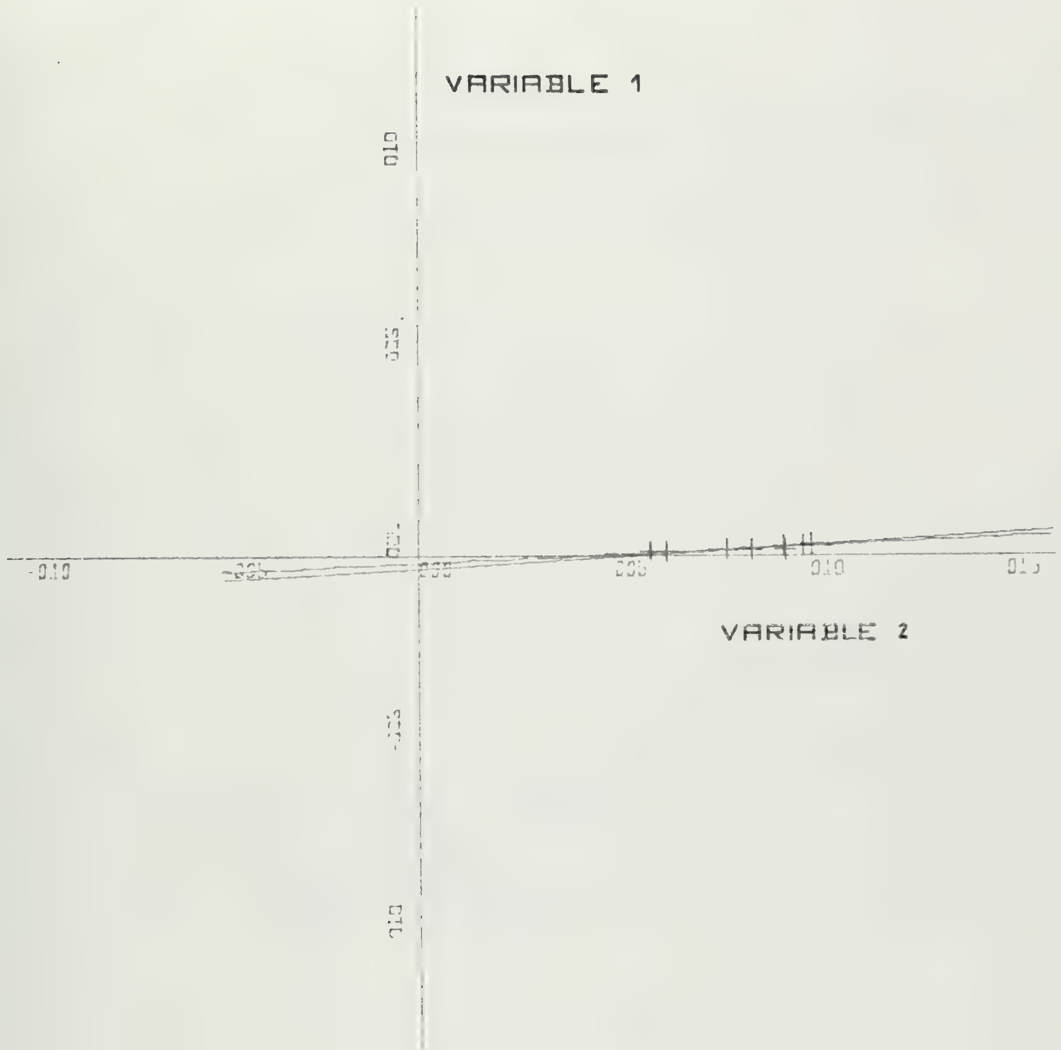


Figure 18. Computer Output Plot of the Least Square Regression Lines and the Scatter Diagram (+) for Two Given Parameters: Variable 1 and Variable 2

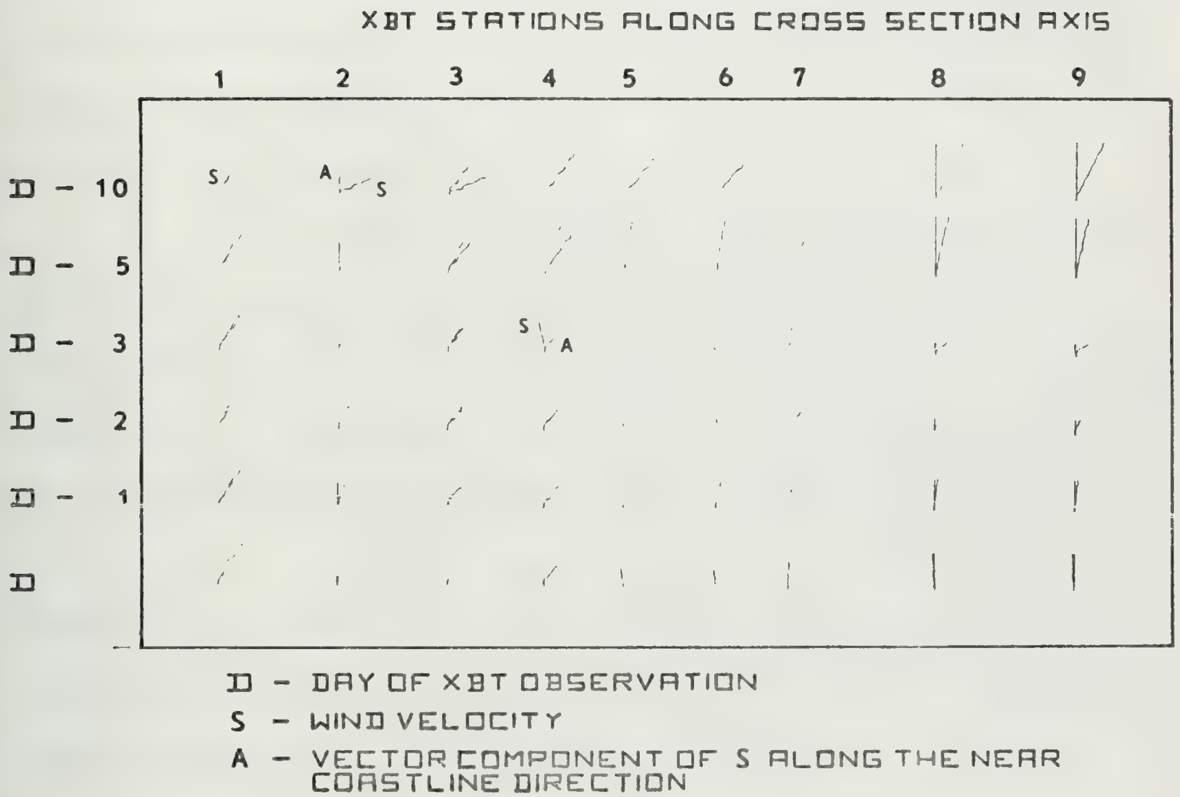


Figure 19. Computer Output Plot of the Wind Vector and its Components (along and perpendicular to the direction of the near coastline) Measured at Same Station and Day of XBT Observation and One, Two, Three, Five and Ten Days Before

(2) Subroutines and Functions Subprograms

The computer program uses subroutines and functions subprograms that had been compiled or been added to a library under OS/MVT. If the source deck of subroutine MISR had not been precompiled, it might be accessed (in Naval Postgraduate School Computer Facility) by concatenation in the FORTRAN compile step using the Job Control Language cards shown in Appendix N.d,e.

(3) Core

The basic computer program processes information for twenty cruises, each containing twelve stations and requires 288 K in core dimension, demanding the FORTRAN compilation step 150 K which must be requested by means of the Job Control Language card presented in Appendix N.b.

The space available is divided in five "COMMON" labelled areas which enables (a) the exchange of information between the main program, and (b) an economic and profitable use of the computer memory.

(4) Time

The time required to process (without statistical study) a set of seventeen cruises, each with nine stations, is about three minutes and thirty-four seconds. This time is distributed by (a) the compilation (one minute and forty-one seconds), (b) the linkage (three seconds), and (c) the execution (one minute and fifty seconds).

(5) Disk

The storage of the initial and processed information in a disk is obtained by the use of the Job Control Language cards shown in Appendix O.b,c,d. The requested space in disk corresponds to one cylinder or IBM 2314.

b. Equipment

The basic equipment used was:

(1) IBM 360/Model 67, OS - MVT with a

(2) CALCOMP Plotter, Model 765.

III. APPLICATION OF THE OBJECTIVE MODEL

The computer model was tested by the use of bathythermographic and wind information available from Fleet Numerical Weather Central for the geographical area shown in Figure 1.

The XBTs were taken during three years of passages between Canary Islands and Ferrol (Bay of Biscay) and Canary Islands and Barcelona (Mediterranean Sea) by Spanish ships. The XBT traces, in digital form and FORTRAN format compatible with this program, were organized in three cross sections, each one describing the normal routing of the survey ships and containing ten stations. In total, about 750 XBT traces were ready for further processing. Values for the filtering parameters of the thermocline analysis were previously chosen, namely

ALPHA = 0.3°C

BETA = 0.015°C/meter

DELTA - 10.0 meters

THETA = 0.027°C/meter

SIGMA = 25.0 meters

and applicable to all seasons and stations of the area. Finally, the objective analysis of the bathythermograph information was performed.

For each station of the cross sections, geostrophic wind data were computed from surface pressure analysis in FNWC. The wind obtained was not corrected for instability due to lack of information concerning the air-sea te perature

difference and according to Cardone [1969], such correction is fundamental to assure reliable wind data, specifically "friction velocity". Without this transformation, further handling of wind was decided to be inappropriate.

The final part of the application of the model, which is a statistical study between the wind and the thermal fields, as well as the computation of the first, second, third and fourth statistical moments (to infer known statistical distributions) of each field for a given station of the area could not be physically concluded due to the lack of proper wind data. Otherwise, the difficulties pointed out the importance of valid wind information and motivated the inclusion of subroutines, developed by Cardone [1969], to calculate "friction velocity" from large scale synoptic information (ITRATE) or ship reports (PROPAR).

IV. CONCLUSIONS

A FORTRAN IV program was developed and tested to handle large amounts of wind and bathythermographic information in an automatic way without distorting the initial data and restricting the human participation to a preparation phase dealing with organization of the information available and selection of values for the parameters of the model. The program performs objective analysis of XBT/BT digital traces, processes wind information, studies the statistical characteristics of the original and processed data and plots the generated fields.

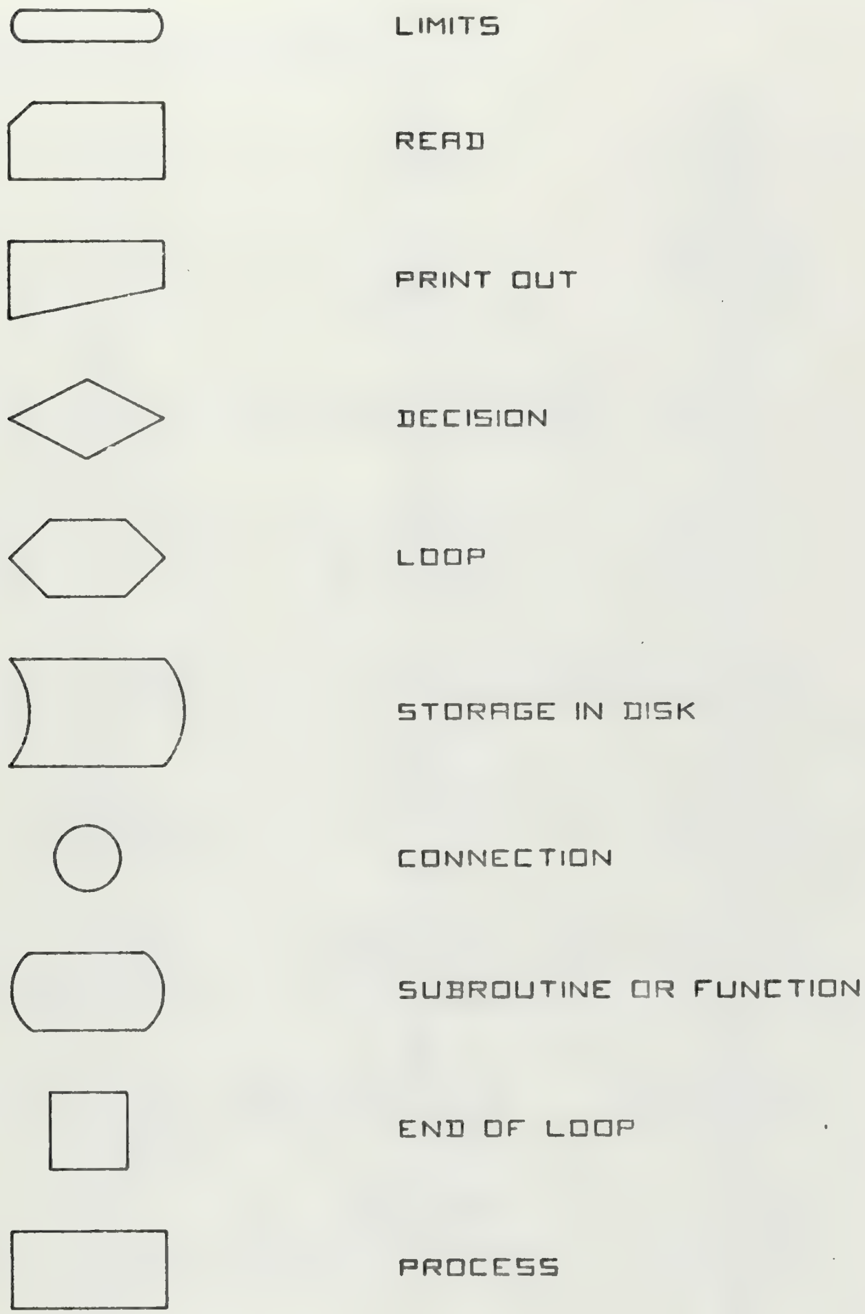
The objective analysis of XBT/BT traces makes use of finite difference schemes which requires preceeding transformation of the analog trace into digital form. The same concepts may be extended to analyze the distribution of any environmental variable (similar to the bathythermograph) along vertical or horizontal directions.

V. RECOMMENDATIONS

It is recommended that bathythermograph data be analyzed by an objective method before any subjective interpretations are made.

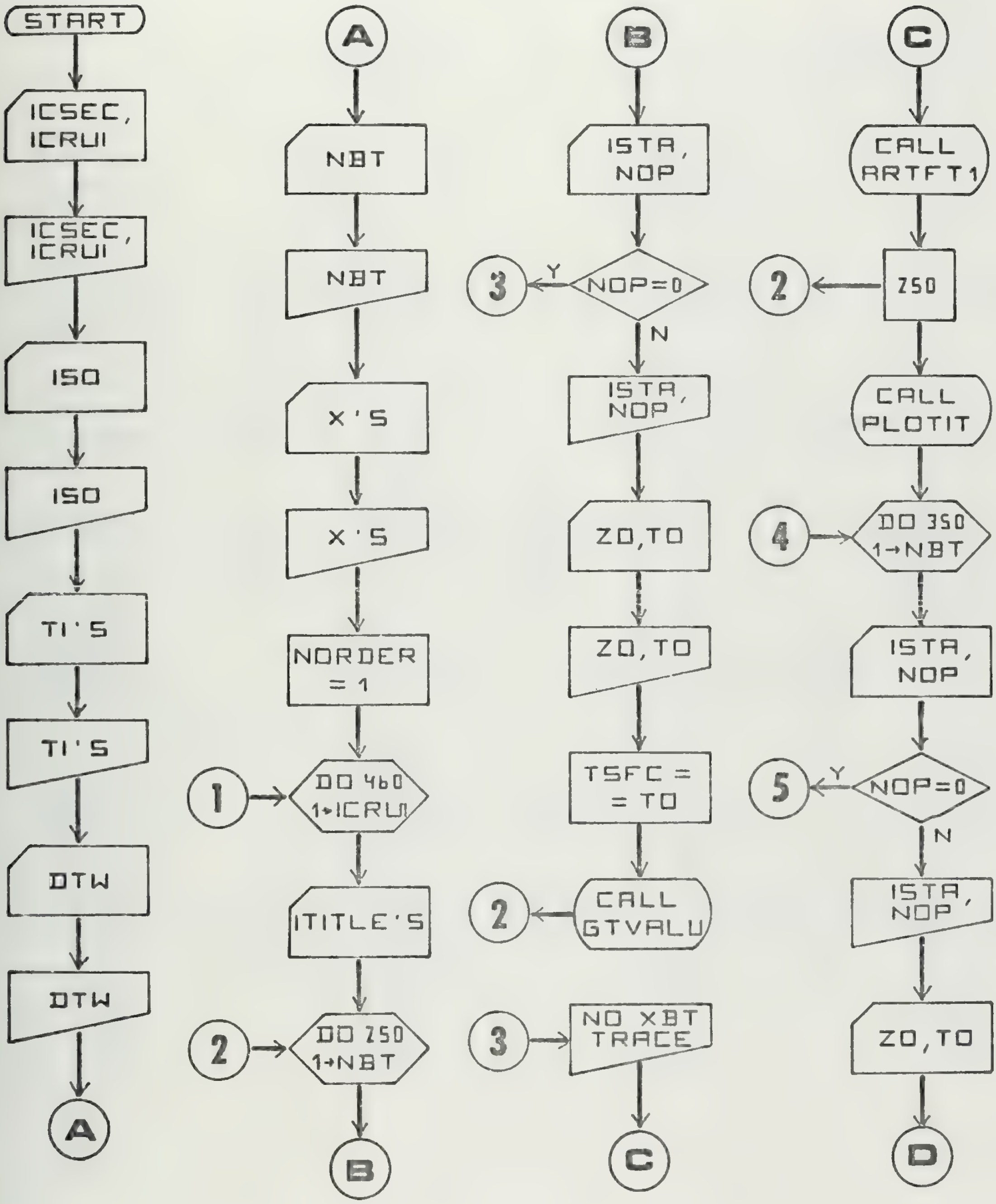
The objective analysis of XBT/BT traces may be performed aboard ships by the use of digital computers requiring a small memory but, the pre-digitization of the analog trace constitutes a fundamental prerequisite for processing.

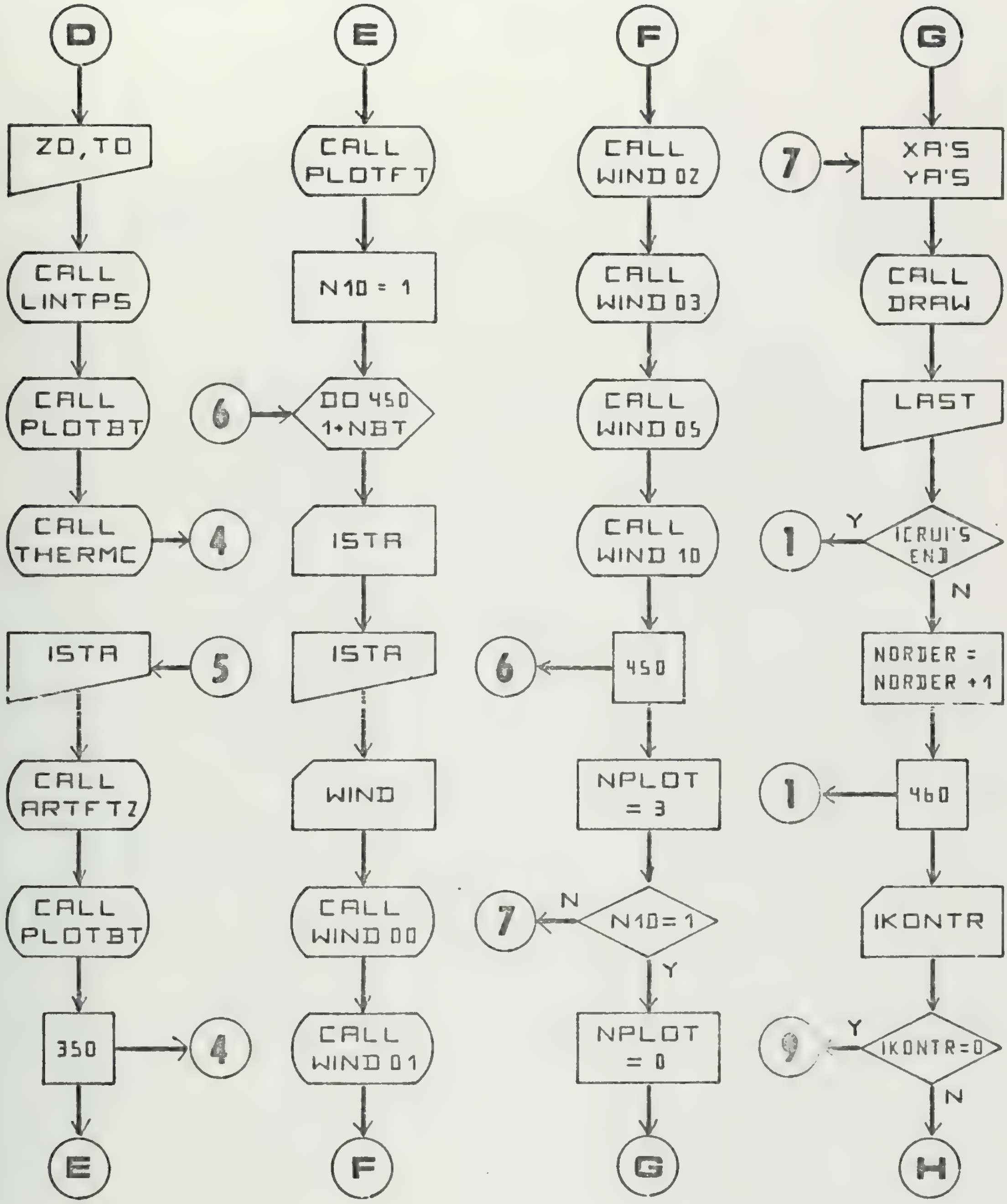
FLOW DIAGRAM SYMBOLS

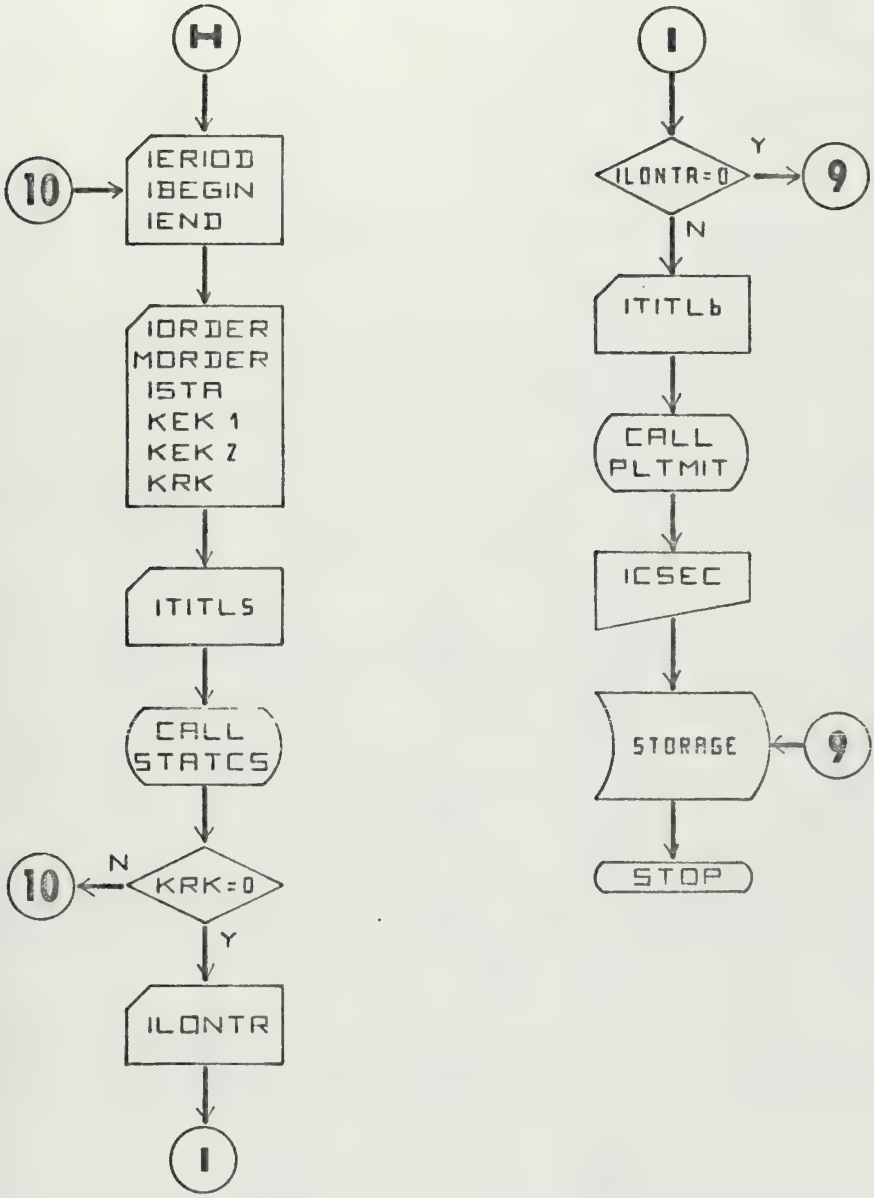


APPENDIX B

FLOW DIAGRAM OF THE FORTRAN IV BASIC COMPUTER PROGRAM

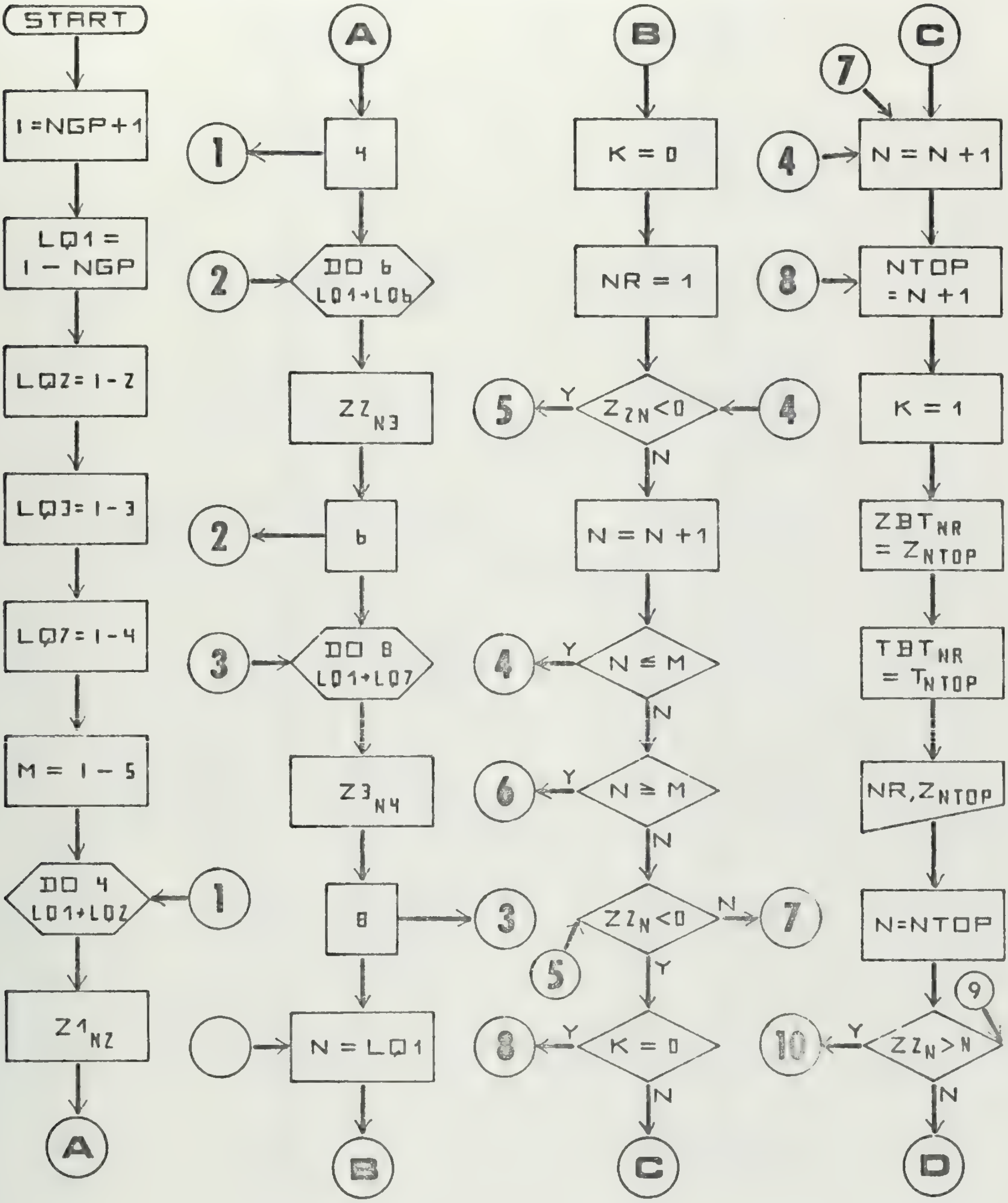


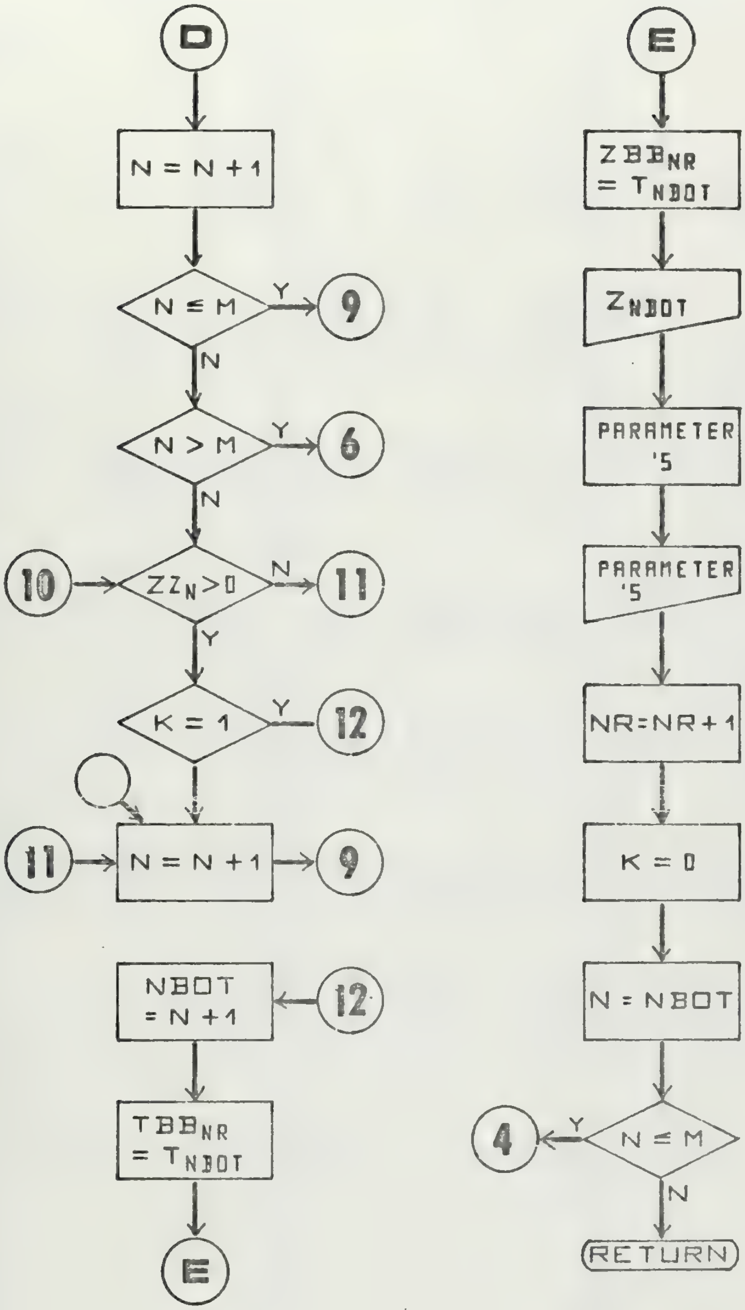

D
ZD,TD
CALL LINTPS
CALL PLOTBT
CALL THERMC
4
ISTA
5
CALL ARTFTZ
CALL PLOTBT
350
4
E
CALL PLOTFT
N10 = 1
6
DO 450 1→NBT
ISTA
ISTA
WIND
CALL WIND 00
CALL WIND 01
F
CALL WIND 02
CALL WIND 03
CALL WIND 05
CALL WIND 10
6
450
NPLOT = 3
7
N
N10=1
Y
NPLOT = 0
G
7
XA'S YA'S
CALL DRAW
LAST
1
Y
ICRUI'S END
N
NORDER = NORDER +1
1
460
IKONTR
9
Y
IKONTR=0
N
H


H
10
IERIOD
IBEGIN
IEND
IORDER
MORDER
ISTA
KEK 1
KEK 2
KRK
ITITLS
CALL
STATCS
KRK=0
N
Y
ILONTR
I
ILONTR=0
9
ITITL6
CALL
PLTMIT
ICSEC
STORAGE
STOP

APPENDIX C

FLOW DIAGRAM OF THE THERMOCLINE ANALYSIS

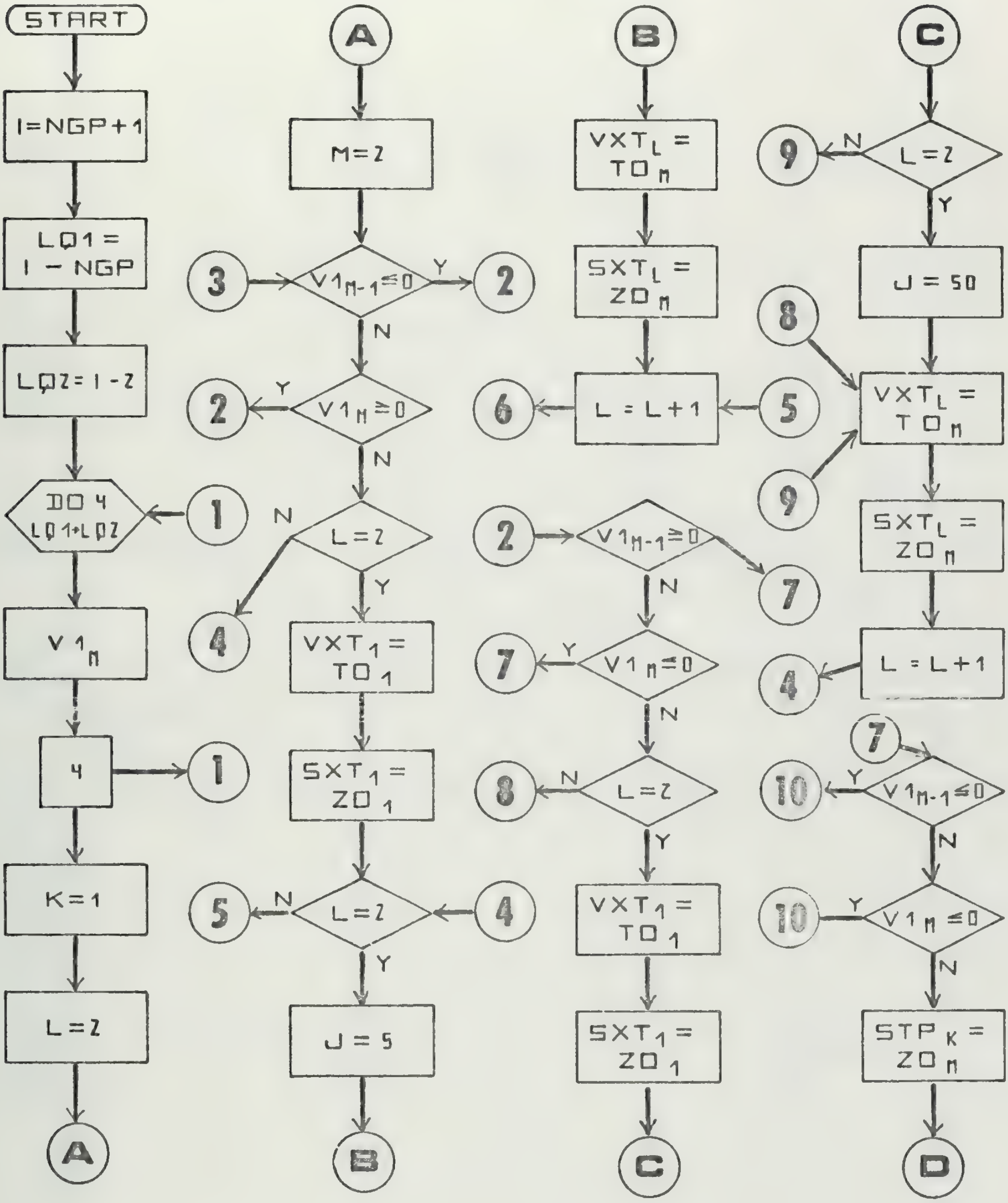


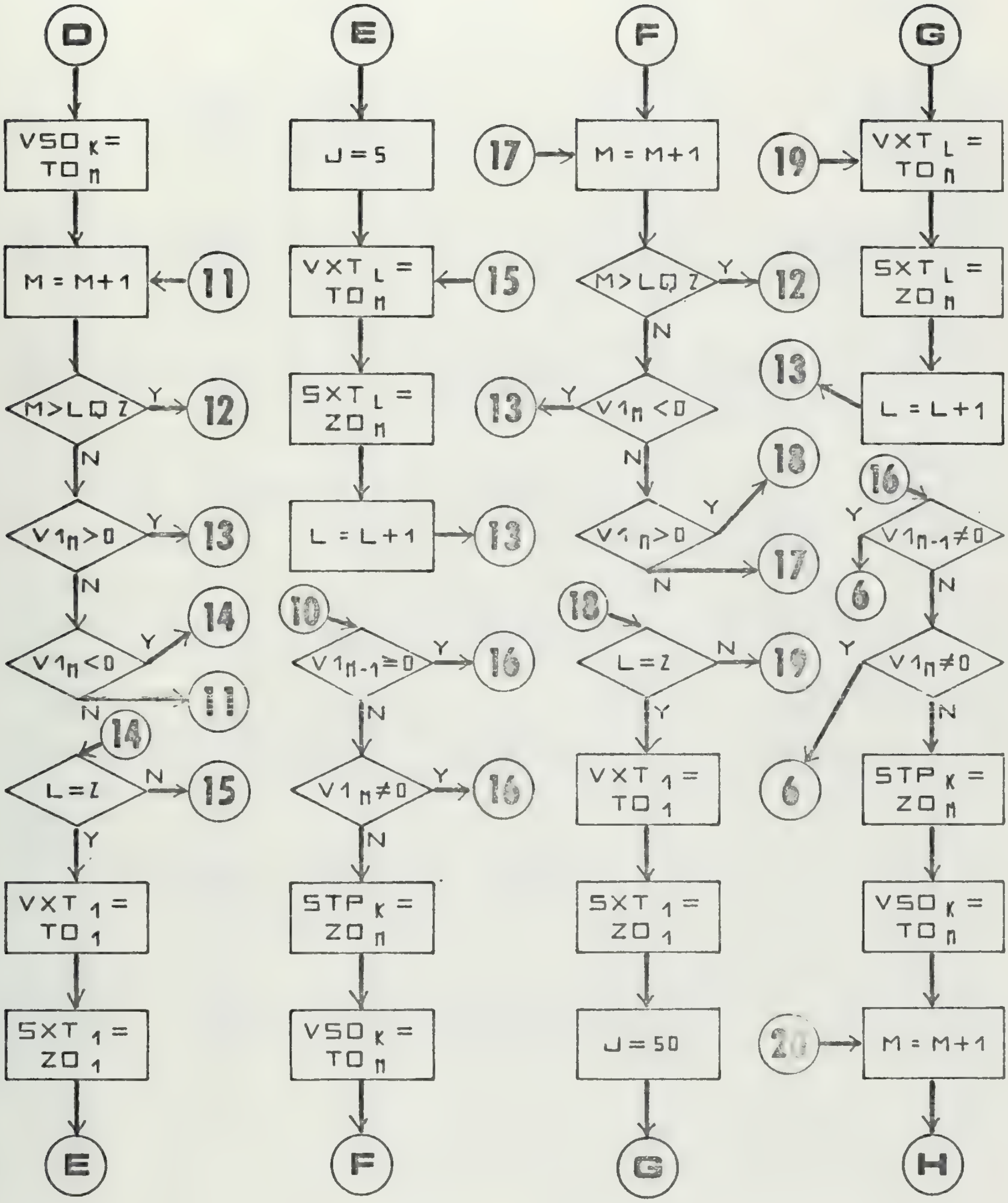


D
N = N + 1
N ≤ M
Y
9
N
N > M
Y
6
N
10
ZZN > 0
N
11
Y
K = 1
Y
12
11
N = N + 1
9
NBOT = N + 1
12
TBBNR = TNBOT
E
E
ZBBNR = TNBOT
ZNBOT
PARAMETER '5
PARAMETER '5
NR = NR + 1
K = 0
N = NBOT
4
Y
N ≤ M
N
RETURN

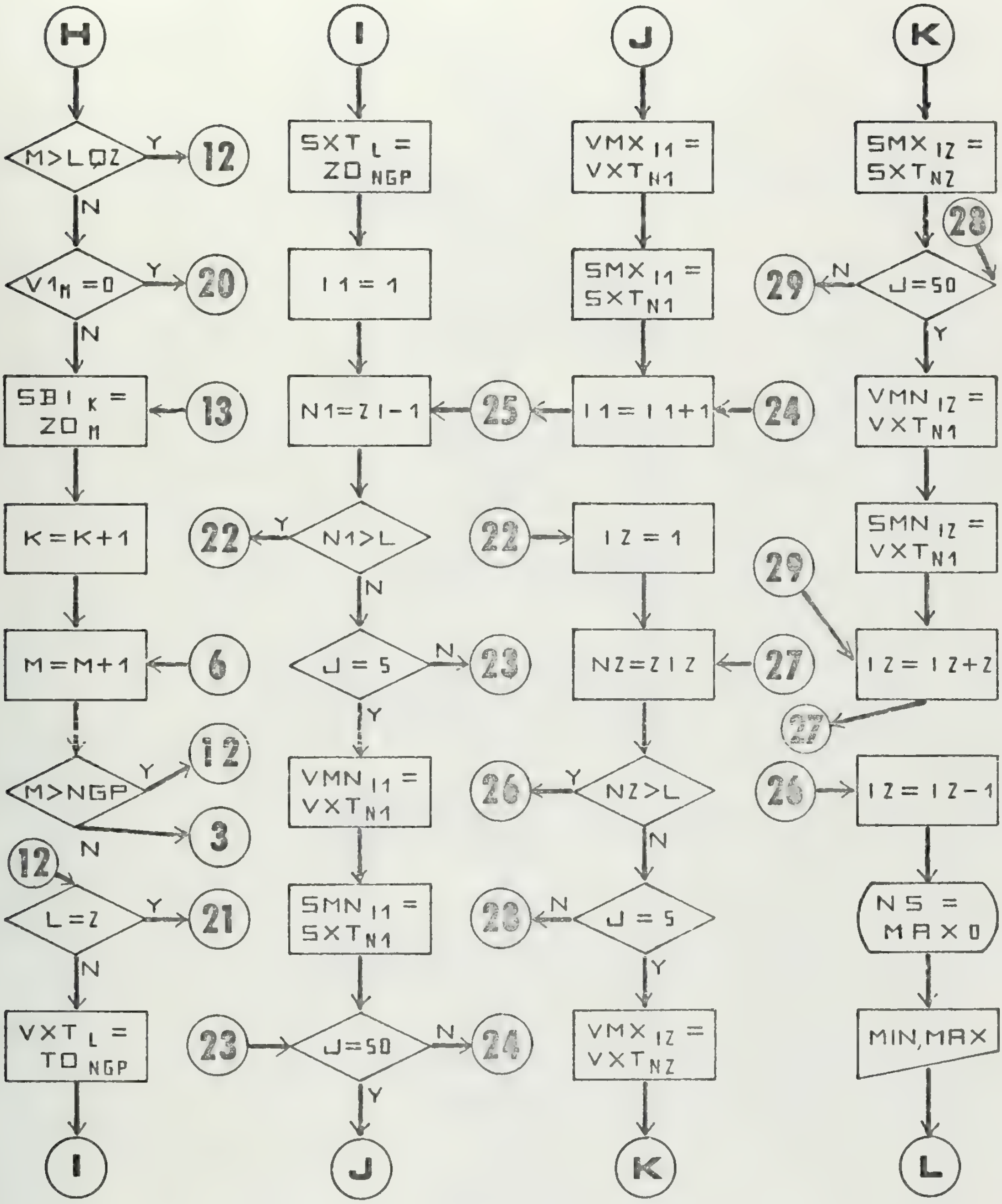
APPENDIX D

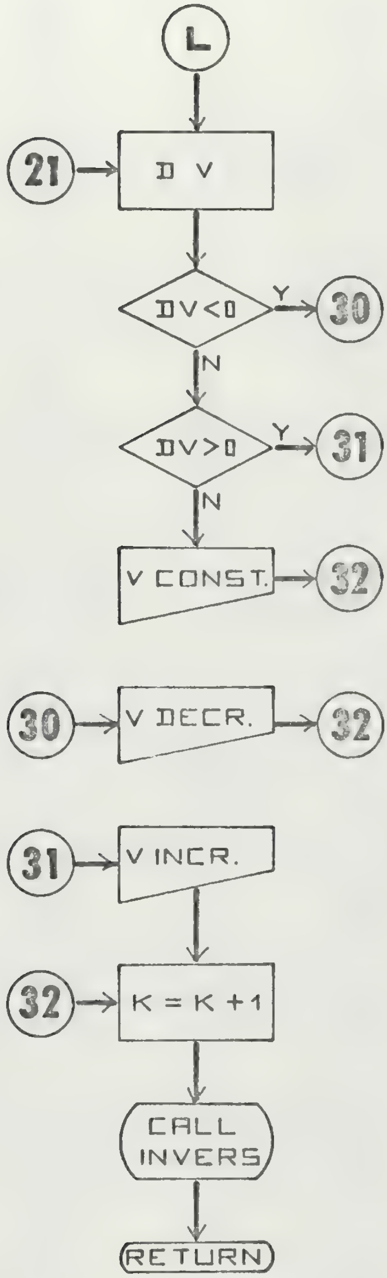
FLOW DIAGRAM OF THE TREND ANALYSIS





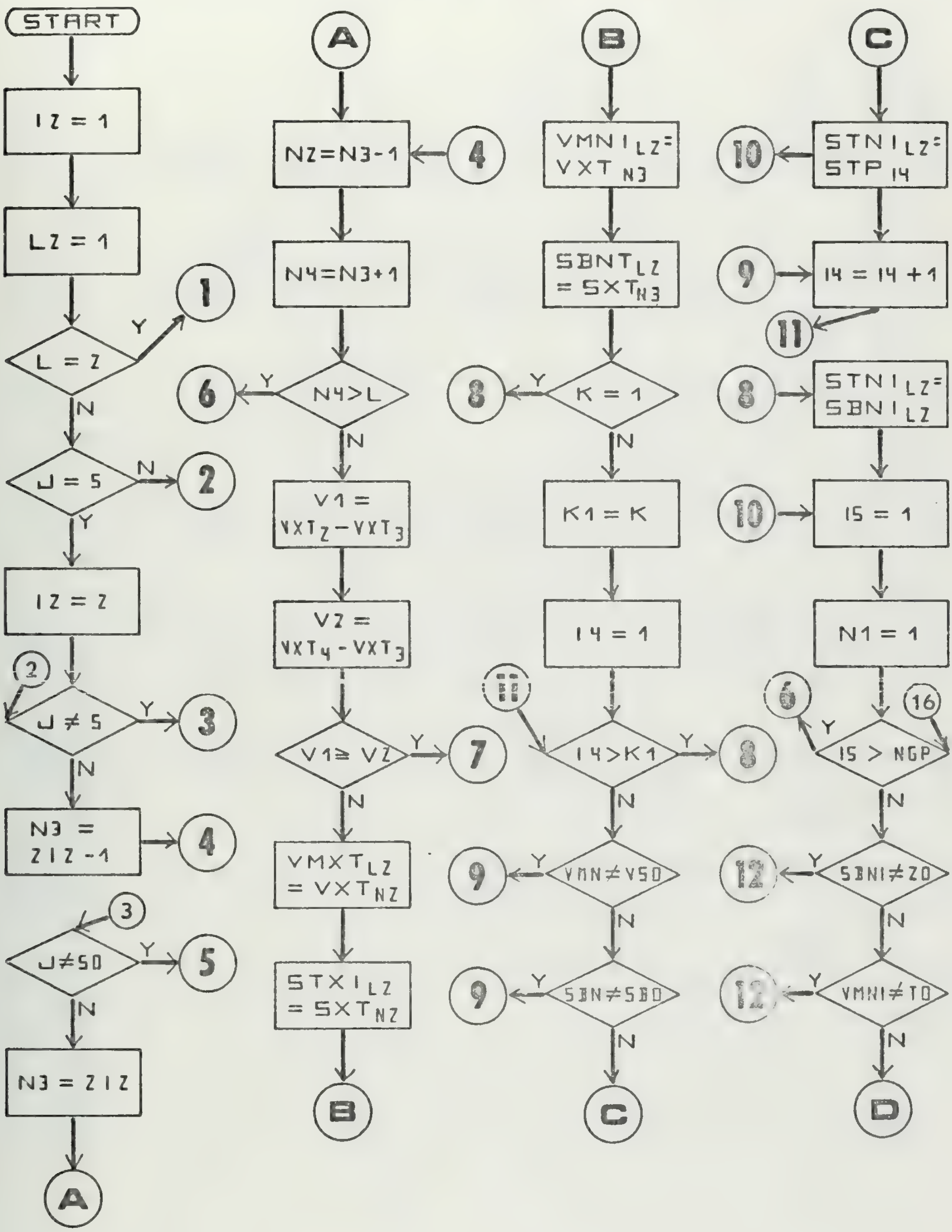
D
VSO K = TO M
M = M+1
11
M>LQZ
Y
12
N
V1M>0
Y
13
N
V1M<0
Y
14
N
11
14
L=Z
N
15
Y
VXT 1 = TO 1
SXT 1 = ZO 1
E
E
J=5
VXT L = TO M
15
SXT L = ZO M
L = L+1
13
10
V1M-1=0
Y
16
N
V1M≠0
Y
16
N
STP K = ZO M
VSO K = TO M
F
F
17
M = M+1
M>LQZ
Y
12
N
13
Y
V1M<0
N
V1M>0
Y
18
N
17
18
L=Z
N
19
Y
VXT 1 = TO 1
SXT 1 = ZO 1
J=50
G
G
19
VXT L = TO M
SXT L = ZO M
13
L = L+1
16
Y
V1M-1≠0
6
N
Y
V1M≠0
6
N
STP K = ZO M
VSO K = TO M
20
M = M+1
H

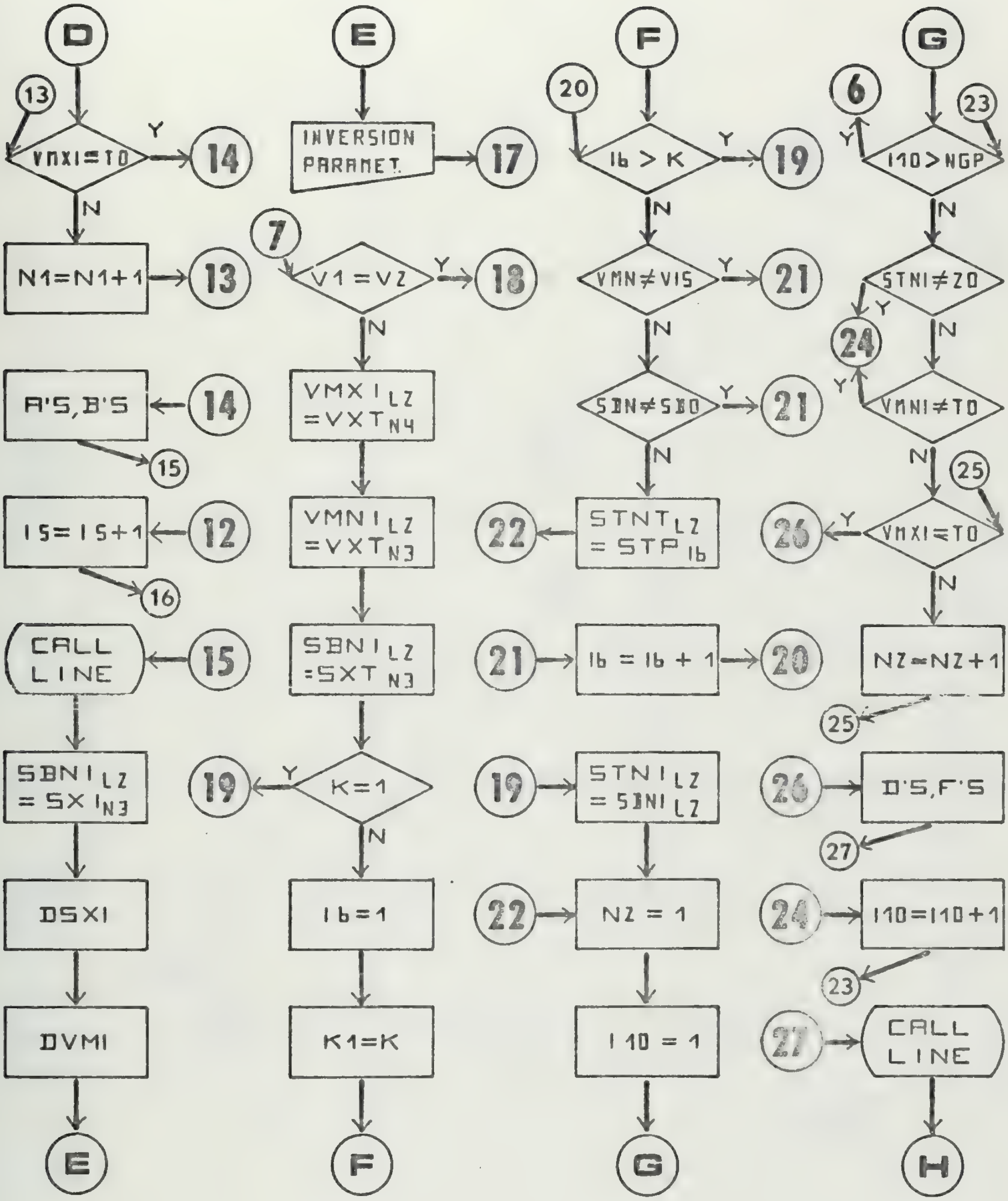

H
M>LQZ
12
V1M = 0
20
SBI K = ZO M
13
K = K+1
M = M+1
6
M>NGP
12
3
12
L = 2
21
VXT L = TO NGP
I
SXT L = ZO NGP
I1 = 1
N1 = 2I-1
25
N1>L
22
J = 5
23
VMN I1 = VXT N1
SMN I1 = SXT N1
23
J = 50
24
J
VMX I1 = VXT N1
SMX I1 = SXT N1
I1 = I1+1
24
22
I2 = 1
N2 = 2I2
27
N2>L
26
J = 5
28
VMX I2 = VXT N2
K
SMX I2 = SXT N2
28
J = 50
29
VMN I2 = VXT N1
SMN I2 = VXT N1
29
I2 = I2+2
27
26
I2 = I2-1
N5 = MAX0
MIN, MAX
L
Y
N

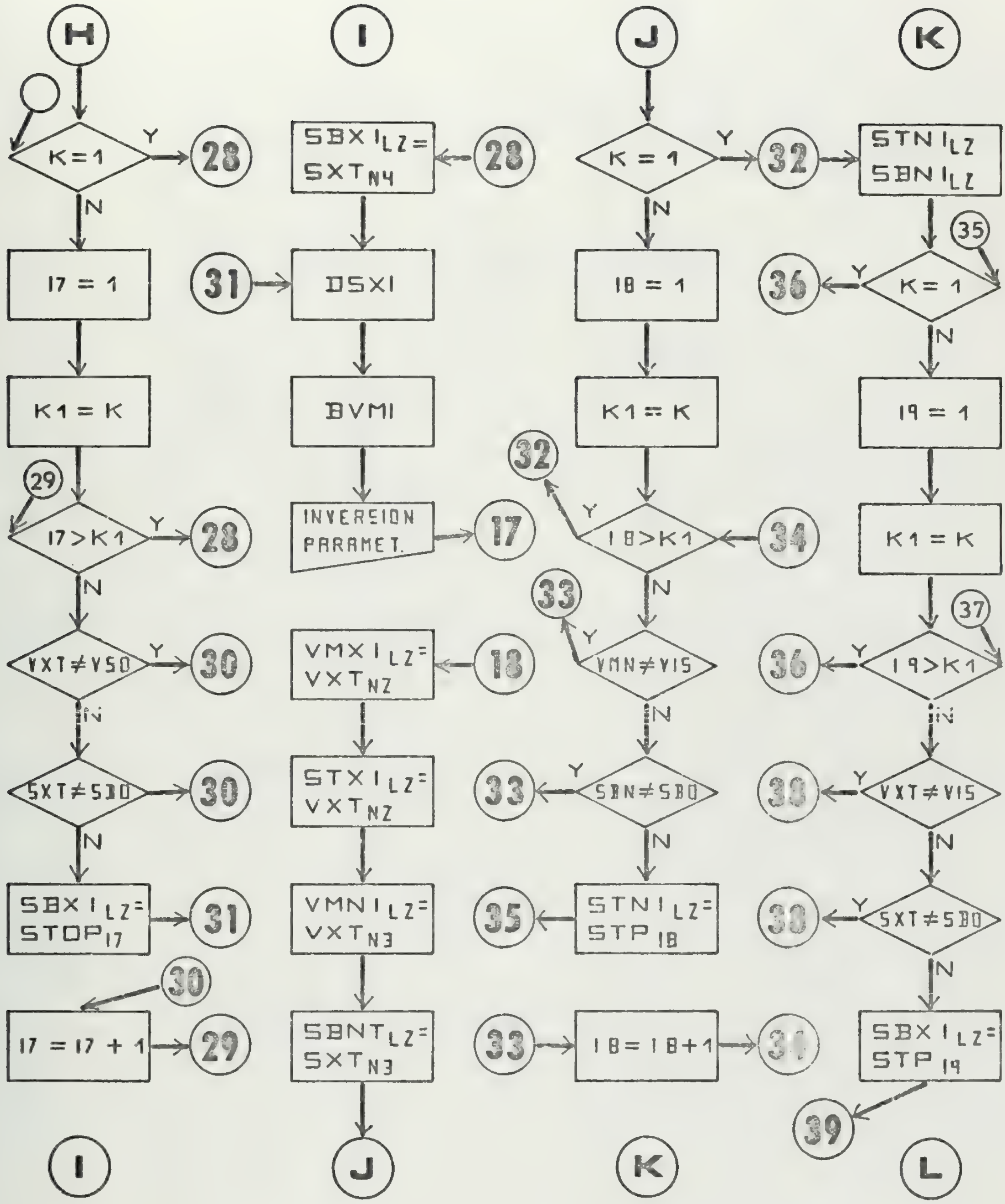

L
21
D V
DV<0
Y
30
N
DV>0
Y
31
N
V CONST.
32
30
V DECR.
32
31
V INCR.
32
K = K + 1
CALL INVERS
RETURN

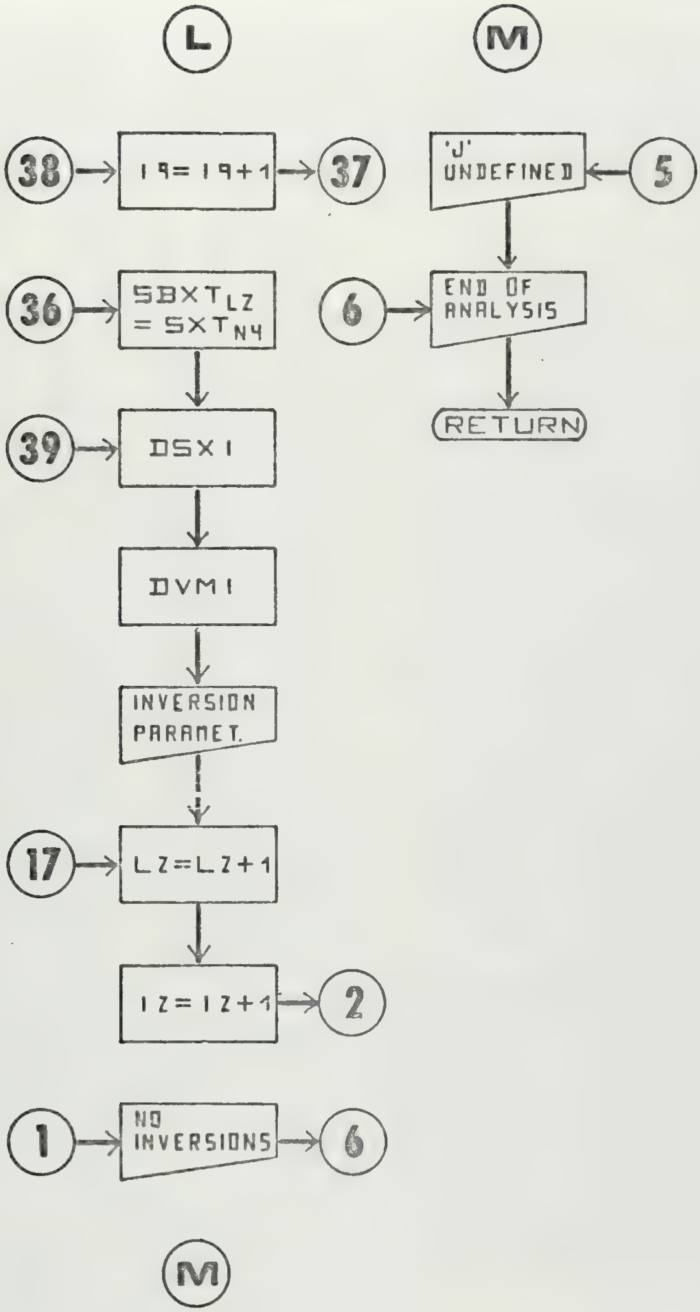
APPENDIX E

FLOW DIAGRAM OF THE INVERSION ANALYSIS




D
E
F
G
VMXI=TO
N1=N1+1
A'S,B'S
IS=IS+1
CALL LINE
SBNI LZ = SX I N3
DSXI
DVMI
INVERSION PARAMET.
V1=V2
VMXI LZ =VXT N4
VMNI LZ =VXT N3
SBNI LZ =SXT N3
K=1
Ib=1
K1=K
Ib > K
VMN≠VIS
SBN≠SBO
STNT LZ = STP Ib
Ib = Ib + 1
STNI LZ = SBNI LZ
NZ = 1
I10 = 1
I10>NGP
STNI≠ZO
VMNI≠TO
VMXI≠TO
NZ=NZ+1
D'S,F'S
I10=I10+1
CALL LINE
H


H
I
J
K
K=1
Y
N
28
17 = 1
K1 = K
29
17>K1
VXT≠VSO
30
SXT≠SBO
SBXI LZ= STOP 17
31
17 = 17 + 1
SBXI LZ= SXT N4
DSXI
BVMI
INVERSION PARAMET.
17
VMXI LZ= VXT NZ
18
STXI LZ= VXT NZ
VMNI LZ= VXT N3
SBNT LZ= SXT N3
32
18 = 1
18>K1
34
33
VMN≠VIS
SBN≠SBO
STNI LZ= STP 18
35
18=18+1
STNI LZ
SBNI LZ
36
19 = 1
37
19>K1
VXT≠VIS
SXT≠SBO
SBXI LZ= STP 19
39
L


L
M
38
I9= I9+1
37
'J'
UNDEFINED
5
36
SBXT LZ
= SXT N4
6
END OF
ANALYSIS
RETURN
39
DSXI
DVMI
INVERSION
PARAMET.
17
LZ=LZ+1
IZ=IZ+1
2
1
NO
INVERSIONS
6
M

APPENDIX F

FORTRAN IV BASIC PROGRAM FOR IBM 360/OS

```
C                                                                       MAIN0010
C     ..................................................................MAIN0020
C                                                                       MAIN0030
C                                                                       MAIN0040
C     MAIN PROGRAM                                                      MAIN0050
C                                                                       MAIN0060
C                                                                       MAIN0070
C     GENERAL PURPOSE                                                   MAIN0080
C        (1) TWO-DIMENSIONAL OBJECTIVE ANALYSIS OF BT/XBT DIGITAL TRACE MAIN0090
C        TO DETERMINE THE THERMOCLINES AND THE DEPTH OF THE MIXED LAYER MAIN0100
C        USING THE FINITE DIFFERENCE TECHNIQUE. (THE ANALYSIS MAY BE    MAIN0110
C        EXTENDED TO INVESTIGATE THE TREND AND INVERSIONS)              MAIN0120
C        (2) REPRESENTATION OF THE DIGITAL TRACES, THE ISOTHERMAL FIELD MAIN0130
C        AND THE FIELD OF ISOPLETHS OF THE THERMOCLINE ANALYSIS AT GIVENMAIN0140
C        POSITIONS ALONG A LENGTH (CROSS SECTION OR TIME AXIS) ALLOWING MAIN0150
C        A SUBJECTIVE ANALYSIS.                                         MAIN0160
C        (3) WIND DATA (INTENSITY AND DIRECTION) RELATIVE TO THE BT/XBT MAIN0170
C        GEOGRAPHICAL POSITION OF OBSERVATION IS SEPARATED IN TWO       MAIN0180
C        COMPONENTS: (A) ALONG AND (B) PERPENDICULAR TO THE DIRECTION   MAIN0190
C        OF THE NEAR COASTLINE. THE WIND AND ITS COMPONENTS ARE         MAIN0200
C        PRESENTED AT THE GIVEN POSITION ALONG A LENGTH OR TIME AXIS    MAIN0210
C        FOR A SUBJECTIVE INVESTIGATION.                                MAIN0220
C        (4) THE ORIGINAL DATA WITH THE INFORMATION RESULTANT FROM THE  MAIN0230
C        ABOVE PROCESS ARE STATISTICALLY INTERRELATED TO EVIDENCE       MAIN0240
C        ANY CONNECTION.                                                MAIN0250
C                                                                       MAIN0260
C     DESCRIPTION OF PARAMETERS                                         MAIN0270
C        DTW   -INPUT VALUE OF THE THERMAL INTERVAL OF THE MESH         MAIN0280
C        IBEGIN-INPUT VALUE CONTAINING THE MONTH,DAY AND YEAR OF THE    MAIN0290
C               BEGINNING OF THE STUDY                                  MAIN0300
C        ICRUI -INPUT NUMBER OF CRUISES PER CROSS SECTION               MAIN0310
C        ICSEC -INPUT NUMBER OF IDENTIFICATION OF THE CROSS SECTION     MAIN0320
C        IEND  -INPUT VALUE CONTAINING THE MONTH,DAY AND YEAR OF THE    MAIN0330
C               END OF THE STUDY                                        MAIN0340
C        IERIOD-INPUT VALUE CONTAINING THE NUMBER OF DAYS CONCERNING    MAIN0350
C               THE DURATION OF THE STUDY                               MAIN0360
C        IKONTR-INPUT VALUE CONTROLLING THE EXECUTION OF THE            MAIN0370
C               STATISTICAL STUDY (IF IKONTR.EQ.0 : 'NO STUDY')         MAIN0380
C        ILONTR-INPUT VALUE CONTROLLING THE PLOT OF THE MEAN LOCATIONS  MAIN0390
C               OF ISOTHERMS (FOR A GIVEN PERIOD) ALONG A CROSS         MAIN0400
C               SECTION OR TIME AXIS                                    MAIN0410
C        IORDER-INPUT VALUE CONTAINING THE NUMBER OF THE CRUISE TO      MAIN0420
C               INITIATE THE STATISTICAL STUDY                          MAIN0430
C        ISO   -INPUT NUMBER OF ISOTHERMS                               MAIN0440
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS        MAIN0450
C               SECTION                                                 MAIN0460
C        ITITL1-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR    MAIN0470
C               PLOT OF DIGITAL TRACES ALONG A CROSS SECTION OR TIME    MAIN0480
```

```
C           AXIS                                                          MAIN0490
C     ITITL2-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR         MAIN0500
C           PLOT OF ISOTHERMS ALONG A CROSS SECTION OR TIME AXIS          MAIN0510
C     ITITL3-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR         MAIN0520
C           PLOT OF ISOPLETHS OF THE THERMOCLINE ANALYSIS ALONG A         MAIN0530
C           CROSS SECTION OR TIME AXIS                                    MAIN0540
C     ITITL4-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR         MAIN0550
C           PLOT OF WIND DATA ALONG A CROSS SECTION OR TIME AXIS          MAIN0560
C     ITITL5-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR         MAIN0570
C           PLOT OF SCATTER DIAGRAM AND REGRESSION LINES AS THE           MAIN0580
C           RESULT OF STATISTICAL STUDY BETWEEN TWO VARIABLES             MAIN0590
C     ITITL6-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR         MAIN0600
C           PLOT OF THE AVERAGE DEPTHS OF ISOTHERMS (FOR A GIVEN          MAIN0610
C           PERIOD) ALONG A CROSS SECTION OR TIME AXIS                    MAIN0620
C     J3    -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED AT A GIVEN         MAIN0630
C           CRUISE                                                        MAIN0640
C     J7    -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED AT A GIVEN         MAIN0650
C           CRUISE                                                        MAIN0660
C     J8    -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED AT A GIVEN         MAIN0670
C           CRUISE                                                        MAIN0680
C     KAMMA1-INPUT VALUE CONTAINING THE DIRECTION OF THE NEAR             MAIN0690
C           COASTLINE                                                     MAIN0700
C     KANG00-INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND         MAIN0710
C           DIRECTION AT DAY OF BT/XBT OBSERVATION, FOR A GIVEN           MAIN0720
C           STATION AND CRUISE NUMBER                                     MAIN0730
C           KEK2=10                                                       MAIN0740
C     KANG01-INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND         MAIN0750
C           DIRECTION AT ONE DAY BEFORE THE DAY OF BT/XBT                 MAIN0760
C           OBSERVATION, FOR A GIVEN STATION AND CRUISE NUMBER            MAIN0770
C           KEK2=11                                                       MAIN0780
C     KANG02-INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND         MAIN0790
C           DIRECTION AT TWO DAYS BEFORE THE DAY OF BT/XBT                MAIN0800
C           OBSERVATION, FOR A GIVEN STATION AND CRUISE NUMBER            MAIN0810
C           KEK2=12                                                       MAIN0820
C     KANG03-INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND         MAIN0830
C           DIRECTION AT THREE DAYS BEFORE THE DAY OF BT/XBT              MAIN0840
C           OBSERVATION, FOR A GIVEN STATION AND CRUISE NUMBER            MAIN0850
C           KEK2=13                                                       MAIN0860
C     KANG05-INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND         MAIN0870
C           DIRECTION AT FIVE DAYS BEFORE THE DAY OF BT/XBT               MAIN0880
C           OBSERVATION, FOR A GIVEN STATION AND CRUISE NUMBER            MAIN0890
C           KEK2=14                                                       MAIN0900
C     KANG10-INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND         MAIN0910
C           DIRECTION AT TEN DAYS BEFORE THE DAY OF BT/XBT                MAIN0920
C           OBSERVATION, FOR A GIVEN STATION AND CRUISE NUMBER            MAIN0930
C           KEK2=15                                                       MAIN0940
C     KEK1  -INPUT VALUE INDICATING THE FIRST OF TWO VARIABLES TO         MAIN0950
C           BE INTERRELATED                                               MAIN0960
```

```
C     KEK2   -INPUT VALUE INDICATING THE SECOND OF TWO VARIABLES TO   MAIN0970
C             BE INTERRELATED                                          MAIN0980
C     KKK    -INPUT VALUE CONTROLLING THE PLOT OF THE WIND VECTOR      MAIN0990
C             (IF K.NE.0 : 'NO PLOT')                                  MAIN1000
C     KRK    -INPUT VALUE CONTROLLING THE CONTINUATION OF THE          MAIN1010
C             STATISTICAL STUDY. (IF KRK.EQ.0 : 'NO MORE STUDY')       MAIN1020
C     K98    -ACTUAL NUMBER OF PROCESSED CRUISES                       MAIN1030
C     LABEL  -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE            MAIN1040
C     LAST   -OUTPUT VALUE CONTAINING AN INFORMATION ABOUT THE         MAIN1050
C             SUCCESS OF A PLOT (IF K.EQ.0 : 'COMPLETED')              MAIN1060
C     MORDER-INPUT VALUE CONTAINING THE NUMBER OF THE CRUISE TO        MAIN1070
C             END THE STATISTICAL STUDY                                MAIN1080
C     M60    -OUTPUT NUMBER OF PAIRS (T,Z) COMPUTED FROM THE DIGITAL   MAIN1090
C             TRACE BY (A) A CONTINUOUS INCREMENT 'DTW' OF             MAIN1100
C             TEMPERATURE BEGINNING AT SEA SURFACE AND (B) THE         MAIN1110
C             DEPTHS OF THE ABOVE PAIRS ARE OBTAINED BY LINEAR         MAIN1120
C             INTERPOLATION BETWEEN NEAR POINTS OF THE ORIGINAL        MAIN1130
C             DIGITAL TRACE                                            MAIN1140
C     NBT    -INPUT NUMBER OF STATIONS ALONG THE CROSS SECTION OR      MAIN1150
C             THE TIME AXIS                                            MAIN1160
C     NOP    -INPUT NUMBER OF PAIRS (T0,Z0) DEFINING A DIGITAL TRACE   MAIN1170
C     NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                        MAIN1180
C     NPLOT  -INPUT VALUE INDICATING THE NUMBER OF CURVES PER PLOT     MAIN1190
C             (INFORMATION REQUIRED BY SUBROUTINE 'DRAW')              MAIN1200
C     N10    -ACTUAL NUMBER OF WIND PLOTS                              MAIN1210
C     R00    -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1220
C             SPEED AT DAY OF BT/XBT OBSERVATION, FOR A GIVEN          MAIN1230
C             STATION AND CRUISE NUMBER                                MAIN1240
C             KEK2=16                                                  MAIN1250
C     R001   -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1260
C             SPEED (ALONG THE COAST) AT DAY OF BT/XBT OBSERVATION,    MAIN1270
C             FOR A GIVEN STATION AND CRUISE NUMBER                    MAIN1280
C             KEK2=25                                                  MAIN1290
C     R002   -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1300
C             SPEED (PERP. TO THE COAST) AT DAY OF BT/XBT OBSERVATION MAIN1310
C             ,FOR A GIVEN STATION AND CRUISE NUMBER                   MAIN1320
C             KEK2=26                                                  MAIN1330
C     R01    -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1340
C             SPEED AT ONE DAY BEFORE THE DAY OF BT/XBT OBSERVATION,   MAIN1350
C             FOR A GIVEN STATION AND CRUISE NUMBER                    MAIN1360
C             KEK2=17                                                  MAIN1370
C     R011   -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1380
C             SPEED (ALONG THE COAST) AT ONE DAY BEFORE THE DAY OF     MAIN1390
C             BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE       MAIN1400
C             NUMBER                                                   MAIN1410
C             KEK2=27                                                  MAIN1420
C     R012   -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1430
C             SPEED (PERP. TO THE COAST) AT ONE DAY BEFORE THE DAY     MAIN1440
```

```
C           OFBT/XBT CBSERVATION, FOR A GIVEN STATION AND CRUISE     MAIN1450
C           NUMBER                                                   MAIN1460
C           KEK2=28                                                  MAIN1470
C     R02  -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1480
C           SPEED AT TWO DAYS BEFORE THE DAY OF BT/XBT OBSERVATION,  MAIN1490
C           FOR A GIVEN STATION AND CRUISE NUMBER                    MAIN1500
C           KEK2=18                                                  MAIN1510
C     R021 -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1520
C           SPEED (ALONG THE COAST) AT TWO DAYS BEFORE THE DAY OF    MAIN1530
C           BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE       MAIN1540
C           NUMBER                                                   MAIN1550
C           KEK2=29                                                  MAIN1560
C     R022 -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1570
C           SPEED (PERP. TO THE COAST) AT TWO DAYS BEFORE THE DAY    MAIN1580
C           CF BT/XBT OSERVATION, FOR A GIVEN STATION AND CRUISE     MAIN1590
C           NUMBER                                                   MAIN1600
C           KEK2=30                                                  MAIN1610
C     R03  -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1620
C           SPEED AT THREE DAYS BEFORE THE DAY OF BT/XBT             MAIN1630
C           CBSERVATICN, FOR A GIVEN STATION AND CRUISE NUMBER       MAIN1640
C           KEK2=19                                                  MAIN1650
C     R031 -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1660
C           SPEED (ALONG THE COAST) AT THREE DAYS BEFORE THE DAY     MAIN1670
C           OF BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE    MAIN1680
C           NUMBER                                                   MAIN1690
C           KEK2=31                                                  MAIN1700
C     R032 -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1710
C           SPEED (PERP. TO THE COAST) AT THREE DAYS BEFORE THE DAY MAIN1720
C           OF BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE    MAIN1730
C           NUMBER                                                   MAIN1740
C           KEK2=32                                                  MAIN1750
C     R05  -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1760
C           SPEED AT FIVE DAYS BEFORE THE DAY OF BT/XBT OBSERVATION MAIN1770
C           , FOR A GIVEN STATION AND CRUISE NUMBER                  MAIN1780
C           KEK2=20                                                  MAIN1790
C     R051 -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1800
C           SPEED (ALONG THE COAST) AT FIVE DAYS BEFORE THE DAY      MAIN1810
C           OF BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE    MAIN1820
C           NUMBER                                                   MAIN1830
C           KEK2=33                                                  MAIN1840
C     R052 -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1850
C           SPEED (PERP. TO THE CCAST) AT FIVE CAYS BEFORE THE DAY   MAIN1860
C           OF BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE    MAIN1870
C           NUMBER                                                   MAIN1880
C           KEK2=34                                                  MAIN1890
C     R10  -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND     MAIN1900
C           SPEED AT TEN DAYS BEFORE THE DAY OF BT/XBT OBSERVATION,  MAIN1910
C           FOR A GIVEN STATION AND CRUISE NUMBER                    MAIN1920
```

```
C           KEK2=24                                                    MAIN1930
C     R101  -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND      MAIN1940
C           SPEED (ALONG THE COAST) AT TEN DAYS BEFORE THE DAY         MAIN1950
C           OF BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE      MAIN1960
C           NUMBER                                                     MAIN1970
C           KEK2=35                                                    MAIN1980
C     R102  -INPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE WIND      MAIN1990
C           SPEED (PERP. TO THE COAST) AT TEN DAYS BEFORE THE DAY      MAIN2000
C           OF BT/XBT OBSERVATION, FOR A GIVEN STATION AND CRUISE      MAIN2010
C           NUMBER                                                     MAIN2020
C           KEK2=36                                                    MAIN2030
C     TI    -INPUT VECTOR OF LENGTH 'ISO' CONTAINING THE VALUES OF     MAIN2040
C           THE ISOTHERMS                                              MAIN2050
C     TO    -INPUT VECTOR OF LENGTH 'NCP' CONTAINING THE               MAIN2060
C           TEMPERATURES OF POINTS DEFINING A DIGITAL TRACE            MAIN2070
C     TSFC  -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE          MAIN2080
C           TEMPERATURE OF THE SEA SURFACE AT A GIVEN STATION AND      MAIN2090
C           CRUISE NUMBER                                              MAIN2100
C           KEK1=29                                                    MAIN2110
C     X     -INPUT VECTOR OF LENGTH 'NBT' CONTAINING THE LOCATION OF MAIN2120
C           EACH STATION FROM A COMMON ORIGIN (IN LENGTH OR TIME)      MAIN2130
C           AND ALONG THE CROSS SECTION OR TIME AXIS                   MAIN2140
C     XA    -VECTOR OF LENGTH TWO CONTAINING THE ABCISSAS OF TWO       MAIN2150
C           ARTIFICIAL POINTS TO COMPLETE THE WIND PLOTS               MAIN2160
C     XGRID -INPUT VALUE CONTAINING THE LOCATION OF THE STATION        MAIN2170
C           (ALONG THE CROSS SECTION OR TIME AXIS) WHERE THE WIND      MAIN2180
C           DATA IS REPORTED TO                                        MAIN2190
C     YGRID -INPUT CONTAINING THE VALUE OF THE ORDINATE TO LOCATE      MAIN2200
C           THE WIND VECTORS IN THE PLOT                               MAIN2210
C     ZO    -INPUT VECTOR OF LENGTH 'NCP' CONTAINING THE DEPTHS OF     MAIN2220
C           POINTS DEFINING A DIGITAL TRACE                            MAIN2230
C     ZTH1S -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH    MAIN2240
C           OF THE TOP OF SHALLOW THERMOCLINE                          MAIN2250
C           KEK1=10                                                    MAIN2260
C     ZTH1I -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH    MAIN2270
C           OF THE TOP OF INTERMEDIATE THERMOCLINE                     MAIN2280
C           KEK1=11                                                    MAIN2290
C     ZTH1D -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH    MAIN2300
C           OF THE TOP OF DEEP THERMOCLINE                             MAIN2310
C           KEK1=12                                                    MAIN2320
C     ZTH2S -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE          MAIN2330
C           MEAN DEPTH OF THE SHALLOW THERMOCLINE                      MAIN2340
C           KEK1=13                                                    MAIN2350
C     ZTH2I -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE          MAIN2360
C           MEAN DEPTH OF THE INTERMEDIATE THERMOCLINE                 MAIN2370
C           KEK1=14                                                    MAIN2380
C     ZTH2D -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE          MAIN2390
C           MEAN DEPTH OF THE DEEP THERMOCLINE                         MAIN2400
```

```
C           KEK1=15                                                       MAIN2410
C     ZTH3S -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2420
C           OF THE BOTTOM OF SHALLOW THERMOCLINE                          MAIN2430
C           KEK1=16                                                       MAIN2440
C     ZTH3I -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2450
C           OF THE BOTTOM OF INTERMEDIATE THERMOCLINE                     MAIN2460
C           KEK1=17                                                       MAIN2470
C     ZTH3D -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2480
C           OF THE BOTTOM OF DEEP THERMOCLINE                             MAIN2490
C           KEK1=18                                                       MAIN2500
C     ZTHWS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE RANGE       MAIN2510
C           OF THE SHALLOW THERMOCLINE                                    MAIN2520
C           KEK1=19                                                       MAIN2530
C     ZTHWI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE RANGE       MAIN2540
C           OF THE INTERMEDIATE THERMOCLINE                               MAIN2550
C           KEK1=20                                                       MAIN2560
C     ZTHWD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE RANGE       MAIN2570
C           OF THE DEEP THERMOCLINE                                       MAIN2580
C           KEK1=24                                                       MAIN2590
C     ZTHVS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE             MAIN2600
C           THERMAL WIDTH OF THE SHALLOW THERMOCLINE                      MAIN2610
C           KEK1=25                                                       MAIN2620
C     ZTHVI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE             MAIN2630
C           THERMAL WIDTH OF THE INTERMEDIATE THERMOCLINE                 MAIN2640
C           KEK1=26                                                       MAIN2650
C     ZTHVD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE             MAIN2660
C           THERMAL WIDTH OF THE DEEP THERMOCLINE                         MAIN2670
C           KEK1=27                                                       MAIN2680
C     ZMIXLR-OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2690
C           OF THE MIXED LAYER                                            MAIN2700
C           KEK1=28                                                       MAIN2710
C     YA    -VECTOR OF LENGTH TWO CONTAINING THE ORDINATES OF TWO         MAIN2720
C           ARTIFICIAL POINTS TO COMPLETE THE WIND PLOTS                  MAIN2730
C     Z11CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2740
C           OF THE 11 DEG CELSIUS SHALLOW ISOTHERM                        MAIN2750
C     Z11CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2760
C           OF THE 11 DEG CELSIUS INTERMEDIATE ISOTHERM                   MAIN2770
C     Z11CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2780
C           OF THE 11 DEG CELSIUS DEEP ISOTHERM                           MAIN2790
C           KEK1=35                                                       MAIN2800
C     Z12CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2810
C           OF THE 12 DEG CELSIUS SHALLOW ISOTHERM                        MAIN2820
C     Z12CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2830
C           OF THE 12 DEG CELSIUS INTERMEDIATE ISOTHERM                   MAIN2840
C     Z12CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2850
C           OF THE 12 DEG CELSIUS DEEP ISOTHERM                           MAIN2860
C           KEK1=30                                                       MAIN2870
C     Z13CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH       MAIN2880
```

```
C          OF THE 13 DEG CELSIUS SHALLOW ISOTHERM                     MAIN2890
C    Z13CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN2900
C          OF THE 13 DEG CELSIUS INTERMEDIATE ISOTHERM                MAIN2910
C    Z13CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN2920
C          OF THE 13 DEG CELSIUS DEEP ISOTHERM                        MAIN2930
C          KEK1=36                                                    MAIN2940
C    Z14CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN2950
C          OF THE 14 DEG CELSIUS SHALLOW ISOTHERM                     MAIN2960
C    Z14CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN2970
C          OF THE 14 DEG CELSIUS INTERMEDIATE ISOTHERM                MAIN2980
C    Z14CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN2990
C          OF THE 14 DEG CELSIUS DEEP ISOTHERM                        MAIN3000
C          KEK1=37                                                    MAIN3010
C    Z15CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3020
C          OF THE 15 DEG CELSIUS SHALLOW ISOTHERM                     MAIN3030
C    Z15CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3040
C          OF THE 15 DEG CELSIUS INTERMEDIATE ISOTHERM                MAIN3050
C    Z15CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3060
C          OF THE 15 DEG CELSIUS DEEP ISOTHERM                        MAIN3070
C          KEK1=38                                                    MAIN3080
C    Z16CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3090
C          OF THE 16 DEG CELSIUS SHALLOW ISOTHERM                     MAIN3100
C    Z16CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3110
C          OF THE 16 DEG CELSIUS INTERMEDIATE ISOTHERM                MAIN3120
C    Z16CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3130
C          OF THE 16 DEG CELSIUS DEEP ISOTHERM                        MAIN3140
C          KEK1=39                                                    MAIN3150
C    Z17CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3160
C          OF THE 17 DEG CELSIUS SHALLOW ISOTHERM                     MAIN3170
C    Z17CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3180
C          OF THE 17 DEG CELSIUS INTERMEDIATE ISOTHERM                MAIN3190
C    Z17CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3200
C          OF THE 17 DEG CELSIUS DEEP ISOTHERM                        MAIN3210
C          KEK1=40                                                    MAIN3220
C    Z18CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3230
C          OF THE 18 DEG CELSIUS SHALLOW ISOTHERM                     MAIN3240
C    Z18CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3250
C          OF THE 18 DEG CELSIUS INTERMEDIATE ISOTHERM                MAIN3260
C    Z18CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3270
C          OF THE 18 DEG CELSIUS DEEP ISOTHERM                        MAIN3280
C          KEK1=41                                                    MAIN3290
C    Z19CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3300
C          OF THE 19 DEG CELSIUS SHALLOW ISOTHERM                     MAIN3310
C    Z19CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3320
C          OF THE 19 DEG CELSIUS INTERMEDIATE ISOTHERM                MAIN3330
C    Z19CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3340
C          OF THE 19 DEG CELSIUS DEEP ISOTHERM                        MAIN3350
C          KEK1=42                                                    MAIN3360
```

```
C        Z20CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3370
C              OF THE 20 DEG CELSIUS SHALLOW ISOTHERM                    MAIN3380
C        Z20CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3390
C              OF THE 20 DEG CELSIUS INTERMEDIATE ISOTHERM               MAIN3400
C        Z20CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3410
C              OF THE 20 DEG CELSIUS DEEP ISOTHERM                       MAIN3420
C              KEK1=43                                                   MAIN3430
C        Z21CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3440
C              OF THE 21 DEG CELSIUS SHALLOW ISOTHERM                    MAIN3450
C        Z21CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3460
C              OF THE 21 DEG CELSIUS INTERMEDIATE ISOTHERM               MAIN3470
C        Z21CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3480
C              OF THE 21 DEG CELSIUS DEEP ISOTHERM                       MAIN3490
C              KEK1=44                                                   MAIN3500
C        Z22CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3510
C              OF THE 22 DEG CELSIUS SHALLOW ISOTHERM                    MAIN3520
C        Z22CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3530
C              OF THE 22 DEG CELSIUS INTERMEDIATE ISOTHERM               MAIN3540
C        Z22CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3550
C              OF THE 22 DEG CELSIUS DEEP ISOTHERM                       MAIN3560
C              KEK1=45                                                   MAIN3570
C        Z23CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3580
C              OF THE 23 DEG CELSIUS SHALLOW ISOTHERM                    MAIN3590
C        Z23CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3600
C              OF THE 23 DEG CELSIUS INTERMEDIATE ISOTHERM               MAIN3610
C        Z23CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3620
C              OF THE 23 DEG CELSIUS DEEP ISOTHERM                       MAIN3630
C              KEK1=46                                                   MAIN3640
C        Z24CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3650
C              OF THE 24 DEG CELSIUS SHALLOW ISOTHERM                    MAIN3660
C        Z24CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3670
C              OF THE 24 DEG CELSIUS INTERMEDIATE ISOTHERM               MAIN3680
C        Z24CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3690
C              OF THE 24 DEG CELSIUS DEEP ISOTHERM                       MAIN3700
C              KEK1=47                                                   MAIN3710
C        Z25CS -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3720
C              OF THE 25 DEG CELSIUS SHALLOW ISOTHERM                    MAIN3730
C        Z25CI -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3740
C              OF THE 25 DEG CELSIUS INTERMEDIATE ISOTHERM               MAIN3750
C        Z25CD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3760
C              OF THE 25 DEG CELSIUS DEEP ISOTHERM                       MAIN3770
C              KEK1=48                                                   MAIN3780
C        ZSUCD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3790
C              OF THE ISOTHERM (SU) DEFINING THE TOP OF A GIVEN LAYER    MAIN3800
C              (S)                                                       MAIN3810
C              KEK1=31                                                   MAIN3820
C        ZSLCD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH   MAIN3830
C              OF THE ISOTHERM (SL) DEFINING THE BOTTOM OF A GIVEN       MAIN3840
```

```
C             LAYER (S)
C             KEK1=32
C        ZTUCD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH
C             OF THE ISOTHERM (TU) DEFINING THE TOP OF A GIVEN LAYER
C             (T)
C             KEK1=33
C        ZTLCD -OUTPUT MATRIX OF SIZE 'NBT*ICRUI' CONTAINING THE DEPTH
C             OF THE ISOTHERM (TL) DEFINING THE BOTTOM OF A GIVEN
C             LAYER (T)
C             KEK1=34
C        ZTTMD -OUTPUT MATRIX OF SIZE 'ISC*NBT' CONTAINING THE AVERAGE
C             DEPTH OF THE DEEP ISOTHERMS ALONG THE CROSS SECTION
C
C     INPUT,OUTPUT AND OPERATION
C        SEE SALDANHA,J.M.,1971.'OBJECTIVE DIGITAL ANALYSIS OF A
C        TRACE'.M.S.THESIS, NAVAL POSTGRADUATE SCHOOL,
C        MONTEREY, CALIFORNIA 93940
C
C     SUBROUTINES AND FUNCTIONS REQUIRED
C        ABS (IBM SCIENTIC LIBRARY)
C        AMAX1 (IBM SCIENTIFIC LIBRARY)
C        AMOD (IBM SCIENTIFIC LIBRARY)
C        ARTFT1
C        ARTFT2
C        COS (IBM SCIENTIFIC LIBRARY)
C        DRAW (IBM SCIENTIFIC LIBRARY)
C        GTVALU
C        LINE
C        LINTPS
C        LINTP1
C        LINTP2
C        MISR (IBM SCIENTIFIC SUBROUTINE LIBRARY)
C        PLOTBT
C        PLOTIT
C        PLOTFT
C        PLTMIT
C        PRINT
C        RLINE
C        SIN (IBM SCIENTIFIC LIBRARY)
C        STATCS
C        THERMC
C        TINC1
C        TINC2
C        VECTOR
C        WINDO0
C        WINDO1
C        WINDO2
C        WINDO3
```

```
C        WINDO5
C        WINDO10
C
C     ..........................................................................
C
      REAL*8 ITITL1(12)
      REAL*8 ITITL2(12)
      REAL*8 ITITL3(12)
      REAL*8 ITITL4(12)
      REAL*8 ITITL5(12)
      REAL*8 ITITL6(12)
      REAL LABEL/4H    /
      DIMENSION XA(2),YA(2)
      COMMON/WORKA/TO(60),ZO(60),BLANK(1332)
      COMMON/WORKB/X(20),TI(19)
      COMMON/WORKC/ZTHIS(20,12),ZTH1I(20,12),ZTH1D(20,12),ZTH2S(20,12),
     *ZTH2I(20,12),ZTH2D(20,12),ZTH3S(20,12),ZTH3I(20,12),ZTH3D(20,12),
     *ZTHWS(20,12),ZTHWI(20,12),ZTHWD(20,12),TTHVS(20,12),TTHVI(20,12),
     *TTHVD(20,12),ZMIXLR(20,12),TSFC(20,12)
      COMMON/WORKD/R00(20,12),KANG00(20,12),R01(20,12),KANG01(20,12),
     *R02(20,12),KANG02(20,12),R03(20,12),KANG03(20,12),R05(20,12),
     *KANG05(20,12),R10(20,12),KANG10(20,12),R001(20,12),R002(20,12),
     *R011(20,12),R012(20,12),R021(20,12),R022(20,12),R031(20,12),
     *R032(20,12),R051(20,12),R052(20,12),R101(20,12),R102(20,12)
      COMMON/WORKE/Z11CS(20,12),Z11CI(20,12),Z11CD(20,12),Z12CS(20,12),
     *Z12CI(20,12),Z12CD(20,12),Z13CS(20,12),Z13CI(20,12),Z13CD(20,12),
     *Z14CS(20,12),Z14CI(20,12),Z14CD(20,12),Z15CS(20,12),Z15CI(20,12),
     *Z15CD(20,12),Z16CS(20,12),Z16CI(20,12),Z16CD(20,12),Z17CS(20,12),
     *Z17CI(20,12),Z17CD(20,12),Z18CS(20,12),Z18CI(20,12),Z18CD(20,12),
     *Z19CS(20,12),Z19CI(20,12),Z19CD(20,12),Z20CS(20,12),Z20CI(20,12),
     *Z20CD(20,12),Z21CS(20,12),Z21CI(20,12),Z21CD(20,12),Z22CS(20,12),
     *Z22CI(20,12),Z22CD(20,12),Z23CS(20,12),Z23CI(20,12),Z23CD(20,12),
     *Z24CS(20,12),Z24CI(20,12),Z24CD(20,12),Z25CS(20,12),Z25CI(20,12),
     *Z25CD(20,12),ZTUCS(20,12),ZTUCI(20,12),ZTUCD(20,12),ZTLCS(20,12),
     *ZTLCI(20,12),ZTLCD(20,12),ZSUCS(20,12),ZSUCI(20,12),ZSUCD(20,12),
     *ZSLCS(20,12),ZSLCI(20,12),ZSLCD(20,12),ZTTMD(19,12)
C
C     CROSS SECTION INFORMATION
C
      READ(5,13) ICSEC,ICRUI
   13 FORMAT(2I3)
      WRITE(6,3) ICSEC,ICRUI
    3 FORMAT('1',//////////,40X,'*   BEGINNING OF THE STUDY FOR CROSS SE
     *CTION   ',I2,'   *',////,30X,'NO. OF CRUISES ALONG CSECTION =',I3)
      READ(5,5) ISO
    5 FORMAT(I3)
      WRITE(6,7) ISO
    7 FORMAT('0',30X,'NO. OF ISOTHERMS =',I3)
```

```
      READ(5,9)(TI(I10),I10=1,ISO)                                       MAIN4810
    9 FCRMAT(8F10.0)                                                     MAIN4820
      WRITE(6,10)(I10,TI(I10),I10=1,ISO)                                 MAIN4830
   10 FCRMAT(' ',30X,'TI(',I2,')=',F5.1)                                 MAIN4840
      READ(5,11) DTW                                                     MAIN4850
   11 FCRMAT(F10.0)                                                      MAIN4860
      WRITE(6,15) DTW                                                    MAIN4870
   15 FORMAT(' ',30X,'THERMAL INCREMENT =',F5.2)                         MAIN4880
      READ(5,17) NBT                                                     MAIN4890
   17 FORMAT(I3)                                                         MAIN4900
      WRITE(6,19) NBT                                                    MAIN4910
   19 FORMAT(' ',30X,'NO. CF STATIONS ALONG THE CROSS SECTION =',I3)     MAIN4920
      READ(5,21)(X(I12),I12=1,NBT)                                       MAIN4930
   21 FORMAT(8F10.0)                                                     MAIN4940
      WRITE(6,23)(I12,X(I12),I12=1,NBT)                                  MAIN4950
   23 FCRMAT(' ',30X,'X(',I2,')=',F5.2)                                  MAIN4960
C                                                                        MAIN4970
C     BEGINNING CF CROSS SECTION STUDY                                   MAIN4980
C                                                                        MAIN4990
      NCRDER=1                                                           MAIN5000
      DC 460 K98=1,ICRUI                                                 MAIN5010
C                                                                        MAIN5020
C     CRUISE INFORMATICN                                                 MAIN5030
C                                                                        MAIN5040
      READ(5,25)(ITITL1(I),I=1,12)                                       MAIN5050
   25 FCRMAT(6A8)                                                        MAIN5060
      READ(5,27)(ITITL2(I),I=1,12)                                       MAIN5070
   27 FCRMAT(6A8)                                                        MAIN5080
      READ(5,29)(ITITL3(I),I=1,12)                                       MAIN5090
   29 FORMAT(6A8)                                                        MAIN5100
      READ(5,31)(ITITL4(I),I=1,12)                                       MAIN5110
   31 FCRMAT(6A8)                                                        MAIN5120
C                                                                        MAIN5130
C     PROCESSING THE FIELD OF ISOTHERMS                                  MAIN5140
C                                                                        MAIN5150
      DC 250 J7=1,NBT                                                    MAIN5160
      READ(5,33) ISTA,NOP                                                MAIN5170
   33 FORMAT(2I3)                                                        MAIN5180
      IF(NOP.EQ.0) GO TO 240                                             MAIN5190
      WRITE(6,35) ISTA,NOP                                               MAIN5200
   35 FCRMAT('0',30X,'STATION NO. =',I3,10X,'NO. CF POINTS DEFINING XBT  MAIN5210
     * TRACE =',I3)                                                      MAIN5220
      READ(5,41)(ZO(I50),TO(I50),I50=1,NOP)                              MAIN5230
   41 FCRMAT(F6.0,F9.0,F6.0,F9.0,F6.0,F9.0,F6.0,F9.0,F6.0,F9.0)          MAIN5240
      WRITE(6,42)                                                        MAIN5250
   42 FORMAT(///,20X,'    * * *  OBSERVED PAIRS FRCM XBT TRACE  * * *    MAIN5260
     * ',//)                                                             MAIN5270
      WRITE(6,43)(I40,TO(I40),I40,ZO(I40),I40=1,NOP)                     MAIN5280
```

```
   43 FORMAT(' ',30X,'TO(',I2,')=',F5.2,10X,'ZO(',I2,')=',F6.2)
      TSFC(ISTA,NORDER)=TO(1)
      CALL GTVALU(ISO,NOP,J7,ISTA,NORDER)
      GO TO 250
  240 WRITE(6,244) ISTA
  244 FORMAT('0',2X,'ISTA =',I3,10X,'NO XBT TRACE')
      CALL ARTFT1(ISO,J7,ISTA,NORDER)
  250 CONTINUE
      CALL PLOTIT(ISO,NBT,LABEL,ITITL2)
C
C     PROCESSING THE DIGITAL TRACES AND THE THERMOCLINE ANALYSIS
C
      DO 350 J8=1,NBT
      READ(5,33) ISTA,NOP
      IF(NOP.EQ.0) GO TO 340
      WRITE(6,35) ISTA,NOP
      READ(5,41)(ZO(I50),TO(I50),I50=1,NOP)
      WRITE(6,42)
      WRITE(6,43)(I40,TO(I40),I40,ZO(I40),I40=1,NOP)
      CALL LINTPS(NOP,DTW,M60)
      CALL PLOTBT(M60,J8,NBT,ITITL1,LABEL)
      CALL THERMC(NORDER,ISTA,J8,M60)
      GO TO 350
  340 WRITE(6,344) ISTA
  344 FORMAT('0',2X,'ISTA =',I3,10X,'NO XBT TRACE')
      CALL ARTFT2(NORDER,ISTA,J8)
      M60=0
      CALL PLOTBT(M60,J8,NBT,ITITL1,LABEL)
  350 CONTINUE
      CALL PLOTFT(NBT,LABEL,ITITL3)
      N10=1
      DO 450 J3=1,NBT
C
C     WIND INFORMATION
C
      READ(5,37) ISTA
   37 FORMAT(I3)
      WRITE(6,39) ISTA
   39 FORMAT('0',30X,'STATION NO. =',I3)
      READ(5,310)  R00(NORDER,ISTA),KANG00(NORDER,ISTA),R01(NORDER,ISTA)
     *,KANG01(NORDER,ISTA),R02(NORDER,ISTA),KANG02(NORDER,ISTA)
     *,R03(NORDER,ISTA),KANG03(NORDER,ISTA),
     *R05(NORDER,ISTA),KANG05(NORDER,ISTA),R10(NORDER,ISTA),
     *KANG10(NORDER,ISTA),XGRID,YGRID,KAMMA1,KKK
  310 FORMAT(F7.0,I3,F7.0,I3,F7.0,I3,F7.0,I3,F7.0,I3,F7.0,I3,2F5.0,I3,
     *I1)
C
C     PROCESSING THE WIND FIELD
```

```
C
      CALL WINDOC(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,
     *N10)
      CALL WINDO1(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,
     *N10)
      CALL WINDO2(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,
     *N10)
      CALL WINDO3(NORDER,ISTA,XGRID,YGRID KKK,N10,LABEL,ITITL4,KAMMA1,
     *N10)
      CALL WINDO5(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,
     *N10)
      CALL WIND10(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,
     *N10)
  450 CONTINUE
      NPLOT=3
      IF(N10.EQ.1) NPLOT=0
      XA(1)=0.
      XA(2)=1.
      YA(1)=1.
      YA(2)=0.
      CALL DRAW(2,XA,YA,NPLOT,5,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,LAST)
      WRITE(6,257) LAST
  257 FORMAT('0',10X,'LAST =',I1)
      IF(K98.EQ.ICRU1) GO TO 460
      NORDER=NORDER+1
  460 CONTINUE
C
C     END OF CROSS SECTION STUDY
C
      READ(5,600) IKONTR
  600 FORMAT(I3)
C
C     BEGINNING OF STATISTICAL STUDY
C
      WRITE(6,601)
  601 FORMAT(///,50X,'*   BEGINNING OF THE STATISTICAL STUDY   *',//)
      IF(IKONTR.EQ.0) GO TO 1000
C
C     STATISTICAL STUDY INFORMATION
C
      READ(5,608) IERIOD,IBEGIN,IEND
  608 FORMAT(3I8)
  602 READ(5,604) IORDER,MORDER,ISTA,KEK1,KEK2,KRK
  604 FORMAT(6I8)
      READ(5,606)(ITITL5(I),I=1,12)
  606 FORMAT(6A8)
C
C     PROCESSING THE STATISTICAL STUDY
```

```
C
      CALL STATCS(ISTA,IORDER,MORDER,KEK1,KEK2,ITITL5,LABEL,ICSEC,IERIOD
     *,IBEGIN,IEND)
      IF(KRK.NE.0) GO TO 602
      READ(5,612) ILONTR
  612 FORMAT(I3)
      IF(ILONTR.EQ.0) GO TO 1000
      READ(5,900)(ITITL6(I),I=1,12)
  900 FORMAT(6A8)
      CALL PLTMIT(ISO,NBT,LABEL,ITITL6)
C
C     END OF STATISTICAL STUDY
C
      WRITE(6,995) ICSEC
  995 FORMAT(//,50X,'*   END OF STATISTICAL STUDY   *',////,40X,'*   END
     * OF STUDY FOR CROSS SECTION  ',I2,'   *')
C
C     STORAGE OF PROCESSED AND ORIGINAL DATA
C
 1000 WRITE (9) ZTH1S,ZTH1I,ZTH1D
      WRITE (9) ZTH2S,ZTH2I,ZTH2D
      WRITE (9) ZTH3S,ZTH3I,ZTH3D
      WRITE (9) ZTHWS,ZTHWI,ZTHWD
      WRITE (9) TTHVS,TTHVI,TTHVD
      WRITE (9) ZMIXLR,TSFC,R00
      WRITE (9) KANG00,R01,KANG01
      WRITE (9) R02,KANG02,R03
      WRITE (9) KANG03,R05,KANG05
      WRITE (9) R10,KANG10,R001
      WRITE (9) R002,R011,R012
      WRITE (9) R021,R022,R031
      WRITE (9) R032,R051,R052
      WRITE (9) R101,R102,Z11CS
      WRITE (9) Z11CI,Z11CD,Z12CS
      WRITE (9) Z12CI,Z12CD,Z13CS
      WRITE (9) Z13CI,Z13CD,Z14CS
      WRITE (9) Z14CI,Z14CD,Z15CS
      WRITE (9) Z15CI,Z15CD,Z16CS
      WRITE (9) Z16CI,Z16CD,Z17CS
      WRITE (9) Z17CI,Z17CD,Z18CS
      WRITE (9) Z18CI,Z18CD,Z19CS
      WRITE (9) Z19CI,Z19CD,Z20CS
      WRITE (9) Z20CI,Z20CD,Z21CS
      WRITE (9) Z21CI,Z21CD,Z22CS
      WRITE (9) Z22CI,Z22CD,Z23CS
      WRITE (9) Z23CI,Z23CD,Z24CS
      WRITE (9) Z24CI,Z24CD,Z25CS
      WRITE (9) Z25CI,Z25CD,ZTUCS
```

```
      WRITE (9) ZTUCI,ZTUCD,ZTLCS                                       MAIN6730
      WRITE (9) ZTLCI,ZTLCD,ZSUCS                                       MAIN6740
      WRITE (9) ZSUCI,ZSUCD,ZSLCS                                       MAIN6750
      WRITE (9) ZSLCI,ZSLCD,ZTTMD                                       MAIN6760
      STOP                                                              MAIN6770
      END                                                               MAIN6780
C                                                                       ART10010
C     ..................................................................ART10020
C                                                                       ART10030
C     SUBROUTINE ARTFT1                                                 ART10040
C                                                                       ART10050
C     PURPOSE                                                           ART10060
C        IN THE ABSENCE OF A DIGITAL TRACE, ARTIFICIAL INFORMATION IS   ART10070
C        GENERATED TO FILL IN THE ALLOCATED SPACE FOR THE FIELD OF      ART10080
C        ISOTHERMS                                                      ART10090
C                                                                       ART10100
C     USAGE                                                             ART10110
C        CALL ARTFT1(ISO,J1,ISTA,NORDER)                                ART10120
C                                                                       ART10130
C     DESCRIPTION OF PARAMETERS                                         ART10140
C        ISO   -INPUT NUMBER OF ISOTHERMS                               ART10150
C        J1    -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED AT A GIVEN    ART10160
C               CRUISE                                                  ART10170
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS        ART10180
C               SECTION                                                 ART10190
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                      ART10200
C                                                                       ART10210
C     SUBROUTINES REQUIRED                                              ART10220
C        NONE                                                           ART10230
C                                                                       ART10240
C     ..................................................................ART10250
C                                                                       ART10260


      SUBROUTINE ARTFT1(ISO,J1,ISTA,NORDER)                             ART10270
      COMMON/WORKA/TO(60),ZO(60),ZSHAL(20,20),ZINTR(20,20),ZDEEP(20,20),ART10280
     *ZA(10),BLANS1(122)                                                ART10290
      COMMON/WORKC/ZTH1S(20,12),ZTH1I(20,12),ZTH1D(20,12),ZTH2S(20,12), ART10300
     *ZTH2I(20,12),ZTH2D(20,12),ZTH3S(20,12),ZTH3I(20,12),ZTH3D(20,12), ART10310
     *ZTHWS(20,12),ZTHWI(20,12),ZTHWD(20,12),TTHVS(20,12),TTHVI(20,12), ART10320
     *TTHVD(20,12),ZMIXLR(20,12),TSFC(20,12)                            ART10330
      DO 40 J2=1,ISO                                                    ART10340
      ZSHAL(J1,J2)=-5.                                                  ART10350
      ZINTR(J1,J2)=-5.                                                  ART10360
      ZDEEP(J1,J2)=-5.                                                  ART10370
      WRITE(6,41) J1,J2,ZSHAL(J1,J2),J1,J2,ZINTR(J1,J2),J1,J2,          ART10380
     *ZDEEP(J1,J2)                                                      ART10390
   41 FORMAT(' ',2X,'ZSHAL(',I2,',',I2,') =',2X,F11.7,10X,'ZINTR(',I2,  ART10400
```

```
     *',',I2,') =',2X,F11.7,10X,'ZDEEP(',I2,',',I2,') =',2X,F11.7)      ART10410
   40 CONTINUE                                                          ART10420
      TSFC(ISTA,NORDER)=-5.                                             ART10430
      RETURN                                                            ART10440
      END                                                               ART10450
C                                                                       ART20010
C     ..................................................................ART20020
C                                                                       ART20030
C     SUBROUTINE ARTFT2                                                 ART20040
C                                                                       ART20050
C     PURPOSE                                                           ART20060
C        IN THE ABSENCE OF A DIGITAL TRACE, ARTIFICIAL INFORMATION IS   ART20070
C        GENERATED TO FILL IN THE ALLOCATED SPACE FOR THE THERMOCLINE   ART20080
C        DATA OBTAINED FROM SUBROUTINE THERMC                           ART20090
C                                                                       ART20100
C     USAGE                                                             ART20110
C        CALL ARTFT2(NORDER,ISTA,J1)                                    ART20120
C                                                                       ART20130
C     DESCRIPTION OF PARAMETERS                                         ART20140
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                      ART20150
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS        ART20160
C               SECTION                                                 ART20170
C        J1    -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED AT A GIVEN    ART20180
C               CRUISE                                                  ART20190
C                                                                       ART20200
C     SUBROUTINES REQUIRED                                              ART20210
C        NONE                                                           ART20220
C                                                                       ART20230
C     ..................................................................ART20240
C                                                                       ART20250


      SUBROUTINE ARTFT2(NORDER,ISTA,J1)                                 ART20260
      COMMON/WORKA/TO(60),ZO(60),T(120),Z(120),ZAH1S(20),ZAH1I(20),     ART20270
     *ZAH1D(20),ZAH2S(20),ZAH2I(20),ZAH2D(20),ZAH3S(20),ZAH3I(20),      ART20280
     *ZAH3D(20),ZAIXLR(20),Z1(150),Z2(150),Z3(150),ZAUXT(15),ZAUXB(15), ART20290
     *ZAUXC(15),ZAUXP(15),TAUXR(15),TAUXG(15),ZANGE1(15),ZANGE(15),     ART20300
     *TAUXT(15),TAUXG(15),ZAUBT(15),ZAUBB(15),ZAUBC(15),ZAUBR(15),      ART20310
     *TAUBR(15),TAUBG(15),ZANBE1(15),ZANBE(15),TAUBT(15),TAUBB(15),     ART20320
     *ZAUCT(15),ZAUCB(15),ZAUCR(15),ZANCE1(15),ZAUCC(15),TAUCT(15),     ART20330
     *TAUCB(15),TAUCR(15),TAUCG(15),BLAN73(7)                           ART20340
      COMMON/WORKC/ZTH1S(20,12),ZTH1I(20,12),ZTH1D(20,12),ZTH2S(20,12), ART20350
     *ZTH2I(20,12),ZTH2D(20,12),ZTH3S(20,12),ZTH3I(20,12),ZTH3D(20,12), ART20360
     *ZTHWS(20,12),ZTHWI(20,12),ZTHWD(20,12),TTHVS(20,12),TTHVI(20,12), ART20370
     *TTHVD(20,12),ZMIXLR(20,12),TSFC(20,12)                            ART20380
      ZAH1S(J1)=-5.                                                     ART20390
      ZAH1I(J1)=-5.                                                     ART20400
      ZAH1D(J1)=-5.                                                     ART20410
```

```
      ZAH2S(J1)=-5.                                                     ART20420
      ZAH2I(J1)=-5.                                                     ART20430
      ZAH2D(J1)=-5.                                                     ART20440
      ZAH3S(J1)=-5.                                                     ART20450
      ZAH3I(J1)=-5.                                                     ART20460
      ZAH3D(J1)=-5.                                                     ART20470
      ZTH1S(NORDER,ISTA)=-5.                                            ART20480
      ZTH1I(NORDER,ISTA)=-5.                                            ART20490
      ZTH1D(NCRDER,ISTA)=-5.                                            ART20500
      ZTH2S(NORDER,ISTA)=-5.                                            ART20510
      ZTH2I(NORDER,ISTA)=-5.                                            ART20520
      ZTH2D(NORDER,ISTA)=-5.                                            ART20530
      ZTH3S(NCRDER,ISTA)=-5.                                            ART20540
      ZTH3I(NCRDER,ISTA)=-5.                                            ART20550
      ZTH3D(NCRDER,ISTA)=-5.                                            ART20560
      ZTHWS(NCRDER,ISTA)=-5.                                            ART20570
      ZTHWI(NORDER,ISTA)=-5.                                            ART20580
      ZTHWD(NORDER,ISTA)=-5.                                            ART20590
      TTHVS(NORDER,ISTA)=-5.                                            ART20600
      TTHVI(NORDER,ISTA)=-5.                                            ART20610
      TTHVD(NORDER,ISTA)=-5.                                            ART20620
      ZMIXLR(NCRDER,ISTA)=-5.                                           ART20630
      ZAIXLR(J1)=-5.                                                    ART20640
      RETURN                                                            ART20650
      END                                                               ART20660
C                                                                       GTVL0010
C     ..................................................................GTVL0020
C                                                                       GTVL0030
C     SUBROUTINE GTVALU                                                 GTVL0040
C                                                                       GTVL0050
C     PURPOSE                                                           GTVL0060
C        TO ANALYZE A DIGITAL TRACE TO LOCATE THE DEPTHS OF GIVEN       GTVL0070
C        THERMAL VALUES. THE INVESTIGATION IS EXTENDED TO ALL DIGITAL   GTVL0080
C        TRACES OF A GIVEN CRUISE AND ASSUMING A LINEAR VARIATION       GTVL0090
C        BETWEEN CONSECUTIVE TRACES FOR THE SAME THERMAL VALUES, THE    GTVL0100
C        FIELD OF ISOTHERMS ALONG THE CROSS SECTION IS OBTAINED         GTVL0110
C                                                                       GTVL0120
C     USAGE                                                             GTVL0130
C        CALL GTVALU(ISO,NPT,J1,ISTA,NORDER)                            GTVL0140
C                                                                       GTVL0150
C     DESCRIPTION OF PARAMETERS                                         GTVL0160
C        ISO    -INPUT NUMBER OF ISOTHERMS                              GTVL0170
C        NPT    -INPUT NUMBER OF PAIRS (TO,ZO) DEFINING A DIGITAL TRACE GTVL0180
C        J1     -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED AT A GIVEN   GTVL0190
C                CRUISE                                                 GTVL0200
C        ISTA   -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS       GTVL0210
C                SECTION                                                GTVL0220
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                      GTVL0230
```

```
C
C       REMARKS
C          (1)EACH ISOTHERM IS CHARACTERIZED AT THREE POSSIBLE DEPTHS
C          (A) SHALLOW, (B) INTERMEDIATE AND (C) DEEP  TO TAKE INTO
C          ACCOUNT THE OCCURRENCE OF INVERSIONS. IF SUCH FEATURE DOES
C          NOT EXIST, THE SINGLE VALUE IS TAKEN AS 'DEEP' AND TO THE
C          SHALLOW AND INTERMEDIATE DEPTHS ARE ASSIGNED NEGATIVE
C          ARTIFICIAL VALUES.
C          (2)THIS SUBROUTINE USES 'COMMON' LABELLED AREAS TO RECEIVE
C          INFORMATION FROM THE MAIN PROGRAM AND TO GIVE BACK PROCESSED
C          DATA
C
C       SUBROUTINES REQUIRED
C          LINE
C
C       METHOD
C          LINEAR INTERPOLATION BETWEEN TWO CONSECUTIVE POSITIONS OF A
C          DIGITAL TRACE
C
C       ..................................................................
C

      SUBROUTINE GTVALU(ISO,NPT,J1,ISTA,NORDER)
      COMMON/WORKA/TO(60),ZO(60),ZSHAL(20,20),ZINTR(20,20),ZDEEP(20,20),
     *ZA(10),BLAN1(122)
      COMMON/WORKB/X(20),TI(19)
      COMMON/WORKE/Z11CS(20,12),Z11CI(20,12),Z11CD(20,12),Z12CS(20,12),
     *Z12CI(20,12),Z12CD(20,12),Z13CS(20,12),Z13CI(20,12),Z13CD(20,12),
     *Z14CS(20,12),Z14CI(20,12),Z14CD(20,12),Z15CS(20,12),Z15CI(20,12),
     *Z15CD(20,12),Z16CS(20,12),Z16CI(20,12),Z16CD(20,12),Z17CS(20,12),
     *Z17CI(20,12),Z17CD(20,12),Z18CS(20,12),Z18CI(20,12),Z18CD(20,12),
     *Z19CS(20,12),Z19CI(20,12),Z19CD(20,12),Z20CS(20,12),Z20CI(20,12),
     *Z20CD(20,12),Z21CS(20,12),Z21CI(20,12),Z21CD(20,12),Z22CS(20,12),
     *Z22CI(20,12),Z22CD(20,12),Z23CS(20,12),Z23CI(20,12),Z23CD(20,12),
     *Z24CS(20,12),Z24CI(20,12),Z24CD(20,12),Z25CS(20,12),Z25CI(20,12),
     *Z25CD(20,12),ZTUCS(20,12),ZTUCI(20,12),ZTUCD(20,12),ZTLCS(20,12),
     *ZTLCI(20,12),ZTLCD(20,12),ZSUCS(20,12),ZSUCI(20,12),ZSUCD(20,12),
     *ZSLCS(20,12),ZSLCI(20,12),ZSLCD(20,12),ZTTMC(19,12)
      J2=0
      DO 40 J2=1,ISO
C
C       PROCESSING A GIVEN ISOTHERMAL VALUE
C
      K1=1
      J3=0
      DO 30 J3=1,NPT
      IF(J3.EQ.NPT) GO TO 25
```

```
      IF(((TO(J3+1).LT.TI(J2)).AND.(TI(J2).LT.TO(J3))).OR.((TO(J3+1).
     *GT.TI(J2)).AND.(TI(J2).GT.TO(J3)))) GO TO 28
      IF(TO(J3).NE.TI(J2)) GO TO 30
      ZA(K1)=ZO(J3)
      K1=K1+1
      GO TO 30
   25 IF(TO(NPT).NE.TI(J2)) GO TO 30
      ZA(K1)=ZO(NPT)
      GO TO 30
C
C     LINEAR INTERPOLATION
C
   28 CALL LINE(TO(J3),ZO(J3),TO(J3+1),ZO(J3+1),TI(J2),QW)
      ZA(K1)=QW
      K1=K1+1
   30 CONTINUE
C
C     CLASSIFICATION OF THE PROCESSED DATA
C
      IF(K1.GT.1) GO TO 32
      ZSHAL(J1,J2)=-5.
      ZINTR(J1,J2)=-5.
      ZDEEP(J1,J2)=-5.
      GO TO 39
   32 IF(K1.GT.2) GO TO 33
      K1=K1-1
      ZSHAL(J1,J2)=-5.
      ZINTR(J1,J2)=-5.
      ZDEEP(J1,J2)=ZA(K1)
      GO TO 39
   33 IF(K1.GT.3) GO TO 34
      K1=K1-2
      ZSHAL(J1,J2)=ZA(K1)
      ZINTR(J1,J2)=-5.
      ZDEEP(J1,J2)=ZA(K1+1)
      GO TO 39
   34 K1=1
      ZSHAL(J1,J2)=ZA(K1)
      ZINTR(J1,J2)=ZA(K1+1)
      ZDEEP(J1,J2)=ZA(K1+2)
   39 WRITE(6,41) J1,J2,ZSHAL(J1,J2),J1,J2,ZINTR(J1,J2),J1,J2,
     *ZDEEP(J1,J2)
   41 FORMAT(' ',2X,'ZSHAL(',I2,',',I2,') =',2X,F11.7,10X,'ZINTR(',I2,
     *',',I2,') =',2X,F11.7,10X,'ZDEEP(',I2,',',I2,') =',2X,F11.7)
      IF(J2.EQ.1) GO TO 50
      IF(J2.EQ.2) GO TO 60
      IF(J2.EQ.3) GO TO 70
      IF(J2.EQ.4) GO TO 80
```

```
      IF(J2.EQ.5) GO TO 90                                              GTVL1180
      IF(J2.EQ.6) GO TO 100                                             GTVL1190
      IF(J2.EQ.7) GO TO 110                                             GTVL1200
      IF(J2.EQ.8) GO TO 120                                             GTVL1210
      IF(J2.EQ.9) GO TO 130                                             GTVL1220
      IF(J2.EQ.10) GO TO 140                                            GTVL1230
      IF(J2.EQ.11) GO TO 150                                            GTVL1240
      IF(J2.EQ.12) GO TO 160                                            GTVL1250
      IF(J2.EQ.13) GO TO 170                                            GTVL1260
      IF(J2.EQ.14) GO TO 180                                            GTVL1270
      IF(J2.EQ.15) GO TO 190                                            GTVL1280
      IF(J2.EQ.16) GO TO 200                                            GTVL1290
      IF(J2.EQ.17) GO TO 210                                            GTVL1300
      IF(J2.EQ.18) GO TO 220                                            GTVL1310
      IF(J2.EQ.19) GO TO 230                                            GTVL1320
      GO TO 40                                                          GTVL1330
   50 Z25CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1340
      Z25CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1350
      Z25CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1360
      GO TO 40                                                          GTVL1370
   60 Z24CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1380
      Z24CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1390
      Z24CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1400
      GO TO 40                                                          GTVL1410
   70 Z23CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1420
      Z23CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1430
      Z23CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1440
      GO TO 40                                                          GTVL1450
   80 Z22CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1460
      Z22CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1470
      Z22CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1480
      GO TO 40                                                          GTVL1490
   90 Z21CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1500
      Z21CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1510
      Z21CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1520
      GO TO 40                                                          GTVL1530
  100 Z20CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1540
      Z20CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1550
      Z20CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1560
      GO TO 40                                                          GTVL1570
  110 Z19CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1580
      Z19CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1590
      Z19CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1600
      GO TO 40                                                          GTVL1610
  120 Z18CS(NORDER,ISTA)=ZSHAL(J1,J2)                                   GTVL1620
      Z18CI(NORDER,ISTA)=ZINTR(J1,J2)                                   GTVL1630
      Z18CD(NORDER,ISTA)=ZDEEP(J1,J2)                                   GTVL1640
      GO TO 40                                                          GTVL1650
```

```
  130 Z17CS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1660
      Z17CI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1670
      Z17CD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1680
      GO TO 40                                                           GTVL1690
  140 Z16CS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1700
      Z16CI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1710
      Z16CD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1720
      GO TO 40                                                           GTVL1730
  150 Z15CS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1740
      Z15CI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1750
      Z15CD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1760
      GO TO 40                                                           GTVL1770
  160 Z14CS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1780
      Z14CI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1790
      Z14CD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1800
      GO TO 40                                                           GTVL1810
  170 Z13CS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1820
      Z13CI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1830
      Z13CD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1840
      GO TO 40                                                           GTVL1850
  180 Z12CS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1860
      Z12CI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1870
      Z12CD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1880
      GO TO 40                                                           GTVL1890
  190 Z11CS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1900
      Z11CI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1910
      Z11CD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1920
      GO TO 40                                                           GTVL1930
  200 ZTLCS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1940
      ZTLCI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1950
      ZTLCD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL1960
      GO TO 40                                                           GTVL1970
  210 ZTUCS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL1980
      ZTUCI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL1990
      ZTUCD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL2000
      GO TO 40                                                           GTVL2010
  220 ZSLCS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL2020
      ZSLCI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL2030
      ZSLCD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL2040
      GO TO 40                                                           GTVL2050
  230 ZSUCS(NORDER,ISTA)=ZSHAL(J1,J2)                                    GTVL2060
      ZSUCI(NORDER,ISTA)=ZINTR(J1,J2)                                    GTVL2070
      ZSUCD(NORDER,ISTA)=ZDEEP(J1,J2)                                    GTVL2080
   40 CONTINUE                                                           GTVL2090
      RETURN                                                             GTVL2100
      END                                                                GTVL2110
C                                                                        LINE0010
C     ..................................................................LINE0020
```

```
C
C     SUBROUTINE LINE
C
C     PURPOSE
C        LINEAR INTERPOLATION BETWEEN TWO GIVEN POSITIONS
C
C     USAGE
C        CALL LINE(AO1,QO1,AO2,QO2,AW,QW)
C
C     DESCRIPTION OF PARAMETERS
C        AO1    -INPUT VALUE CONTAINING THE ABCISSA OF THE FIRST OF
C                TWO GIVEN POSITIONS
C        QO1    -INPUT VALUE CONTAINING THE ORDINATE OF THE FIRST
C                OF TWO  GIVEN POSITIONS
C        AO2    -INPUT VALUE CONTAINING THE ABCISSA OF THE SECOND OF
C                TWO GIVEN POSITIONS
C        QO2    -INPUT VALUE CONTAINING THE ORDINATE OF THE FIRST
C                OF TWO  GIVEN POSITIONS
C        AW     -INPUT VALUE CONTAINING THE ABCISSA OF THE INTERPOLATED
C                POSITION
C        QW     -OUTPUT VALUE CONTAINING THE ORDINATE OF THE
C                INTERPOLATED POSITION
C
C     SUBROUTINES REQUIRED
C        NONE
C
C     METHOD
C        SYSTEM OF TWO LINEAR EQUATIONS SATISFYING THE INPUT PARAMETERS
C
C     ..................................................................
C

      SUBROUTINE LINE(AO1,QO1,AO2,QO2,AW,QW)
      DQ6=QO1-QO2
      DA6=AO1-AO2
      Q2A1=QO2*AO1
      Q1A2=QO1*AO2
      QW=DQ6*(AW/DA6)+(Q2A1-Q1A2)/DA6
      RETURN
      END
C
C     ..................................................................
C
C     SUBROUTINE LINTPS
C
C     PURPOSE
C        TO GENERATE A MESH FROM A DIGITAL TRACE FOR A GIVEN INCREMENT
```

```
C                                                                       LNTS0080
C     USAGE                                                             LNTS0090
C        CALL LINTPS(NPT,DTW,M60)                                       LNTS0100
C                                                                       LNTS0110
C     DESCRIPTION OF PARAMETERS                                         LNTS0120
C        NPT    -INPUT NUMBER OF PAIRS (TO,ZO) DEFINING A DIGITAL TRACE LNTS0130
C        DTW    -INPUT VALUE OF THE THERMAL INTERVAL OF THE MESH        LNTS0140
C        M60    -OUTPUT NUMBER OF PAIRS (T,Z) COMPUTED FROM THE DIGITAL LNTS0150
C                TRACE BY (A) A CONTINUOUS INCREMENT 'DTW' OF           LNTS0160
C                TEMPERATURE BEGINNING AT SEA SURFACE AND (B) THE       LNTS0170
C                DEPTHS OF THE ABOVE PAIRS ARE OBTAINED BY LINEAR       LNTS0180
C                INTERPOLATION BETWEEN NEAR POINTS OF THE ORIGINAL      LNTS0190
C                DIGITAL TRACE                                          LNTS0200
C                                                                       LNTS0210
C     REMARKS                                                           LNTS0220
C        THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO RECEIVE THE   LNTS0230
C        SET OF ORDERED PAIRS (ZO,TO) DEFINING THE DIGITAL TRACE FROM   LNTS0240
C        THE MAIN PROGRAM AND TO GIVE BACK THE ORDERED PAIRS OF THE     LNTS0250
C        GENERATED MESH. THE SAME 'COMMON' LABELLED AREA ENABLES AN     LNTS0260
C        EXCHANGE OF INFORMATION BETWEEN THIS SUBROUTINE AND THE        LNTS0270
C        CALLED SUBROUTINES.                                            LNTS0280
C                                                                       LNTS0290
C     SUBROUTINES AND FUNCTIONS REQUIRED                                LNTS0300
C        ABS (IBM SCIENTIC LIBRARY)                                     LNTS0310
C        LINTP1                                                         LNTS0320
C        LINTP2                                                         LNTS0330
C                                                                       LNTS0340
C     METHOD                                                            LNTS0350
C        LINEAR INTERPOLATION BETWEEN TWO CONSECUTIVE POSITIONS OF THE  LNTS0360
C        DIGITAL TRACE                                                  LNTS0370
C                                                                       LNTS0380
C     ..................................................................LNTS0390
C                                                                       LNTS0400


      SUBROUTINE LINTPS(NPT,DTW,M60)                                    LNTS0410
      COMMON/WORKA/TO(60),ZO(60),T(120),Z(120),ZAH1S(20),ZAH1I(20),     LNTS0420
     *ZAH1D(20),ZAH2S(20),ZAH2I(20),ZAH2D(20),ZAH3S(20),ZAH3I(20),      LNTS0430
     *ZAH3D(20),ZAIXLR(20),BLAN3(892)                                   LNTS0440
      TW=TO(1)                                                          LNTS0450
      ZW=ZO(1)                                                          LNTS0460
      M51=1                                                             LNTS0470
      NPT1=NPT-1                                                        LNTS0480
C                                                                       LNTS0490
C     LOOP SET UP TO DESCRIBE THE DIGITAL TRACE BY A CONTINUOUS         LNTS0500
C     INCREMENT OF THE TEMPERATURE FROM THE SEA SURFACE                 LNTS0510
C                                                                       LNTS0520
      DO 8018 I51=1,40                                                  LNTS0530
```

```
      IF(I51.NE.1) GO TO 8024
      T(M51)=TO(1)
      Z(M51)=ZO(1)
      M51=M51+1
      IF(I51.EQ.NPT1) GO TO 8017
      GO TO 8028
 8024 IF(I51.GT.NPT1) GO TO 8018
      IF(I51.EQ.NPT1) GO TO 8016
      IF((ABS(TO(I51)-TO(I51+1)).LE.0.005).AND.(ABS(TO(I51+2)-TO(I51+1))
     *.LE.0.005)) GO TO 8018
C     IF(TO(I51).EQ.TO(I51+1))
      IF(ABS(TO(I51)-TO(I51+1)).LE.0.005) GO TO 8019
      IF(TO(I51+1).GT.TO(I51).AND.TW.GT.TO(I51+1)) GO TO 8030
      IF(TO(I51).GT.TO(I51+1).AND.TW.LT.TO(I51+1)) GO TO 8035
      GO TO 8028
 8030 IF(TO(I51+2).GT.TO(I51+1)) GO TO 8018
      IF(TO(I51+2).LT.TO(I51+1).AND.TW.GT.TO(I51+1)) GO TO 8032
      GO TO 8018
 8032 TW=TW-2.*ABS(TW-TO(I51+1))
      GO TO 8018
 8035 IF(TO(I51+2).LT.TO(I51+1)) GO TO 8018
      IF(TO(I51+2).GT.TO(I51+1).AND.TW.LT.TO(I51+1)) GO TO 8036
      GO TO 8018
 8036 TW=TW+2.*ABS(TO(I51)-TW)
      GO TO 8018
C
C     GENERATION OF THE MESH BETWEEN TWO CONSECUTIVE POSITIONS OF THE
C     DIGITAL TRACE
C
 8028 CALL LINTP1(TO(I51),ZO(I51),TO(I51+1),ZO(I51+1),TO(I51+2),
     *ZO(I51+2),TW,DTW,I51,M51,TW2,M52)
      M51=M52
      TW=TW2
      GO TO 8018
 8016 IF(ABS(TO(NPT1)-TO(NPT)).LE.0.005) GO TO 8230
      IF(TO(NPT).GT.TO(NPT1)) GO TO 8232
      IF(TO(NPT).GT.TW) GO TO 8013
      GO TO 8017
 8230 IF(TW.GT.TO(NPT).OR.TW.LT.TO(NPT)) GO TO 8014
      GO TO 8017
 8232 IF(TW.GT.TO(NPT)) GO TO 8014
C
C     GENERATION OF THE MESH BETWEEN THE TWO LAST POSITIONS OF THE
C     DIGITAL TRACE
C
 8017 CALL LINTP2(TO(NPT1),ZO(NPT1),TO(NPT),ZO(NPT),TW,DTW,
     *NPT1,M51,TW3,M53)
      M51=M53
```

```
      TW=TW3                                                            LNTS1020
      GO TO 8018                                                        LNTS1030
 8014 M51=M51-1                                                         LNTS1040
      GC TO 8018                                                        LNTS1050
 8019 IF(TO(I51+1).GT.TO(I51+2).AND.TW.GT.TO(I51+1)) GO TO 500          LNTS1060
      IF(TO(I51+2).GT.TO(I51+1).AND.TW.LT.TO(I51+1)) GO TO 502          LNTS1070
      GO TO 8018                                                        LNTS1080
  500 TW=TW-2.*ABS(TW-TO(I51+1))                                        LNTS1090
      GC TO 8018                                                        LNTS1100
  502 TW=TW+2.*ABS(TW-TO(I51+1))                                        LNTS1110
      GC TO 8018                                                        LNTS1120
 8013 M51=M51-1                                                         LNTS1130
 8018 CCNTINUE                                                          LNTS1140
      M60=M51                                                           LNTS1150
      WRITE(6,30)(K,T(K),K,Z(K),K=1,M60)                                LNTS1160
   30 FORMAT(' ',10X,'T(',I3,') =',F11.7,4X,'Z(',I3,') =',F11.7)        LNTS1170
      RETURN                                                            LNTS1180
      END                                                               LNTS1190
C                                                                       LNT10010
C     ..................................................................LNT10020
C                                                                       LNT10030
C     SUBROUTINE LINTP1                                                 LNT10040
C                                                                       LNT10050
C     PURPOSE                                                           LNT10060
C        TO GENERATE A MESH BETWEEN TWO GIVEN POSITIONS OF A DIGITAL    LNT10070
C        TRACE. THIS METHOD IS NOT APPLICABLE TO THE LAST TWO LOCATIONS LNT10080
C        OF THE TRACE                                                   LNT10090
C                                                                       LNT10100
C     USAGE                                                             LNT10110
C        CALL LINPT1(AO1,QO1,AO2,QO2,AO3,QO3,AW,DAW,I81,M81,AW2,M82)    LNT10120
C                                                                       LNT10130
C     DESCRIPTION OF PARAMETERS                                         LNT10140
C        AO1    -INPUT VALUE CONTAINING THE ABCISSA (TEMPERATURE) OF    LNT10150
C                THE FIRST OF THREE GIVEN POSITICNS                     LNT10160
C        QO1    -INPUT VALUE CONTAINING THE ORDINATE (DEPTH) OF THE     LNT10170
C                FIRST OF THREE POSITIONS                               LNT10180
C        AO2    -INPUT VALUE CONTAINING THE ABCISSA OF THE SECOND OF    LNT10190
C                THREE GIVEN POSITIONS                                  LNT10200
C        QO2    -INPUT VALUE CONTAINING THE ORDINATE OF THE SECOND OF   LNT10210
C                THREE GIVEN POSITIONS                                  LNT10220
C        AO3    -INPUT VALUE CONTAINING THE ABCISSA OF THE THIRD OF     LNT10230
C                THREE GIVEN POSITIONS                                  LNT10240
C        QO3    -INPUT VALUE CONTAINING THE ORDINATE OF THE THIRD OF    LNT10250
C                THREE GIVEN POSITIONS                                  LNT10260
C        AW     -INPUT VALUE CONTAINING THE ABCISSA OF THE INITIAL      LNT10270
C                INTERPOLATED POSITION BETWEEN THE FIRST TWO OF THREE   LNT10280
C                GIVEN LOCATIONS OF THE DIGITAL TRACE                   LNT10290
C        DAW    -INPUT VALUE CONTAINING THE INCREMENT OF THE MESH       LNT10300
```

```
C        I81    -INPUT VALUE CONTAINING THE ACTUAL NUMBER OF GROUPS OF
C                THREE DIGITAL POSITIONS ALREADY PROCESSED
C        M81    -INPUT VALUE CONTAINING THE ACTUAL NUMBER OF POINTS
C                GENERATED FOR THE MESH
C        AW2    -OUTPUT CONTAINING THE LAST VALUE OF ABCISSA, LOCATED
C                BETWEEN THE SECOND AND THIRD GIVEN POSITIONS OF THE
C                DIGITAL TRACE
C        M82    -OUTPUT VALUE CONTAINING THE ACTUAL NUMBER OF POSITIONS
C                GENERATED FOR THE MESH
C
C     REMARKS
C        (1)THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO GIVE BACK
C        TO THE SUBROUTINE LINTPS THE GENERATED MESH
C        (2)THIS SUBROUTINE USES THREE POSITIONS FROM THE DIGITAL
C        TRACE; IT INTERPOLATES A MESH BETWEEN THE FIRST TWO POSITIONS;
C        THE LAST TWO POSITIONS DEFINE THE ACTUAL ABCISSA OF THE
C        INCREMENT
C
C     SUBROUTINES AND FUNCTIONS REQUIRED
C        LINE
C        TINC1
C        TINC2
C        ABS (IBM SCIENTIC LIBRARY)
C
C     METHOD
C        LINEAR INTERPOLATION BETWEEN TWO CONSECUTIVE POSITIONS OF THE
C        DIGITAL TRACE
C
C     ..................................................................
C

      SUBROUTINE LINTP1(AO1,QO1,AO2,QO2,AO3,QO3,AW,DAW,I81,M81,
     *AW2,M82)
      COMMON/WORKA/TO(60),ZO(60),T(120),Z(120),ZAH1S(20),ZAH1I(20),
     *ZAH1D(20),ZAH2S(20),ZAH2I(20),ZAH2D(20),ZAH3S(20),ZAH3I(20),
     *ZAH3D(20),ZAIXLR(20),BLAN31(892)
      IF(I81.EQ.1) GO TO 9020
      IF(AW.EQ.AO1) GO TO 9008
 9005 CALL LINE(AO1,QO1,AO2,QO2,AW,QW)
      GO TO 9010
 9008 AW=AO1
      QW=QO1
 9010 T(M81)=AW
      Z(M81)=QW
      M81=M81+1
 9020 AW2=AW
C
```

```
C       DA1-DIFFERENCE BETWEEN WORKING TEMPERATURE (AW2) AND THE NEXT      LNT10770
C           OBSERVED TEMPERATURE (AO2)                                     LNT10780
C                                                                          LNT10790
        DA1=AW2-AO2                                                        LNT10800
        IF((DA1.LT.(DAW+0.005).AND.DA1.GT.(DAW-0.005)).OR.(DA1.GT.(-DAW-   LNT10810
       *0.005).AND.DA1.LT.(-DAW+0.005))) GO TO 9085                        LNT10820
C                                                                          LNT10830
C       INCREMENT DAW OF THE ACTUAL TEMPERATURE VALUE                      LNT10840
C                                                                          LNT10850
        CALL TINC1(AO1,AO2,AW,DAW,M81,AW2)                                 LNT10860
        SA=AW2                                                             LNT10870
        AW=SA                                                              LNT10880
        IF(((AO2.GT.AO1.AND.DA1.LT.0.0).OR.(AO1.GT.AO2.AND.DA1.GT.0.0)).   LNT10890
       *AND.(ABS(DA1).GE.(DAW-0.005))) GO TO 9005                          LNT10900
C                                                                          LNT10910
C       DA2-ABSOLUTE VALUE OF DA1                                          LNT10920
C                                                                          LNT10930
        AW2=AW                                                             LNT10940
        DA2=ABS(DA1)                                                       LNT10950
C                                                                          LNT10960
C       INCREMENT DAW OF THE ACTUAL TEMPERATURE VALUE                      LNT10970
C                                                                          LNT10980
        CALL TINC2(AO2,AO3,DA2,DAW,AW2)                                    LNT10990
        GO TO 9095                                                         LNT11000
 9085   IF(AO2.EQ.AO3) GO TO 9094                                          LNT11010
        T(M81)=AO2                                                         LNT11020
        Z(M81)=QO2                                                         LNT11030
        AW2=AO2                                                            LNT11040
C                                                                          LNT11050
C       INCREMENT DAW OF THE ACTUAL TEMPERATURE VALUE                      LNT11060
C                                                                          LNT11070
        CALL TINC1(AO2,AO3,AW2,DAW,M81,AW3)                                LNT11080
        M81=M81+1                                                          LNT11090
        AW2=AW3                                                            LNT11100
        GO TO 9095                                                         LNT11110
 9094   AW2=AO3                                                            LNT11120
 9095   M82=M81                                                            LNT11130
        RETURN                                                             LNT11140
        END                                                                LNT11150
C                                                                          LNT20010
C       ..................................................................LNT20020
C                                                                          LNT20030
C       SUBROUTINE LINTP2                                                  LNT20040
C                                                                          LNT20050
C       PURPOSE                                                            LNT20060
C          TO GENERATE A MESH BETWEEN THE TWO LAST GIVEN POSITIONS OF A    LNT20070
C          DIGITAL TRACE                                                   LNT20080
C                                                                          LNT20090
```

```
C       LSAGE
C          CALL LINTP2(AO1,QO1,AO2,QO2,AW,DAW,N81,M81,AW3,M83)
C
C       DESCRIPTICN OF PARAMETERS
C          QO1    -INPUT VALUE CONTAINING THE ORDINATE OF THE FIRST OF
C                  THE TWO LAST POSITIONS
C          AO1    -INPUT VALUE CONTAINING THE ABCISSA CF THE FIRST OF
C                  THE TWO LAST POSITIONS
C          QO2    -INPUT VALUE CONTAINING THE ORDINATE OF THE LAST OF
C                  THE TWO LAST POSITIONS
C          AO2    -INPUT VALUE CONTAINING THE ABCISSA OF THE LAST OF THE
C                  TWO LAST POSITIONS
C          AW     -INPUT VALUE CONTAINING THE ABCISSA OF THE INITIAL
C                  INTERPOLATED POSITION BETWEEN THE TWO LAST LOCATIONS
C                  OF THE DIGITAL TRACE
C          DAW    -INPUT VALUE CONTAINING THE INCREMENT OF THE MESH
C          N81    -INPUT VALUE CONTAINING THE ACTUAL NUMBER OF GROUPS CF
C                  THREE DIGITAL POSITIONS ALREADY PRCCESSED
C          M81    -INPUT VALUE CONTAINING THE ACTUAL NUMBER CF POINTS
C                  GENERATED FOR THE MESH
C          AW3    -OUTPUT VALUE CONTAINING THE LAST ABCISSA LOCATED IN
C                  BETWEEN THE TWO LAST DIGITAL PCINTS
C
C       REMARKS
C          (1)THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO GIVE BACK
C          TO THE SUBROUTINE LINTPS THE GENERATED MESH
C          (2)THIS SUBROUTINE USES THE TWO LAST POSITIONS FROM THE
C          DIGITAL TRACE AND INTERPOLATES A MESH IN
C
C       SUBRCUTINES AND FUNCTIONS REQUIRED
C          LINE
C          TINC1
C          ABS (IBM SCIENTIC LIBRARY)
C
C       METHOD
C          LINEAR INTERPOLATION BETWEEN THE LAST TWO POSITIONS OF THE
C          DIGITAL TRACE
C
C       ..................................................................
C


        SUBROUTINE LINTP2(AO1,QO1,AO2,QO2,AW,DAW,N81,M81,
       *AW3,M83)
        CCMMCN/WORKA/TO(60),ZO(60),T(120),Z(120),ZAH1S(20),ZAH1I(20),
       *ZAH1D(20),ZAH2S(20),ZAH2I(20),ZAH2D(20),ZAH3S(20),ZAH3I(20),
       *ZAH3D(20),ZAIXLR(20),BLAN3(892)
        IF(N81.EQ.1) GO TO 9120
```

```
      IF(AW.EQ.A01) GO TO 9108
      IF((A01.GE.A02.AND.AW.LT.A02).OR.(A02.GE.A01.AND.AW.GT.A02)) GO TO
     *9194
 9105 CALL LINE(AC1,Q01,AC2,Q02,AW,QW)
      GO TO 9110
 9108 AW=A01
      QW=Q01
 9110 T(M81)=AW
      Z(M81)=QW
      GO TO 9121
 9120 M81=M81-1
 9121 AW3=AW
C
C     DA11-DIFFERENCE BETWEEN WORKING TEMPERATURE (AW3) AND THE NEXT
C          OBSERVED TEMPERATURE (A02)
C
      DA11=AW3-A02
      IF((DA11.LT.(DAW+0.005).AND.DA11.GT.(DAW-0.005)).OR.(DA11.GE.
     *(-DAW-0.005).AND.DA11.LT.(-DAW+0.005))) GO TO 9190
C
C     INCREMENT DAW OF THE ACTUAL TEMPERATURE VALUE
C
      CALL TINC1(A01,A02,AW3,DAW,M81,AW)
      IF((A01.GE.A02.AND.AW.LT.A02).OR.(A02.GE.A01.AND.AW.GT.A02))
     *GO TO 9195
      M81=M81+1
      IF(ABS(DA11).GE.(DAW-0.005)) GO TO 9105
      CALL LINE(A01,Q01,AC2,Q02,AW,QW)
      T(M81)=AW
      Z(M81)=QW
      GO TO 9195
 9190 M81=M81+1
      T(M81)=AC2
      Z(M81)=Q02
      GO TO 9195
 9194 M81=M81-1
 9195 M83=M81
      RETURN
      END
C
C     ...................................................................
C
C        SUBROUTINE MISR
C
C        PURPOSE
C           COMPUTE MEANS, STANDARD DEVIATIONS, SKEWNESS AND KURTOSIS,
C           CORRELATION COEFFICIENTS, REGRESSION COEFFICIENTS, AND
C           STANDARD ERRORS OF REGRESSION COEFFICIENTS WHEN THERE ARE
```

```
C           MISSING DATA POINTS.  THE USER IDENTIFIES THE MISSING DATA   MISR 100
C           BY MEANS OF A NUMERIC CODE.  THOSE VALUES HAVING THIS CODE   MISR 110
C           ARE SKIPPED IN COMPUTING THE STATISTICS.  IN THE CASE OF THEMISR 120
C           CORRELATION COEFFICIENTS, ANY PAIR OF VALUES ARE SKIPPED IF  MISR 130
C           EITHER ONE OF THEM ARE MISSING.                              MISR 140
C                                                                        MISR 150
C        USAGE                                                           MISR 160
C           CALL MISR (NO,M,X,CODE,XBAR,STD,SKEW,CURT,R,N,A,B,S,IER)     MISR 170
C                                                                        MISR 180
C        DESCRIPTION OF PARAMETERS                                       MISR 190
C           NO   - NUMBER OF OBSERVATIONS                                MISR 200
C           M    - NUMBER OF VARIABLES                                   MISR 210
C           X    - INPUT DATA MATRIX OF SIZE NO X M.                     MISR 220
C           CODE - INPUT VECTOR OF LENGTH M, WHICH CONTAINS A NUMERIC    MISR 230
C                  MISSING DATA CODE FOR EACH VARIABLE. ANY OBSERVATION MISR 240
C                  FOR A GIVEN VARIABLE HAVING A VALUE EQUAL TO THE CODEMISR 250
C                  WILL BE DROPPED FOR THE COMPUTATIONS.                 MISR 260
C           XBAR - OUTPUT VECTOR OF LENGTH M CONTAINING MEANS            MISR 270
C           STD  - OUTPUT VECTOR OF LENGTH M CONTAINING STANDARD DEVI-   MISR 280
C                  ATIONS                                                MISR 290
C           SKEW - OUTPUT VECTOR OF LENGTH M CONTAINING SKEWNESS         MISR 300
C           CURT - OUTPUT VECTOR OF LENGTH M CONTAINING KURTOSIS         MISR 310
C           R    - OUTPUT MATRIX OF PRODUCT-MOMENT CORRELATION           MISR 320
C                  COEFFICIENTS.  THIS WILL BE THE UPPER TRIANGULAR      MISR 330
C                  MATRIX ONLY, SINCE THE M X M MATRIX OF COEFFICIENTS   MISR 340
C                  IS SYMMETRIC. (STORAGE MODE 1)                        MISR 350
C           N    - OUTPUT MATRIX OF NUMBER OF PAIRS OF OBSERVATIONS USEDMISR 360
C                  IN COMPUTING THE CORRELATION COEFFICIENTS.  ONLY THE MISR 370
C                  UPPER TRIANGULAR PORTION OF THE MATRIX IS GIVEN.      MISR 380
C                  (STORAGE MODE 1)                                      MISR 390
C           A    - OUTPUT MATRIX (M BY M)  CONTAINING INTERCEPTS OF      MISR 400
C                  REGRESSION LINES (A) OF THE FORM Y=A+BX.  THE FIRST   MISR 410
C                  SUBSCRIPT OF THIS MATRIX REFERS TO THE INDEPENDENT    MISR 420
C                  VARIABLE AND THE SECOND TO THE DEPENDENT VARIABLE.    MISR 430
C                  FOR EXAMPLE, A(1,3) CONTAINS THE INTERCEPT OF THE     MISR 440
C                  REGRESSION LINE FOR TWO VARIABLES WHERE VARIABLE 1    MISR 450
C                  IS INDEPENDENT AND VARIABLE 3 IS DEPENDENT.  NOTE     MISR 460
C                  THAT MATRIX A IS STORED IN A VECTOR FORM.             MISR 470
C           B    - OUTPUT MATRIX (M BY M)  CONTAINING REGRESSION         MISR 480
C                  COEFFICIENTS (B) CORRESPONDING TO THE VALUES OF       MISR 490
C                  INTERCEPTS CONTAINED IN THE OUTPUT MATRIX A.          MISR 500
C           S    - OUTPUT MATRIX (M BY M)  CONTAINING STANDARD ERRORS    MISR 510
C                  OF REGRESSION COEFFICIENTS CORRESPONDING TO THE       MISR 520
C                  COEFFICIENTS CONTAINED IN THE OUTPUT MATRIX B.        MISR 530
C           IER  - 0, NO ERROR.                                          MISR 540
C                  1, IF NUMBER OF NON-MISSING DATA ELEMENTS FOR J-TH    MISR 550
C                     VARIABLE IS TWO OR LESS.  IN THIS CASE, STD(J),    MISR 560
C                     SKEW(J), AND CURT(J) ARE SET TO 10**75.  ALL       MISR 570
```

```
C                 VALUES OF R, A, B, AND S RELATED TO THIS VARIABLE
C                 ARE ALSO SET TO 10**75.
C              2, IF VARIANCE OF J-TH VARIABLE IS LESS THAN
C                 10**(-20).  IN THIS CASE, STD(J), SKEW(J), AND
C                 CURT(J) ARE SET TO 10**75.  ALL VALUES OF R, A,
C                 B, AND S RELATED TO THIS VARIABLE ARE ALSO SET TO
C                 10**75.
C
C        REMARKS
C           THIS SUBROUTINE CANNOT DISTINGUISH A BLANK AND A ZERO.
C           THEREFORE, IF A BLANK IS SPECIFIED AS A MISSING DATA CODE IN
C           INPUT CARDS, IT WILL BE TREATED AS 0 (ZERO).
C
C        SUBROUTINES AND FUNCTION SUBPROGRAMS REQUIRED
C           NONE
C
C        METHOD
C           LEAST SQUARES REGRESSION LINES AND PRODUCT-MOMENT CORRE-
C           LATION COEFFICIENTS ARE COMPUTED.
C
C     ..................................................................
C

      SUBROUTINE MISR (NO,M,X,CODE,XBAR,STD,SKEW,CURT,R,N,A,B,S,IER)
C
      DIMENSION X(1),CODE(1),XBAR(1),STD(1),SKEW(1),CURT(1),R(1),N(1)
      DIMENSION A(1),B(1),S(1)
C
C        COMPUTE MEANS
C
      IER=0
      L=0
      DO 20 J=1,M
      FN=0.0
      XBAR(J)=0.0
      DO 15 I=1,NO
      L=L+1
      IF(X(L)-CODE(J)) 12, 15, 12
   12 FN=FN+1.0
      XBAR(J)=XBAR(J)+X(L)
   15 CONTINUE
      IF(FN) 16, 16, 17
   16 XBAR(J)=0.0
      GO TO 20
   17 XBAR(J)=XBAR(J)/FN
   20 CONTINUE
C
```

```
C        SET-UP WORK AREAS AND TEST WHETHER DATA IS MISSING                MISR1040
C                                                                          MISR1050
      L=0                                                                  MISR1060
      DO 55 J=1,M                                                          MISR1070
      LJJ=NO*(J-1)                                                         MISR1080
      SKEW(J)=0.0                                                          MISR1090
      CURT(J)=0.0                                                          MISR1100
      KI=M*(J-1)                                                           MISR1110
      KJ=J-M                                                               MISR1120
      DO 54 I=1,J                                                          MISR1130
      KI=KI+1                                                              MISR1140
      KJ=KJ+M                                                              MISR1150
      SUMX=0.0                                                             MISR1160
      SUMY=0.0                                                             MISR1170
      TI=0.0                                                               MISR1180
      TJ=0.0                                                               MISR1190
      TII=0.0                                                              MISR1200
      TJJ=0.0                                                              MISR1210
      TIJ=0.0                                                              MISR1220
      NIJ=0                                                                MISR1230
      LI=NO*(I-1)                                                          MISR1240
      LJ=LJJ                                                               MISR1250
      L=L+1                                                                MISR1260
      DO 38 K=1,NC                                                         MISR1270
      LI=LI+1                                                              MISR1280
      LJ=LJ+1                                                              MISR1290
      IF(X(LI)-CODE(I)) 30, 38, 30                                         MISR1300
   30 IF(X(LJ)-CODE(J)) 35, 38, 35                                         MISR1310
C                                                                          MISR1320
C        BOTH DATA ARE PRESENT                                             MISR1330
C                                                                          MISR1340
   35 XX=X(LI)-XBAR(I)                                                     MISR1350
      YY=X(LJ)-XBAR(J)                                                     MISR1360
      TI=TI+XX                                                             MISR1370
      TII=TII+XX**2                                                        MISR1380
      TJ=TJ+YY                                                             MISR1390
      TJJ=TJJ+YY**2                                                        MISR1400
      TIJ=TIJ+XX*YY                                                        MISR1410
      NIJ=NIJ+1                                                            MISR1420
      SUMX=SUMX+X(LI)                                                      MISR1430
      SUMY=SUMY+X(LJ)                                                      MISR1440
      IF(I-J) 38, 37, 37                                                   MISR1450
   37 SKEW(J)=SKEW(J)+YY**3                                                MISR1460
      CURT(J)=CURT(J)+YY**4                                                MISR1470
   38 CONTINUE                                                             MISR1480
C                                                                          MISR1490
C        COMPUTE SUM OF CROSS-PRODUCTS OF DEVIATIONS                       MISR1500
C                                                                          MISR1510
```

```
      IF(NIJ) 40, 40, 39                                              MISR1520
   39 FN=NIJ                                                          MISR1530
      R(L)=TIJ-TI*TJ/FN                                               MISR1540
      N(L)=NIJ                                                        MISR1550
      TII=TII-TI*TI/FN                                                MISR1560
      TJJ=TJJ-TJ*TJ/FN                                                MISR1570
C                                                                     MISR1580
C        COMPUTE STANDARD DEVIATION, SKEWNESS, AND KURTOSIS           MISR1590
C                                                                     MISR1600
   40 IF(I-J) 47, 41, 47                                              MISR1610
   41 IF(NIJ-2) 42,42,43                                              MISR1620
   42 IER=1                                                           MISR1630
      R(L)=1.0E75                                                     MISR1640
      A(KI)=1.0E75                                                    MISR1650
      B(KI)=1.0E75                                                    MISR1660
      S(KI)=1.0E75                                                    MISR1670
      GO TO 45                                                        MISR1680
C                                                                     MISR1690
   43 STD(J)=R(L)                                                     MISR1700
      R(L)=1.0                                                        MISR1710
      A(KI)=0.0                                                       MISR1720
      B(KI)=1.0                                                       MISR1730
      S(KI)=0.0                                                       MISR1740
C                                                                     MISR1750
      IF(STD(J)-(1.0E-20)) 44,44,46                                   MISR1760
   44 IER=2                                                           MISR1770
   45 STD(J)=1.0E75                                                   MISR1780
      SKEW(J)=1.0E75                                                  MISR1790
      CURT(J)=1.0E75                                                  MISR1800
      GO TO 55                                                        MISR1810
C                                                                     MISR1820
   46 WORK=STD(J)/FN                                                  MISR1830
      SKEW(J)=(SKEW(J)/FN)/(WORK*SQRT(WORK))                          MISR1840
      CURT(J)=((CURT(J)/FN)/WORK**2)-3.0                              MISR1850
      STD(J)=SQRT(STD(J)/(FN-1.0))                                    MISR1860
      GO TO 55                                                        MISR1870
C                                                                     MISR1880
C        COMPUTE REGRESSION COEFFICIENTS                              MISR1890
C                                                                     MISR1900
   47 IF(NIJ-2) 48,48,50                                              MISR1910
   48 IER=1                                                           MISR1920
   49 R(L)=1.0E75                                                     MISR1930
      A(KI)=1.0E75                                                    MISR1940
      B(KI)=1.0E75                                                    MISR1950
      S(KI)=1.0E75                                                    MISR1960
      A(KJ)=1.0E75                                                    MISR1970
      B(KJ)=1.0E75                                                    MISR1980
      S(KJ)=1.0E75                                                    MISR1990
```

```
      GC TO 54                                                          MISR2000
C                                                                       MISR2010
   50 IF(TII-(1.0E-20)) 52,52,51                                        MISR2020
   51 IF(TJJ-(1.0E-20)) 52,52,53                                        MISR2030
   52 IER=2                                                             MISR2040
      GC TO 49                                                          MISR2050
C                                                                       MISR2060
   53 SUMX=SUMX/FN                                                      MISR2070
      SUMY=SUMY/FN                                                      MISR2080
      B(KI)=R(L)/TII                                                    MISR2090
      A(KI)=SUMY-B(KI)*SUMX                                             MISR2100
      B(KJ)=R(L)/TJJ                                                    MISR2110
      A(KJ)=SUMX-B(KJ)*SUMY                                             MISR2120
C                                                                       MISR2130
C        COMPUTE CORRELATICN COEFFICIENTS                               MISR2140
C                                                                       MISR2150
      R(L)=R(L)/(SQRT(TII)*SQRT(TJJ))                                   MISR2160
C                                                                       MISR2170
C        COMPUTE STANDARD ERRORS OF REGRESSION COEFFICIENTS             MISR2180
C                                                                       MISR2190
      RR=R(L)**2                                                        MISR2200
      SUMX=(TJJ-TJJ*RR)/(FN-2)                                          MISR2210
      S(KI)=SQRT(SUMX/TII)                                              MISR2220
      SUMY=(TII-TII*RR)/(FN-2)                                          MISR2230
      S(KJ)=SQRT(SUMY/TJJ)                                              MISR2240
C                                                                       MISR2250
   54 CCNTINUE                                                          MISR2260
   55 CCNTINUE                                                          MISR2270
C                                                                       MISR2280
      RETURN                                                            MISR2290
      END                                                               MISR2300
C                                                                       PLTT0010
C     SUBROUTINE PLOTIT                                                 PLTT0020
C                                                                       PLTT0030
C     PURPOSE                                                           PLTT0040
C        TO PLOT THE LINEAR FIELD OF ISOTHERMS GENERATED BY             PLTT0050
C        SUBROUTINE GTVALU (FOR A GIVEN CRUISE)                         PLTT0060
C                                                                       PLTT0070
C     USAGE                                                             PLTT0080
C        CALL PLCTIT(ISO,NBT,LABEL,ITITL2)                              PLTT0090
C                                                                       PLTT0100
C     DESCRIPTICN OF PARAMETERS                                         PLTT0110
C        ISO    -INPUT NUMBER OF ISOTHERMS                              PLTT0120
C        NBT    -INPUT NUMBER OF STATIONS ALONG THE CROSS SECTION OR    PLTT0130
C                THE TIME AXIS                                          PLTT0140
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLCTTED CURVE           PLTT0150
C        ITITL2-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR    PLTT0160
C                PLCT OF ISOTHERMS ALONG A CROSS SECTION OR TIME AXIS   PLTT0170
```

```
C                                                                           PLTT0180
C     REMARKS                                                               PLTT0190
C        (1)THIS SUBROUTINE RECEIVES THE REQUIRED INFORMATION FROM THE      PLTT0200
C        MAIN PROGRAM BY MEANS OF 'COMMON' LABELLED AREAS                   PLTT0210
C        (2)SUBROUTINE GTVALU COMPUTES (FOR EACH STATION OF A GIVEN         PLTT0220
C        CROSS SECTION) THE DEPTH OF SPECIFIED ISOTHERMS. THE PRESENT       PLTT0230
C        SUBROUTINE (PLOTIT) PLOTS THE FIELD OF ISOTHERMS ALONG A GIVEN     PLTT0240
C        CROSS SECTION IN THE FOLLOWING WAY: CONNECTING BY STRAIGHT         PLTT0250
C        LINES THE POINTS WITH THE SAME ISOTHERMAL VALUE LOCATED AT         PLTT0260
C        SUCCESSIVE GEOGRAPHICAL POSITIONS                                  PLTT0270
C                                                                           PLTT0280
C     SUBROUTINES REQUIRED                                                  PLTT0290
C        DRAW (IBM SCIENTIFIC LIBRARY)                                      PLTT0300
C                                                                           PLTT0310
C     ..................................................................... PLTT0320
C                                                                           PLTT0330
      SUBROUTINE PLOTIT(ISC,NBT,LABEL,ITITL2)                               PLTT0340
      REAL*8 ITITL2(12)                                                     PLTT0350
      COMMON/WORKA/TO(60),ZO(60),ZSHAL(20,20),ZINTR(20,20),ZDEEP(20,20),    PLTT0360
     *XU(20),XV(20),XW(20),ZU(20),ZV(20),ZW(20),XX(6),ZZ(6)                 PLTT0370
      COMMON/WORKB/X(20),TI(19)                                             PLTT0380
      L31=1                                                                 PLTT0390
      I9=0                                                                  PLTT0400
      DO 400 I9=1,ISO                                                       PLTT0410
C                                                                           PLTT0420
C     PROCESSING A GIVEN THERMAL VALUE                                      PLTT0430
C                                                                           PLTT0440
      L13=1                                                                 PLTT0450
      K9=0                                                                  PLTT0460
C                                                                           PLTT0470
C     PLOT OF THE DEEP ISOTHERM                                             PLTT0480
C                                                                           PLTT0490
      DO 100 K9=1,NBT                                                       PLTT0500
      IF(ZDEEP(K9,I9).LT.0..OR.ZDEEP(K9,I9).GT.457.) GO TO 70               PLTT0510
      ZU(L13)=-(ZDEEP(K9,I9)/230.+1.)                                       PLTT0520
      XU(L13)=X(K9)                                                         PLTT0530
      GO TO 72                                                              PLTT0540
   70 L12=L13-1                                                             PLTT0550
      IF(L12.GE.2) GO TO 80                                                 PLTT0560
      GO TO 90                                                              PLTT0570
   72 L13=L13+1                                                             PLTT0580
      IF(K9.NE.NBT) GO TO 100                                               PLTT0590
      L12=L13-1                                                             PLTT0600
   80 IF(L12.EQ.1) GO TO 90                                                 PLTT0610
      IF(L31.GE.2) L31=2                                                    PLTT0620
      CALL DRAW(L12,XU,ZU,L31,0,LABEL,ITITL2,1.0,1.0,5,0,2,2,8,5,0,LAST)PLTT0630
```

```
      L31=L31+1                                                         PLTT0640
      WRITE(6,1000) LAST                                                PLTT0650
      WRITE(6,82)(I9,L81,L12,ZU(L81),XU(L81),L81=1,L12)                 PLTT0660
   82 FORMAT(' ',2X,'ISOLINE (',I2,')',2X,'DEFINED BY',2X,I2,2X,'OF',   PLTT0670
     *2X,I2,2X,'POINTS AT Z=',F14.7,2X,'AT X=',F10.7)                  PLTT0680
   90 L13=1                                                             PLTT0690
  100 CONTINUE                                                          PLTT0700
      K10=0                                                             PLTT0710
      L23=1                                                             PLTT0720
C                                                                       PLTT0730
C     PLOT OF THE SHALLOW ISOTHERM                                      PLTT0740
C                                                                       PLTT0750
      DO 200 K10=1,NBT                                                  PLTT0760
      IF(ZSHAL(K10,I9).LT.0..OR.ZSHAL(K10,I9).GT.457.) GO TO 170        PLTT0770
      ZV(L23)=-(ZSHAL(K10,I9)/230.+1.)                                  PLTT0780
      XV(L23)=X(K10)                                                    PLTT0790
      GO TO 172                                                         PLTT0800
  170 L22=L23-1                                                         PLTT0810
      IF(L22.GE.2) GO TO 180                                            PLTT0820
      GO TO 190                                                         PLTT0830
  172 L23=L23+1                                                         PLTT0840
      IF(K10.NE.NBT) GO TO 200                                          PLTT0850
      L22=L23-1                                                         PLTT0860
  180 IF(L22.EQ.1) GO TO 190                                            PLTT0870
      IF(L31.GE.2) L31=2                                                PLTT0880
      CALL DRAW(L22,XV,ZV,L31,2,LABEL,ITITL2,1.0,1.0,5,0,2,2,8,5,0,LAST)PLTT0890
      L31=L31+1                                                         PLTT0900
      WRITE(6,1000) LAST                                                PLTT0910
      L21=0                                                             PLTT0920
      WRITE(6,102)(I9,L21,L22,ZV(L21),XV(L21),L21=1,L22)                PLTT0930
  102 FORMAT(' ',2X,'ISOVALUES (',I2,')',2X,'DEFINED BY',2X,I2,2X,'OF', PLTT0940
     *2X,I2,2X,'POINTS AT Z=',F14.7,2X,'AT X=',F10.7)                  PLTT0950
  190 L23=1                                                             PLTT0960
  200 CONTINUE                                                          PLTT0970
      K11=0                                                             PLTT0980
      L33=1                                                             PLTT0990
C                                                                       PLTT1000
C     PLOT OF THE INTERMEDIATE ISOTHERM                                 PLTT1010
C                                                                       PLTT1020
      DO 300 K11=1,NBT                                                  PLTT1030
      IF(ZINTR(K11,I9).LT.0..OR.ZINTR(K11,I9).GT.457.) GO TO 270        PLTT1040
      ZW(L33)=-(ZINTR(K11,I9)/230.+1.)                                  PLTT1050
      XW(L33)=X(K11)                                                    PLTT1060
      GO TO 272                                                         PLTT1070
  270 L32=L33-1                                                         PLTT1080
      IF(L32.GE.2) GO TO 280                                            PLTT1090
      GO TO 290                                                         PLTT1100
  272 L33=L33+1                                                         PLTT1110
```

```
      IF(K11.NE.NBT) GO TO 300                                          PLTT1120
      L32=L33-1                                                         PLTT1130
  280 IF(L32.EQ.1) GO TO 290                                            PLTT1140
      IF(L31.GE.2) L31=2                                                PLTT1150
      CALL DRAW(L32,XW,ZW,L31,5,LABEL,ITITL2,1.0,1.0,5,0,2,2,8,5,0,LAST)PLTT1160
      L31=L31+1                                                         PLTT1170
      WRITE(6,1000) LAST                                                PLTT1180
      L61=0                                                             PLTT1190
      WRITE(6,202)(I9,L61,L32,ZW(L61),XW(L61),L61=1,L32)               PLTT1200
  202 FORMAT(' ',2X,'ISOVALUES (',I2,')',2X,'DEFINED BY',2X,I2,2X,'OF', PLTT1210
     *2X,I2,2X,'POINTS AT Z=',F14.7,2X,'AT X=',F10.7)                   PLTT1220
  290 L33=1                                                             PLTT1230
  300 CONTINUE                                                          PLTT1240
  400 CONTINUE                                                          PLTT1250
C                                                                       PLTT1260
C     ARTIFICIAL COMPLETION OF THE PLOTS                                PLTT1270
C                                                                       PLTT1280
      ZZ(1)=-1.                                                         PLTT1290
      ZZ(2)=-(100./230.+1.)                                             PLTT1300
      ZZ(3)=-(200./230.+1.)                                             PLTT1310
      ZZ(4)=-(300./230.+1.)                                             PLTT1320
      ZZ(5)=-(400./230.+1.)                                             PLTT1330
      ZZ(6)=-(500./230.+1.)                                             PLTT1340
      XX(1)=0.                                                          PLTT1350
      XX(2)=0.                                                          PLTT1360
      XX(3)=0.                                                          PLTT1370
      XX(4)=0.                                                          PLTT1380
      XX(5)=0.                                                          PLTT1390
      XX(6)=0.                                                          PLTT1400
      CALL DRAW(6,XX,ZZ,3,4,LABEL,ITITL2,1.0,1.0,5,0,2,2,8,5,0,LAST)    PLTT1410
      WRITE(6,1000) LAST                                                PLTT1420
 1000 FORMAT('0',30X,'LAST =',I1)                                       PLTT1430
      RETURN                                                            PLTT1440
      END                                                               PLTT1450
C                                                                       PRNT0010
C     ..................................................................PRNT0020
C                                                                       PRNT0030
C     SUBROUTINE PRINT                                                  PRNT0040
C                                                                       PRNT0050
C     PURPOSE                                                           PRNT0060
C        TO PRINT THE TITLE OF THE OUTPUT OF THE STATISTICAL STUDY      PRNT0070
C                                                                       PRNT0080
C     USAGE                                                             PRNT0090
C        SUBROUTINE PRINT(IBEGIN,IEND,ISTA,ICSEC,IERIOD)                PRNT0100
C                                                                       PRNT0110
C     DESCRIPTION OF PARAMETERS                                         PRNT0120
C        IBEGIN-INPUT VALUE CONTAINING THE MONTH,DAY AND YEAR OF THE    PRNT0130
C               BEGINNING OF THE STUDY                                  PRNT0140
```

```
C         IEND  -INPUT VALUE CONTAINING THE MONTH,DAY AND YEAR OF THE     PRNT0150
C                END OF THE STUDY                                          PRNT0160
C         ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS          PRNT0170
C                SECTION                                                   PRNT0180
C         ICSEC -INPUT NUMBER OF IDENTIFICATION CF THE CROSS SECTION       PRNT0190
C         IERIOD-INPUT VALUE CONTAINING THE NUMBER CF DAYS CONCERNING      PRNT0200
C                THE DURATION OF THE STUDY                                 PRNT0210
C                                                                          PRNT0220
C     SUBROUTINES REQUIRED                                                 PRNT0230
C        NONE                                                              PRNT0240
C                                                                          PRNT0250
C     ..................................................................PRNT0260
C                                                                          PRNT0270
C
      SUBROUTINE PRINT(IBEGIN,IEND,ISTA,ICSEC,IERIOD)                      PRNT0280
      L1=IBEGIN/100                                                        PRNT0290
      M1=IEND/100                                                          PRNT0300
      IYEARB=IBEGIN-L1*100                                                 PRNT0310
      IYEARE=IEND-M1*100                                                   PRNT0320
      ICAYB=L1/100                                                         PRNT0330
      ICAYE=M1/100                                                         PRNT0340
      L2=IDAYB*100                                                         PRNT0350
      M2=IDAYE*100                                                         PRNT0360
      IMONTB=L1-L2                                                         PRNT0370
      IMONTE=M1-M2                                                         PRNT0380
      WRITE(6,10) ICSEC,ISTA,IDAYB,IMONTB,IYEARB,IDAYE,IMONTE,IYEARE,     PRNT0390
     *IERIOD                                                               PRNT0400
   10 FORMAT('1',////////,1X,'CROSS SECTION NO. ',I1,27X,'STATICN NO. ',PRNT0410
     *I2,/,1X,'BEGIN:',I2,'/',I2,'/',I2,9X,'END:',I2,'/',I2,'/',I2,9X,    PRNT0420
     *'PERIOD: ',I3,1X,'DAYS',/,1X,'SOURCE OF DATA: FNWC (MONTEREY-USA)'PRNT0430
     *,////,6X,'LINEAR STATISTICAL STUDY BETWEEN THE VARIABLES  :',/)      PRNT0440
      RETURN                                                               PRNT0450
      END                                                                  PRNT0460
C                                                                          PTBT0010
C     ..................................................................PTBT0020
C                                                                          PTBT0030
C     SUBROUTINE PLOTBT                                                    PTBT0040
C                                                                          PTBT0050
C     PURPOSE                                                              PTBT0060
C        TO PLOT THE TRACES GENERATED BY SUBROUTINE LINTPS FOR A GIVEN     PTBT0070
C        CRUISE                                                            PTBT0080
C                                                                          PTBT0090
C     USAGE                                                                PTBT0100
C        CALL PLOTBT(N,L50,NBT,ITITL1,LABEL)                               PTBT0110
C                                                                          PTBT0120
C     DESCRIPTICN OF PARAMETERS                                            PTBT0130
C        N      -OUTPUT NUMBER OF PAIRS (T,Z) COMPUTED FROM THE DIGITAL    PTBT0140
```

```
C               TRACE BY (A) A CONTINUOUS INCREMENT 'DTW' OF                PTBT0150
C               TEMPERATURE BEGINNING AT SEA SURFACE AND (B) THE            PTBT0160
C               DEPTHS OF THE ABOVE PAIRS ARE OBTAINED BY LINEAR            PTBT0170
C               INTERPOLATION BETWEEN NEAR POINTS CF THE ORIGINAL           PTBT0180
C               DIGITAL TRACE                                               PTBT0190
C        L50    -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED FOR A GIVEN      PTBT0200
C               CRUISE                                                      PTBT0210
C        NBT    -INPUT NUMBER OF STATIONS ALONG THE CROSS SECTION OR        PTBT0220
C               THE TIME AXIS                                               PTBT0230
C        ITITL1-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR        PTBT0240
C               PLOT OF DIGITAL TRACES ALONG A CROSS SECTION OR TIME        PTBT0250
C               AXIS                                                        PTBT0260
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLCTTED CURVE               PTBT0270
C                                                                           PTBT0280
C     REMARKS                                                               PTBT0290
C        (1)THIS SUBROUTINE RECEIVES THE REQUIRED INFORMATION FROM THE      PTBT0300
C        MAIN PROGRAM BY MEANS OF A 'COMMON' LABELLED AREA                  PTBT0310
C        (2)THIS SUBROUTINE CONDENSES THREE-DIMENSIONAL INFORMATION         PTBT0320
C        (T,Z,X) INTO TWO-DIMENSIONAL (T/X,Z) BY USING THE HORIZONTAL       PTBT0330
C        AXIS SIMULTANEOUSLY AS TEMPERATURE AND CRUISE LOCATION AXIS        PTBT0340
C                                                                           PTBT0350
C     SUBRCUTINES REQUIRED                                                  PTBT0360
C        DRAW (IBM SCIENTIFIC SUBROUTINE LIBRARY)                           PTBT0370
C                                                                           PTBT0380
C     ..............................................................        PTBT0390
C                                                                           PTBT0400

      SUBROUTINE PLOTBT(N,L50,NBT,ITITL1,LABEL)                             PTBT0410
      REAL*8 ITITL1(12)                                                     PTBT0420
      CCMMON/WCRKA/TO(60),ZO(60),T(120),Z(120),ZAH1S(20),ZAH1I(20),         PTBT0430
     *ZAH1D(20),ZAH2S(20),ZAH2I(20),ZAH2D(20),ZAH3S(20),ZAH3I(20),          PTBT0440
     *ZAH3D(20),ZAIXLR(20),XA(200),ZA(200),BLAN5(492)                       PTBT0450
      CCMMCN/WORKB/X(20),TI(19)                                             PTBT0460
      IF(N.EQ.0) GO TO 50                                                   PTBT0470
C                                                                           PTBT0480
C     PROCESSING THE REQUIRED INFORMATION                                   PTBT0490
C                                                                           PTBT0500
      XA(1)=X(L50)                                                          PTBT0510
      DXA=X(L50)-T(1)/15.                                                   PTBT0520
      ZA(1)=-(Z(1)/230.+1.)                                                 PTBT0530
      DC 1 I49=2,N                                                          PTBT0540
      XA(I49)=T(I49)/15.+DXA                                                PTBT0550
      ZA(I49)=-(Z(I49)/230.+1.)                                             PTBT0560
    1 CONTINUE                                                              PTBT0570
      GO TO 2                                                               PTBT0580
   50 N=2                                                                   PTBT0590
      XA(1)=0.                                                              PTBT0600
```

```
      XA(2)=0.1                                                          PTBT0610
      ZA(1)=-5.                                                          PTBT0620
      ZA(2)=-5.                                                          PTBT0630
    2 IF(L50.NE.1.OR.L50.NE.NBT) KCURV=2                                 PTBT0640
      IF(L50.EQ.1) KCURV=1                                               PTBT0650
      IF(L50.EQ.NBT) KCURV=3                                             PTBT0660
    3 CALL DRAW(N,XA,ZA,KCURV,0,LABEL,ITITL1,1.,1.,5,0,2,2,8,5,0,LAST)   PTBT0670
      WRITE(6,4) LAST                                                    PTBT0680
    4 FORMAT('0',30X,'LAST =',I1)                                        PTBT0690
      RETURN                                                             PTBT0700
      END                                                                PTBT0710
C                                                                        PTFT0010
C     ...................................................................PTFT0020
C                                                                        PTFT0030
C     SUBROUTINE PLOTFT                                                  PTFT0040
C                                                                        PTFT0050
C     PURPOSE                                                            PTFT0060
C        TO PLOT THE LINEAR FIELD OF ISOPLETHS DEFINING THE TOP, THE     PTFT0070
C        MEAN DEPTH AND THE BOTTOM OF THE THERMOCLINES IDENTIFIED BY     PTFT0080
C        SUBROUTINE THERMC FOR A GIVEN CRUISE                            PTFT0090
C                                                                        PTFT0100
C     USAGE                                                              PTFT0110
C        CALL PLOTFT(NBT,LABEL,ITITL3)                                   PTFT0120
C                                                                        PTFT0130
C     DESCRIPTION OF PARAMETERS                                          PTFT0140
C        NBT    -INPUT NUMBER OF STATIONS ALONG THE CROSS SECTION OR     PTFT0150
C                THE TIME AXIS                                           PTFT0160
C        ITITL3-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR     PTFT0170
C                PLOT OF ISOPLETHS OF THE THERMOCLINE ANALYSIS ALONG A   PTFT0180
C                CROSS SECTION OR TIME AXIS                              PTFT0190
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE            PTFT0200
C                                                                        PTFT0210
C     REMARKS                                                            PTFT0220
C        (1)THIS SUBROUTINE RECEIVES THE REQUIRED INFORMATION FROM THE   PTFT0230
C        MAIN PROGRAM BY MEANS OF 'COMMON' LABELLED AREAS.               PTFT0240
C        (2)A CRUISE CONTAINS SEVERAL DIGITAL TRACES DISTRIBUTED AT      PTFT0250
C        KNOWN GEOGRAPHICAL POSITIONS. THE SUBROUTINE THERMC             PTFT0260
C        IDENTIFIES THE THERMOCLINES IN EACH DIGITAL TRACE AND           PTFT0270
C        ORDERED THEM FROM THE SEA SURFACE. SUBROUTINE PLOTFT PLOTS THE  PTFT0280
C        FIELD OF ISOPLETHS OF THERMOCLINES ALONG A GIVEN CROSS SECTION  PTFT0290
C        IN THE FOLLOWING WAY: CONNECTING BY STRAIGHT LINES THE POINTS   PTFT0300
C        LOCATED AT SUCCESSIVE GEOGRAPHICAL POSITIONS WHICH SHOW THE     PTFT0310
C        SAME FEATURE (TOP, BOTTOM OR MEAN) AND ORDER OF THERMOCLINE.    PTFT0320
C                                                                        PTFT0330
C     SUBROUTINES REQUIRED                                               PTFT0340
C        DRAW (IBM SCIENTIFIC LIBRARY)                                   PTFT0350
C                                                                        PTFT0360
C     ...................................................................PTFT0370
```

```
C                                                                        PTFT0380
      SUBROUTINE PLOTFT(NBT,LABEL,ITITL3)                                PTFT0390
      REAL*8 ITITL3(12)                                                  PTFT0400
      COMMON/WORKA/TO(60),ZO(60),T(120),Z(120),ZAH1S(20),ZAH1I(20),      PTFT0410
     *ZAH1D(20),ZAH2S(20),ZAH2I(20),ZAH2D(20),ZAH3S(20),ZAH3I(20),       PTFT0420
     *ZAH3D(20),ZAIXLR(20),XA(20),ZA(20),XB(20),ZB(20),XC(20),ZC(20),    PTFT0430
     *XD(20),ZD(20),XE(20),ZE(20),XF(20),ZF(20),XG(20),ZG(20),XU(20),    PTFT0440
     *ZU(20),XV(20),ZV(20),XW(20),ZW(20),XX(6),ZZ(6),BLAN8(480)          PTFT0450
      COMMON/WORKB/X(20),TI(19)                                          PTFT0460
      L31=1                                                              PTFT0470
      L13=1                                                              PTFT0480
      K9=0                                                               PTFT0490
C                                                                        PTFT0500
C     PROCESSING THE ISOPLETH DEFINING THE TOP OF DEEP THERMOCLINE       PTFT0510
C                                                                        PTFT0520
      DO 100 K9=1,NBT                                                    PTFT0530
      IF(ZAH1D(K9).LT.0..OR.ZAH1D(K9).GT.457.) GO TO 70                  PTFT0540
      ZU(L13)=-(ZAH1D(K9)/230.+1.)                                       PTFT0550
      XU(L13)=X(K9)                                                      PTFT0560
      GO TO 72                                                           PTFT0570
   70 L12=L13-1                                                          PTFT0580
      IF(L12.GE.2) GO TO 80                                              PTFT0590
      GO TO 90                                                           PTFT0600
   72 L13=L13+1                                                          PTFT0610
      IF(K9.NE.NBT) GO TO 100                                            PTFT0620
      L12=L13-1                                                          PTFT0630
   80 IF(L12.EQ.1) GO TO 90                                              PTFT0640
      IF(L31.GE.2) L31=2                                                 PTFT0650
      CALL DRAW(L12,XU,ZU,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)PTFT0660
      L31=L31+1                                                          PTFT0670
      WRITE(6,1000) LAST                                                 PTFT0680
      WRITE(6,82)(L81,L12,ZU(L81),XU(L81),L81=1,L12)                     PTFT0690
   82 FORMAT(' ',2X,'TOP OF DEEP THERMOCLINE DEFINED BY',2X,I2           PTFT0700
     *,2X,'OF',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)             PTFT0710
   90 L13=1                                                              PTFT0720
  100 CONTINUE                                                           PTFT0730
      K10=0                                                              PTFT0740
      L23=1                                                              PTFT0750
C                                                                        PTFT0760
C     PROCESSING THE ISOPLETH DEFINING THE BOTTOM OF DEEP THERMOCLINE    PTFT0770
C                                                                        PTFT0780
      DO 200 K10=1,NBT                                                   PTFT0790
      IF(ZAH3D(K10).LT.0..OR.ZAH3D(K10).GT.457.) GO TO 170               PTFT0800
      ZV(L23)=-(ZAH3D(K10)/230.+1.)                                      PTFT0810
      XV(L23)=X(K10)                                                     PTFT0820
      GO TO 172                                                          PTFT0830
```

```
  170 L22=L23-1
      IF(L22.GE.2) GO TO 180
      GO TO 190
  172 L23=L23+1
      IF(K10.NE.NBT) GO TO 200
      L22=L23-1
  180 IF(L22.EQ.1) GO TO 190
      IF(L31.GE.2) L31=2
      CALL DRAW(L22,XV,ZV,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)
      L31=L31+1
      WRITE(6,1000) LAST
      L21=0
      WRITE(6,102)(L21,L22,ZV(L21),XV(L21),L21=1,L22)
  102 FORMAT(' ',2X,'BOTTOM OF DEEP THERMOCLINE DEFINED BY',2X
     *,I2,2X,'OF',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)
  190 L23=1
  200 CONTINUE
      K11=0
      L33=1
C
C     PROCESSING THE ISOPLETH DEFINING THE AVERAGE DEPTH OF DEEP
C     THERMOCLINE
C
      DO 300 K11=1,NBT
      IF(ZAH2D(K11).LT.0..OR.ZAH2D(K11).GT.457.) GO TO 270
      ZW(L33)=-(ZAH2D(K11)/230.+1.)
      XW(L33)=X(K11)
      GO TO 272
  270 L32=L33-1
      IF(L32.GE.2) GO TO 280
      GO TO 290
  272 L33=L33+1
      IF(K11.NE.NBT) GO TO 300
      L32=L33-1
  280 IF(L32.EQ.1) GO TO 290
      IF(L31.GE.2) L31=2
      CALL DRAW(L32,XW,ZW,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)
      L31=L31+1
      WRITE(6,1000) LAST
      L61=0
      WRITE(6,202)(L61,L32,ZW(L61),XW(L61),L61=1,L32)
  202 FORMAT(' ',2X,'MEAN DEPTH OF DEEP THERMOCLINE DEFINED BY
     *',2X,I2,2X,'OF',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)
  290 L33=1
  300 CONTINUE
      L43=1
      K12=0
C
```

```
C     PROCESSING THE ISOPLETH DEFINING THE TOP OF SHALLOW THERMOCLINE   PTFT1320
C                                                                       PTFT1330
      DO 400 K12=1,NBT                                                  PTFT1340
      IF(ZAH1S(K12).LT.0..OR.ZAH1S(K12).GT.457.) GO TO 370              PTFT1350
      ZA(L43)=-(ZAH1S(K12)/230.+1.)                                     PTFT1360
      XA(L43)=X(K12)                                                    PTFT1370
      GO TO 372                                                         PTFT1380
  370 L42=L43-1                                                         PTFT1390
      IF(L42.GE.2) GO TO 380                                            PTFT1400
      GO TO 390                                                         PTFT1410
  372 L43=L43+1                                                         PTFT1420
      IF(K12.NE.NBT) GO TO 400                                          PTFT1430
      L42=L43-1                                                         PTFT1440
  380 IF(L42.EQ.1) GO TO 390                                            PTFT1450
      IF(L31.GE.2) L31=2                                                PTFT1460
      CALL DRAW(L42,XA,ZA,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)PTFT1470
      L31=L31+1                                                         PTFT1480
      WRITE(6,1000) LAST                                                PTFT1490
      WRITE(6,382)(L81,L42,ZA(L81),XA(L81),L81=1,L42)                   PTFT1500
  382 FORMAT(' ',2X,'TOP OF SHALLOW THERMOCLINE DEFINED BY',2X,I2,2X,'OFPTFT1510
     *',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)                   PTFT1520
  390 L43=1                                                             PTFT1530
  400 CONTINUE                                                          PTFT1540
      L53=1                                                             PTFT1550
      K13=0                                                             PTFT1560
C                                                                       PTFT1570
C     PROCESSING THE ISOPLETH DEFINING THE BOTTOM OF SHALLOW            PTFT1580
C     THERMOCLINE                                                       PTFT1590
C                                                                       PTFT1600
      DO 500 K13=1,NBT                                                  PTFT1610
      IF(ZAH3S(K13).LT.0..OR.ZAH3S(K13).GT.457.) GO TO 470              PTFT1620
      ZB(L53)=-(ZAH3S(K13)/230.+1.)                                     PTFT1630
      XB(L53)=X(K13)                                                    PTFT1640
      GO TO 472                                                         PTFT1650
  470 L52=L53-1                                                         PTFT1660
      IF(L52.GE.2) GO TO 480                                            PTFT1670
      GO TO 490                                                         PTFT1680
  472 L53=L53+1                                                         PTFT1690
      IF(K13.NE.NBT) GO TO 500                                          PTFT1700
      L52=L53-1                                                         PTFT1710
  480 IF(L52.EQ.1) GO TO 490                                            PTFT1720
      IF(L31.GE.2) L31=2                                                PTFT1730
      CALL DRAW(L52,XB,ZB,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)PTFT1740
      L31=L31+1                                                         PTFT1750
      WRITE(6,1000) LAST                                                PTFT1760
      WRITE(6,482)(L21,L52,ZB(L21),XB(L21),L21=1,L52)                   PTFT1770
  482 FORMAT(' ',2X,'BOTTOM OF SHALLOW THERMOCLINE DEFINED BY',2X,I2,2X,PTFT1780
     *'OF',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)                PTFT1790
```

```
  490 L53=1                                                             PTFT1800
  500 CONTINUE                                                          PTFT1810
      L63=1                                                             PTFT1820
      K14=0                                                             PTFT1830
C                                                                       PTFT1840
C     PROCESSING THE ISOPLETH DEFINING THE MEAN DEPTH OF SHALLOW        PTFT1850
C     THERMOCLINE                                                       PTFT1860
C                                                                       PTFT1870
      DO 600 K14=1,NBT                                                  PTFT1880
      IF(ZAH2S(K14).LT.0..OR.ZAH2S(K13).GT.457.) GO TO 570              PTFT1890
      ZC(L63)=-(ZAH2S(K14)/230.+1.)                                     PTFT1900
      XC(L63)=X(K14)                                                    PTFT1910
      GO TO 572                                                         PTFT1920
  570 L62=L63-1                                                         PTFT1930
      IF(L62.GE.2) GO TO 580                                            PTFT1940
      GO TO 590                                                         PTFT1950
  572 L63=L63+1                                                         PTFT1960
      IF(K14.NE.NBT) GO TO 600                                          PTFT1970
      L62=L63-1                                                         PTFT1980
  580 IF(L62.EQ.1) GO TO 590                                            PTFT1990
      IF(L31.GE.2) L31=2                                                PTFT2000
      CALL DRAW(L62,XC,ZC,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)PTFT2010
      L31=L31+1                                                         PTFT2020
      WRITE(6,1000) LAST                                                PTFT2030
      WRITE(6,582)(L61,L62,ZC(L61),XC(L61),L61=1,L62)                   PTFT2040
  582 FORMAT(' ',2X,'MEAN DEPTH OF SHALLOW THERMOCLINE DEFINED BY',2X,I2PTFT2050
     *,2X,'OF',2X,I2,2X,'POINTS AT Z=',F5,1,2X,'AT X=',F5.1)            PTFT2060
  590 L63=1                                                             PTFT2070
  600 CONTINUE                                                          PTFT2080
      L73=1                                                             PTFT2090
      K15=0                                                             PTFT2100
C                                                                       PTFT2110
C     PROCESSING THE ISOPLETH DEFINING THE TOP OF INTERMEDIATE          PTFT2120
C     THERMOCLINE                                                       PTFT2130
C                                                                       PTFT2140
      DO 700 K15=1,NBT                                                  PTFT2150
      IF(ZAH1I(K15).LT.0..OR.ZAH1I(K15).GT.457.) GO TO 670              PTFT2160
      ZD(L73)=-(ZAH1I(K15)/230.+1.)                                     PTFT2170
      XD(L73)=X(K15)                                                    PTFT2180
      GO TO 672                                                         PTFT2190
  670 L72=L73-1                                                         PTFT2200
      IF(L72.GE.2) GO TO 680                                            PTFT2210
      GO TO 690                                                         PTFT2220
  672 L73=L73+1                                                         PTFT2230
      IF(K15.NE.NBT) GO TO 700                                          PTFT2240
      L72=L73-1                                                         PTFT2250
  680 IF(L72.EQ.1) GO TO 690                                            PTFT2260
      IF(L31.GE.2) L31=2                                                PTFT2270
```

```
      CALL DRAW(L72,XD,ZD,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)PTFT2280
      L31=L31+1                                                         PTFT2290
      WRITE(6,1000) LAST                                                PTFT2300
      WRITE(6,682)(L81,L72,ZD(L81),XD(L81),L81=1,L72)                   PTFT2310
  682 FORMAT(' ',2X,'TOP OF INTERMEDIATE THERMOCLINE DEFINED BY',2X,I2, PTFT2320
     *2X,'OF',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)             PTFT2330
  690 L73=1                                                             PTFT2340
  700 CONTINUE                                                          PTFT2350
      L83=1                                                             PTFT2360
      K16=0                                                             PTFT2370
C                                                                       PTFT2380
C     PROCESSING THE ISOPLETH DEFINING THE BOTTOM OF INTERMEDIATE       PTFT2390
C     THERMOCLINE                                                       PTFT2400
C                                                                       PTFT2410
      DO 800 K16=1,NBT                                                  PTFT2420
      IF(ZAH3I(K16).LT.0..OR.ZAH3I(K16).GT.457.) GO TO 770              PTFT2430
      ZE(L83)=-(ZAH3I(K16)/230.+1.)                                     PTFT2440
      XE(L83)=X(K16)                                                    PTFT2450
      GO TO 772                                                         PTFT2460
  770 L82=L83-1                                                         PTFT2470
      IF(L82.GE.2) GO TO 780                                            PTFT2480
      GO TO 790                                                         PTFT2490
  772 L83=L83+1                                                         PTFT2500
      IF(K16.NE.NBT) GO TO 800                                          PTFT2510
      L82=L83-1                                                         PTFT2520
  780 IF(L82.EQ.1) GO TO 790                                            PTFT2530
      IF(L31.GE.2) L31=2                                                PTFT2540
      CALL DRAW(L82,XE,ZE,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)PTFT2550
      L31=L31+1                                                         PTFT2560
      WRITE(6,1000) LAST                                                PTFT2570
      WRITE(6,782)(L21,L82,ZE(L21),XE(L21),L21=1,L82)                   PTFT2580
  782 FORMAT(' ',2X,'BOTTOM OF INTERMEDIATE THERMOCLINE DEFINED BY',2X, PTFT2590
     *I2,2X,'OF',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)          PTFT2600
  790 L83=1                                                             PTFT2610
  800 CONTINUE                                                          PTFT2620
      L93=1                                                             PTFT2630
      K17=0                                                             PTFT2640
C                                                                       PTFT2650
C     PROCESSING THE ISOPLETH DEFINING THE MEAN DEPTH OF INTERMEDIATE   PTFT2660
C     THERMOCLINE                                                       PTFT2670
C                                                                       PTFT2680
      DO 900 K17=1,NBT                                                  PTFT2690
      IF(ZAH2I(K17).LT.0..OR.ZAH2I(K17).GT.457.) GO TO 870              PTFT2700
      ZF(L93)=-(ZAH2I(K17)/230.+1.)                                     PTFT2710
      XF(L93)=X(K17)                                                    PTFT2720
      GO TO 872                                                         PTFT2730
  870 L92=L93-1                                                         PTFT2740
      IF(L92.GE.2) GO TO 880                                            PTFT2750
```

```
      GO TO 890
  872 L93=L93+1
      IF(K17.NE.NBT) GO TO 900
      L92=L93-1
  880 IF(L92.EQ.1) GO TO 890
      IF(L31.GE.2) L31=2
      CALL DRAW(L92,XF,ZF,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)
      L31=L31+1
      WRITE(6,1000) LAST
      WRITE(6,882)(L61,L92,ZF(L61),XF(L61),L61=1,L92)
  882 FORMAT(' ',2X,'MEAN DEPTH OF INTERMEDIATE THERMOCLINE DEFINED BY',
     *2X,I2,2X,'OF',2X,I2,2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)
  890 L93=1
  900 CONTINUE
      N93=1
      M17=0
C
C     PROCESSING THE ISOPLETH DEFINING THE MIXED LAYER DEPTH
C
      DO 935 M17=1,NBT
      IF(ZAIXLR(M17).LT.0..OR.ZAIXLR(M17).GT.457.) GO TO 915
      ZG(N93)=-(ZAIXLR(M17)/230.+1.)
      XG(N93)=X(M17)
      GO TO 920
  915 N92=N93-1
      IF(N92.GE.2) GO TO 925
      GO TO 930
  920 N93=N93+1
      IF(M17.NE.NBT) GO TO 935
      N92=N93-1
  925 IF(N92.EQ.1) GO TO 930
      IF(L31.GE.2) L31=2
      CALL DRAW(N92,XG,ZG,L31,0,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)
      L31=L31+1
      WRITE(6,1000) LAST
      WRITE(6,932)(L61,N92,ZG(L61),XG(L61),L61=1,N92)
  932 FORMAT(' ',2X,'MIXED LAYER DEPTH DEFINED BY',2X,I2,2X,'OF',2X,I2,
     *2X,'POINTS AT Z=',F5.1,2X,'AT X=',F5.1)
  930 N93=1
  935 CONTINUE
C
C     ARTIFICIAL VALUES TO COMPLETE THE PLOT
C
  940 ZZ(1)=-1.
      ZZ(2)=-(100./230.+1.)
      ZZ(3)=-(200./230.+1.)
      ZZ(4)=-(300./230.+1.)
      ZZ(5)=-(400./230.+1.)
```

```
      ZZ(6)=-(500./230.+1.)                                             PTFT3240
      XX(1)=0.                                                          PTFT3250
      XX(2)=0.                                                          PTFT3260
      XX(3)=0.                                                          PTFT3270
      XX(4)=0.                                                          PTFT3280
      XX(5)=0.                                                          PTFT3290
      XX(6)=0.                                                          PTFT3300
      IF(L31.EQ.1) GO TO 960                                            PTFT3310
      CALL DRAW(6,XX,ZZ,3,4,LABEL,ITITL3,1.0,1.0,5,0,2,2,8,5,0,LAST)    PTFT3320
      WRITE(6,1000) LAST                                                PTFT3330
 1000 FORMAT('0',30X,'LAST =',I1)                                       PTFT3340
  960 RETURN                                                            PTFT3350
      END                                                               PTFT3360
C                                                                       PTIT0010
C     SUBROUTINE PLTMIT                                                 PTIT0020
C                                                                       PTIT0030
C     PURPOSE                                                           PTIT0040
C        TO PLOT THE AVERAGE LINEAR FIELD OF ISOTHERMS GENERATED BY     PTIT0050
C        SUBROUTINE STATCS (FOR A SPECIFIED NUMBER OF CRUISES)          PTIT0060
C                                                                       PTIT0070
C     USAGE                                                             PTIT0080
C        CALL PLTMIT(ISO,NBT,LABEL,ITITL6)                              PTIT0090
C                                                                       PTIT0100
C     DESCRIPTION OF PARAMETERS                                         PTIT0110
C        ISO    -INPUT NUMBER OF ISOTHERMS                              PTIT0120
C        NBT    -INPUT NUMBER OF STATIONS ALONG THE CROSS SECTION OR    PTIT0130
C                THE TIME AXIS                                          PTIT0140
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE           PTIT0150
C        ITITL6-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR    PTIT0160
C                PLOT OF THE AVERAGE DEPTHS OF ISOTHERMS (FOR A GIVEN   PTIT0170
C                PERIOD) ALONG A CROSS SECTION OR TIME AXIS             PTIT0180
C                                                                       PTIT0190
C     REMARKS                                                           PTIT0200
C        (1)THIS SUBROUTINE RECEIVES THE REQUIRED INFORMATION FROM THE  PTIT0210
C        MAIN PROGRAM AND SUBROUTINE STATCS BY MEANS OF 'COMMON'        PTIT0220
C        LABELLED AREAS.                                                PTIT0230
C        (2)SUBROUTINE STATCS COMPUTES (FOR EACH STATION OF A GIVEN     PTIT0240
C        CROSS SECTION) THE AVERAGE DEPTH OF SPECIFIED ISOTHERMS. THE   PTIT0250
C        CROSS SECTION) THE AVERAGE DEPTH OF SPECIFIED ISOTHERMS. THE   PTIT0260
C        PRESENT SUBROUTINE (PLTMIT) PLOTS THE FIELD OF ISOTHERMS       PTIT0270
C        ALONG A GIVEN CROSS SECTION IN THE FOLLOWING WAY: CONNECTING   PTIT0280
C        BY STRAIGHT LINES THE POINTS WITH THE SAME ISOTHERMAL VALUE    PTIT0290
C        LOCATED AT SUCCESSIVE GEOGRAPHICAL POSITIONS                   PTIT0300
C                                                                       PTIT0310
C     SUBROUTINES REQUIRED                                              PTIT0320
C        DRAW (IBM SCIENTIFIC LIBRARY)                                  PTIT0330
C                                                                       PTIT0340
C     ..................................................................PTIT0350
```

```
C                                                                          PTIT0360
      SUBROUTINE PLTMIT(ISC,NBT,LABEL,ITITL6)                              PTIT0370
      REAL*8 ITITL6(12)                                                    PTIT0380
      COMMON/WORKA/TO(60),ZO(60),ZSHAL(20,20),ZINTR(20,20),ZDEEP(20,20),PTIT0390
     *XU(20),XV(20),XW(20),ZU(20),ZV(20),ZW(20),XX(6),ZZ(6)               PTIT0400
      COMMON/WORKB/X(20),TI(19)                                            PTIT0410
      COMMON/WORKE/Z11CS(20,12),Z11CI(20,12),Z11CD(20,12),Z12CS(20,12),   PTIT0420
     *Z12CI(20,12),Z12CD(20,12),Z13CS(20,12),Z13CI(20,12),Z13CD(20,12),   PTIT0430
     *Z14CS(20,12),Z14CI(20,12),Z14CD(20,12),Z15CS(20,12),Z15CI(20,12),   PTIT0440
     *Z15CD(20,12),Z16CS(20,12),Z16CI(20,12),Z16CD(20,12),Z17CS(20,12),   PTIT0450
     *Z17CI(20,12),Z17CD(20,12),Z18CS(20,12),Z18CI(20,12),Z18CD(20,12),   PTIT0460
     *Z19CS(20,12),Z19CI(20,12),Z19CD(20,12),Z20CS(20,12),Z20CI(20,12),   PTIT0470
     *Z20CD(20,12),Z21CS(20,12),Z21CI(20,12),Z21CD(20,12),Z22CS(20,12),   PTIT0480
     *Z22CI(20,12),Z22CD(20,12),Z23CS(20,12),Z23CI(20,12),Z23CD(20,12),   PTIT0490
     *Z24CS(20,12),Z24CI(20,12),Z24CD(20,12),Z25CS(20,12),Z25CI(20,12),   PTIT0500
     *Z25CD(20,12),ZTUCS(20,12),ZTUCI(20,12),ZTUCD(20,12),ZTLCS(20,12),   PTIT0510
     *ZTLCI(20,12),ZTLCD(20,12),ZSUCS(20,12),ZSUCI(20,12),ZSUCD(20,12),   PTIT0520
     *ZSLCS(20,12),ZSLCI(20,12),ZSLCD(20,12),ZTTMD(19,12)                 PTIT0530
      L31=1                                                                PTIT0540
      K9=0                                                                 PTIT0550
      DO 400 K9=1,ISO                                                      PTIT0560
      L13=1                                                                PTIT0570
      I9=0                                                                 PTIT0580
      DO 100 I9=1,NBT                                                      PTIT0590
      IF(ZTTMD(K9,I9).LT.0..OR.ZTTMD(K9,I9).GT.457.) GO TO 70             PTIT0600
      XU(L13)=X(I9)                                                        PTIT0610
      ZU(L13)=-(ZTTMD(K9,I9)/230.+1.)                                      PTIT0620
      GO TO 72                                                             PTIT0630
   70 L12=L13-1                                                            PTIT0640
      IF(L12.GE.2) GO TO 80                                                PTIT0650
      GO TO 90                                                             PTIT0660
   72 L13=L13+1                                                            PTIT0670
      IF(I9.NE.NBT) GO TO 100                                              PTIT0680
      L12=L13-1                                                            PTIT0690
   80 IF(L12.EQ.1) GO TO 90                                                PTIT0700
      IF(L31.GE.2) L31=2                                                   PTIT0710
      CALL DRAW(L12,XU,ZU,L31,0,LABEL,ITITL2,1.0,1.0,5,0,2,2,8,5,0,LAST)PTIT0720
      L31=L31+1                                                            PTIT0730
      WRITE(6,1000) LAST                                                   PTIT0740
      WRITE(6,82)(I9,L81,L12,ZU(L81),XU(L81),L81=1,L12)                    PTIT0750
   82 FORMAT(' ',2X,'ISOLINE (',I2,')',2X,'DEFINED BY',2X,I2,2X,'OF',     PTIT0760
     *2X,I2,2X,'POINTS AT Z=',F14.7,2X,'AT X=',F10.7)                      PTIT0770
   90 L13=1                                                                PTIT0780
  100 CONTINUE                                                             PTIT0790
  400 CONTINUE                                                             PTIT0800
      ZZ(1)=-1.                                                            PTIT0810
```

```
      ZZ(2)=-(100./230.+1.)
      ZZ(3)=-(200./230.+1.)
      ZZ(4)=-(300./230.+1.)
      ZZ(5)=-(400./230.+1.)
      ZZ(6)=-(500./230.+1.)
      XX(1)=0.
      XX(2)=0.
      XX(3)=0.
      XX(4)=0.
      XX(5)=0.
      XX(6)=0.
      CALL DRAW(6,XX,ZZ,3,4,LABEL,ITITL2,1.0,1.0,5,0,2,2,8,5,0,LAST)
      WRITE(6,1000) LAST
 1000 FCRMAT('0',30X,'LAST =',I1)
      RETURN
      END
C
C     ...............................................................
C
C     SLBRCUTINE RLINE
C
C     PURPOSE
C        TO PLOT THE LEAST SQUARE REGRESSION LINES
C
C     LSAGE
C        SUBROUTINE RLINE(ITITL5,LABEL,SCALE3)
C
C     DESCRIPTICN OF PARAMETERS
C        ITITL5-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR
C               PLOT OF SCATTER DIAGRAM AND REGRESSION LINES AS THE
C               RESULT OF STATISTICAL STUDY BETWEEN TWO VARIABLES
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLCTTED CURVE
C        SCALE3-INPUT CONTAINING A SCALE VALUE WHICH ALLOWS THE PLOT
C               OF THE SCATTER DIAGRAM AND REGRESSICN LINES IN THE
C               SAME GRAPH
C
C     REMARKS
C        (1)THIS SUBROUTINE USES A 'CCMMON' LABELLED AREA TO RECEIVE
C        THE REQUIRED DATA FROM SUBROUTINE STATCS.
C        (2)THE SCATTER DIAGRAM (PLOTTED BY STATCS) AND THE REGRESSION
C        LINES (PLOTTED BY RLINE) ARE DRAWN IN THE SAME DIAGRAM.
C
C     SUBRCUTINES REQUIRED
C        DRAW (IBM SCIENTIFIC LIBRARY)
C
C     ...............................................................
C
```

```
      SUBROUTINE RLINE(ITITL5,LABEL,SCALE3)                             RLN 0320
      REAL*8 ITITL5(12)                                                 RLN 0330
      COMMON/WORKA/CODE(2),AUX(400),XBAR(10),STD(10),SKEW(10),CURT(10), RLN 0340
     *RR(4),NN(4),AA(4),BB(2,2),SS(2,2),X2(100),Y2(100),X3(100),Y3(100),RLN 0350
     *XX2(100),YY3(100),BLAN30(370)                                     RLN 0360
      NRP=50                                                            RLN 0370
      XX2(1)=-0.5*SCALE3                                                RLN 0380
      YY3(1)=-0.5*SCALE3                                                RLN 0390
      X2(1)=-0.5                                                        RLN 0400
      Y3(1)=-0.5                                                        RLN 0410
      DO 10 N13=1,NRP                                                   RLN 0420
      IF(N13.EQ.1) GO TO 8                                              RLN 0430
      X2(N13)=XX2(N13)/SCALE3                                           RLN 0440
      Y3(N13)=YY3(N13)/SCALE3                                           RLN 0450
    8 Y2(N13)=AA(2)/SCALE3+BB(2,1)*X2(N13)                              RLN 0460
      X3(N13)=AA(3)/SCALE3+BB(1,2)*Y3(N13)                              RLN 0470
      IF(N13.EQ.NRP) GO TO 10                                           RLN 0480
      XX2(N13+1)=XX2(1)+N13*50.                                         RLN 0490
      YY3(N13+1)=YY3(1)+N13*50.                                         RLN 0500
   10 CONTINUE                                                          RLN 0510
      CALL DRAW(NRP,X2,Y2,2,0,LABEL,ITITL5,0.5,0.5,4,4,2,2,8,8,0,LAST1) RLN 0520
      CALL DRAW(NRP,X3,Y3,3,0,LABEL,ITITL5,0.5,0.5,4,4,2,2,8,8,0,LAST2) RLN 0530
      WRITE(6,15) LAST1,LAST2                                           RLN 0540
   15 FORMAT('0',10X,'1ST REGRESSION LINE =',I1,10X,'2ND REGRESSION LINERLN 0550
     * =',I1)                                                           RLN 0560
      RETURN                                                            RLN 0570
      END                                                               RLN 0580
C                                                                       STCS0010
C     ..................................................................STCS0020
C                                                                       STCS0030
C     SUBROUTINE STATCS                                                 STCS0040
C                                                                       STCS0050
C     PURPOSE                                                           STCS0060
C        TO STUDY BY MEANS OF STATISTICS ONE GIVEN PARAMETER (1ST,2ND,  STCS0070
C        3RD AND 4TH MOMENTS) OR THE INTERRELATION BETWEEN TWO GIVEN    STCS0080
C        PARAMETERS (CORRELATION AND REGRESSION COEFFICIENTS,LEAST      STCS0090
C        SQUARE REGRESSION LINES AND SCATTER DIAGRAM)                   STCS0100
C                                                                       STCS0110
C     USAGE                                                             STCS0120
C        SUBROUTINE STATCS(ISTA,IORDER,MORDER,KEK1,KEK2,ITITL5,LABEL,   STCS0130
C        ICSEC,IERIOD,IBEGIN,IEND)                                      STCS0140
C                                                                       STCS0150
C     DESCRIPTION OF PARAMETERS                                         STCS0160
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS        STCS0170
C               SECTION                                                 STCS0180
C        IORDER-INPUT VALUE CONTAINING THE NUMBER OF THE CRUISE TO      STCS0190
C               INITIATE THE STATISTICAL STUDY                          STCS0200
C        MORDER-INPUT VALUE CONTAINING THE NUMBER OF THE CRUISE TO      STCS0210
```

```
C          END THE STATISTICAL STUDY                                   STCS0220
C      KEK1  -INPUT VALUE INDICATING THE FIRST OF TWO VARIABLES TO     STCS0230
C            BE INTERRELATED                                           STCS0240
C      KEK2  -INPUT VALUE INDICATING THE SECONC OF TWO VARIABLES TO    STCS0250
C            BE INTERRELATED                                           STCS0260
C      ITITL5-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR     STCS0270
C            PLOT OF SCATTER DIAGRAM AND REGRESSION LINES AS THE       STCS0280
C            RESULT OF STATISTICAL STUDY BETWEEN TWO VARIABLES         STCS0290
C      LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE            STCS0300
C      ICSEC -INPUT NUMBER OF IDENTIFICATION OF THE CROSS SECTION      STCS0310
C      IERIOD-INPUT VALUE CONTAINING THE NUMBER OF DAYS CONCERNING     STCS0320
C            THE DURATION OF THE STUDY                                 STCS0330
C      IBEGIN-INPUT VALUE CONTAINING THE MONTH,DAY AND YEAR OF THE     STCS0340
C            BEGINNING OF THE STUDY                                    STCS0350
C      IEND  -INPUT VALUE CONTAINING THE MONTH,DAY AND YEAR OF THE     STCS0360
C            END OF THE STUDY                                          STCS0370
C                                                                      STCS0380
C   REMARKS                                                            STCS0390
C      (1)THIS SUBROUTINE USES 'COMMON' LABELLED AREAS TO RECEIVE THE  STCS0400
C      REQUIRED DATA FROM THE MAIN PROGRAM AND TO GIVE PROCESSED       STCS0410
C      INFORMATION TO SUBROUTINE RLINE.                                STCS0420
C      (2)THE SCATTER DIAGRAM (PLOTTED BY STATCS) AND THE REGRESSION   STCS0430
C      LINES (PLOTTED BY RLINE) ARE DRAWN IN THE SAME DIAGRAM.         STCS0440
C      (3)THIS SUBROUTINE IDENTIFIES NEGATIVE VALUES AS MISSING DATA.  STCS0450
C                                                                      STCS0460
C   SUBROUTINES AND FUNCTIONS REQUIRED                                 STCS0470
C      PRINT                                                           STCS0480
C      MISR (IBM SCIENTIFIC SUBROUTINE LIBRARY)                        STCS0490
C      AMOD (IBM SCIENTIFIC LIBRARY)                                   STCS0500
C      AMAX1 (IBM SCIENTIFIC LIBRARY)                                  STCS0510
C      DRAW (IBM SCIENTIFIC LIBRARY)                                   STCS0520
C      RLINE                                                           STCS0530
C                                                                      STCS0540
C   METHOD                                                             STCS0550
C      SEE SUBROUTINE MISR                                             STCS0560
C                                                                      STCS0570
C   ...................................................................STCS0580
C                                                                      STCS0590


      SUBROUTINE STATCS(ISTA,IORDER,MORDER,KEK1,KEK2,ITITL5,LABEL,ICSEC,STCS0600
     *IERIOD,IBEGIN,IEND)                                               STCS0610
      REAL*8 ITITL5(12)                                                 STCS0620
      COMMON/WORKA/CUDE(2),AUX(400),XBAR(10),STD(10),SKEW(10),CURT(10), STCS0630
     *RR(4),NN(4),AA(4),BB(2,2),SS(2,2),BUX(400),X1(100),Y1(100),       STCS0640
     *BLAN20(390)                                                       STCS0650
      COMMON/WORKC/ZTH1S(20,12),ZTH1I(20,12),ZTH1D(20,12),ZTH2S(20,12), STCS0660
     *ZTH2I(20,12),ZTH2D(20,12),ZTH3S(20,12),ZTH3I(20,12),ZTH3D(20,12), STCS0670
```

```
     *ZTHWS(20,12),ZTHWI(20,12),ZTHWD(20,12),TTHVS(20,12),TTHVI(20,12),
     *TTHVD(20,12),ZMIXLR(20,12),TSFC(20,12)
      COMMON/WORKD/R00(20,12),KANG00(20,12),R01(20,12),KANG01(20,12),
     *R02(20,12),KANG02(20,12),R03(20,12),KANG03(20,12),R05(20,12),
     *KANG05(20,12),R10(20,12),KANG10(20,12),R001(20,12),R002(20,12),
     *R011(20,12),R012(20,12),R021(20,12),R022(20,12),R031(20,12),
     *R032(20,12),R051(20,12),R052(20,12),R101(20,12),R102(20,12)
      COMMON/WORKE/Z11CS(20,12),Z11CI(20,12),Z11CD(20,12),Z12CS(20,12),
     *Z12CI(20,12),Z12CD(20,12),Z13CS(20,12),Z13CI(20,12),Z13CD(20,12),
     *Z14CS(20,12),Z14CI(20,12),Z14CD(20,12),Z15CS(20,12),Z15CI(20,12),
     *Z15CD(20,12),Z16CS(20,12),Z16CI(20,12),Z16CD(20,12),Z17CS(20,12),
     *Z17CI(20,12),Z17CD(20,12),Z18CS(20,12),Z18CI(20,12),Z18CD(20,12),
     *Z19CS(20,12),Z19CI(20,12),Z19CD(20,12),Z20CS(20,12),Z20CI(20,12),
     *Z20CD(20,12),Z21CS(20,12),Z21CI(20,12),Z21CD(20,12),Z22CS(20,12),
     *Z22CI(20,12),Z22CD(20,12),Z23CS(20,12),Z23CI(20,12),Z23CD(20,12),
     *Z24CS(20,12),Z24CI(20,12),Z24CD(20,12),Z25CS(20,12),Z25CI(20,12),
     *Z25CD(20,12),ZTUCS(20,12),ZTUCI(20,12),ZTUCD(20,12),ZTLCS(20,12),
     *ZTLCI(20,12),ZTLCD(20,12),ZSUCS(20,12),ZSUCI(20,12),ZSUCD(20,12),
     *ZSLCS(20,12),ZSLCI(20,12),ZSLCD(20,12),ZTTMD(19,12)
C
C     TITLE OF THE OUTPUT
C
      CALL PRINT(IBEGIN,IEND,ISTA,ICSEC,IERIOD)
      JORDER=MORDER
C
C     PROCESSING THE FIRST OF TWO VARIABLES TO BE INTERRELATED
C
      IND=1
      IF(KEK1.EQ.10) GO TO 608
      IF(KEK1.EQ.11) GO TO 612
      IF(KEK1.EQ.12) GO TO 616
      IF(KEK1.EQ.13) GO TO 620
      IF(KEK1.EQ.14) GO TO 624
      IF(KEK1.EQ.15) GO TO 628
      IF(KEK1.EQ.16) GO TO 632
      IF(KEK1.EQ.17) GO TO 636
      IF(KEK1.EQ.18) GO TO 640
      IF(KEK1.EQ.19) GO TO 644
      IF(KEK1.EQ.20) GO TO 648
      IF(KEK1.EQ.24) GO TO 652
      IF(KEK1.EQ.25) GO TO 656
      IF(KEK1.EQ.26) GO TO 660
      IF(KEK1.EQ.27) GO TO 664
      IF(KEK1.EQ.28) GO TO 668
      IF(KEK1.EQ.29) GO TO 672
      IF(KEK1.EQ.30) GO TO 676
      IF(KEK1.EQ.31) GO TO 680
      IF(KEK1.EQ.32) GO TO 684
```

```
      IF(KEK1.EQ.33) GO TO 688
      IF(KEK1.EQ.34) GO TO 692
      IF(KEK1.EQ.35) GO TO 710
      IF(KEK1.EQ.36) GO TO 714
      IF(KEK1.EQ.37) GO TO 718
      IF(KEK1.EQ.38) GO TO 722
      IF(KEK1.EQ.39) GO TO 726
      IF(KEK1.EQ.40) GO TO 730
      IF(KEK1.EQ.41) GO TO 734
      IF(KEK1.EQ.42) GO TO 738
      IF(KEK1.EQ.43) GO TO 742
      IF(KEK1.EQ.44) GO TO 746
      IF(KEK1.EQ.45) GO TO 750
      IF(KEK1.EQ.46) GO TO 754
      IF(KEK1.EQ.47) GO TO 758
      IF(KEK1.EQ.48) GO TO 762
      GO TO 990
  608 WRITE(6,610)
  610 FORMAT(9X,'(1) DEPTH OF THE TOP OF SHALLOW THERMOCLINE,',/)
      DO 609 NORDER=IORDER,MORDER
      AUX(NORDER)=ZTH1S(NORDER,ISTA)
  609 CONTINUE
      GO TO 704
  612 WRITE(6,614)
  614 FORMAT(9X,'(1) DEPTH OF THE TOP OF INTERMEDIATE THERMOCLINE',/)
      DO 613 NORDER=IORDER,MORDER
      AUX(NORDER)=ZTH1I(NORDER,ISTA)
  613 CONTINUE
      GO TO 704
  616 WRITE(6,618)
  618 FORMAT(9X,'(1) DEPTH OF THE TOP OF DEEP OR UNIQUE THERMOCLINE,',/)
      DO 617 NORDER=IORDER,MORDER
      AUX(NORDER)=ZTH1D(NORDER,ISTA)
  617 CONTINUE
      GO TO 704
  620 WRITE(6,622)
  622 FORMAT(9X,'(1) MEAN DEPTH OF THE SHALLOW THERMOCLINE,',/)
      DO 621 NORDER=IORDER,MORDER
      AUX(NORDER)=ZTH2S(NORDER,ISTA)
  621 CONTINUE
      GO TO 704
  624 WRITE(6,626)
  626 FORMAT(9X,'(1) MEAN DEPTH OF THE INTERMEDIATE THERMOCLINE,',/)
      DO 625 NORDER=IORDER,MORDER
      AUX(NORDER)=ZTH2I(NORDER,ISTA)
  625 CONTINUE
      GO TO 704
  628 WRITE(6,630)
```

```
630 FORMAT(9X,'(1) MEAN DEPTH OF THE DEEP OR UNIQUE THERMOCLINE,',/)  STCS1640
    DO 629 NORDER=IORDER,MORDER                                       STCS1650
    AUX(NORDER)=ZTH2D(NORDER,ISTA)                                    STCS1660
629 CONTINUE                                                          STCS1670
    GO TO 704                                                         STCS1680
632 WRITE(6,634)                                                      STCS1690
634 FORMAT(9X,'(1) DEPTH OF THE BOTTOM OF SHALLOW THERMOCLINE,',/)    STCS1700
    DO 633 NORDER=IORDER,MORDER                                       STCS1710
    AUX(NORDER)=ZTH3S(NORDER,ISTA)                                    STCS1720
633 CONTINUE                                                          STCS1730
    GO TO 704                                                         STCS1740
636 WRITE(6,638)                                                      STCS1750
638 FORMAT(9X,'(1) DEPTH OF THE BOTTOM OF INTERMEDIATE THERMOCLINE,',/STCS1760
   *)                                                                 STCS1770
    DO 637 NORDER=IORDER,MORDER                                       STCS1780
    AUX(NORDER)=ZTH3I(NORDER,ISTA)                                    STCS1790
637 CONTINUE                                                          STCS1800
    GO TO 704                                                         STCS1810
640 WRITE(6,642)                                                      STCS1820
642 FORMAT(9X,'(1) DEPTH OF THE BOTTOM OF DEEP OR UNIQUE THERMOCLINE,'STCS1830
   *,/)                                                               STCS1840
    DO 641 NORDER=IORDER,MORDER                                       STCS1850
    AUX(NORDER)=ZTH3D(NORDER,ISTA)                                    STCS1860
641 CONTINUE                                                          STCS1870
    GO TO 704                                                         STCS1880
644 WRITE(6,646)                                                      STCS1890
646 FORMAT(9X,'(1) WIDTH OF THE SHALLOW THERMOCLINE,',/)              STCS1900
    DO 645 NORDER=IORDER,MORDER                                       STCS1910
    AUX(NORDER)=ZTHWS(NORDER,ISTA)                                    STCS1920
645 CONTINUE                                                          STCS1930
    GO TO 704                                                         STCS1940
648 WRITE(6,650)                                                      STCS1950
650 FORMAT(9X,'(1) WIDTH OF THE INTERMEDIATE THERMOCLINE,',/)         STCS1960
    DO 649 NORDER=IORDER,MORDER                                       STCS1970
    AUX(NORDER)=ZTHWI(NORDER,ISTA)                                    STCS1980
649 CONTINUE                                                          STCS1990
    GO TO 704                                                         STCS2000
652 WRITE(6,654)                                                      STCS2010
654 FORMAT(9X,'(1) WIDTH OF THE DEEP OR UNIQUE THERMOCLINE,',/)       STCS2020
    DO 653 NORDER=IORDER,MORDER                                       STCS2030
    AUX(NORDER)=ZTHWD(NORDER,ISTA)                                    STCS2040
653 CONTINUE                                                          STCS2050
    GO TO 704                                                         STCS2060
656 WRITE(6,658)                                                      STCS2070
658 FORMAT(9X,'(1) THERMAL VARIATION OF THE SHALLOW THERMOCLINE,',/)  STCS2080
    DO 657 NORDER=IORDER,MORDER                                       STCS2090
    AUX(NORDER)=TTHVS(NORDER,ISTA)                                    STCS2100
657 CONTINUE                                                          STCS2110
```

```
      GO TO 704
  660 WRITE(6,662)
  662 FCRMAT(9X,'(1) THERMAL VARIATION OF THE INTERMEDIATE THERMOCLINE,'
     *,/)
      DC 661 NCRDER=ICRDER,MORDER
      AUX(NORDER)=TTHVI(NORDER,ISTA)
  661 CCNTINUE
      GO TO 704
  664 WRITE(6,666)
  666 FCRMAT(9X,'(1) THERMAL VARIATION OF THE DEEP OR UNIQUE THERMOCLINE
     *,',/)
      DC 665 NCRDER=ICRDER,MCRDER
      AUX(NORDER)=TTHVD(NCRDER,ISTA)
  665 CCNTINUE
      GC TO 704
  668 WRITE(6,670)
  670 FCRMAT(9X,'(1) DEPTH OF THE MIXED LAYER,',/)
      DO 669 NCRDER=IORDER,MORDER
      AUX(NORDER)=ZMIXLR(NORDER,ISTA)
  669 CCNTINUE
      GC TO 704
  672 WRITE(6,674)
  674 FORMAT(9X,'(1) TEMPERATURE AT SEA SURFACE,',/)
      DC 673 NCRDER=ICRDER,MORDER
      AUX(NORDER)=TSFC(NCRDER,ISTA)
  673 CONTINUE
      GC TO 704
  676 WRITE(6,678)
  678 FORMAT(9X,'(1) DEPTH OF THE 12 DEG CELSIUS ISCTHERM,',/)
      DC 677 NORDER=ICRDER,MORDER
      AUX(NORDER)=Z12CD(NCRDER,ISTA)
  677 CCNTINUE
      GC TO 704
  680 WRITE(6,682)
  682 FCRMAT(9X,'(1) DEPTH OF THE TOP OF SARDINE HABITAT,',/)
      DC 681 NCRDER=ICRDER,MORDER
      AUX(NORDER)=ZSUCD(NORDER,ISTA)
  681 CCNTINUE
      GC TO 704
  684 WRITE(6,686)
  686 FCRMAT(9X,'(1) DEPTH OF THE BOTTOM OF SARCINE HABITAT,',/)
      DC 685 NCRDER=ICRDER,MCRDER
      AUX(NORDER)=ZSLCD(NORDER,ISTA)
  685 CCNTINUE
      GO TO 704
  688 WRITE(6,690)
  69C FCRMAT(9X,'(1) DEPTH OF THE TOP OF TUNA HABITAT,',/)
      DC 689 NCRDER=ICRDER,MORDER
```

```
      AUX(NORDER)=ZTUCD(NORDER,ISTA)                                     STCS2600
  689 CCNTINUE                                                           STCS2610
      GC TO 704                                                          STCS2620
  692 WRITE(6,694)                                                       STCS2630
  694 FORMAT(9X,'(1) DEPTH OF THE BOTTOM OF TUNA HABITAT,',/)            STCS2640
      DC 693 NCRDER=ICRDER,MORDER                                        STCS2650
      AUX(NORDER)=ZTLCD(NORDER,ISTA)                                     STCS2660
  693 CCNTINUE                                                           STCS2670
      GO TO 704                                                          STCS2680
  710 WRITE(6,712)                                                       STCS2690
  712 FORMAT(9X,'(1) DEPTH OF THE 11 DEG CELSIUS ISOTHERM,',/)           STCS2700
      DC 711 NCRDER=ICRDER,MORDER                                        STCS2710
      AUX(NOPDER)=Z11CD(NCRDER,ISTA)                                     STCS2720
  711 CCNTINUE                                                           STCS2730
      GC TO 704                                                          STCS2740
  714 WRITE(6,716)                                                       STCS2750
  716 FCRMAT(9X,'(1) DEPTH OF THE 13 DEG CELSIUS ISOTHERM,',/)           STCS2760
      DC 715 NCRDER=ICRDER,MORDER                                        STCS2770
      AUX(NOPDER)=Z13CD(NCRDER,ISTA)                                     STCS2780
  715 CCNTINUE                                                           STCS2790
      GC TO 704                                                          STCS2800
  718 WRITE(6,720)                                                       STCS2810
  720 FORMAT(9X,'(1) DEPTH OF THE 14 DEG CELSIUS ISCTHERM,',/)           STCS2820
      DC 719 NCRDER=ICRDER,MORDER                                        STCS2830
      AUX(NORDER)=Z14CD(NCRDER,ISTA)                                     STCS2840
  719 CONTINUE                                                           STCS2850
      GC TO 704                                                          STCS2860
  722 WRITE(6,724)                                                       STCS2870
  724 FORMAT(9X,'(1) DEPTH OF THE 15 DEG CELSIUS ISCTHERM,',/)           STCS2880
      DC 723 NCRDER=ICRDER,MORDER                                        STCS2890
      AUX(NORDER)=Z15CD(NCRDER,ISTA)                                     STCS2900
  723 CCNTINUE                                                           STCS2910
      GC TO 704                                                          STCS2920
  726 WRITE(6,728)                                                       STCS2930
  728 FCRMAT(9X,'(1) DEPTH OF THE 16 DEG CELSIUS ISOTHERM,',/)           STCS2940
      DC 727 NCRDER=ICRDER,MORDER                                        STCS2950
      AUX(NORDER)=Z16CD(NCRDER,ISTA)                                     STCS2960
  727 CCNTINUE                                                           STCS2970
      GC TO 704                                                          STCS2980
  730 WRITE(6,732)                                                       STCS2990
  732 FORMAT(9X,'(1) DEPTH OF THE 17 DEG CELSIUS ISOTHERM,',/)           STCS3000
      DC 731 NCRDER=ICRDER,MCRDER                                        STCS3010
      AUX(NORDER)=Z17CD(NCRDER,ISTA)                                     STCS3020
  731 CCNTINUE                                                           STCS3030
      GO TO 704                                                          STCS3040
  734 WRITE(6,736)                                                       STCS3050
  736 FCRMAT(9X,'(1) DEPTH OF THE 18 DEG CELSIUS ISOTHERM,',/)           STCS3060
      DC 735 NORDER=ICRDER,MORDER                                        STCS3070
```

```
      AUX(NORDER)=Z18CD(NORDER,ISTA)                                    STCS3080
  735 CONTINUE                                                          STCS3090
      GO TO 704                                                         STCS3100
  738 WRITE(6,740)                                                      STCS3110
  740 FORMAT(9X,'(1) DEPTH OF THE 19 DEG CELSIUS ISOTHERM,',/)          STCS3120
      DO 739 NORDER=IORDER,MORDER                                       STCS3130
      AUX(NORDER)=Z19CD(NORDER,ISTA)                                    STCS3140
  739 CONTINUE                                                          STCS3150
      GO TO 704                                                         STCS3160
  742 WRITE(6,744)                                                      STCS3170
  744 FORMAT(9X,'(1) DEPTH OF THE 20 DEG CELSIUS ISOTHERM,',/)          STCS3180
      DO 743 NORDER=IORDER,MORDER                                       STCS3190
      AUX(NORDER)=Z20CD(NORDER,ISTA)                                    STCS3200
  743 CONTINUE                                                          STCS3210
      GO TO 704                                                         STCS3220
  746 WRITE(6,748)                                                      STCS3230
  748 FORMAT(9X,'(1) DEPTH OF THE 21 DEG CELSIUS ISOTHERM,',/)          STCS3240
      DO 747 NORDER=IORDER,MORDER                                       STCS3250
      AUX(NORDER)=Z21CD(NORDER,ISTA)                                    STCS3260
  747 CONTINUE                                                          STCS3270
      GO TO 704                                                         STCS3280
  750 WRITE(6,752)                                                      STCS3290
  752 FORMAT(9X,'(1) DEPTH OF THE 22 DEG CELSIUS ISOTHERM,',/)          STCS3300
      DO 751 NORDER=IORDER,MORDER                                       STCS3310
      AUX(NORDER)=Z22CD(NORDER,ISTA)                                    STCS3320
  751 CONTINUE                                                          STCS3330
  754 WRITE(6,756)                                                      STCS3340
  756 FORMAT(9X,'(1) DEPTH OF THE 23 DEG CELSIUS ISOTHERM,',/)          STCS3350
      DO 755 NORDER=IORDER,MORDER                                       STCS3360
      AUX(NORDER)=Z23CD(NORDER,ISTA)                                    STCS3370
  755 CONTINUE                                                          STCS3380
      GO TO 704                                                         STCS3390
  758 WRITE(6,760)                                                      STCS3400
  760 FORMAT(9X,'(1) DEPTH OF THE 24 DEG CELSIUS ISOTHERM,',/)          STCS3410
      DO 759 NORDER=IORDER,MORDER                                       STCS3420
      AUX(NORDER)=Z24CD(NORDER,ISTA)                                    STCS3430
  759 CONTINUE                                                          STCS3440
      GO TO 704                                                         STCS3450
  762 WRITE(6,764)                                                      STCS3460
  764 FORMAT(9X,'(1) DEPTH OF THE 25 DEG CELSIUS ISOTHERM,',/)          STCS3470
      DO 763 NORDER=IORDER,MORDER                                       STCS3480
      AUX(NORDER)=Z25CD(NORDER,ISTA)                                    STCS3490
  763 CONTINUE                                                          STCS3500
C                                                                       STCS3510
C     PROCESSING THE SECOND OF TWO VARIABLES TO BE INTERRELATED         STCS3520
C                                                                       STCS3530
  704 IF(KEK2.EQ.10) GO TO 808                                          STCS3540
      IF(KEK2.EQ.11) GO TO 812                                          STCS3550
```

```
      IF(KEK2.EQ.12) GO TO 816                                          STCS3560
      IF(KEK2.EQ.13) GO TO 820                                          STCS3570
      IF(KEK2.EQ.14) GO TO 824                                          STCS3580
      IF(KEK2.EQ.15) GO TO 828                                          STCS3590
      IF(KEK2.EQ.16) GO TO 832                                          STCS3600
      IF(KEK2.EQ.17) GO TO 836                                          STCS3610
      IF(KEK2.EQ.18) GO TO 840                                          STCS3620
      IF(KEK2.EQ.19) GO TO 844                                          STCS3630
      IF(KEK2.EQ.20) GO TO 848                                          STCS3640
      IF(KEK2.EQ.24) GO TO 852                                          STCS3650
      IF(KEK2.EQ.25) GO TO 856                                          STCS3660
      IF(KEK2.EQ.26) GO TO 860                                          STCS3670
      IF(KEK2.EQ.27) GO TO 864                                          STCS3680
      IF(KEK2.EQ.28) GO TO 868                                          STCS3690
      IF(KEK2.EQ.29) GO TO 872                                          STCS3700
      IF(KEK2.EQ.30) GO TO 876                                          STCS3710
      IF(KEK2.EQ.31) GO TO 880                                          STCS3720
      IF(KEK2.EQ.32) GO TO 884                                          STCS3730
      IF(KEK2.EQ.33) GO TO 888                                          STCS3740
      IF(KEK2.EQ.34) GO TO 892                                          STCS3750
      IF(KEK2.EQ.35) GO TO 896                                          STCS3760
      IF(KEK2.EQ.36) GO TO 900                                          STCS3770
      GO TO 990                                                         STCS3780
  808 WRITE(6,810)                                                      STCS3790
  810 FORMAT(9X,'(2) DIRECTION OF THE SURFACE WIND AT DAY XBT OBS.')    STCS3800
      DO 809 NORDER=IORDER,MORDER                                       STCS3810
      AUX(NORDER+JORDER)=KANG00(NORDER,ISTA)                            STCS3820
      IF(NORDER.NE.IORDER) GO TO 1808                                   STCS3830
      AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.                        STCS3840
      GO TO 809                                                         STCS3850
 1808 DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS3860
 1809 IF(DAUX.LE.(-180.)) AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.    STCS3870
      DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS3880
      IF(DAUX.LE.(-180.)) GO TO 1809                                    STCS3890
  809 CONTINUE                                                          STCS3900
      IND=0                                                             STCS3910
      GO TO 904                                                         STCS3920
  812 WRITE(6,814)                                                      STCS3930
  814 FORMAT(9X,'(2) DIRECTION OF THE SURFACE WIND AT 1 DAY BEFORE DAY XSTCS3940
     *BT OBS.',/)                                                       STCS3950
      DO 813 NORDER=IORDER,MORDER                                       STCS3960
      AUX(NORDER+JORDER)=KANG01(NORDER,ISTA)                            STCS3970
      IF(NORDER.NE.IORDER) GO TO 1812                                   STCS3980
      AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.                        STCS3990
      GO TO 813                                                         STCS4000
 1812 DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4010
 1813 IF(DAUX.LE.(-180.)) AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.    STCS4020
      DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4030
```

```
      IF(DAUX.LE.(-180.)) GO TO 1813                                   STCS4040
  813 CONTINUE                                                          STCS4050
      IND=0                                                             STCS4060
      GO TO 904                                                         STCS4070
  816 WRITE(6,818)                                                      STCS4080
  818 FORMAT(9X,'(2) DIRECTION OF THE SURFACE WIND AT 2 DAYS BEFORE DAY STCS4090
     *XBT OBS.',/)                                                      STCS4100
      DO 817 NORDER=IORDER,MORDER                                       STCS4110
      AUX(NORDER+JORDER)=KANGO2(NORDER,ISTA)                            STCS4120
      IF(NORDER.NE.IORDER) GO TO 1816                                   STCS4130
      AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.                        STCS4140
      GO TO 817                                                         STCS4150
 1816 DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4160
 1817 IF(DAUX.LE.(-180.)) AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.    STCS4170
      DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4180
      IF(DAUX.LE.(-180.)) GO TO 1817                                    STCS4190
  817 CONTINUE                                                          STCS4200
      IND=0                                                             STCS4210
      GO TO 904                                                         STCS4220
  820 WRITE(6,822)                                                      STCS4230
  822 FORMAT(9X,'(2) DIRECTION OF THE SURFACE WIND AT 3 DAYS BEFORE DAY STCS4240
     *XBT OBS.',/)                                                      STCS4250
      DO 821 NORDER=IORDER,MORDER                                       STCS4260
      AUX(NORDER+JORDER)=KANGO3(NORDER,ISTA)                            STCS4270
      IF(NORDER.NE.IORDER) GO TO 1820                                   STCS4280
      AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.                        STCS4290
      GO TO 821                                                         STCS4300
 1820 DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4310
 1821 IF(DAUX.LE.(-180.)) AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.    STCS4320
      DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4330
      IF(DAUX.LE.(-180.)) GO TO 1821                                    STCS4340
  821 CONTINUE                                                          STCS4350
      IND=0                                                             STCS4360
      GO TO 904                                                         STCS4370
  824 WRITE(6,826)                                                      STCS4380
  826 FORMAT(9X,'(2) DIRECTION OF THE SURFACE WIND AT 5 DAYS BEFORE DAY STCS4390
     *XBT OBS.',/)                                                      STCS4400
      DO 825 NORDER=IORDER,MORDER                                       STCS4410
      AUX(NORDER+JORDER)=KANGO5(NORDER,ISTA)                            STCS4420
      IF(NORDER.NE.IORDER) GO TO 1824                                   STCS4430
      AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.                        STCS4440
      GO TO 825                                                         STCS4450
 1824 DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4460
 1825 IF(DAUX.LE.(-180.)) AUX(NORDER+JORDER)=AUX(NORDER+JORDER)+360.    STCS4470
      DAUX=AUX(NORDER+JORDER)-AUX(NORDER+JORDER-1)                      STCS4480
      IF(DAUX.LE.(-180.)) GO TO 1825                                    STCS4490
  825 CONTINUE                                                          STCS4500
      IND=0                                                             STCS4510
```

```
      GC TO 904                                                          STCS4520
  828 WRITE(6,830)                                                       STCS4530
  830 FCRMAT(9X,'(2) DIRECTION OF THE SURFACE WIND AT 10 DAYS BEFORE DAYSTCS4540
     * XBT OBS.',/)                                                      STCS4550
      DC 829 NCRDER=ICRDER,MCRDER                                        STCS4560
      AUX(NCRDER+JORDER)=KANG10(NORDER,ISTA)                             STCS4570
      IF(NORDER.NE.IORDER) GO TO 1828                                    STCS4580
      AUX(NCRDER+JORDER)=AUX(NCRDER+JCRDER)+360.                         STCS4590
      GO TO 829                                                          STCS4600
 1828 DAUX=AUX(NCRDER+JORDER)-AUX(NORDER+JCRDER-1)                       STCS4610
 1829 IF(DAUX.LE.(-18C.)) AUX(NORDER+JORDER)=AUX(NCRDER+JORDER)+36C.     STCS4620
      CAUX=AUX(NCRDER+JORDER)-AUX(NORDER+JCRDER-1)                       STCS4630
      IF(DAUX.LE.(-180.)) GO TC 1829                                     STCS4640
  829 CCNTINUE                                                           STCS4650
      IND=0                                                              STCS4660
      GO TO 904                                                          STCS4670
  832 WRITE(6,834)                                                       STCS4680
  834 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND AT DAY XBT OBS.',/)   STCS4690
      DC 833 NCRDER=ICRDER,MORDER                                        STCS4700
      AUX(NCRDER+JORDER)=R00(NORDER,ISTA)                                STCS4710
  833 CCNTINUE                                                           STCS4720
      GC TO 904                                                          STCS4730
  836 WRITE(6,838)                                                       STCS4740
  838 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND AT 1 DAY BEFORE DAY XSTCS4750
     *BT OBS.',/)                                                        STCS4760
      DC 837 NCRDER=ICRDER,MORDER                                        STCS4770
      AUX(NORDER+JORDER)=R01(NORDER,ISTA)                                STCS4780
  837 CCNTINUE                                                           STCS4790
      GO TO 904                                                          STCS4800
  840 WRITE(6,842)                                                       STCS4810
  842 FCRMAT(9X,'(2) INTENSITY OF THE SURFACE WIND AT 2 DAYS BEFORE DAY STCS4820
     *XBT OBS.',/)                                                       STCS4830
      DC 841 NORDER=ICRDER,MORDER                                        STCS4840
      AUX(NCPDER+JORDER)=R02(NORDER,ISTA)                                STCS4850
  841 CCNTINUE                                                           STCS4860
      GL TO 904                                                          STCS4870
  844 WRITE(6,846)                                                       STCS4880
  846 FCRMAT(9X,'(2) INTENSITY OF THE SURFACE WIND AT 3 DAYS BEFORE DAY STCS4890
     *XBT OBS.',/)                                                       STCS4900
      DC 845 NCRDER=IGRDER,MORDER                                        STCS4910
      AUX(NORDER+JORDER)=R03(NORDER,ISTA)                                STCS4920
  845 CCNTINUE                                                           STCS4930
      GO TO 904                                                          STCS4940
  848 WRITE(6,850)                                                       STCS4950
  850 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND AT 5 DAYS BEFORE DAY STCS4960
     *XBT OBS.',/)                                                       STCS4970
      DC 849 NORDER=ICRDER,MCRDER                                        STCS4980
      AUX(NORDER+JORDER)=R05(NORDER,ISTA)                                STCS4990
```

```
    * XBT OBS.',/)                                                       STCS5040
      DC 853 NORDER=IORDER,MORDER                                        STCS5050
      AUX(NORDER+JORDER)=R10(NORDER,ISTA)                                STCS5060
  853 CONTINUE                                                           STCS5070
      GO TO 904                                                          STCS5080
  856 WRITE(6,858)                                                       STCS5090
  858 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (ALONG THE COAST) AT  STCS5100
    *DAY XBT OBS.',/)                                                    STCS5110
      DO 857 NORDER=IORDER,MORDER                                        STCS5120
      AUX(NORDER+JORDER)=R001(NORDER,ISTA)                               STCS5130
  857 CONTINUE                                                           STCS5140
      GO TO 904                                                          STCS5150
  860 WRITE(6,862)                                                       STCS5160
  862 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (PERP. TO THE COAST)  STCS5170
    *AT DAY OF XBT OBS.',/)                                              STCS5180
      DO 861 NORDER=IORDER,MORDER                                        STCS5190
      AUX(NORDER+JORDER)=R002(NORDER,ISTA)                               STCS5200
  861 CONTINUE                                                           STCS5210
      GO TO 904                                                          STCS5220
  864 WRITE(6,866)                                                       STCS5230
  866 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (ALONG THE COAST) AT  STCS5240
    *1 DAY BEFORE DAY XBT OBS.',/)                                       STCS5250
      DO 865 NORDER=IORDER,MORDER                                        STCS5260
      AUX(NORDER+JORDER)=R011(NORDER,ISTA)                               STCS5270
  865 CONTINUE                                                           STCS5280
      GO TO 904                                                          STCS5290
  868 WRITE(6,870)                                                       STCS5300
  870 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (PERP. TO THE COAST)  STCS5310
    *AT 1 DAY BEFORE DAY XBT OBS.',/)                                    STCS5320
      DO 869 NORDER=IORDER,MORDER                                        STCS5330
      AUX(NORDER+JORDER)=R012(NORDER,ISTA)                               STCS5340
  869 CONTINUE                                                           STCS5350
      GO TO 904                                                          STCS5360
  872 WRITE(6,874)                                                       STCS5370
  874 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (ALONG THE COAST) AT  STCS5380
    *2 DAYS BEFORE DAY XBT OBS.',/)                                      STCS5390
      DO 873 NORDER=IORDER,MORDER                                        STCS5400
      AUX(NORDER+JORDER)=R021(NORDER,ISTA)                               STCS5410
  873 CONTINUE                                                           STCS5420
      GO TO 904                                                          STCS5430
  876 WRITE(6,878)                                                       STCS5440
  878 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (PERP. TO THE COAST)  STCS5450
    *AT 2 DAYS BEFORE DAY XBT OBS.',/)                                   STCS5460
      DO 877 NORDER=IORDER,MORDER                                        STCS5470
```

```
      AUX(NORDER+JORDER)=R022(NORDER,ISTA)                              STCS5480
  877 CONTINUE                                                          STCS5490
      GO TO 904                                                         STCS5500
  880 WRITE(6,882)                                                      STCS5510
  882 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (ALONG THE COAST) AT STCS5520
     *3 DAYS BEFORE DAY XBT OBS.',/)                                    STCS5530
      DO 881 NORDER=IORDER,MORDER                                       STCS5540
      AUX(NORDER+JORDER)=R031(NORDER,ISTA)                              STCS5550
  881 CONTINUE                                                          STCS5560
      GO TO 904                                                         STCS5570
  884 WRITE(6,886)                                                      STCS5580
  886 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (PERP. TO THE COAST) STCS5590
     *AT 3 DAYS BEFORE DAY XBT OBS.',/)                                 STCS5600
      DO 885 NORDER=IORDER,MORDER                                       STCS5610
      AUX(NORDER+JORDER)=R032(NORDER,ISTA)                              STCS5620
  885 CONTINUE                                                          STCS5630
      GO TO 904                                                         STCS5640
  888 WRITE(6,890)                                                      STCS5650
  890 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (ALONG THE COAST) AT STCS5660
     *5 DAYS BEFORE DAY XBT OBS.',/)                                    STCS5670
      DO 889 NORDER=IORDER,MORDER                                       STCS5680
      AUX(NORDER+JORDER)=R051(NORDER,ISTA)                              STCS5690
  889 CONTINUE                                                          STCS5700
      GO TO 904                                                         STCS5710
  892 WRITE(6,894)                                                      STCS5720
  894 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (PERP. TO THE COAST) STCS5730
     *AT 5 DAYS BEFORE DAY XBT OBS.',/)                                 STCS5740
      DO 893 NORDER=IORDER,MORDER                                       STCS5750
      AUX(NORDER+JORDER)=R052(NORDER,ISTA)                              STCS5760
  893 CONTINUE                                                          STCS5770
      GO TO 904                                                         STCS5780
  896 WRITE(6,898)                                                      STCS5790
  898 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (ALONG THE COAST) AT STCS5800
     *10 DAYS BEFORE DAY XBT OBS.',/)                                   STCS5810
      DO 897 NORDER=IORDER,MORDER                                       STCS5820
      AUX(NORDER+JORDER)=R101(NORDER,ISTA)                              STCS5830
  897 CONTINUE                                                          STCS5840
      GO TO 904                                                         STCS5850
  900 WRITE(6,902)                                                      STCS5860
  902 FORMAT(9X,'(2) INTENSITY OF THE SURFACE WIND (PERP. TO THE COAST) STCS5870
     *AT 10 DAYS BEFORE DAY XBT OBS.',/)                                STCS5880
      DO 901 NORDER=IORDER,MORDER                                       STCS5890
      AUX(NORDER+JORDER)=R102(NORDER,ISTA)                              STCS5900
  901 CONTINUE                                                          STCS5910
      GO TO 904                                                         STCS5920
  904 NOBS=(MORDER-IORDER)+1                                            STCS5930
      CODE(1)=-5.                                                       STCS5940
      CODE(2)=-5.                                                       STCS5950
```

```
      CALL MISR(NOBS,2,AUX,CODE,XBAR,STD,SKEW,CURT,RR,NN,AA,BB,SS,IER)
C
C     REDUCING THE ANGULAR PROCESSED DATA TO AN ANGLE LESS THAN 360
C
      IF(IND.EQ.0.AND.XBAR(1).GE.360.) XBAR(1)=AMOD(XBAR(1),360.)
      IF(IND.EQ.0.AND.XBAR(2).GE.360.) XBAR(2)=AMOD(XBAR(2),360.)
C
C     AVERAGE DEPTH OF A GIVEN ISOTHERM (AT A KNOWN STATION)
C
      IF(KEK1.EQ.30) ZTTMD(14,ISTA)=XBAR(1)
      IF(KEK1.EQ.31) ZTTMD(16,ISTA)=XBAR(1)
      IF(KEK1.EQ.32) ZTTMD(17,ISTA)=XBAR(1)
      IF(KEK1.EQ.33) ZTTMD(18,ISTA)=XBAR(1)
      IF(KEK1.EQ.34) ZTTMD(19,ISTA)=XBAR(1)
      IF(KEK1.EQ.35) ZTTMD(15,ISTA)=XBAR(1)
      IF(KEK1.EQ.36) ZTTMD(13,ISTA)=XBAR(1)
      IF(KEK1.EQ.37) ZTTMD(12,ISTA)=XBAR(1)
      IF(KEK1.EQ.38) ZTTMD(11,ISTA)=XBAR(1)
      IF(KEK1.EQ.39) ZTTMD(10,ISTA)=XBAR(1)
      IF(KEK1.EQ.40) ZTTMD(9,ISTA)=XBAR(1)
      IF(KEK1.EQ.41) ZTTMD(8,ISTA)=XBAR(1)
      IF(KEK1.EQ.42) ZTTMD(7,ISTA)=XBAR(1)
      IF(KEK1.EQ.43) ZTTMD(6,ISTA)=XBAR(1)
      IF(KEK1.EQ.44) ZTTMD(5,ISTA)=XBAR(1)
      IF(KEK1.EQ.45) ZTTMD(4,ISTA)=XBAR(1)
      IF(KEK1.EQ.46) ZTTMD(3,ISTA)=XBAR(1)
      IF(KEK1.EQ.47) ZTTMD(2,ISTA)=XBAR(1)
      IF(KEK1.EQ.48) ZTTMD(1,ISTA)=XBAR(1)
      WRITE(6,910)
  910 FORMAT(////,1X,'VARIABLES    MEAN    STD  DEVIATION    SKEWNESS
     *  KURTOSIS')
      WRITE(6,912)(I13,XBAR(I13),STD(I13),SKEW(I13),CURT(I13),I13=1,2)
  912 FORMAT(' ',3X,'(',I1,')',5X,F10.1,8X,F10.1,9X,F10.1,6X,F10.1)
      I32=1
      J32=1
      WRITE(6,922) NN(I32),NN(I32+1),NN(I32+1),NN(I32+2)
  922 FORMAT(///,1X,'NO. OF PAIRS OF OBS.',14X,I3,9X,I3,9X,I3,9X,I3)
      WRITE(6,918) RR(I32),RR(I32+1),RR(I32+1),RR(I32+2)
  918 FORMAT(' ','CORRELATION COEFFICIENTS',5X,F11.2,1X,F11.2,1X,F11.2,
     *1X,F11.2)
      WRITE(6,930) BB(I32,J32),BB(I32+1,J32),BB(I32,J32+1),BB(I32+1,J32+
     *1)
  930 FORMAT(' ','REGRESSION COEFFICIENTS',6X,F11.2,1X,F11.2,1X,F11.2,1X
     *,F11.2)
      WRITE(6,934) SS(I32,J32),SS(I32+1,J32),SS(I32,J32+1),SS(I32+1,J32+
     *1)
  934 FORMAT(' ','STD ERRORS REGRES. COEF.',5X,F11.2,1X,F11.2,1X,F11.2,
     *1X,F11.2)
```

```
      WRITE(6,938) AA(I32+1),BB(I32+1,J32),AA(I32+2),BB(I32,J32+1)
  938 FORMAT(' ','REGRESSION LINES',12X,'VAR(1) = ',F11.2,' +',F11.2,
     *' VAR(2)',/,29X,'VAR(2) = ',F11.2,' +',F11.2,' VAR(1)')
C
C     COMPUTING A SCALE FACTOR FOR THE COMMON PLOT OF SCATTER DIAGRAM
C     AND LEAST SQUARE REGRESSION LINES
C
      DO 940 NORDER=IORDER,MORDER
      BUX(NORDER)=AUX(NORDER)
      BUX(NORDER+JORDER)=AUX(NORDER+JORDER)
      IF(IND.EQ.0.AND.BUX(NORDER).GE.360.) BUX(NORDER)=AMOD(BUX(NORDER),
     *360.)
      IF(IND.EQ.0.AND.BUX(NORDER+JORDER).GE.360.) BUX(NORDER+JORDER)=
     *AMOD(BUX(NORDER+JORDER),360.)
  940 CONTINUE
      STEMP1=AMAX1(BUX(IORDER),BUX(IORDER+1))
      STEMP2=AMAX1(BUX(JORDER+1),BUX(JORDER+2))
      LORDER=IORDER+2
      DO 944 NORDER=LORDER,MORDER
      SCALE1=AMAX1(STEMP1,BUX(NORDER))
      STEMP1=SCALE1
      SCALE2=AMAX1(STEMP2,BUX(JORDER+NORDER))
      STEMP2=SCALE2
  944 CONTINUE
      SCALE3=AMAX1(SCALE1,SCALE2)
      WRITE(6,945) SCALE3,SCALE3
  945 FORMAT(////,28X,'.',//,1X,'VAR(1) / ',F7.2,////////////////////////
     */,28X,'.',31X,'.',/////,45X,'VAR(2) / ',F7.2,//////,11X,'+--------
     *------------------------------+',/,11X,'| SCATTER DIAGRAM AND REGR
     *ESSION LINES |',/,11X,'+---------------------------------------+')
      L31=1
C
C     PROCESSING THE DATA FOR THE PLOT OF THE SCATTER DIAGRAM
C
      DO 946 NORDER=IORDER,MORDER
      IF((BUX(NORDER+JORDER).LT.0.).OR.(BUX(NORDER).LT.0.)) GO TO 946
      X1(L31)=BUX(NORDER+JORDER)/SCALE3
      Y1(L31)=BUX(NORDER)/SCALE3
      IF(NORDER.EQ.MORDER) GO TO 946
      L31=L31+1
  946 CONTINUE
      IF(L31.EQ.1) GO TO 958
      CALL DRAW( L31,X1,Y1,1,2,LABEL,ITITL5,0.5,0.5,4,4,2,2,8,8,0,LAST)
      WRITE(6,950) LAST
  950 FORMAT('1',20X,'SCATTER DIAGRAM =',I1)
      GO TO 959
  958 WRITE(6,960)
  960 FORMAT('1',20X,'NO DATA AVAILABLE')
```

```
  959 WRITE(6,908) IER                                                STCS6920
  908 FORMAT('0',10X,'IER =',I1,/)                                    STCS6930
      CALL RLINE(ITITL5,LABEL,SCALE3)                                 STCS6940
  954 IND=1                                                           STCS6950
      GO TO 993                                                       STCS6960
  990 WRITE(6,991)                                                    STCS6970
  991 FORMAT(9X,'     NO VARIABLE(S). PLEASE CHECK DATA.',/)          STCS6980
  993 RETURN                                                          STCS6990
      END                                                             STCS7000
C                                                                     THMC0010
C     ................................................................THMC0020
C                                                                     THMC0030
C     SUBROUTINE THERMC                                               THMC0040
C                                                                     THMC0050
C     PURPOSE                                                         THMC0060
C        TWO DIMENSIONAL (Z,T) OBJECTIVE ANALYSIS OF A DIGITAL TRACE  THMC0070
C        (BEGINNING AT SEA SURFACE) TO IDENTIFY THERMOCLINES          THMC0080
C                                                                     THMC0090
C     LSAGE                                                           THMC0100
C        CALL THERMC(NORDER,ISTA,J1,NGP)                              THMC0110
C                                                                     THMC0120
C     DESCRIPTION OF PARAMETERS                                       THMC0130
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                    THMC0140
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS      THMC0150
C               SECTION                                               THMC0160
C        J1    -ACTUAL NUMBER OF DIGITAL TRACES PROCESSED FOR A GIVEN THMC0170
C               CRUISE                                                THMC0180
C        NGP   -OUTPUT NUMBER OF PAIRS (T,Z) COMPUTED FROM THE DIGITAL THMC0190
C               TRACE BY (A) A CONTINUOUS INCREMENT 'DTW' OF          THMC0200
C               TEMPERATURE BEGINNING AT SEA SURFACE AND (B) THE      THMC0210
C               DEPTHS OF THE ABOVE PAIRS ARE OBTAINED BY LINEAR      THMC0220
C               INTERPOLATION BETWEEN NEAR POINTS OF THE ORIGINAL     THMC0230
C               DIGITAL TRACE                                         THMC0240
C                                                                     THMC0250
C     REMARKS                                                         THMC0260
C        (1)THIS SUBROUTINE USES 'COMMON' LABELLED AREAS TO EXCHANGE  THMC0270
C        INFORMATION WITH THE MAIN PROGRAM.                           THMC0280
C        (2)THIS SUBROUTINE REQUIRES THE PROCESSING OF THE DIGITAL    THMC0290
C        TRACE INTO A MESH AS PERFORMED BY SUBROUTINE LINTPS.         THMC0300
C        (3)THE SETTING UP OF THE CONDITIONS TO DEFINE (A) THE TOP AND THMC0310
C        THE BOTTOM OF THE THERMOCLINE AND (B) THE FILTERING PARAMETERS THMC0320
C        REQUIRES CONSULTATION OF:                                    THMC0330
C            SALDANHA,J.M.,1971.'OBJECTIVE DIGITAL ANALYSIS OF A      THMC0340
C            TRACE'.M.S.THESIS, NAVAL POSTGRADUATE SCHOOL,            THMC0350
C            MONTEREY,CALIFORNIA                                      THMC0360
C        (4)THE SELECTED VALUES FOR THE FILTERING PARAMETERS MUST BE  THMC0370
C        INTRODUCED IN THE SUBROUTINE. THE FILTERING PARAMETERS ARE:  THMC0380
C            (A) ALPHA-MINIMUM THERMAL MAGNITUDE DEFINING THERMOCLINE THMC0390
```

```
C           (B) BETA -MINIMUM GRADIENT DEFINING THERMOCLINE                     THMC0400
C           (C) DELTA-MAXIMUM SEPARATION (VERTICAL DISTANCE)                    THMC0410
C           (D) THETA-MINIMUM GRADIENT SELECTING THERMOCLINE (WITH              THMC0420
C                     SIGMA PARAMETER)                                          THMC0430
C           (E) SIGMA-MAXIMUM SEPARATICN BETWEEN THERMOCLINES TO BE             THMC0440
C                     MERGED (WITH THETA PARAMETER)                             THMC0450
C       (5)THIS SUBROUTINE STORES THE PROCESSED INFORMATION RELATIVE            THMC0460
C       TO THE THREE SHALLOWEST THERMOCLINES.                                   THMC0470
C                                                                               THMC0480
C    FUNCTICNS REQUIRED                                                         THMC0490
C       ABS (IBM SCIENTIC LIBRARY)                                              THMC0500
C                                                                               THMC0510
C    METHOD                                                                     THMC0520
C       ITERATIVE PROCESS TO APPLY THE FINITE DIFFERENCE TECHNIQUE TO           THMC0530
C       THE MESH                                                                THMC0540
C                                                                               THMC0550
C    ...........................................................................THMC0560
C                                                                               THMC0570

      SUBROUTINE THERMC(NCRDER,ISTA,J1,NGP)                                     THMC0580
      CCMMCN/WORKA/TO(60),ZO(60),T(120),Z(120),ZAH1S(20),ZAH1I(20),             THMC0590
     *ZAH1D(20),ZAH2S(20),ZAH2I(20),ZAH2D(20),ZAH3S(20),ZAH3I(20),              THMC0600
     *ZAH3D(20),ZAIXLR(20),Z1(150),Z2(150),Z3(150),ZAUXT(15),ZAUXB(15),         THMC0610
     *ZAUXC(15),ZAUXR(15),TAUXR(15),TAUXG(15),ZANGE1(15),ZANGE(15),             THMC0620
     *TAUXT(15),TAUXB(15),ZAUBT(15),ZAUBB(15),ZAUBC(15),ZAUBR(15),              THMC0630
     *TAUBR(15),TAUBG(15),ZANBE1(15),ZANBE(15),TAUBT(15),TAUBB(15),             THMC0640
     *ZAUCT(15),ZAUCB(15),ZAUCR(15),ZANCE1(15),ZAUCC(15),TAUCT(15),             THMC0650
     *TAUCB(15),TAUCR(15),TAUCG(15),BLAN7(7)                                    THMC0660
      CCMMCN/WCRKC/ZTH1S(20,12),ZTH1I(20,12),ZTH1C(20,12),ZTH2S(20,12),         THMC0670
     *ZTH2I(20,12),ZTH2D(20,12),ZTH3S(20,12),ZTH3I(20,12),ZTH3D(20,12),         THMC0680
     *ZTHWS(20,12),ZTHWI(20,12),ZTHWD(20,12),TTHVS(20,12),TTHVI(20,12),         THMC0690
     *TTHVD(20,12),ZMIXLP(20,12),TSFC(20,12)                                    THMC0700
C                                                                               THMC0710
C    ALPHA: MINIMUM THERMAL MAGNITUDE DEFINING THERMOCLINE                      THMC0720
C                                                                               THMC0730
      ALPHA=0.3                                                                 THMC0740
C                                                                               THMC0750
C    BETA: MINIMUM GRADIENT DEFINING THERMOCLINE                                THMC0760
C                                                                               THMC0770
      BETA=0.015                                                                THMC0780
C                                                                               THMC0790
C    DELTA: MAXIMUM SEPARATION (VERTICAL DISTANCE) BETWEEN THERMOCLINESTHMC0800
C           TO BE MERGED                                                        THMC0810
      DELTA=10.                                                                 THMC0820
C                                                                               THMC0830
C    THETA: MINIMUM GRADIENT SELECTING THERMOCLINE                              THMC0840
C                                                                               THMC0850
```

```
      THETA=0.027                                                       THMC0860
C                                                                       THMC0870
C     SIGMA: MAXIMUM SEPARATION BETWEEN THERMOCLINES TO BE MERGED       THMC0880
C            TOGETHER WITH THETA PARAMETER                              THMC0890
C                                                                       THMC0900
      SIGMA=25.                                                         THMC0910
      WRITE(6,3)                                                        THMC0920
    3 FORMAT('1',///,15X,'* * *  THE FOLLCWING CUTPUT FIGURES REPRESENTSTHMC0930
     * THE RESULTS FOR THERMOCLINE ANALYSIS IN THIS XBT TRACE * * *',//)THMC0940
      I=NGP+1                                                           THMC0950
      LQ1=I-NGP                                                         THMC0960
      LQ2=I-2                                                           THMC0970
      LQ3=I-3                                                           THMC0980
      LQ7=I-4                                                           THMC0990
      M=I-5                                                             THMC1000
C                                                                       THMC1010
C     CCMPUTING 1ST DEPTH DIFFERENCES (Z1)                              THMC1020
C                                                                       THMC1030
      DC 4 N2=LQ1,LQ2                                                   THMC1040
      Z1(N2)=Z(N2+1)-Z(N2)                                              THMC1050
    4 CONTINUE                                                          THMC1060
C                                                                       THMC1070
C     CCMPUTING 2ND DEPTH DIFFERENCES                                   THMC1080
C                                                                       THMC1090
      DC 6 N3=LQ1,LQ3                                                   THMC1100
      Z2(N3)=Z1(N3+1)-Z1(N3)                                            THMC1110
    6 CCNTINUE                                                          THMC1120
C                                                                       THMC1130
C     CCMPUTING 3RD DEPTH DIFERENCES                                    THMC1140
C                                                                       THMC1150
      DC 8 N4=LQ1,LQ7                                                   THMC1160
      Z3(N4)=Z2(N4+1)-Z2(N4)                                            THMC1170
    8 CCNTINUE                                                          THMC1180
    9 N=LQ1                                                             THMC1190
C                                                                       THMC1200
C     ESTABLISHING AN ITERATIVE ANALYSIS OF XBT TRACE TO FIND           THMC1210
C     THERMOCLINES                                                      THMC1220
C                                                                       THMC1230
C     ANALYSIS FOR THE TOP CF A THERMOCLINE                             THMC1240
C                                                                       THMC1250
C                                                                       THMC1260
C     K: IS A THERMOCLINE TOP ALREADY FCUND?  IF K=0, NO. IF K=1,       THMC1270
C     YES YOU ARE LOOKING FOR THE BOTTCM OF THIS THERMOCLINE            THMC1280
C                                                                       THMC1290
      K=0                                                               THMC1300
      NR=1                                                              THMC1310
   10 IF(Z2(N).LT.0.) GO TO 12                                          THMC1320
      N=N+1                                                             THMC1330
```

```
      IF(N.LE.M) GO TO 10
      IF(N.GT.M) GO TO 41
C
C     DEFINING THE CONDITIONS TO LOCATE THE TOP OF A THERMOCLINE
C
   12 IF(Z2(N).LT.0..AND.K.EQ.0) GO TO 16
      N=N+1
      GO TO 10
   16 NTOP=N+1
      K=1
      ZAUBT(NR)=Z(NTOP)
      TAUBT(NR)=T(NTOP)
      WRITE(6,18) NR,Z(NTOP)
   18 FORMAT(///30X,'THERMOCLINE NO.',I2,/31X,'TOP AT DEPTH =',F5.1)
C
C     ANALYSIS OF THE BOTTOM OF THERMOCLINE
C
      N=NTOP
   20 IF(Z2(N).GT.0.) GO TO 22
      N=N+1
      IF(N.LE.M) GO TO 20
      IF(N.GT.M) GO TO 41
C
C     DEFINING THE CONDITIONS TO LOCATE THE BOTTOM OF A THERMOCLINE
C
   22 IF((Z2(N).GT.0..AND.K.EQ.1).AND.Z2(N+1).GE.0.) GO TO 26
   24 N=N+1
      GO TO 20
   26 NBOT=N+1
      TAUBB(NR)=T(NBOT)
      ZAUBB(NR)=Z(NBOT)
      WRITE(6,28) Z(NBOT)
   28 FORMAT(' ',30X,'BOTTOM AT DEPTH =',F5.1)
C
C     COMPUTATION OF THERMOCLINE PARAMETERS
C
C     VERTICAL SPACEMENT OF THERMOCLINE
C
      ZANBE(NR)=ABS(ZAUBB(NR)-ZAUBT(NR))
      ZAUBR(NR)=ZANBE(NR)
      WRITE(6,30) ZANBE(NR)
   30 FORMAT(' ',30X,'VERTICAL SPACEMENT BETWEEN TOP AND BOTTOM =',F5.1)
C
C     HALF SPACEMENT
C
      ZANBE1(NR)=0.5*ZANBE(NR)
C
C     DEPTH OF THERMOCLINE MEAN (DEFINED BY LINEAR AVERAGE BETWEEN TOP
```

```
C        AND BOTTCM)                                                    THMC1820
C                                                                       THMC1830
         ZAUBC(NR)=Z(NTOP)+ZANBE1(NR)                                   THMC1840
C                                                                       THMC1850
C        TEMPERATURE VARIATION OF THERMOCLINE                           THMC1860
C                                                                       THMC1870
         TAUBR(NR)=ABS(TAUBT(NR)-TAUBB(NR))                             THMC1880
         WRITE(6,32) TAUBR(NR)                                          THMC1890
   32    FCRMAT(' ',30X,'TEMPERATURE VARIATION =',F5.1)                 THMC1900
         WRITE(6,34) ZAUBC(NR)                                          THMC1910
   34    FCRMAT(' ',30X,'DEPTH OF THERMOCLINE MEAN =',F5.1)             THMC1920
C                                                                       THMC1930
C        TAUXG:THERMOCLINE GRADIENT (DEGREES CENT./METERS)              THMC1940
C                                                                       THMC1950
         TAUBG(NR)=TAUBR(NR)/ZANBE(NR)                                  THMC1960
         WRITE(6,40) TAUBG(NR)                                          THMC1970
   40    FCRMAT(' ',30X,'THERMCCLINE GRADIENT =',F10.6)                 THMC1980
         NR=NR+1                                                        THMC1990
         K=0                                                            THMC2000
         N=NBOT                                                         THMC2010
         IF(N.LE.M) GO TO 10                                            THMC2020
   41    NR=NR-1                                                        THMC2030
         IF(NR.GT.1) GO TO 42                                           THMC2040
         IF(NR.EQ.0) JCOUNT=1                                           THMC2050
         IF(NR.EQ.1) JCOUNT=2                                           THMC2060
C                                                                       THMC2070
C        PROCESSING THE FILTERING OF THE IDENTIFIED THERMOCLINES        THMC2080
C                                                                       THMC2090
         GO TO 46                                                       THMC2100
   42    NS=1                                                           THMC2110
         NZ=0                                                           THMC2120
         NW=0                                                           THMC2130
C                                                                       THMC2140
C        SELECTICN OF THERMOCLINES BASED UPON (A) VERTICAL SEPARATION OR THMC2150
C        (B) VERTICAL SEPARATION AND MINIMUM GRADIENT                   THMC2160
C                                                                       THMC2170
         DO 44 I10=1,NR                                                 THMC2180
         IF(I10.EQ.NR) GO TO 43                                         THMC2190
         DELTZ=ABS(ZAUBT(I10+1)-ZAUBB(I10))                             THMC2200
         IF(DELTZ.LE.DELTA) GO TO 39                                    THMC2210
         IF(DELTZ.LE.SIGMA) GO TO 48                                    THMC2220
         GO TO 43                                                       THMC2230
   48    IF((NZ.EQ.0).AND.((TAUBG(I10).GE.THETA).AND.(TAUBG(I10+1).GE.TAUBGTHMC2240
        *(I10)))) GO TO 39                                              THMC2250
         IF((NZ.EQ.1).AND.((TAUCG(NS).GE.THETA).AND.(TAUBG(I10+1).GE.TAUCG(THMC2260
        *NS)))) GO TO 39                                                THMC2270
         GO TO 43                                                       THMC2280
   39    IF(NZ.EQ.0.AND.I10.GT.1) NS=NS+1                               THMC2290
```

```
      IF(NZ.EQ.0) ZAUCT(NS)=ZAUBT(I10)
      IF(NZ.EQ.0) TAUCT(NS)=TAUBT(I10)
      IF(I10.EC.(NR-1)) NW=1
      ZAUCB(NS)=ZAUBB(I10+1)
      ZAUCR(NS)=ABS(ZAUCB(NS)-ZAUCT(NS))
      ZANCE1(NS)=0.5*ZAUCR(NS)
      ZAUCC(NS)=ZAUCT(NS)+ZANCE1(NS)
      TAUCB(NS)=TAUBB(I10+1)
      TAUCR(NS)=ABS(TAUCT(NS)-TAUCB(NS))
      TAUCG(NS)=TAUCR(NS)/ZAUCR(NS)
      WRITE(6,18) NS,ZAUCT(NS)
      WRITE(6,28) ZAUCB(NS)
      WRITE(6,30) ZAUCR(NS)
      WRITE(6,32) TAUCR(NS)
      WRITE(6,34) ZAUCC(NS)
      WRITE(6,40) TAUCG(NS)
      NZ=1
      GC TO 44
   43 IF(NZ.EQ.1.AND.NW.EQ.1) GO TO 44
      IF(NZ.EQ.1.AND.I10.EQ.NR) GO TO 50
      IF(NZ.EQ.1) GO TO 47
   50 IF(I10.GT.1) NS=NS+1
      I11=I10
      ZAUCT(NS)=ZAUBT(I11)
      ZAUCB(NS)=ZAUBB(I11)
      ZAUCR(NS)=ABS(ZAUCB(NS)-ZAUCT(NS))
      ZANCE1(NS)=0.5*ZAUCR(NS)
      ZAUCC(NS)=ZAUCT(NS)+ZANCE1(NS)
      TAUCT(NS)=TAUBT(I11)
      TAUCB(NS)=TAUBB(I11)
      TAUCR(NS)=ABS(TAUCT(NS)-TAUCB(NS))
      TAUCG(NS)=TAUBG(I11)
      WRITE(6,18) NS,ZAUCT(NS)
      WRITE(6,28) ZAUCB(NS)
      WRITE(6,30) ZAUCR(NS)
      WRITE(6,32) TAUCR(NS)
      WRITE(6,34) ZAUCC(NS)
      WRITE(6,40) TAUCG(NS)
   47 NZ=0
   44 CONTINUE
      JCOUNT=1
C
C     SELECTION OF FILTERED THERMOCLINES BASED ON (A) MINIMUM THERMAL
C     MAGNITUDE OR (B) MINIMUM GRADIENT
C
      DC 45 I12=1,NS
      IF(TAUCR(I12).LT.ALPHA) GO TO 45
      IF(TAUCG(I12).LT.BETA) GO TO 45
```

```
      ZAUXT(JCCUNT)=ZAUCT(I12)                                          THMC2780
      ZAUXB(JCCUNT)=ZAUCB(I12)                                          THMC2790
      ZAUXR(JCCUNT)=ZAUCR(I12)                                          THMC2800
      ZANGE1(JCOUNT)=ZANCE1(I12)                                        THMC2810
      ZAUXC(JCCUNT)=ZAUCC(I12)                                          THMC2820
      TAUXT(JCCUNT)=TAUCT(I12)                                          THMC2830
      TAUXB(JCOUNT)=TAUCB(I12)                                          THMC2840
      TAUXR(JCCUNT)=TAUCR(I12)                                          THMC2850
      TAUXG(JCOUNT)=TAUCG(I12)                                          THMC2860
      WRITE(6,18) JCOUNT,ZAUXT(JCOUNT)                                  THMC2870
      WRITE(6,28) ZAUXB(JCOUNT)                                         THMC2880
      WRITE(6,30) ZAUXR(JCCUNT)                                         THMC2890
      WRITE(6,32) TAUXR(JCOUNT)                                         THMC2900
      WRITE(6,34) ZAUXC(JCOUNT)                                         THMC2910
      WRITE(6,40) TAUXG(JCOUNT)                                         THMC2920
      JCOUNT=JCOUNT+1                                                   THMC2930
   45 CONTINUE                                                          THMC2940
   46 JTCUNT=JCCUNT-1                                                   THMC2950
      WRITE(6,55) JTOUNT                                                THMC2960
   55 FORMAT(//,30X,'NO. OF THERMOCLINES =',I2,////)                    THMC2970
C                                                                       THMC2980
C     END OF THERMOCLINE ANALYSIS                                       THMC2990
C                                                                       THMC3000
C                                                                       THMC3010
C     STORAGE OF THE THERMOCLINE PROCESSED DATA                         THMC3020
C                                                                       THMC3030
C          (1) NO THERMOCLINE  (NEGATIVE ARTIFICIAL VALUES)             THMC3040
C          (2) 1 THERMOCLINE  (SHALLOW)                                 THMC3050
C          (3) 2 THERMOCLINES  (SHALLOW+INTERMEDIATE)                   THMC3060
C          (4) 3 THERMOCLINES  (SHALLOW+INTERMEDIATE+DEEP)              THMC3070
C                                                                       THMC3080
      JCOUNT=JTOUNT+1                                                   THMC3090
      IF(JCOUNT.GT.1) GO TO 60                                          THMC3100
      ZAH1S(J1)=-5.                                                     THMC3110
      ZAH1I(J1)=-5.                                                     THMC3120
      ZAH1D(J1)=-5.                                                     THMC3130
      ZAH2S(J1)=-5.                                                     THMC3140
      ZAH2I(J1)=-5.                                                     THMC3150
      ZAH2D(J1)=-5.                                                     THMC3160
      ZAH3S(J1)=-5.                                                     THMC3170
      ZAH3I(J1)=-5.                                                     THMC3180
      ZAH3D(J1)=-5.                                                     THMC3190
      ZTH1S(NORDER,ISTA)=-5.                                            THMC3200
      ZTH1I(NORDER,ISTA)=-5.                                            THMC3210
      ZTH1D(NORDER,ISTA)=-5.                                            THMC3220
      ZTH2S(NORDER,ISTA)=-5.                                            THMC3230
      ZTH2I(NORDER,ISTA)=-5.                                            THMC3240
      ZTH2D(NORDER,ISTA)=-5.                                            THMC3250
```

```
      ZTH3S(NORDER,ISTA)=-5.                                            THMC3260
      ZTH3I(NORDER,ISTA)=-5.                                            THMC3270
      ZTH3D(NORDER,ISTA)=-5.                                            THMC3280
      ZTHWS(NORDER,ISTA)=-5.                                            THMC3290
      ZTHWI(NORDER,ISTA)=-5.                                            THMC3300
      ZTHWD(NORDER,ISTA)=-5.                                            THMC3310
      TTHVS(NORDER,ISTA)=-5.                                            THMC3320
      TTHVI(NORDER,ISTA)=-5.                                            THMC3330
      TTHVD(NORDER,ISTA)=-5.                                            THMC3340
      ZMIXLR(NORDER,ISTA)=-5.                                           THMC3350
      ZAIXLR(J1)=-5.                                                    THMC3360
      GO TO 100                                                         THMC3370
   60 IF(JCOUNT.GT.2) GO TO 65                                          THMC3380
      JCOUNT=JCOUNT-1                                                   THMC3390
      ZAH1S(J1)=ZAUXT(JCOUNT)                                           THMC3400
      ZAH1I(J1)=-5.                                                     THMC3410
      ZAH1D(J1)=-5.                                                     THMC3420
      ZAH2S(J1)=ZAUXC(JCOUNT)                                           THMC3430
      ZAH2I(J1)=-5.                                                     THMC3440
      ZAH2D(J1)=-5.                                                     THMC3450
      ZAH3S(J1)=ZAUXB(JCOUNT)                                           THMC3460
      ZAH3I(J1)=-5.                                                     THMC3470
      ZAH3D(J1)=-5.                                                     THMC3480
      ZAIXLR(J1)=ZAUXT(JCOUNT)                                          THMC3490
      ZTH1S(NORDER,ISTA)=ZAUXT(JCOUNT)                                  THMC3500
      ZTH1I(NORDER,ISTA)=-5.                                            THMC3510
      ZTH1D(NORDER,ISTA)=-5.                                            THMC3520
      ZTH2S(NORDER,ISTA)=ZAUXC(JCOUNT)                                  THMC3530
      ZTH2I(NORDER,ISTA)=-5.                                            THMC3540
      ZTH2D(NORDER,ISTA)=-5.                                            THMC3550
      ZTH3S(NORDER,ISTA)=ZAUXB(JCOUNT)                                  THMC3560
      ZTH3I(NORDER,ISTA)=-5.                                            THMC3570
      ZTH3D(NORDER,ISTA)=-5.                                            THMC3580
      ZTHWS(NORDER,ISTA)=ZAUXR(JCOUNT)                                  THMC3590
      ZTHWI(NORDER,ISTA)=-5.                                            THMC3600
      ZTHWD(NORDER,ISTA)=-5.                                            THMC3610
      TTHVS(NORDER,ISTA)=TAUXR(JCOUNT)                                  THMC3620
      TTHVI(NORDER,ISTA)=-5.                                            THMC3630
      TTHVD(NORDER,ISTA)=-5.                                            THMC3640
      ZMIXLR(NORDER,ISTA)=ZAUXT(JCOUNT)                                 THMC3650
      GO TO 100                                                         THMC3660
   65 IF(JCOUNT.GT.3) GO TO 70                                          THMC3670
      JCOUNT=JCOUNT-2                                                   THMC3680
      ZAH1S(J1)=ZAUXT(JCOUNT)                                           THMC3690
      ZAH1I(J1)=ZAUXT(JCOUNT+1)                                         THMC3700
      ZAH1D(J1)=-5.                                                     THMC3710
      ZAH2S(J1)=ZAUXC(JCOUNT)                                           THMC3720
      ZAH2I(J1)=ZAUXC(JCOUNT+1)                                         THMC3730
```

```
      ZAH2D(J1)=-5.                                                    THMC3740
      ZAH3S(J1)=ZAUXB(JCOUNT)                                          THMC3750
      ZAH3I(J1)=ZAUXB(JCOUNT+1)                                        THMC3760
      ZAH3D(J1)=-5.                                                    THMC3770
      ZAIXLR(J1)=ZAUXT(JCOUNT)                                         THMC3780
      ZTH1S(NORDER,ISTA)=ZAUXT(JCOUNT)                                 THMC3790
      ZTH1I(NORDER,ISTA)=ZAUXT(JCOUNT+1)                               THMC3800
      ZTH1D(NORDER,ISTA)=-5.                                           THMC3810
      ZTH2S(NORDER,ISTA)=ZAUXC(JCOUNT)                                 THMC3820
      ZTH2I(NORDER,ISTA)=ZAUXC(JCOUNT+1)                               THMC3830
      ZTH2D(NORDER,ISTA)=-5.                                           THMC3840
      ZTH3S(NORDER,ISTA)=ZAUXB(JCOUNT)                                 THMC3850
      ZTH3I(NORDER,ISTA)=ZAUXB(JCOUNT+1)                               THMC3860
      ZTH3D(NORDER,ISTA)=-5.                                           THMC3870
      ZTHWS(NORDER,ISTA)=ZAUXR(JCOUNT)                                 THMC3880
      ZTHWI(NORDER,ISTA)=ZAUXR(JCOUNT+1)                               THMC3890
      ZTHWD(NORDER,ISTA)=-5.                                           THMC3900
      TTHVS(NORDER,ISTA)=TAUXR(JCOUNT)                                 THMC3910
      TTHVI(NORDER,ISTA)=TAUXR(JCOUNT+1)                               THMC3920
      TTHVD(NORDER,ISTA)=-5.                                           THMC3930
      ZMIXLR(NORDER,ISTA)=ZAUXT(JCOUNT)                                THMC3940
      GO TO 100                                                        THMC3950
   70 JCOUNT=1                                                         THMC3960
      ZAH1S(J1)=ZAUXT(JCOUNT)                                          THMC3970
      ZAH1I(J1)=ZAUXT(JCOUNT+1)                                        THMC3980
      ZAH1D(J1)=ZAUXT(JCOUNT+2)                                        THMC3990
      ZAH2S(J1)=ZAUXC(JCOUNT)                                          THMC4000
      ZAH2I(J1)=ZAUXC(JCOUNT+1)                                        THMC4010
      ZAH2D(J1)=ZAUXC(JCOUNT+2)                                        THMC4020
      ZAH3S(J1)=ZAUXB(JCOUNT)                                          THMC4030
      ZAH3I(J1)=ZAUXB(JCOUNT+1)                                        THMC4040
      ZAH3D(J1)=ZAUXB(JCOUNT+2)                                        THMC4050
      ZAIXLR(J1)=ZAUXT(JCOUNT)                                         THMC4060
      ZTH1S(NORDER,ISTA)=ZAUXT(JCOUNT)                                 THMC4070
      ZTH1I(NORDER,ISTA)=ZAUXT(JCOUNT+1)                               THMC4080
      ZTH1D(NORDER,ISTA)=ZAUXT(JCOUNT+2)                               THMC4090
      ZTH2S(NORDER,ISTA)=ZAUXC(JCOUNT)                                 THMC4100
      ZTH2I(NORDER,ISTA)=ZAUXC(JCOUNT+1)                               THMC4110
      ZTH2D(NORDER,ISTA)=ZAUXC(JCOUNT+2)                               THMC4120
      ZTH3S(NORDER,ISTA)=ZAUXB(JCOUNT)                                 THMC4130
      ZTH3I(NORDER,ISTA)=ZAUXB(JCOUNT+1)                               THMC4140
      ZTH3D(NORDER,ISTA)=ZAUXB(JCOUNT+2)                               THMC4150
      ZTHWS(NORDER,ISTA)=ZAUXR(JCOUNT)                                 THMC4160
      ZTHWI(NORDER,ISTA)=ZAUXR(JCOUNT+1)                               THMC4170
      ZTHWD(NORDER,ISTA)=ZAUXR(JCOUNT+2)                               THMC4180
      TTHVS(NORDER,ISTA)=TAUXR(JCOUNT)                                 THMC4190
      TTHVI(NORDER,ISTA)=TAUXR(JCOUNT+1)                               THMC4200
      TTHVD(NORDER,ISTA)=TAUXR(JCOUNT+2)                               THMC4210
```

```
      ZMIXLR(NORDER,ISTA)=ZAUXT(JCOUNT)
  100 WRITE(6,9999) ZAH1S(J1),ZAH1I(J1),ZAH1D(J1),ZAH2S(J1),ZAH2I(J1),
     *ZAH2D(J1),ZAH3S(J1),ZAH3I(J1),ZAH3D(J1),ZAIXLR(J1)
 9999 FORMAT(' ',5X,10F6.2)
      WRITE(6,9998) ZTH1S(NORDER,ISTA),ZTH1I(NORDER,ISTA),
     *ZTH1D(NORDER,ISTA),ZTH2S(NORDER,ISTA),ZTH2I(NORDER,ISTA),
     *ZTH2D(NORDER,ISTA),ZTH3S(NORDER,ISTA),ZTH3I(NORDER,ISTA),
     *ZTH3D(NORDER,ISTA),ZTHWS(NORDER,ISTA),ZTHWI(NORDER,ISTA),
     *ZTHWD(NORDER,ISTA),TTHVS(NORDER,ISTA),TTHVI(NORDER,ISTA),
     *TTHVD(NORDER,ISTA),ZMIXLR(NORDER,ISTA)
 9998 FORMAT(' ',5X,10F6.2,/6X,6F6.2)
      WRITE(6,9997)
 9997 FORMAT(///,40X,'*    END OF THE THERMOCLINE ANALYSIS    *')
      RETURN
      END
C
C     ..................................................................
C
C     SUBROUTINE TINC1
C
C     PURPOSE
C        TO INCREMENT (MESH INTERVAL) A GIVEN ABCISSA (TEMPERATURE)
C        LOCATED BETWEEN TWO KNOWN POSITIONS OF THE DIGITAL TRACE. THE
C        RESULTANT ABCISSA REMAINS IN BETWEEN THE TWO GIVEN POSITIONS
C
C     USAGE
C        CALL TINC1(AO1,AO2,AW,DAW,M81,AW3)
C
C     DESCRIPTION OF PARAMETERS
C        AO1    -INPUT VALUE CONTAINING THE ABCISSA (TEMPERATURE) OF
C                THE FIRST OF TWO GIVEN POSITIONS
C        AO2    -INPUT VALUE CONTAINING THE ABCISSA OF THE LAST OF
C                TWO GIVEN POSITIONS
C        DAW    -INPUT VALUE CONTAINING THE INCREMENT OF THE MESH
C        M81    -INPUT VALUE CONTAINING THE ACTUAL NUMBER OF POINTS
C                GENERATED FOR THE MESH
C        AW3    -OUTPUT VALUE CONTAINING THE ACTUAL ABCISSA (TEMP.)
C                RESULTANT FROM THE INCREMENT OF A GIVEN ABCISSA
C
C     SUBROUTINES REQUIRED
C        NONE
C
C     METHOD
C        ACCORDING TO THE ORIENTATION OF THE TREND (DEFINED BY TWO
C        KNOWN DIGITAL POSITIONS) THE VALUE OF A GIVEN ABCISSA,
C        LOCATED IN BETWEEN THE ABOVE POSITIONS, IS INCREASED OR
C        DECREASED BY THE MESH INTERVAL
C
```

```
C     ..................................................................
C
      SUBROUTINE TINC1(AO1,AO2,AW,DAW,M81;AW3)
C
C     DA -DIFFERENCE BETWEEN CONSECUTIVE OBSERVED TEMPERATURES, AO1
C         AND AO2
C
      DA=AO2-AO1
C
C     COMPARE  DA WITH ZERO. ZERO IS DEFINED BY THE OPEN INTERVAL
C     BETWEEN -0.005 AND 0.005
C
      IF(DA.GE.0.005.OR.DA.LE.(-0.005)) GO TO 9050
      AW3=AO2
      GO TO 9060
 9050 IF(DA.GT.0.0) GO TO 9054
      AW3=AW-DAW
      GO TO 9060
 9054 AW3=AW+DAW
 9060 RETURN
      END
C
C     ..................................................................
C
C     SUBROUTINE TINC2
C
C     PURPOSE
C        TO INCREMENT (MESH INTERVAL) A GIVEN ABCISSA (TEMPERATURE)
C        LOCATED BETWEEN THE FIRST TWO OF THREE KNOWN POSITIONS OF THE
C        DIGITAL TRACE. THE RESULTANT ABCISSA IS FOUND BETWEEN THE
C        SECOND AND THIRD GIVEN DIGITAL POINTS
C
C     USAGE
C        CALL TINC2(AO2,AO3,DA2,DAW,AW2)
C     DESCRIPTION OF PARAMETERS
C        AO2    -INPUT VALUE CONTAINING THE ABCISSA (TEMPERATURE) OF
C                THE FIRST OF TWO GIVEN POSITIONS
C        AO3    -INPUT VALUE CONTAINING THE ABCISSA OF THE LAST OF
C                TWO GIVEN POSITIONS
C        DA2    -INPUT CONTAINING THE ABSOLUTE VALUE OF THE DIFFERENCE
C                BETWEEN THE INITIAL GIVEN ABCISSA (LOCATED BETWEEN THE
C                FIRST TWO OF THREE KNOWN POSITIONS OF THE DIGITAL
C                TRACE) AND THE ABCISSA OF THE THIRD GIVEN DIGITAL
C                POSITION
C        DAW    -INPUT VALUE CONTAINING THE INCREMENT OF THE MESH
C        AW2    -OUTPUT VALUE CONTAINING THE ABCISSA (TEMPERATURE)
```

```
C           RESULTANT FROM THE INCREMENT OF A GIVEN ABCISSA             TNC20260
C                                                                       TNC20270
C     SUBROUTINES REQUIRED                                              TNC20280
C        NONE                                                           TNC20290
C                                                                       TNC20300
C     METHOD                                                            TNC20310
C        ACCORDING TO THE ORIENTATION OF THE TREND (DEFINED BY THREE    TNC20320
C        KNOWN DIGITAL POSITIONS) THE VALUE OF A GIVEN ABCISSA,         TNC20330
C        LOCATED IN BETWEEN THE FIRST TWO POSITIONS, IS INCREASED OR    TNC20340
C        DECREASED BY THE MESH INTERVAL AND THE FINAL ABCISSA IS        TNC20350
C        LOCATED IN BETWEEN THE TWO LAST DIGITAL POSITIONS              TNC20360
C                                                                       TNC20370
C     ..................................................................TNC20380
C                                                                       TNC20390


      SUBROUTINE TINC2(AO2,AO3,DA2,DAW,AW2)                             TNC20400
      DA5=AO3-AO2                                                       TNC20410
C                                                                       TNC20420
C     COMPARE  DA5 WITH ZERO. ZERO IS DEFINED BY THE OPEN INTERVAL      TNC20430
C     BETWEEN -0.005 AND 0.005                                          TNC20440
C                                                                       TNC20450
      IF(DA5.GE.0.005.OR.DA5.LE.(-0.005)) GO TO 9070                    TNC20460
      GO TO 9080                                                        TNC20470
 9070 IF(DA5.GT.0.0) GO TO 9074                                         TNC20480
      AW2=AO2-(DAW-DA2)                                                 TNC20490
      GO TO 9080                                                        TNC20500
 9074 AW2=AO2+(DAW-DA2)                                                 TNC20510
 9080 RETURN                                                            TNC20520
      END                                                               TNC20530
C                                                                       VCTR0010
C     ..................................................................VCTR0020
C                                                                       VCTR0030
C     SUBROUTINE VECTOR                                                 VCTR0040
C                                                                       VCTR0050
C     PURPOSE                                                           VCTR0060
C        TO COMPUTE THE SINE AND THE COSINE OF A GIVEN ANGLE            VCTR0070
C                                                                       VCTR0080
C     USAGE                                                             VCTR0090
C        CALL VECTOR(M77,A,B)                                           VCTR0100
C                                                                       VCTR0110
C     DESCRIPTION OF PARAMETERS                                         VCTR0120
C        M77    -INPUT CONTAINING THE VALUE OF THE ANGLE (DEGREES)      VCTR0130
C        A      -OUTPUT CONTAINING THE VALUE OF THE SIN(M77)            VCTR0140
C        B      -OUTPUT CONTAINING THE VALUE OF THE COS(M77)            VCTR0150
C                                                                       VCTR0160
C     FUNCTIONS REQUIRED                                                VCTR0170
C        SIN (IBM SCIENTIFIC LIBRARY)                                   VCTR0180
```

```
C        COS (IBM SCIENTIFIC LIBRARY)                                    VCTR0190
C                                                                        VCTR0200
C     ...................................................................VCTR0210
C                                                                        VCTR0220
      SUBROUTINE VECTOR(M77,A,B)                                         VCTR0230
      PI=3.14159265                                                      VCTR0240
      PI1=PI/2.                                                          VCTR0250
      PI3=3.*PI1                                                         VCTR0260
    1 IF(M77.GE.360) M77=M77-360                                         VCTR0270
C                                                                        VCTR0280
C     ASSIGNING A QUADRANT                                               VCTR0290
C                                                                        VCTR0300
      IF(M77.GE.0.AND.M77.LT.90) GO TO 4                                 VCTR0310
      IF(M77.GE.90.AND.M77.LT.180) GO TO 5                               VCTR0320
      IF(M77.GE.180.AND.M77.LT.270) GO TO 6                              VCTR0330
      IF(M77.GE.270.AND.M77.LT.360) GO TO 7                              VCTR0340
      IF(M77.GE.360) GO TO 1                                             VCTR0350
      GO TO 9                                                            VCTR0360
C                                                                        VCTR0370
C     COMPUTING THE SINE AND COSINE                                      VCTR0380
C                                                                        VCTR0390
    4 A77=M77*PI/180.                                                    VCTR0400
      A=SIN(A77)                                                         VCTR0410
      B=COS(A77)                                                         VCTR0420
      GO TO 10                                                           VCTR0430
    5 A77=M77*PI/180.                                                    VCTR0440
      A=COS(A77-PI1)                                                     VCTR0450
      B=-SIN(A77-PI1)                                                    VCTR0460
      GO TO 10                                                           VCTR0470
    6 A77=M77*PI/180.                                                    VCTR0480
      A=-SIN(A77-PI)                                                     VCTR0490
      B=-COS(A77-PI)                                                     VCTR0500
      GO TO 10                                                           VCTR0510
    7 A77=M77*PI/180.                                                    VCTR0520
      A=-COS(A77-PI3)                                                    VCTR0530
      B=SIN(A77-PI3)                                                     VCTR0540
      GO TO 10                                                           VCTR0550
    9 A=0.                                                               VCTR0560
      B=0.                                                               VCTR0570
   10 RETURN                                                             VCTR0580
      END                                                                VCTR0590
C                                                                        WOO 0010
C     ...................................................................WOO 0020
C     SUBROUTINE WINDOO                                                  WOO 0030
C                                                                        WOO 0040
C     PURPOSE                                                            WOO 0050
```

```
C      TC PROCESS THE WIND VELOCITY MEASURED AT THE SAME DAY OF THE    WOO 0060
C      DIGITAL TRACE AND AT THE SAME GEOGRAPHICAL POSITION. TO PLOT    WOO 0070
C      THE WIND VECTOR AND ITS COMPONENTS (ALONG AND PERPENDICULAR     WOO 0080
C      TO THE DIRECTION OF THE NEAR CCAST)                             WOO 0090
C                                                                      WOO 0100
C   USAGE                                                              WOO 0110
C      CALL WINDOO(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,       WOO 0120
C      KAMMA1,N41)                                                     WOO 0130
C                                                                      WOO 0140
C   DESCRIPTICN OF PARAMETERS                                          WOO 0150
C      NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                       WOO 0160
C      ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS         WOO 0170
C             SECTION                                                  WOO 0180
C      XGRID -INPUT VALUE CONTAINING THE LOCATICN OF THE STATION       WOO 0190
C             (ALONG THE CROSS SECTION OR TIME AXIS) WHERE THE WIND    WOO 0200
C             DATA IS REPORTED TO                                      WOO 0210
C      YGRID -INPUT CONTAINING THE VALUE OF THE ORDINATE TO LOCATE     WOO 0220
C             THE WIND VECTORS IN THE PLOT                             WOO 0230
C      KKK   -INPUT VALUE CONTROLLING THE PLOT OF THE WIND VECTOR      WOO 0240
C             (IF K.NE.0 : 'NO PLOT')                                  WOO 0250
C      N40   -INPUT ACTUAL NUMBER OF WIND PLOTS                        WOO 0260
C      LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE            WOO 0270
C      ITITL4-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR     WOO 0280
C             PLOT OF WIND DATA ALONG A CROSS SECTION OR TIME AXIS     WOO 0290
C      KAMMA1-INPUT VALUE CONTAINING THE DIRECTION OF THE NEAR         WOO 0300
C             CCASTLINE                                                WOO 0310
C      N41   -OUTPUT ACTUAL NUMBER OF WIND PLOTS                       WOO 0320
C                                                                      WOO 0330
C   REMARKS                                                            WOO 0340
C      (1)THIS SUBROUTINE USES A 'CCMMON' LABELLED AREA TO RECEIVE     WOO 0350
C      THE REQUIRED DATA FROM THE MAIN PROGRAM AND TO GIVE BACK        WOO 0360
C      PROCESSED INFORMATICN.                                          WOO 0370
C      (2)THE WIND VECTOR AND ITS CCMPCNENTS ARE PLOTTED IN A TWO      WOO 0380
C      DIMENSICNAL DIAGRAM: (A)THE ABCISSA OR THE POSITION ALONG THE   WOO 0390
C      CRUISE WHERE THE WIND IS REPCRTED TO AND (B)THE CRDINATE OR     WOO 0400
C      THE RELATIVE TIME (IN DAYS) THAT ELAPSES FROM THE DAY WHEN      WOO 0410
C      THE WIND WAS MEASURED UNTIL THE DAY THE DIGITAL TRACE IS        WOO 0420
C      OBTAINED.                                                       WOO 0430
C                                                                      WOO 0440
C   SUBRCUTINES REQUIRED                                               WOO 0450
C      VECTOR                                                          WOO 0460
C      DRAW (IBM SCIENTIFIC LIBRARY)                                   WOO 0470
C                                                                      WOO 0480
C   METHOD                                                             WOO 0490
C      THE CCMPCNENTS CF THE WIND VECTOR ARE COMPUTED USING            WOO 0500
C      TRIGONCMETRIC RELATIONSHIPS                                     WOO 0510
C                                                                      WOO 0520
C   ..............................................................    WOO 0530
```

```
C
      SUBROUTINE WINDOC(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
     *KAMMA1,N41)
      REAL*8 ITITL4(12)
      DIMENSION XAUX1(2),YAUX1(2),XAUX3(2),YAUX3(2),XAUX4(2),YAUX4(2)
      COMMON/WORKD/ROO(20,12),KANGOO(20,12),RO1(20,12),KANGO1(20,12),
     *RO2(20,12),KANGO2(20,12),RO3(20,12),KANGO3(20,12),RO5(20,12),
     *KANGO5(20,12),R10(20,12),KANG10(20,12),ROO1(20,12),ROO2(20,12),
     *RO11(20,12),RO12(20,12),RO21(20,12),RO22(20,12),RO31(20,12),
     *RO32(20,12),RO51(20,12),RO52(20,12),R101(20,12),R102(20,12)
C
C     COMPUTATION OF COORDINATES DEFINING THE LOCAL WIND VECTOR
C
      IF((ROO(NORDER,ISTA).LT.0.).OR.(KANGOO(NORDER,ISTA).LT.0)) GO TO
     *300
      IF(KKK.GT.0) GO TO 3
      KANG1=KANGOO(NORDER,ISTA)
      CALL VECTOR(KANG1,ALPH1,BET1)
      DELTX1=ROO(NORDER,ISTA)*ALPH1/45.
      DELTY1=ROO(NORDER,ISTA)*BET1/45.
      XAUX1(1)=XGRID
      XAUX1(2)=XAUX1(1)+DELTX1
      YAUX1(1)=YGRID
      YAUX1(2)=YAUX1(1)+DELTY1
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX1,YAUX1,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
C
C     FINDING THE COMPONENTS OF THE LOCAL WIND VECTOR ALONG TWO PRE-
C     -DEFINED DIRECTIONS (I.E., ALONG THE COAST AND PERPENDICULAR TO
C     THIS DIRECTION)
C
C     KAMMA: ANGULAR DIFFERENCE BETWEEN THE WIND VECTOR (KANG) AND THE
C     DIRECTION ALONG THE COAST (KAMMA1)
C
      IF(KAMMA1.LT.0) GO TO 300
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
    3 KAMMA=KANGOO(NORDER,ISTA)-KAMMA1
      KAMMA5=IABS(KAMMA)
      IF(KAMMA5.LE.90) GO TO 4
      IF(KAMMA.LT.(-90)) GO TO 6
      KAMMA1=KAMMA1+180
      GO TO 3
```

```
    6 KAMMA1=KAMMA1-180
      GC TO 3
    4 CALL VECTOR(KAMMA5,ALPH2,BET2)
C
C     RXX1: ALONG THE COAST COMPONENT (INTENSITY) OF LOCAL WIND
C     RXX2:PERPENDICULAR TO COAST CCMPONENT (INTENSITY) OF LOCAL WIND
C
      R001(NORDER,ISTA)=R00(NORDER,ISTA)*BET2
      R002(NORDER,ISTA)=R00(NORDER,ISTA)*ALPH2
      IF(KAMMA1.LT.0) KAMMA1=KAMMA1+360
      IHETA1=KAMMA1
      IF(IHETA1.GE.360) IHETA1=IHETA1-360
      IF(KAMMA5.EQ.0) GO TO 11
      IF(KKK.GT.0) GO TO 8
      CALL VECTOR(IHETA1,ALPH3,BET3)
      DELTX3=R001(NORDER,ISTA)*ALPH3/45.
      DELTY3=R001(NORDER,ISTA)*BET3/45.
      XAUX3(1)=XGRID
      XAUX3(2)=XAUX3(1)+DELTX3
      YAUX3(1)=YGRID
      YAUX3(2)=YAUX3(1)+DELTY3
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX3,YAUX3,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    8 IF(KANGOO(NORDER,ISTA).LT.KAMMA1) IHETA2=KAMMA1-90
      IF(KANGOO(NORDER,ISTA).GT.KAMMA1) IHETA2=KAMMA1+90
      IF(IHETA2.GE.360) IHETA2=IHETA2-360
      IF(IHETA2.LT.0) IHETA2=IHETA2+360
      KAMMA7=IABS(KANGOO(NORDER,ISTA)-IHETA2)
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
      IF(KANGOO(NORDER,ISTA).GE.360) KANGOO(NORDER,ISTA)=KANGOO(NORDER,
     *ISTA)-360
      IF(KAMMA7.EQ.0) GO TO 14
      IF(KKK.GT.0) GO TO 10
      CALL VECTOR(IHETA2,ALPH4,BET4)
      DELTX4=R002(NORDER,ISTA)*ALPH4/45.
      DELTY4=R002(NORDER,ISTA)*BET4/45.
      XAUX4(1)=XGRID
      XAUX4(2)=XAUX4(1)+DELTX4
      YAUX4(1)=YGRID
      YAUX4(2)=YAUX4(1)+DELTY4
      WRITE(6,511) XAUX1(1),XAUX1(2),XAUX3(1),XAUX3(2),XAUX4(1),XAUX4(2)
     *,YAUX1(1),YAUX1(2),YAUX3(1),YAUX3(2),YAUX4(1),YAUX4(2)
  511 FCRMAT(8F10.3)
      IF(N4C.EQ.1) NPLOT=1
```

```
      IF(N40.GT.1) NPLOT=2                                              WOO 1480
      CALL DRAW(2,XAUX4,YAUX4,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0, WOO 1490
     *LAST)                                                             WOO 1500
      N40=N40+1                                                         WOO 1510
      WRITE(6,2) LAST                                                   WOO 1520
    2 FORMAT('0',30X,'LAST =',I1)                                       WOO 1530
      GO TO 10                                                          WOO 1540
  300 ROO1(NORDER,ISTA)=-5.                                             WOO 1550
      ROO2(NORDER,ISTA)=-5.                                             WOO 1560
      IHETA1=-5.                                                        WOO 1570
      IHETA2=-5.                                                        WOO 1580
   10 N41=N40                                                           WOO 1590
      WRITE(6,12) NORDER,ISTA,KAMMA1,ROO(NORDER,ISTA),                  WOO 1600
     *KANGOO(NORDER,ISTA),ROO1(NORDER,ISTA),IHETA1,ROO2(NORDER,ISTA),   WOO 1610
     *IHETA2                                                            WOO 1620
   12 FORMAT('1',///,45X,'***    SURFACE WIND AT DAY OF XBT OBSERVATION WOO 1630
     *    ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,     WOO 1640
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',WOO 1650
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,   WOO 1660
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =', WOO 1670
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,   WOO 1680
     *'DIRECTION =',I3)                                                 WOO 1690
      GO TO 16                                                          WOO 1700
   11 N41=N40                                                           WOO 1710
      WRITE(6,13) NORDER,ISTA,KAMMA1,ROO(NORDER,ISTA),                  WOO 1720
     *KANGOO(NORDER,ISTA),ROO1(NORDER,ISTA),KANGOO(NORDER,ISTA),        WOO 1730
     *ROO2(NORDER,ISTA),KANGOO(NORDER,ISTA)                             WOO 1740
   13 FORMAT('1',///,45X,'***    SURFACE WIND AT DAY OF XBT OBSERVATION WOO 1750
     *    ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,     WOO 1760
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',WOO 1770
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,   WOO 1780
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =', WOO 1790
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,   WOO 1800
     *'DIRECTION =',I3,' (+ OR -90)')                                   WOO 1810
      GO TO 16                                                          WOO 1820
   14 N41=N40                                                           WOO 1830
      WRITE(6,15) NORDER,ISTA,KAMMA1,ROO(NORDER,ISTA),                  WOO 1840
     *KANGOO(NORDER,ISTA),ROO1(NORDER,ISTA),KANGOO(NORDER,ISTA),        WOO 1850
     *ROO2(NORDER,ISTA),KANGOO(NORDER,ISTA)                             WOO 1860
   15 FORMAT('1',///,45X,'***    SURFACE WIND AT DAY OF XBT OBSERVATION WOO 1870
     *    ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,     WOO 1880
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',WOO 1890
     *F5.1,25X,'DIRECTION =',I3,            //,20X,'SURFACE WIND COMPOWOO 1900
     *NENTS  :',/,35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,   WOO 1910
     *'DIRECTION =',I3,' (+ OR -90)',/,25X,'PERPENDICULAR TO COAST :',  WOO 1920
     *10X,'INTENSITY =',F5.1,10X,'DIRECTION =',I3)                      WOO 1930
   16 RETURN                                                            WOO 1940
      END                                                               WOO 1950
```

```
C
C     ..................................................................
C     SUBROUTINE WIND01
C
C     PURPOSE
C        TO PROCESS THE WIND VELOCITY MEASURED AT ONE DAY BEFORE THE
C        DAY OF THE DIGITAL TRACE AND AT THE SAME GEOGRAPHIC POSITION.
C        TO PLOT THE WIND VECTOR AND ITS COMPONENTS (ALONG AND
C        PERPENDICULAR TO THE DIRECTION OF THE NEAR COASTLINE)
C
C     USAGE
C        CALL WIND01(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
C        KAMMA1,N41)
C
C     DESCRIPTION OF PARAMETERS
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS
C               SECTION
C        XGRID -INPUT VALUE CONTAINING THE LOCATION OF THE STATION
C               (ALONG THE CROSS SECTION OR TIME AXIS) WHERE THE WIND
C               DATA IS REPORTED TO
C        YGRID -INPUT CONTAINING THE VALUE OF THE ORDINATE TO LOCATE
C               THE WIND VECTORS IN THE PLOT
C        KKK   -INPUT VALUE CONTROLLING THE PLOT OF THE WIND VECTOR
C               (IF K.NE.0 : 'NO PLOT')
C        N40   -INPUT ACTUAL NUMBER OF WIND PLOTS
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE
C        ITITL4-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR
C               PLOT OF WIND DATA ALONG A CROSS SECTION OR TIME AXIS
C        KAMMA1-INPUT VALUE CONTAINING THE DIRECTION OF THE NEAR
C               COASTLINE
C        N41   -OUTPUT ACTUAL NUMBER OF WIND PLOTS
C
C     REMARKS
C        (1)THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO RECEIVE
C        THE REQUIRED DATA FROM THE MAIN PROGRAM AND TO GIVE BACK
C        PROCESSED INFORMATION.
C        (2)THE WIND VECTOR AND ITS COMPONENTS ARE PLOTTED IN A TWO
C        DIMENSIONAL DIAGRAM: (A)THE ABCISSA OR THE POSITION ALONG THE
C        CRUISE WHERE THE WIND IS REPORTED TO AND (B)THE ORDINATE OR
C        THE RELATIVE TIME (IN DAYS) THAT ELAPSES FROM THE DAY WHEN
C        THE WIND WAS MEASURED UNTIL THE DAY THE DIGITAL TRACE IS
C        OBTAINED.
C
C     SUBROUTINES REQUIRED
C        VECTOR
C        DRAW (IBM SCIENTIFIC LIBRARY)
C
```

```
C     METHOD
C        THE COMPONENTS OF THE WIND VECTOR ARE COMPUTED USING
C        TRIGONOMETRIC RELATIONSHIPS
C
C     ..................................................................
C

      SUBROUTINE WIND01(NORDER,ISTA,XGRID YGRID,KKK,N40,LABEL,ITITL4,
     *KAMMA1,N41)
      REAL*8 ITITL4(12)
      DIMENSION XAUX1(2),YAUX1(2),XAUX3(2),YAUX3(2),XAUX4(2),YAUX4(2)
      COMMON/WORKD/R00(20,12),KANG00(20,12),R01(20,12),KANG01(20,12),
     *R02(20,12),KANG02(20,12),R03(20,12),KANG03(20,12),R05(20,12),
     *KANG05(20,12),R10(20,12),KANG10(20,12),R001(20,12),R002(20,12),
     *R011(20,12),R012(20,12),R021(20,12),R022(20,12),R031(20,12),
     *R032(20,12),R051(20,12),R052(20,12),R101(20,12),R102(20,12)
C
C     COMPUTATION OF COORDINATES DEFINING THE LOCAL WIND VECTOR
C
      IF((R01(NORDER,ISTA).LT.0.).OR.(KANG01(NORDER,ISTA).LT.0)) GO TO
     *300
      IF(KKK.GT.0) GO TO 3
      KANG1=KANG01(NORDER,ISTA)
      CALL VECTOR(KANG1,ALPH1,BET1)
      DELTX1=R01(NORDER,ISTA)*ALPH1/45.
      DELTY1=R01(NORDER,ISTA)*BET1/45.
      XAUX1(1)=XGRID
      XAUX1(2)=XAUX1(1)+DELTX1
      YAUX1(1)=YGRID+0.4
      YAUX1(2)=YAUX1(1)+DELTY1
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX1,YAUX1,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
C
C     FINDING THE COMPONENTS OF THE LOCAL WIND VECTOR ALONG TWO PRE-
C     -DEFINED DIRECTIONS (I.E., ALONG THE COAST AND PERPENDICULAR TO
C     THIS DIRECTION)
C
C     KAMMA: ANGULAR DIFFERENCE BETWEEN THE WIND VECTOR (KANG) AND THE
C     DIRECTION ALONG THE COAST (KAMMA1)
C
      IF(KAMMA1.LT.0) GO TO 300
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
    3 KAMMA=KANG01(NORDER,ISTA)-KAMMA1
```

```
      KAMMA5=IABS(KAMMA)
      IF(KAMMA5.LE.90) GO TO 4
      IF(KAMMA.LT.(-90)) GO TO 6
      KAMMA1=KAMMA1+180
      GO TO 3
    6 KAMMA1=KAMMA1-180
      GO TO 3
    4 CALL VECTOR(KAMMA5,ALPH2,BET2)
C
C     RXX1: ALONG THE COAST COMPONENT (INTENSITY) OF LOCAL WIND
C     RXX2:PERPENDICULAR TO COAST COMPONENT (INTENSITY) OF LOCAL WIND
C
      RO11(NORDER,ISTA)=RO1(NORDER,ISTA)*BET2
      RO12(NORDER,ISTA)=RO1(NORDER,ISTA)*ALPH2
      IF(KAMMA1.LT.0) KAMMA1=KAMMA1+360
      IHETA1=KAMMA1
      IF(IHETA1.GE.360) IHETA1=IHETA1-360
      IF(KAMMA5.EQ.0) GO TO 11
      IF(KKK.GT.0) GO TO 8
      CALL VECTOR(IHETA1,ALPH3,BET3)
      DELTX3=RO11(NORDER,ISTA)*ALPH3/45.
      DELTY3=RO11(NORDER,ISTA)*BET3/45.
      XAUX3(1)=XGRID
      XAUX3(2)=XAUX3(1)+DELTX3
      YAUX3(1)=YGRID+0.4
      YAUX3(2)=YAUX3(1)+DELTY3
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX3,YAUX3,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    8 IF(KANGO1(NORDER,ISTA).LT.KAMMA1) IHETA2=KAMMA1-90
      IF(KANGO1(NORDER,ISTA).GT.KAMMA1) IHETA2=KAMMA1+90
      IF(IHETA2.GE.360) IHETA2=IHETA2-360
      IF(IHETA2.LT.0) IHETA2=IHETA2+360
      KAMMA7=IABS(KANGO1(NORDER,ISTA)-IHETA2)
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
      IF(KANGO1(NORDER,ISTA).GE.360) KANGO1(NORDER,ISTA)=KANGO1(NORDER,
     *ISTA)-360
      IF(KAMMA7.EQ.0) GO TO 14
      IF(KKK.GT.0) GO TO 10
      CALL VECTOR(IHETA2,ALPH4,BET4)
      DELTX4=RO12(NORDER,ISTA)*ALPH4/45.
      DELTY4=RO12(NORDER,ISTA)*BET4/45.
      XAUX4(1)=XGRID
      XAUX4(2)=XAUX4(1)+DELTX4
      YAUX4(1)=YGRID+0.4
```

```
      YAUX4(2)=YAUX4(1)+DELTY4
      WRITE(6,511) XAUX1(1),XAUX1(2),XAUX3(1),XAUX3(2),XAUX4(1),XAUX4(2)
     *,YAUX1(1),YAUX1(2),YAUX3(1),YAUX3(2),YAUX4(1),YAUX4(2)
  511 FORMAT(8F10.3)
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX4,YAUX4,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    2 FORMAT('0',30X,'LAST =',I1)
      GO TO 10
  300 RO11(NORDER,ISTA)=-5.
      RO12(NORDER,ISTA)=-5.
      IHETA1=-5.
      IHETA2=-5.
   10 N41=N40
      WRITE(6,12) NORDER,ISTA,KAMMA1,RO1(NORDER,ISTA),
     *KANGO1(NORDER,ISTA),RO11(NORDER,ISTA),IHETA1,RO12(NORDER,ISTA),
     *IHETA2
   12 FORMAT('0',///,45X,'***  SURFACE WIND AT  1 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3)
      GO TO 16
   11 N41=N40
      WRITE(6,13) NORDER,ISTA,KAMMA1,RO1(NORDER,ISTA),
     *KANGO1(NORDER,ISTA),RO11(NORDER,ISTA),KANGO1(NORDER,ISTA),
     *RO12(NORDER,ISTA),KANGO1(NORDER,ISTA)
   13 FORMAT('0',///,45X,'***  SURFACE WIND AT  1 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3,' (+ OR -90)')
      GO TO 16
   14 N41=N40
      WRITE(6,15) NORDER,ISTA,KAMMA1,RO1(NORDER,ISTA),
     *KANGO1(NORDER,ISTA),RO11(NORDER,ISTA),KANGO1(NORDER,ISTA),
     *RO12(NORDER,ISTA),KANGO1(NORDER,ISTA)
   15 FORMAT('0',///,45X,'***  SURFACE WIND AT  1 DAY BEFORE DAY XBT CBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,25X,'DIRECTION =',I3,                //,20X,'SURFACE WIND COMPO
```

```
     *NENTS  :',/,35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3,' (+ OR -90)',/,25X,'PERPENDICULAR TO COAST :',
     *10X,'INTENSITY =',F5.1,10X,'DIRECTION =',I3)
   16 RETURN
      END
C
C     ..................................................................
C     SUBROUTINE WINDO2
C
C     PURPOSE
C        TO PROCESS THE WIND VELOCITY MEASURED AT TWO DAYS BEFORE THE
C        DAY OF THE DIGITAL TRACE AND AT THE SAME GEOGRAPHICAL
C        POSITION. TO PLOT THE WIND VECTOR AND ITS COMPONENTS (ALONG
C        AND PERPENDICULAR TO THE DIRECTION OF THE NEAR COASTLINE)
C
C     USAGE
C        CALL WINDO2(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
C        KAMMA1,N41)
C
C     DESCRIPTION OF PARAMETERS
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS
C               SECTION
C        XGRID -INPUT VALUE CONTAINING THE LOCATION OF THE STATION
C               (ALONG THE CROSS SECTION OR TIME AXIS) WHERE THE WIND
C               DATA IS REPORTED TO
C        YGRID -INPUT CONTAINING THE VALUE OF THE ORDINATE TO LOCATE
C               THE WIND VECTORS IN THE PLOT
C        KKK   -INPUT VALUE CONTROLLING THE PLOT OF THE WIND VECTOR
C               (IF K.NE.0 : 'NO PLOT')
C        N40   -INPUT ACTUAL NUMBER OF WIND PLOTS
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE
C        ITITL4-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR
C               PLOT OF WIND DATA ALONG A CROSS SECTION OR TIME AXIS
C        KAMMA1-INPUT VALUE CONTAINING THE DIRECTION OF THE NEAR
C               COASTLINE
C        N41   -OUTPUT ACTUAL NUMBER OF WIND PLOTS
C
C     REMARKS
C        (1)THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO RECEIVE
C        THE REQUIRED DATA FROM THE MAIN PROGRAM AND TO GIVE BACK
C        PROCESSED INFORMATION.
C        (2)THE WIND VECTOR AND ITS COMPONENTS ARE PLOTTED IN A TWO
C        DIMENSIONAL DIAGRAM: (A)THE ABCISSA OR THE POSITION ALONG THE
C        CRUISE WHERE THE WIND IS REPORTED TO AND (B)THE ORDINATE OR
C        THE RELATIVE TIME (IN DAYS) THAT ELAPSES FROM THE DAY WHEN
C        THE WIND WAS MEASURED UNTIL THE DAY THE DIGITAL TRACE IS
C        OBTAINED.
```

```
C      DRAW (IBM-SCIENTIFIC LIBRARY)
C
C      METHOD
C         THE COMPONENTS OF THE WIND VECTOR ARE COMPUTED USING
C         TRIGONOMETRIC RELATIONSHIPS
C
C      ...............................................................
C
      SUBROUTINE WIND02(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
     *KAMMA1,N41)
      REAL*8 ITITL4(12)
      DIMENSION XAUX1(2),YAUX1(2),XAUX3(2),YAUX3(2),XAUX4(2),YAUX4(2)
      COMMON/WORKD/R00(20,12),KANG00(20,12),R01(20,12),KANG01(20,12),
     *R02(20,12),KANG02(20,12),R03(20,12),KANG03(20,12),R05(20,12),
     *KANG05(20,12),R10(20,12),KANG10(20,12),R001(20,12),R002(20,12),
     *R011(20,12),R012(20,12),R021(20,12),R022(20,12),R031(20,12),
     *R032(20,12),R051(20,12),R052(20,12),R101(20,12),R102(20,12)
C
C      COMPUTATION OF COORDINATES DEFINING THE LOCAL WIND VECTOR
C
      IF((R02(NORDER,ISTA).LT.0.).OR.(KANG02(NORDER,ISTA).LT.0)) GO TO
     *300
      IF(KKK.GT.0) GO TO 3
      KANG1=KANG02(NORDER,ISTA)
      CALL VECTOR(KANG1,ALPH1,BET1)
      DELTX1=R02(NORDER,ISTA)*ALPH1/45.
      DELTY1=R02(NORDER,ISTA)*BET1/45.
      XAUX1(1)=XGRID
      XAUX1(2)=XAUX1(1)+DELTX1
      YAUX1(1)=YGRID+0.8
      YAUX1(2)=YAUX1(1)+DELTY1
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX1,YAUX1,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
C
C      FINDING THE COMPONENTS OF THE LOCAL WIND VECTOR ALONG TWO PRE-
C      -DEFINED DIRECTIONS (I.E., ALONG THE COAST AND PERPENDICULAR TO
C      THIS DIRECTION)
C
C      KAMMA: ANGULAR DIFFERENCE BETWEEN THE WIND VECTOR (KANG) AND THE
```

```
C     DIRECTION ALONG THE COAST (KAMMA1)
C
      IF(KAMMA1.LT.0) GO TO 300
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
    3 KAMMA=KANGO2(NORDER,ISTA)-KAMMA1
      KAMMA5=IABS(KAMMA)
      IF(KAMMA5.LE.90) GO TO 4
      IF(KAMMA.LT.(-90)) GO TO 6
      KAMMA1=KAMMA1+180
      GO TO 3
    6 KAMMA1=KAMMA1-180
      GO TO 3
    4 CALL VECTOR(KAMMA5,ALPH2,BET2)
C
C     RXX1: ALONG THE COAST COMPONENT (INTENSITY) OF LOCAL WIND
C     RXX2:PERPENDICULAR TO COAST COMPONENT (INTENSITY) OF LOCAL WIND
C
      RO21(NORDER,ISTA)=RO2(NORDER,ISTA)*BET2
      RO22(NORDER,ISTA)=RO2(NORDER,ISTA)*ALPH2
      IF(KAMMA1.LT.0) KAMMA1=KAMMA1+360
      IHETA1=KAMMA1
      IF(IHETA1.GE.360) IHETA1=IHETA1-360
      IF(KAMMA5.EQ.0) GO TO 11
      IF(KKK.GT.0) GO TO 8
      CALL VECTOR(IHETA1,ALPH3,BET3)
      DELTX3=RO21(NORDER,ISTA)*ALPH3/45.
      DELTY3=RO21(NORDER,ISTA)*BET3/45.
      XAUX3(1)=XGRID
      XAUX3(2)=XAUX3(1)+DELTX3
      YAUX3(1)=YGRID+0.8
      YAUX3(2)=YAUX3(1)+DELTY3
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX3,YAUX3,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    8 IF(KANGO2(NORDER,ISTA).LT.KAMMA1) IHETA2=KAMMA1-90
      IF(KANGO2(NORDER,ISTA).GT.KAMMA1) IHETA2=KAMMA1+90
      IF(IHETA2.GE.360) IHETA2=IHETA2-360
      IF(IHETA2.LT.0) IHETA2=IHETA2+360
      KAMMA7=IABS(KANGO2(NORDER,ISTA)-IHETA2)
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
      IF(KANGO2(NORDER,ISTA).GE.360) KANGO2(NORDER,ISTA)=KANGO2(NORDER,
     *ISTA)-360
      IF(KAMMA7.EQ.0) GO TO 14
      IF(KKK.GT.0) GO TO 10
      CALL VECTOR(IHETA2,ALPH4,BET4)
```

```
      DELTX4=RO22(NORDER,ISTA)*ALPH4/45.
      DELTY4=RO22(NORDER,ISTA)*BET4/45.
      XAUX4(1)=XGRID
      XAUX4(2)=XAUX4(1)+DELTX4
      YAUX4(1)=YGRID+0.8
      YAUX4(2)=YAUX4(1)+DELTY4
      WRITE(6,511) XAUX1(1),XAUX1(2),XAUX3(1),XAUX3(2),XAUX4(1),XAUX4(2)
     *,YAUX1(1),YAUX1(2),YAUX3(1),YAUX3(2),YAUX4(1),YAUX4(2)
  511 FORMAT(8F10.3)
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX4,YAUX4,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    2 FORMAT('0',30X,'LAST =',I1)
      GO TO 10
  300 RO21(NORDER,ISTA)=-5.
      RO22(NORDER,ISTA)=-5.
      IHETA1=-5.
      IHETA2=-5.
   10 N41=N40
      WRITE(6,12) NORDER,ISTA,KAMMA1,RO2(NORDER,ISTA),
     *KANGO2(NORDER,ISTA),RO21(NORDER,ISTA),IHETA1,RO22(NORDER,ISTA),
     *IHETA2
   12 FORMAT('0',///,45X,'***  SURFACE WIND AT  2 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3)
      GO TO 16
   11 N41=N40
      WRITE(6,13) NORDER,ISTA,KAMMA1,RO2(NORDER,ISTA),
     *KANGO2(NORDER,ISTA),RO21(NORDER,ISTA),KANGO2(NORDER,ISTA),
     *RO22(NORDER,ISTA),KANGO2(NORDER,ISTA)
   13 FORMAT('0',///,45X,'***  SURFACE WIND AT  2 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3,' (+ OR -90)')
      GO TO 16
   14 N41=N40
      WRITE(6,15) NORDER,ISTA,KAMMA1,RO2(NORDER,ISTA),
     *KANGO2(NORDER,ISTA),RO21(NORDER,ISTA),KANGO2(NORDER,ISTA),
```

```
     *R022(NORDER,ISTA),KANGO2(NORDER,ISTA)                             WO2 1860
   15 FORMAT('0',///,45X,'***  SURFACE WIND AT  2 DAY BEFORE DAY XBT OBSWO2 1870
     *  ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,        WO2 1880
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',WO2 1890
     *F5.1,25X,'DIRECTION =',I3,         //,20X,'SURFACE WIND COMPOWO2 1900
     *NENTS  :',/,35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,  WO2 1910
     *'DIRECTION =',I3,' (+ CR -90)',/,25X,'PERPENDICULAR TO COAST :',  WO2 1920
     *10X,'INTENSITY =',F5.1,10X,'DIRECTION =',I3)                      WO2 1930
   16 RETURN                                                            WO2 1940
      END                                                               WO2 1950
C                                                                       WO3 0010
C     ..................................................................WO3 0020
C     SUBROUTINE WINDO3                                                 WO3 0030
C                                                                       WO3 0040
C     PURPOSE                                                           WO3 0050
C        TO PROCESS THE WIND VELOCITY MEASURED AT THREE DAYS BEFORE THE WO3 0060
C        DAY OF THE DIGITAL TRACE AND AT THE SAME GEOGRAPHICAL          WO3 0070
C        POSITION. TO PLOT THE WIND VECTOR AND ITS COMPONENTS (ALONG    WO3 0080
C        AND PERPENDICULAR TO THE DIRECTION OF THE NEAR COASTLINE)      WO3 0090
C                                                                       WO3 0100
C     USAGE                                                             WO3 0110
C        CALL WINDO3(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,      WO3 0120
C        KAMMA1,N41)                                                    WO3 0130
C                                                                       WO3 0140
C     DESCRIPTION OF PARAMETERS                                         WO3 0150
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                      WO3 0160
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS        WO3 0170
C               SECTION                                                 WO3 0180
C        XGRID -INPUT VALUE CONTAINING THE LOCATION OF THE STATION      WO3 0190
C               (ALONG THE CROSS SECTION OR TIME AXIS) WHERE THE WIND   WO3 0200
C               DATA IS REPORTED TO                                     WO3 0210
C        YGRID -INPUT CONTAINING THE VALUE OF THE ORDINATE TO LOCATE    WO3 0220
C               THE WIND VECTORS IN THE PLOT                            WO3 0230
C        KKK   -INPUT VALUE CONTROLLING THE PLOT OF THE WIND VECTOR     WO3 0240
C               (IF K.NE.0 : 'NO PLOT')                                 WO3 0250
C        N40   -INPUT ACTUAL NUMBER OF WIND PLOTS                       WO3 0260
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE           WO3 0270
C        ITITL4-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR    WO3 0280
C               PLOT OF WIND DATA ALONG A CROSS SECTION OR TIME AXIS    WO3 0290
C        KAMMA1-INPUT VALUE CONTAINING THE DIRECTION OF THE NEAR        WO3 0300
C               COASTLINE                                               WO3 0310
C        N41   -OUTPUT ACTUAL NUMBER OF WIND PLOTS                      WO3 0320
C                                                                       WO3 0330
C     REMARKS                                                           WO3 0340
C        (1)THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO RECEIVE    WO3 0350
C        THE REQUIRED DATA FROM THE MAIN PROGRAM AND TO GIVE BACK       WO3 0360
C        PROCESSED INFORMATION.                                         WO3 0370
C        (2)THE WIND VECTOR AND ITS COMPONENTS ARE PLOTTED IN A TWO     WO3 0380
```

```
C        DIMENSIONAL DIAGRAM: (A)THE ABCISSA OR THE POSITION ALONG THE
C        CRUISE WHERE THE WIND IS REPORTED TO AND (B)THE ORDINATE OR
C        THE RELATIVE TIME (IN DAYS) THAT ELAPSES FROM THE DAY WHEN
C        THE WIND WAS MEASURED UNTIL THE DAY THE DIGITAL TRACE IS
C        OBTAINED.
C
C     SUBROUTINES REQUIRED
C        VECTOR
C        DRAW (IBM SCIENTIFIC LIBRARY)
C
C     METHOD
C        THE COMPONENTS OF THE WIND VECTOR ARE COMPUTED USING
C        TRIGONOMETRIC RELATIONSHIPS
C
C     ..................................................................
C
      SUBROUTINE WIND03(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
     *KAMMA1,N41)
      REAL*8 ITITL4(12)
      DIMENSION XAUX1(2),YAUX1(2),XAUX3(2),YAUX3(2),XAUX4(2),YAUX4(2)
      COMMON/WORKD/R00(20,12),KANG00(20,12),R01(20,12),KANG01(20,12),
     *R02(20,12),KANG02(20,12),R03(20,12),KANG03(20,12),R05(20,12),
     *KANG05(20,12),R10(20,12),KANG10(20,12),R001(20,12),R002(20,12),
     *R011(20,12),R012(20,12),R021(20,12),R022(20,12),R031(20,12),
     *R032(20,12),R051(20,12),R052(20,12),R101(20,12),R102(20,12)
C
C     COMPUTATION OF COORDINATES DEFINING THE LOCAL WIND VECTOR
C
      IF((R03(NORDER,ISTA).LT.0.).OR.(KANG03(NORDER,ISTA).LT.0)) GO TO
     *300
      IF(KKK.GT.0) GO TO 3
      KANG1=KANG03(NORDER,ISTA)
      CALL VECTOR(KANG1,ALPH1,BET1)
      DELTX1=R03(NORDER,ISTA)*ALPH1/45.
      DELTY1=R03(NORDER,ISTA)*BET1/45.
      XAUX1(1)=XGRID
      XAUX1(2)=XAUX1(1)+DELTX1
      YAUX1(1)=YGRID+1.2
      YAUX1(2)=YAUX1(1)+DELTY1
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX1,YAUX1,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
C
```

```
C     FINDING THE COMPONENTS OF THE LOCAL WIND VECTCR ALONG TWO PRE-
C     -DEFINED DIRECTIONS (I.E., ALONG THE COAST AND PERPENDICULAR TO
C     THIS DIRECTION)
C
C     KAMMA: ANGULAR DIFFERENCE BETWEEN THE WIND VECTOR (KANG) AND THE
C     DIRECTION ALONG THE COAST (KAMMA1)
C
      IF(KAMMA1.LT.0) GO TO 300
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
    3 KAMMA=KANGO3(NORDER,ISTA)-KAMMA1
      KAMMA5=IABS(KAMMA)
      IF(KAMMA5.LE.90) GO TO 4
      IF(KAMMA.LT.(-90)) GO TO 6
      KAMMA1=KAMMA1+180
      GC TO 3
    6 KAMMA1=KAMMA1-180
      GC TO 3
    4 CALL VECTOR(KAMMA5,ALPH2,BET2)
C
C     RXX1: ALONG THE COAST COMPONENT (INTENSITY) OF LOCAL WIND
C     RXX2:PERPENDICULAR TO COAST COMPONENT (INTENSITY) OF LOCAL WIND
C
      RO31(NORDER,ISTA)=RO3(NORDER,ISTA)*BET2
      RO32(NORDER,ISTA)=RO3(NORDER,ISTA)*ALPH2
      IF(KAMMA1.LT.0) KAMMA1=KAMMA1+360
      IHETA1=KAMMA1
      IF(IHETA1.GE.360) IHETA1=IHETA1-360
      IF(KAMMA5.EQ.0) GO TO 11
      IF(KKK.GT.0) GO TO 8
      CALL VECTOR(IHETA1,ALPH3,BET3)
      DELTX3=RO31(NORDER,ISTA)*ALPH3/45.
      DELTY3=RO31(NORDER,ISTA)*BET3/45.
      XAUX3(1)=XGRID
      XAUX3(2)=XAUX3(1)+DELTX3
      YAUX3(1)=YGRID+1.2
      YAUX3(2)=YAUX3(1)+DELTY3
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=[illegible]
      CALL DRAW(2,XAUX3,YAUX3,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    8 IF(KANGO3(NORDER,ISTA).LT.KAMMA1) IHETA2=KAMMA1-90
      IF(KANGO3(NORDER,ISTA).GT.KAMMA1) IHETA2=KAMMA1+90
      IF(IHETA2.GE.360) IHETA2=IHETA2-360
      IF(IHETA2.LT.0) IHETA2=IHETA2+360
      KAMMA7=IABS(KANGO3(NORDER,ISTA)-IHETA2)
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
```

```
      IF(KANGO3(NORDER,ISTA).GE.360) KANGO3(NORDER,ISTA)=KANGO3(NORDER,
     *ISTA)-360
      IF(KAMMA7.EQ.0) GO TO 14
      IF(KKK.GT.0) GO TO 10
      CALL VECTOR(IHETA2,ALPH4,BET4)
      DELTX4=RO32(NORDER,ISTA)*ALPH4/45.
      DELTY4=RO32(NORDER,ISTA)*BET4/45.
      XAUX4(1)=XGRID
      XAUX4(2)=XAUX4(1)+DELTX4
      YAUX4(1)=YGRID+1.2
      YAUX4(2)=YAUX4(1)+DELTY4
      WRITE(6,511) XAUX1(1),XAUX1(2),XAUX3(1),XAUX3(2),XAUX4(1),XAUX4(2)
     *,YAUX1(1),YAUX1(2),YAUX3(1),YAUX3(2),YAUX4(1),YAUX4(2)
  511 FORMAT(8F10.3)
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX4,YAUX4,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    2 FORMAT('0',30X,'LAST =',I1)
      GO TO 10
  300 RO31(NORDER,ISTA)=-5.
      RO32(NORDER,ISTA)=-5.
      IHETA1=-5.
      IHETA2=-5.
   10 N41=N40
      WRITE(6,12) NORDER,ISTA,KAMMA1,RO3(NORDER,ISTA),
     *KANGO3(NORDER,ISTA),RO31(NORDER,ISTA),IHETA1,RO32(NORDER,ISTA),
     *IHETA2
   12 FORMAT('0',///,45X,'***  SURFACE WIND AT  3 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3)
      GO TO 16
   11 N41=N40
      WRITE(6,13) NORDER,ISTA,KAMMA1,RO3(NORDER,ISTA),
     *KANGO3(NORDER,ISTA),RO31(NORDER,ISTA),KANGO3(NORDER,ISTA),
     *RO32(NORDER,ISTA),KANGO3(NORDER,ISTA)
   13 FORMAT('0',///,45X,'***  SURFACE WIND AT  3 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
```

```
     *'DIRECTION =',I3,' (+ OR -90)')
      GC TO 16
   14 N41=N40
      WRITE(6,15) NORDER,ISTA,KAMMA1,RO3(NORDER,ISTA),
     *KANG03(NORDER,ISTA),RO31(NORDER,ISTA),KANG03(NORDER,ISTA),
     *RO32(NORDER,ISTA),KANG03(NORDER,ISTA)
   15 FORMAT('0',///,45X,'***  SURFACE WIND AT  3 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,25X,'DIRECTION =',I3,              //,20X,'SURFACE WIND COMPO
     *NENTS   :',/,35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3,' (+ OR -90)',/,25X,'PERPENDICULAR TO COAST :',
     *10X,'INTENSITY =',F5.1,10X,'DIRECTION =',I3)
   16 RETURN
      END
C
C     ...................................................................
C     SUBROUTINE WINDO5
C
C     PURPOSE
C        TO PROCESS THE WIND VELOCITY MEASURED AT FIVE DAYS BEFORE THE
C        DAY OF THE DIGITAL TRACE AND AT THE SAME GEOGRAPHICAL
C        POSITION. TO PLOT THE WIND VECTOR AND ITS COMPONENTS (ALONG
C        AND PERPENDICULAR TO THE DIRECTION OF THE NEAR COASTLINE)
C
C     USAGE
C        CALL WINDO5(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
C        KAMMA1,N41)
C
C     DESCRIPTION OF PARAMETERS
C        NORDER-ACTUAL NUMBER OF PROCESSED CRUISES
C        ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS
C               SECTION
C        XGRID -INPUT VALUE CONTAINING THE LOCATION OF THE STATION
C               (ALONG THE CROSS SECTION OR TIME AXIS) WHERE THE WIND
C               DATA IS REPORTED TO
C        YGRID -INPUT CONTAINING THE VALUE OF THE ORDINATE TO LOCATE
C               THE WIND VECTORS IN THE PLOT
C        KKK   -INPUT VALUE CONTROLLING THE PLOT OF THE WIND VECTOR
C               (IF K.NE.0 : 'NO PLOT')
C        N40   -INPUT ACTUAL NUMBER OF WIND PLOTS
C        LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE
C        ITITL4-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR
C               PLOT OF WIND DATA ALONG A CROSS SECTION OR TIME AXIS
C        KAMMA1-INPUT VALUE CONTAINING THE DIRECTION OF THE NEAR
C               COASTLINE
C        N41   -CUTPUT ACTUAL NUMBER OF WIND PLOTS
C
```

```
C     REMARKS
C        (1)THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO RECEIVE
C        THE REQUIRED DATA FROM THE MAIN PROGRAM AND TO GIVE BACK
C        PROCESSED INFORMATION.
C        (2)THE WIND VECTOR AND ITS COMPONENTS ARE PLOTTED IN A TWO
C        DIMENSIONAL DIAGRAM: (A)THE ABCISSA OR THE POSITION ALONG THE
C        CRUISE WHERE THE WIND IS REPORTED TO AND (B)THE ORDINATE OR
C        THE RELATIVE TIME (IN DAYS) THAT ELAPSES FROM THE DAY WHEN
C        THE WIND WAS MEASURED UNTIL THE DAY THE DIGITAL TRACE IS
C        OBTAINED.
C
C     SUBROUTINES REQUIRED
C        VECTOR
C        DRAW (IBM SCIENTIFIC LIBRARY)
C
C     METHOD
C        THE COMPONENTS OF THE WIND VECTOR ARE COMPUTED USING
C        TRIGONOMETRIC RELATIONSHIPS
C
C     ...............................................................
C

      SUBROUTINE WINDO5(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
     *KAMMA1,N41)
      REAL*8 ITITL4(12)
      DIMENSION XAUX1(2),YAUX1(2),XAUX3(2),YAUX3(2),XAUX4(2),YAUX4(2)
      COMMON/WORKD/R00(20,12),KANG00(20,12),R01(20,12),KANG01(20,12),
     *R02(20,12),KANG02(20,12),R03(20,12),KANG03(20,12),R05(20,12),
     *KANG05(20,12),R10(20,12),KANG10(20,12),R001(20,12),R002(20,12),
     *R011(20,12),R012(20,12),R021(20,12),R022(20,12),R031(20,12),
     *R032(20,12),R051(20,12),R052(20,12),R101(20,12),R102(20,12)
C
C     COMPUTATION OF COORDINATES DEFINING THE LOCAL WIND VECTOR
C
      IF((R05(NORDER,ISTA).LT.0.).OR.(KANG05(NORDER,ISTA).LT.0)) GO TO
     *300
      IF(KKK.GT.0) GO TO 3
      KANG1=KANG05(NORDER,ISTA)
      CALL VECTOR(KANG1,ALPH1,BET1)
      DELTX1=R05(NORDER,ISTA)*ALPH1/45.
      DELTY1=R05(NORDER,ISTA)*BET1/45.
      XAUX1(1)=XGRID
      XAUX1(2)=XAUX1(1)+DELTX1
      YAUX1(1)=YGRID+1.6
      YAUX1(2)=YAUX1(1)+DELTY1
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
```

```
      CALL DRAW(2,XAUX1,YAUX1,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
C
C     FINDING THE COMPONENTS OF THE LOCAL WIND VECTOR ALONG TWO PRE-
C     -DEFINED DIRECTIONS (I.E., ALONG THE COAST AND PERPENDICULAR TO
C     THIS DIRECTION)
C
C     KAMMA: ANGULAR DIFFERENCE BETWEEN THE WIND VECTOR (KANG) AND THE
C     DIRECTION ALONG THE COAST (KAMMA1)
C
      IF(KAMMA1.LT.0) GO TO 300
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
    3 KAMMA=KANG05(NORDER,ISTA)-KAMMA1
      KAMMA5=IABS(KAMMA)
      IF(KAMMA5.LE.90) GO TO 4
      IF(KAMMA.LT.(-90)) GO TO 6
      KAMMA1=KAMMA1+180
      GO TO 3
    6 KAMMA1=KAMMA1-180
      GO TO 3
    4 CALL VECTOR(KAMMA5,ALPH2,BET2)
C
C     RXX1: ALONG THE COAST COMPONENT (INTENSITY) OF LOCAL WIND
C     RXX2:PERPENDICULAR TO COAST COMPONENT (INTENSITY) OF LOCAL WIND
C
      RO51(NORDER,ISTA)=RO5(NORDER,ISTA)*BET2
      RO52(NORDER,ISTA)=RO5(NORDER,ISTA)*ALPH2
      IF(KAMMA1.LT.0) KAMMA1=KAMMA1+360
      IHETA1=KAMMA1
      IF(IHETA1.GE.360) IHETA1=IHETA1-360
      IF(KAMMA5.EQ.0) GO TO 11
      IF(KKK.GT.0) GO TO 8
      CALL VECTOR(IHETA1,ALPH3,BET3)
      DELTX3=RO51(NORDER,ISTA)*ALPH3/45.
      DELTY3=RO51(NORDER,ISTA)*BET3/45.
      XAUX3(1)=XGRID
      XAUX3(2)=XAUX3(1)+DELTX3
      YAUX3(1)=YGRID+1.6
      YAUX3(2)=YAUX3(1)+DELTY3
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX3,YAUX3,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    8 IF(KANG05(NORDER,ISTA).LT.KAMMA1) IHETA2=KAMMA1-90
```

```
      IF(KANGO5(NORDER,ISTA).GT.KAMMA1) IHETA2=KAMMA1+90
      IF(IHETA2.GE.360) IHETA2=IHETA2-360
      IF(IHETA2.LT.0) IHETA2=IHETA2+360
      KAMMA7=IABS(KANGO5(NORDER,ISTA)-IHETA2)
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
      IF(KANGO5(NORDER,ISTA).GE.360) KANGO5(NORDER,ISTA)=KANGO5(NORDER,
     *ISTA)-360
      IF(KAMMA7.EQ.0) GO TO 14
      IF(KKK.GT.0) GO TO 10
      CALL VECTOR(IHETA2,ALPH4,BET4)
      DELTX4=RO52(NORDER,ISTA)*ALPH4/45.
      DELTY4=RO52(NORDER,ISTA)*BET4/45.
      XAUX4(1)=XGRID
      XAUX4(2)=XAUX4(1)+DELTX4
      YAUX4(1)=YGRID+1.6
      YAUX4(2)=YAUX4(1)+DELTY4
      WRITE(6,511) XAUX1(1),XAUX1(2),XAUX3(1),XAUX3(2),XAUX4(1),XAUX4(2)
     *,YAUX1(1),YAUX1(2),YAUX3(1),YAUX3(2),YAUX4(1),YAUX4(2)
  511 FORMAT(8F10.3)
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX4,YAUX4,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
    2 FORMAT('0',30X,'LAST =',I1)
      GO TO 10
  300 RO51(NORDER,ISTA)=-5.
      RO52(NORDER,ISTA)=-5.
      IHETA1=-5.
      IHETA2=-5.
   10 N41=N40
      WRITE(6,12) NORDER,ISTA,KAMMA1,RO5(NORDER,ISTA),
     *KANGO5(NORDER,ISTA),RO51(NORDER,ISTA),IHETA1,RO52(NORDER,ISTA),
     *IHETA2
   12 FORMAT('0',///,45X,'***  SURFACE WIND AT  5 DAY BEFORE DAY XBT OBS
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
     *'DIRECTION =',I3)
      GO TO 16
   11 N41=N40
      WRITE(6,13) NORDER,ISTA,KAMMA1,RO5(NORDER,ISTA),
     *KANGO5(NORDER,ISTA),RO51(NORDER,ISTA),KANGO5(NORDER,ISTA),
     *RO52(NORDER,ISTA),KANGO5(NORDER,ISTA)
   13 FORMAT('0',///,45X,'***  SURFACE WIND AT  5 DAY BEFORE DAY XB3 O22
```

```
*    ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,        W05 1760
*'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',W05 1770
*F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,     W05 1780
*35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',   W05 1790
*I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,     W05 1800
*'DIRECTION =',I3,' (+ OR -90)')                                      W05 1810
 GO TO 16                                                             W05 1820
14 N41=N40                                                            W05 1830
 WRITE(6,15) NORDER,ISTA,KAMMA1,R05(NORDER,ISTA),                     W05 1840
*KANG05(NORDER,ISTA),R051(NORDER,ISTA),KANG05(NORDER,ISTA),           W05 1850
*R052(NORDER,ISTA),KANG05(NORDER,ISTA)                                W05 1860
15 FORMAT('0',///,45X,'***  SURFACE WIND AT  5 DAY BEFORE DAY XB3 022W05 1870
*    ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,        W05 1880
*'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',W05 1890
*F5.1,25X,'DIRECTION =',I3,             //,20X,'SURFACE WIND COMPOW05 1900
*NENTS  :',/,35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,     W05 1910
*'DIRECTION =',I3,' (+ OR -90)',/,25X,'PERPENDICULAR TO COAST :',    W05 1920
*10X,'INTENSITY =',F5.1,10X,'DIRECTION =',I3)                         W05 1930
16 RETURN                                                             W05 1940
 END                                                                  W05 1950
C                                                                     W10 0010
C   ..................................................................W10 0020
C   SUBROUTINE WIND10                                                 W10 0030
C                                                                     W10 0040
C   PURPOSE                                                           W10 0050
C      TO PROCESS THE WIND VELOCITY MEASURED AT TEN DAYS BEFORE THE   W10 0060
C      DAY OF THE DIGITAL TRACE AND AT THE SAME GEOGRAPHICAL          W10 0070
C      POSITION. TO PLOT THE WIND VECTOR AND ITS COMPONENTS (ALONG    W10 0080
C      AND PERPENDICULAR TO THE DIRECTION OF THE NEAR COASTLINE)      W10 0090
C                                                                     W10 0100
C   USAGE                                                             W10 0110
C      CALL WIND10(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,      W10 0120
C      KAMMA1,N41)                                                    W10 0130
C                                                                     W10 0140
C   DESCRIPTION OF PARAMETERS                                         W10 0150
C      NORDER-ACTUAL NUMBER OF PROCESSED CRUISES                      W10 0160
C      ISTA  -INPUT NUMBER IDENTIFYING A STATION ALONG A CROSS        W10 0170
C             SECTION                                                 W10 0180
C      XGRID -INPUT VALUE CONTAINING THE LOCATION OF THE STATION      W10 0190
C             (ALONG THE CROSS SECTION OR TIME AXIS) WHERE THE WIND   W10 0200
C             DATA IS REPORTED TO                                     W10 0210
C      YGRID -INPUT CONTAINING THE VALUE OF THE ORDINATE TO LOCATE    W10 0220
C             THE WIND VECTORS IN THE PLOT                            W10 0230
C      KKK   -INPUT VALUE CONTROLLING THE PLOT OF THE WIND VECTOR     W10 0240
C             (IF K.NE.0 : 'NO PLOT')                                 W10 0250
C      N40   -INPUT ACTUAL NUMBER OF WIND PLOTS                       W10 0260
C      LABEL -INPUT SHORT IDENTIFICATION OF A PLOTTED CURVE           W10 0270
C      ITITL4-INPUT VECTOR OF LENGTH '12' CONTAINING THE TITLE FOR    W10 0280
```

```
C           PLOT OF WIND DATA ALONG A CROSS SECTION OR TIME AXIS
C      KAMMA1-INPUT VALUE CONTAINING THE DIRECTION OF THE NEAR
C             COASTLINE
C      N41   -OUTPUT ACTUAL NUMBER OF WIND PLOTS
C
C   REMARKS
C      (1)THIS SUBROUTINE USES A 'COMMON' LABELLED AREA TO RECEIVE
C      THE REQUIRED DATA FROM THE MAIN PROGRAM AND TO GIVE BACK
C      PROCESSED INFORMATION.
C      (2)THE WIND VECTOR AND ITS COMPONENTS ARE PLOTTED IN A TWO
C      DIMENSIONAL DIAGRAM: (A)THE ABCISSA OR THE POSITION ALONG THE
C      CRUISE WHERE THE WIND IS REPORTED TO AND (B)THE ORDINATE OR
C      THE RELATIVE TIME (IN DAYS) THAT ELAPSES FROM THE DAY WHEN
C      THE WIND WAS MEASURED UNTIL THE DAY THE DIGITAL TRACE IS
C      OBTAINED.
C
C   SUBROUTINES REQUIRED
C      VECTOR
C      DRAW (IBM SCIENTIFIC LIBRARY)
C
C   METHOD
C      THE COMPONENTS OF THE WIND VECTOR ARE COMPUTED USING
C      TRIGONOMETRIC RELATIONSHIPS
C
C   ...............................................................
C

      SUBROUTINE WIND10(NORDER,ISTA,XGRID,YGRID,KKK,N40,LABEL,ITITL4,
     *KAMMA1,N41)
      REAL*8 ITITL4(12)
      DIMENSION XAUX1(2),YAUX1(2),XAUX3(2),YAUX3(2),XAUX4(2),YAUX4(2)
      COMMON/WORKD/R00(20,12),KANG00(20,12),R01(20,12),KANG01(20,12),
     *R02(20,12),KANG02(20,12),R03(20,12),KANG03(20,12),R05(20,12),
     *KANG05(20,12),R10(20,12),KANG10(20,12),R001(20,12),R002(20,12),
     *R011(20,12),R012(20,12),R021(20,12),R022(20,12),R031(20,12),
     *R032(20,12),R051(20,12),R052(20,12),R101(20,12),R102(20,12)
C
C   COMPUTATION OF COORDINATES DEFINING THE LOCAL WIND VECTOR
C
      IF((R10(NORDER,ISTA).LT.0.).OR.(KANG10(NORDER,ISTA).LT.0)) GO TO
     *300
      IF(KKK.GT.0) GO TO 3
      KANG1=KANG10(NORDER,ISTA)
      CALL VECTOR(KANG1,ALPH1,BET1)
      DELTX1=R10(NORDER,ISTA)*ALPH1/45.
      DELTY1=R10(NORDER,ISTA)*BET1/45.
      XAUX1(1)=XGRID
```

```
      XAUX1(2)=XAUX1(1)+DELTX1
      YAUX1(1)=YGRID+2.0
      YAUX1(2)=YAUX1(1)+DELTY1
      IF(N40.EQ.1) NPLOT=1
      IF(N40.GT.1) NPLOT=2
      CALL DRAW(2,XAUX1,YAUX1,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0,
     *LAST)
      N40=N40+1
      WRITE(6,2) LAST
C
C     FINDING THE COMPONENTS OF THE LOCAL WIND VECTOR ALONG TWO PRE-
C     -DEFINED DIRECTIONS (I.E., ALONG THE COAST AND PERPENDICULAR TO
C     THIS DIRECTION)
C
C     KAMMA: ANGULAR DIFFERENCE BETWEEN THE WIND VECTOR (KANG) AND THE
C     DIRECTION ALONG THE COAST (KAMMA1)
C
      IF(KAMMA1.LT.0) GO TO 300
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360
    3 KAMMA=KANG10(NORDER,ISTA)-KAMMA1
      KAMMA5=IABS(KAMMA)
      IF(KAMMA5.LE.90) GO TO 4
      IF(KAMMA.LT.(-90)) GO TO 6
      KAMMA1=KAMMA1+180
      GO TO 3
    6 KAMMA1=KAMMA1-180
      GO TO 3
    4 CALL VECTOR(KAMMA5,ALPH2,BET2)
C
C     RXX1: ALONG THE COAST COMPONENT (INTENSITY) OF LOCAL WIND
C     RXX2:PERPENDICULAR TO COAST COMPONENT (INTENSITY) OF LOCAL WIND
C
      R101(NORDER,ISTA)=R10(NORDER,ISTA)*BET2
      R102(NORDER,ISTA)=R10(NORDER,ISTA)*ALPH2
      IF(KAMMA1.LT.0) KAMMA1=KAMMA1+360
      IHETA1=KAMMA1
      IF(IHETA1.GE.360) IHETA1=IHETA1-360
      IF(KAMMA5.EQ.0) GO TO 11
      IF(KKK.GT.0) GO TO 8
      CALL VECTOR(IHETA1,ALPH3,BET3)
      DELTX3=R101(NORDER,ISTA)*ALPH3/45.
      DELTY3=R101(NORDER,ISTA)*BET3/45.
      XAUX3(1)=XGRID
      XAUX3(2)=XAUX3(1)+DELTX3
      YAUX3(1)=YGRID+2.0
      YAUX3(2)=YAUX3(1)+DELTY3
      IF(N40.EQ.1) NPLCT=1
      IF(N40.GT.1) NPLOT=2
```

```
      CALL DRAW(2,XAUX3,YAUX3,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0, W10 1230
     *LAST)                                                             W10 1240
      N40=N40+1                                                         W10 1250
      WRITE(6,2) LAST                                                   W10 1260
    8 IF(KANG10(NORDER,ISTA).LT.KAMMA1) IHETA2=KAMMA1-90                W10 1270
      IF(KANG10(NORDER,ISTA).GT.KAMMA1) IHETA2=KAMMA1+90                W10 1280
      IF(IHETA2.GE.360) IHETA2=IHETA2-360                               W10 1290
      IF(IHETA2.LT.0) IHETA2=IHETA2+360                                 W10 1300
      KAMMA7=IABS(KANG10(NORDER,ISTA)-IHETA2)                           W10 1310
      IF(KAMMA1.GE.360) KAMMA1=KAMMA1-360                               W10 1320
      IF(KANG10(NORDER,ISTA).GE.360) KANG10(NORDER,ISTA)=KANG10(NORDER, W10 1330
     *ISTA)-360                                                         W10 1340
      IF(KAMMA7.EQ.0) GO TO 14                                          W10 1350
      IF(KKK.GT.0) GO TO 10                                             W10 1360
      CALL VECTOR(IHETA2,ALPH4,BET4)                                    W10 1370
      DELTX4=R102(NORDER,ISTA)*ALPH4/45.                                W10 1380
      DELTY4=R102(NORDER,ISTA)*BET4/45.                                 W10 1390
      XAUX4(1)=XGRID                                                    W10 1400
      XAUX4(2)=XAUX4(1)+DELTX4                                          W10 1410
      YAUX4(1)=YGRID+2.0                                                W10 1420
      YAUX4(2)=YAUX4(1)+DELTY4                                          W10 1430
      WRITE(6,511) XAUX1(1),XAUX1(2),XAUX3(1),XAUX3(2),XAUX4(1),XAUX4(2)W10 1440
     *,YAUX1(1),YAUX1(2),YAUX3(1),YAUX3(2),YAUX4(1),YAUX4(2)            W10 1450
  511 FORMAT(8F10.3)                                                    W10 1460
      IF(N40.EQ.1) NPLOT=1                                              W10 1470
      IF(N40.GT.1) NPLOT=2                                              W10 1480
      CALL DRAW(2,XAUX4,YAUX4,NPLOT,0,LABEL,ITITL4,1.,1.,0,0,2,2,8,6,0, W10 1490
     *LAST)                                                             W10 1500
      N40=N40+1                                                         W10 1510
      WRITE(6,2) LAST                                                   W10 1520
    2 FORMAT('0',30X,'LAST =',I1)                                       W10 1530
      GO TO 10                                                          W10 1540
  300 R101(NORDER,ISTA)=-5.                                             W10 1550
      R102(NORDER,ISTA)=-5.                                             W10 1560
      IHETA1=-5.                                                        W10 1570
      IHETA2=-5.                                                        W10 1580
   10 N41=N40                                                           W10 1590
      WRITE(6,12) NORDER,ISTA,KAMMA1,R10(NORDER,ISTA),                  W10 1600
     *KANG10(NORDER,ISTA),R101(NORDER,ISTA),IHETA1,R102(NORDER,ISTA),   W10 1610
     *IHETA2                                                            W10 1620
   12 FORMAT('0',///,45X,'***  SURFACE WIND AT 10 DAY BEFORE DAY XBT OBSW10 1630
     *   ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,      W10 1640
     *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',W10 1650
     *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,   W10 1660
     *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =', W10 1670
     *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,   W10 1680
     *'DIRECTION =',I3)                                                 W10 1690
      GO TO 16                                                          W10 1700
```

```
11 N41=N40
   WRITE(6,13) NORDER,ISTA,KAMMA1,R10(NORDER,ISTA),
  *KANG10(NORDER,ISTA),R101(NORDER,ISTA),KANG10(NORDER,ISTA),
  *R102(NORDER,ISTA),KANG10(NORDER,ISTA)
13 FORMAT('0',///,45X,'***  SURFACE WIND AT 10 DAY BEFORE DAY XBT OBS
  *    ***',//,20X,'CRUISE NO. =',I3,20X,'STATICN NO. =',I3,20X,
  *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
  *F5.1,35X,'DIRECTION =',I3,//,20X,'SURFACE WIND COMPONENTS :',/,
  *35X,'ALONG THE COAST :',10X,'INTENSITY =',F5.1,10X,'DIRECTION =',
  *I3,/,35X,'PERPENDICULAR TO COAST :',10X,'INTENSITY =',F5.1,10X,
  *'DIRECTICN =',I3,' (+ OR -90)')
   GC TO 16
14 N41=N40
   WRITE(6,15) NORDER,ISTA,KAMMA1,R10(NORDER,ISTA),
  *KANG10(NORDER,ISTA),R101(NORDER,ISTA),KANG10(NORDER,ISTA),
  *R102(NORDER,ISTA),KANG10(NORDER,ISTA)
15 FORMAT('0',///,45X,'***  SURFACE WIND AT 10 DAY BEFORE DAY XBT OBS
  *    ***',//,20X,'CRUISE NO. =',I3,20X,'STATION NO. =',I3,20X,
  *'COAST DIRECTION =',I3,///,20X,'SURFACE WIND :',20X,'INTENSITY =',
  *F5.1,25X,'DIRECTION =',I3,                 //,20X,'SURFACE WIND COMPO
  *NENTS  :',/,35X,'ALCNG THE COAST :',10X,'INTENSITY =',F5.1,10X,
  *'DIRECTION =',I3,' (+ OR -90)',/,25X,'PERPENDICULAR TO COAST :',
  *10X,'INTENSITY =',F5.1,10X,'DIRECTION =',I3)
16 RETURN
   END
```

APPENDIX G

FORTRAN IV SUBPROGRAM TREND FOR IBM 360/OS

```
C
C     ...................................................................
C
C     SUBROUTINE TREND
C
C     PURPOSE
C        TWO DIMENSIONAL (Z,T) OBJECTIVE ANALYSIS OF A DIGITAL TRACE
C        (BEGINNING AT SEA SURFACE) TO IDENTIFY THE TREND,THE EXTREMA
C        AND THE INVERSIONS
C
C     USAGE
C        CALL TREND(NGP)
C
C     DESCRIPTION OF PARAMETERS
C        NGP   -INPUT NUMBER OF PAIRS (T,Z) DEFINING A TRACE
C
C     REMARKS
C        (1)THIS SUBROUTINE USES 'COMMON' LABELLED AREAS TO RECEIVE
C        INFORMATION FROM THE MAIN PROGRAM AND TO COMUNICATE THE
C        PROCESSED DATA TO THE SUBROUTINE INVERS
C        (2)THIS SUBROUTINE ANALYSIS THE ORIGINAL DIGITAL TRACE OR
C        THE EQUIVALENT MESH
C        (3)THE SETTING UP OF THE CONDITIONS TO DEFINE (A) THE TREND
C        AND (B) THE EXTREMA REQUIRES CONSULTATION OF:
C            SALDANHA,J.M.,1971.'OBJECTIVE DIGITAL ANALYSIS OF A
C            TRACE',M.S.THESIS, NAVAL POSTGRADUATE SCHOOL,
C            MONTEREY,CALIFORNIA  93940
C
C     SUBROUTINES AND FUNCTIONS REQUIRED
C        INVERS
C        MAXO (IBM SCIENTIFIC LIBRARY)
C
C     METHOD
C        ITERATIVE PROCESS TO APPLY THE FINITE DIFFERENCE TECHNIQUE
C        TO THE ORIGINAL DIGITAL TRACE OR MESH
C
C     ...................................................................
C
      SUBROUTINE TREND(NGP)
      COMMON/WORKA/TO(60),ZO(60),BLA37(1332)
      COMMON/WORKF/SEXT(50),VEXT(50),STOP(50),SBOT(50),VISO(50),V1(50),
     *SMAX(50),VMAX(50),SMIN(50),VMIN(50)
C
C     PROCESSING THE FIRST FINITE DIFFERENCES OF T
C
      I=NGP+1
      LQ1=I-NGP
```

```
      LQ2=I-2                                                           TRND0470
      DO 4 N=LQ1,LQ2                                                    TRND0480
      V1(N)=TO(N+1)-TO(N)                                               TRND0490
    4 CONTINUE                                                          TRND0500
C                                                                       TRND0510
C     K: ACTUAL NUMBER OF ISOTHERMAL LAYERS                             TRND0520
C                                                                       TRND0530
      K=1                                                               TRND0540
C                                                                       TRND0550
C     L: ACTUAL NUMBER OF EXTREMA (MAXIMA AND MINIMA)                   TRND0560
C                                                                       TRND0570
      L=2                                                               TRND0580
C                                                                       TRND0590
C     M: ACTUAL NUMBER OF FIRST FINITE DIFFERENCE SEGMENTS USED IN      TRND0600
C        THE ANALYSIS                                                   TRND0610
C                                                                       TRND0620
      M=2                                                               TRND0630
C                                                                       TRND0640
C     ANALYSIS OF EXTREMA AND ISOTHERMAL LAYERS                         TRND0650
C                                                                       TRND0660
    6 IF(V1(M-1).LE.0.) GO TO 10                                        TRND0670
      IF(V1(M).GE.0.) GO TO 10                                          TRND0680
C                                                                       TRND0690
C     (V1(M-1).GT.0) AND (V1(M).LT.0)                                   TRND0700
C                                                                       TRND0710
C     (VEXT,SEXT): POSITION OF MAXIMUM OR MINIMUM                       TRND0720
C                                                                       TRND0730
      IF(L.EQ.2) VEXT(1)=TO(1)                                          TRND0740
      IF(L.EQ.2) SEXT(1)=ZO(1)                                          TRND0750
C                                                                       TRND0760
C     J=5: THE FIRST EXTREMA IS A MAXIMUM AND THE INITIAL POSITION      TRND0770
C        OF THE TRACE IS CONSIDERED A MINIMUM                           TRND0780
C                                                                       TRND0790
      IF(L.EQ.2) J=5                                                    TRND0800
      VEXT(L)=TO(M)                                                     TRND0810
      SEXT(L)=ZO(M)                                                     TRND0820
      L=L+1                                                             TRND0830
      GO TO 40                                                          TRND0840
   10 IF(V1(M-1).GE.0.) GO TO 20                                        TRND0850
      IF(V1(M).LE.0.) GO TO 20                                          TRND0860
C                                                                       TRND0870
C     (V1(M-1).LT.0) AND (V1(M).GT.0)                                   TRND0880
C                                                                       TRND0890
      IF(L.EQ.2) VEXT(1)=TO(1)                                          TRND0900
      IF(L.EQ.2) SEXT(1)=ZO(1)                                          TRND0910
C                                                                       TRND0920
C     J=50: THE FIRST EXTREMA IS A MINIMUM AND THE INITIAL POSITION     TRND0930
C        OF THE TRACE IS CONSIDERED A MAXIMUM                           TRND0940
```

```
C                                                                      TRND0950
      IF(L.EQ.2) J=50                                                  TRND0960
      VEXT(L)=TO(M)                                                    TRND0970
      SEXT(L)=ZO(M)                                                    TRND0980
      L=L+1                                                            TRND0990
      GO TO 40                                                         TRND1000
   20 IF(V1(M-1).LE.0.) GO TO 30                                       TRND1010
      IF(V1(M).NE.0.) GO TO 30                                         TRND1020
C                                                                      TRND1030
C     (V1(M-1).EQ.0) AND (V1(M).EQ.0)                                  TRND1040
C                                                                      TRND1050
C                                                                      TRND1060
C     (VISO,STOP/SBOT): LOCATION OF ISOTHERMAL LAYER                   TRND1070
C                                                                      TRND1080
      STOP(K)=ZO(M)                                                    TRND1090
      VISO(K)=TO(M)                                                    TRND1100
   24 M=M+1                                                            TRND1110
      IF(M.GT.LQ2) GO TO 50                                            TRND1120
      IF(V1(M).GT.0.) GO TO 39                                         TRND1130
      IF(V1(M).LT.0.) GO TO 26                                         TRND1140
      GO TO 24                                                         TRND1150
   26 IF(L.EQ.2) VEXT(1)=TO(1)                                         TRND1160
      IF(L.EQ.2) SEXT(1)=ZO(1)                                         TRND1170
      IF(L.EQ.2) J=5                                                   TRND1180
      VEXT(L)=TO(M)                                                    TRND1190
      SEXT(L)=ZO(M)                                                    TRND1200
      L=L+1                                                            TRND1210
      GO TO 39                                                         TRND1220
   30 IF(V1(M-1).GE.0.) GO TO 37                                       TRND1230
      IF(V1(M).NE.0.) GO TO 37                                         TRND1240
      STOP(K)=ZO(M)                                                    TRND1250
      VISO(K)=TO(M)                                                    TRND1260
   34 M=M+1                                                            TRND1270
      IF(M.GT.LQ2) GO TO 50                                            TRND1280
      IF(V1(M).LT.0.) GO TO 39                                         TRND1290
      IF(V1(M).GT.0.) GO TO 36                                         TRND1300
      GO TO 34                                                         TRND1310
   36 IF(L.EQ.2) VEXT(1)=TO(1)                                         TRND1320
      IF(L.EQ.2) SEXT(1)=ZO(1)                                         TRND1330
      IF(L.EQ.2) J=50                                                  TRND1340
      VEXT(L)=TO(M)                                                    TRND1350
      SEXT(L)=ZO(M)                                                    TRND1360
      L=L+1                                                            TRND1370
      GO TO 39                                                         TRND1380
   37 IF(V1(M-1).NE.0.) GO TO 40                                       TRND1390
      IF(V1(M).NE.0.) GO TO 40                                         TRND1400
      STOP(K)=ZO(M)                                                    TRND1410
      VISO(K)=TO(M)                                                    TRND1420
```

```
   38 M=M+1
      IF(M.GT.LQ2) GO TO 50
      IF(V1(M).EQ.0.) GO TO 38
   39 SBOT(K)=ZO(M)
      K=K+1
   40 M=M+1
      IF(M.GT.NGP) GO TO 50
      GO TO 6
   50 IF(L.LE.2) GO TO 95
      VEXT(L)=TC(NGP)
      SEXT(L)=ZC(NGP)
C
C     SELECTING THE EXTREMA IN MAXIMA (VMAX,SMAX) OR MINIMA (VMIN,
C        SMIN)
C
      I1=1
   80 N1=2*(I1-1)+1
      IF(N1.GT.L) GO TO 85
      IF(J.EQ.5) VMIN(I1)=VEXT(N1)
      IF(J.EQ.5) SMIN(I1)=SEXT(N1)
      IF(J.EQ.50) VMAX(I1)=VEXT(N1)
      IF(J.EQ.50) SMAX(I1)=SEXT(N1)
      I1=I1+1
      GO TO 80
   85 I1=I1-1
      I2=1
   88 N2=2*I2
      IF(N2.GT.L) GO TO 90
      IF(J.EQ.5) VMAX(I2)=VEXT(N2)
      IF(J.EQ.5) SMAX(I2)=SEXT(N2)
      IF(J.EQ.50) VMIN(I2)=VEXT(N2)
      IF(J.EQ.50) SMIN(I2)=SEXT(N2)
      I2=I2+1
      GC TO 88
   90 I2=I2-1
      N5=MAX0(I1,I2)
      WRITE(6,91)(I3,VMIN(I3),SMIN(I3),I3,VMAX(I3),SMAX(I3),I3=1,N5)
   91 FCRMAT('0',10X,'MIN (',I2,') LOCATED AT (',F5.2,',',F6.2,')',/,
     *11X,'MAX (',I2,') LCCATED AT (',F5.2,',',F6.2,')')
      GO TO 100
   95 DV=TC(NGP)-TO(1)
      IF(DV.LT.0.) GO TO 96
      IF(DV.GT.0.) GO TO 97
      WRITE(6,101)
  101 FORMAT('0',10X,'TREND : V CONSTANT')
      GC TO 100
   96 WRITE(6,102)
  102 FORMAT('0',10X,'TREND : V DECREASES')
```

```
      GO TO 100
   97 WRITE(6,103)
  103 FORMAT('0',10X,'TREND : V INCREASES')
  100 K=K-1
C
C     ANALYSIS OF INVERSIONS
C
      CALL INVERS(L,J,NGP,K)
      RETURN
      END
C
C     ...................................................................
C
C     SUBROUTINE INVERS
C
C     PURPOSE
C        TWO DIMENSIONAL (Z,T) OBJECTIVE ANALYSIS OF A DIGITAL TRACE
C        (BEGINNING AT SEA SURFACE) TO IDENTIFY THE INVERSIONS
C
C     USAGE
C        CALL INVERS(L,J,NGP,K)
C
C     DESCRIPTION OF PARAMETERS
C        L      -INPUT VALUE CONTAINING THE NUMBER OF EXTREMA  OF THE
C                TRACE
C        J      -INPUT VALUE CONTAINING AN INFORMATION ABOUT THE FIRST
C                OF THE EXTREMA:(A) MAXIMUM (J=5) OR (B) MINIMUM (J=50)
C        NGP    -INPUT NUMBER OF PAIRS (T,Z) DEFINING A TRACE
C        K      -INPUT VALUE CONTAINING THE NUMBER OF ISOTHERMAL LAYERS
C                IN THE TRACE
C
C     REMARKS
C        (1)THIS SUBROUTINE USES 'COMMON' LABELLED AREAS TO RECEIVE
C        INFORMATION FROM THE MAIN PROGRAM AND THE SUBROUTINE TREND
C        (2)THIS SUBROUTINE ANALYSIS THE ORIGINAL DIGITAL TRACE OR
C        THE EQUIVALENT MESH
C        (3)THE SETTING UP OF THE CONDITIONS TO DEFINE THE INVERSIONS
C        REQUIRES CONSULTATION OF:
C            SALDANHA,J.M.,1971.'OBJECTIVE DIGITAL ANALYSIS OF A
C            TRACE',M.S.THESIS, NAVAL POSTGRADUATE SCHOOL,
C            MONTEREY,CALIFORNIA  93940
C
C     SUBROUTINES AND FUNCTIONS REQUIRED
C        ABS (IBM SCIENTIFIC LIBRARY)
C        LINE
C
C     METHOD
C        ITERATIVE PROCESS TO APPLY THE FINITE DIFFERENCE TECHNIQUE
```

```
C        TO THE ORIGINAL DIGITAL TRACE OR MESH
C
C     .......................................................................
C

      SUBROUTINE INVERS(L,J,NGP,K)
      CCMMON/WORKA/TO(60),ZO(60),BLA38(1332)
      CCMMCN/WORKF/SEXT(50),VEXT(50),STOP(50),SBOT(50),VISO(50),V1(50),
     *VMXI(25),VMNI(25),STXI(25),SBXI(25),STNI(25),SBNI(25),BLA39(50)
      I2=1
      L2=1
C
C     BEGINNING OF THE ITERATIVE PROCESS
C
      IF(L.EQ.2) GO TO 98
      IF(J.EQ.5) I2=2
    2 IF(J.NE.5) GO TO 4
      N3=2*I2-1
      GC TO 6
    4 IF(J.NE.50) GO TO 100
      N3=2*I2
    6 N2=N3-1
      N4=N3+1
      IF(N4.GT.L) GO TO 120
C
C     DEFINING THE FIRST FINITE DIFFERENCES OF THE EXTREMA
C
      V1=ABS(VEXT(N2)-VEXT(N3))
      V2=ABS(VEXT(N4)-VEXT(N3))
      IF(V1.GE.V2) GO TO 34
C
C     V1.LT.V2
C
C
C     VMXI: MAXIMUM THERMAL VALUE OF THE INVERSION
C
      VMXI(L2)=VEXT(N2)
C
C     STXI: DEPTH OF THE TOP OF THE MAXIMUM THERMAL VALUE OF INVERSION
C
      STXI(L2)=SEXT(N2)
C
C     VMNI: MINIMUM THERMAL VALUE OF THE INVERSION
C
      VMNI(L2)=VEXT(N3)
C
C     DEPTH OF THE BOTTOM OF THE MINIMUM THERMAL VALUE OF INVERSION
```

```
C
      SBNI(L2)=SEXT(N3)
      IF(K.EQ.1) GO TO 16
      K1=K
      I4=1
   10 IF(I4.GT.K1) GO TO 16
      IF(VMNI(L2).NE.VISO(I4)) GO TO 15
      IF(SBNI(L2).NE.SBOT(I4)) GO TO 15
C
C     STNI: DEPTH OF THE TOP OF THE MINIMUM THERMAL VALUE OF INVERSION

      STNI(L2)=STOP(I4)
      GO TO 17
   15 I4=I4+1
      GO TO 10
   16 STNI(L2)=SBNI(L2)
   17 I5=1
      N1=1
   18 IF(I5.GT.NGP) GO TO 120
      IF(SBNI(L2).NE.ZO(I5)) GO TO 22
      IF(VMNI(L2).NE.TO(I5)) GO TO 22
   20 IF(VMXI(L2).LE.TO(I5+N1)) GO TO 21
      N1=N1+1
      GO TO 20
   21 A2=TO(I5+N1)
      B2=ZO(I5+N1)
      A1=TO(I5+N1-1)
      B1=ZO(I5+N1-1)
      GO TO 28
   22 I5=I5+1
      GO TO 18
C
C     SBXI: DEPTH OF THE BOTTOM OF THE MAXIMUM THERMAL VALUE OF
C        INVERSION
C
   28 CALL LINE(A1,B1,A2,B2,VEXT(N2),SBXI(L2))
C
C     DSXI: RANGE OF THE INVERSION
C
      DSXI=SBXI(L2)-STXI(L2)
C
C     DVMI: THERMAL WIDTH OF THE INVERSION
C
      DVMI=ABS(VMXI(L2)-VMNI(L2))
      WRITE(6,30)L2,DSXI,VMXI(L2),STXI(L2),VMXI(L2),SBXI(L2)
   30 FORMAT('0',10X,'INVERSION NO. ',I2,/,15X,'RANGE =',F6.2,/,15X,
     *'DEFINING POSITIONS : ',10X,'(',F5.2,',',F6.2,')',10X,'(',F5.2,
     *',',F6.2,')')
```

```
      WRITE(6,32)DVMI,VMNI(L2),STNI(L2),VMNI(L2),SBNI(L2)                 INVR1330
   32 FORMAT(/,15X,'THERMAL WIDTH =',F6.2,/,15X,'MINIMUM THERMAL VALUE DINVR1340
     *EFINED BY THE POSITIONS : ',10X,'(',F5.2,',',F6.2,')',10X,'(',      INVR1350
     *F5.2,',',F6.2,')')                                                  INVR1360
      GO TO 80                                                            INVR1370
   34 IF(ABS(V1-V2).LE.0.005) GO TO 50                                    INVR1380
C                                                                         INVR1390
C     V1.GT.V2                                                            INVR1400
C                                                                         INVR1410
      VMXI(L2)=VEXT(N4)                                                   INVR1420
      VMNI(L2)=VEXT(N3)                                                   INVR1430
      SBNI(L2)=SEXT(N3)                                                   INVR1440
      IF(K.EQ.1) GO TO 36                                                 INVR1450
      I6=1                                                                INVR1460
      K1=K                                                                INVR1470
   37 IF(I6.GT.K1) GO TO 36                                               INVR1480
      IF(VMNI(L2).NE.VISO(I6)) GO TO 35                                   INVR1490
      IF(SBNI(L2).NE.SBOT(I6)) GO TO 35                                   INVR1500
      STNI(L2)=STOP(I6)                                                   INVR1510
      GO TO 38                                                            INVR1520
   35 I6=I6+1                                                             INVR1530
      GO TO 37                                                            INVR1540
   36 STNI(L2)=SBNI(L2)                                                   INVR1550
   38 N2=1                                                                INVR1560
      I10=1                                                               INVR1570
   47 IF(I10.GT.NGP) GO TO 120                                            INVR1580
      IF(STNI(L2).NE.ZO(I10)) GO TO 39                                    INVR1590
      IF(VMNI(L2).NE.TO(I10)) GO TO 39                                    INVR1600
   45 IF(VMXI(L2).LE.TO(I5-N1)) GO TO 46                                  INVR1610
      N2=N2+1                                                             INVR1620
      GO TO 45                                                            INVR1630
   46 D1=TO(I5-N2)                                                        INVR1640
      F1=ZO(I5-N2)                                                        INVR1650
      D2=TO(I5-N2+1)                                                      INVR1660
      F2=ZO(I5-N2+1)                                                      INVR1670
      GO TO 41                                                            INVR1680
   39 I10=I10+1                                                           INVR1690
      GO TO 47                                                            INVR1700
   41 CALL LINE(D1,F1,D2,F2,VEXT(N4),STXI(L2))                            INVR1710
   40 IF(K.EQ.1) GO TO 43                                                 INVR1720
      I7=1                                                                INVR1730
      K1=K                                                                INVR1740
   49 IF(I7.GT.K1) GO TO 43                                               INVR1750
      IF(VEXT(N4).NE.VISO(I7)) GO TO 42                                   INVR1760
      IF(SEXT(N4).NE.SBOT(I7)) GO TO 42                                   INVR1770
      SBXI(L2)=STOP(I7)                                                   INVR1780
      GO TO 44                                                            INVR1790
   42 I7=I7+1                                                             INVR1800
```

```
      GO TO 49                                                          INVR1810
   43 SBXI(L2)=SEXT(N4)                                                 INVR1820
   44 DSXI=SBXI(L2)-STXI(L2)                                            INVR1830
      DVMI=ABS(VMXI(L2)-VMNI(L2))                                       INVR1840
      WRITE(6,30)L2,DSXI,VMXI(L2),STXI(L2),VMXI(L2),SBXI(L2)            INVR1850
      WRITE(6,32)DVMI,VMNI(L2),STNI(L2),VMNI(L2),SBNI(L2)               INVR1860
      GO TO 80                                                          INVR1870
C                                                                       INVR1880
C     V1.EQ.V2                                                          INVR1890
C                                                                       INVR1900
   50 VMXI(L2)=VEXT(N2)                                                 INVR1910
      STXI(L2)=VEXT(N2)                                                 INVR1920
      VMNI(L2)=VEXT(N3)                                                 INVR1930
      SBNI(L2)=SEXT(N3)                                                 INVR1940
      IF(K.EQ.1) GO TO 56                                               INVR1950
      I8=1                                                              INVR1960
      K1=K                                                              INVR1970
   53 IF(I8.GT.K1) GO TO 56                                             INVR1980
      IF(VMNI(L2).NE.VISO(I8)) GO TO 55                                 INVR1990
      IF(SBNI(L2).NE.SBOT(I8)) GO TO 55                                 INVR2000
      STNI(L2)=STOP(I8)                                                 INVR2010
      GO TO 58                                                          INVR2020
   55 I8=I8+1                                                           INVR2030
      GO TO 53                                                          INVR2040
   56 STNI(L2)=SBNI(L2)                                                 INVR2050
   58 IF(K.EQ.1) GO TO 63                                               INVR2060
      K1=K                                                              INVR2070
      I9=1                                                              INVR2080
   60 IF(I9.GT.K1) GO TO 63                                             INVR2090
      IF(VEXT(N4).NE.VISO(I9)) GO TO 62                                 INVR2100
      IF(SEXT(N4).NE.SBOT(I9)) GO TO 62                                 INVR2110
      SBXI(L2)=STOP(I9)                                                 INVR2120
      GO TO 65                                                          INVR2130
   62 I9=I9+1                                                           INVR2140
      GO TO 60                                                          INVR2150
   63 SBXI(L2)=SEXT(N4)                                                 INVR2160
   65 DSXI=SBXI(L2)-STXI(L2)                                            INVR2170
      DVMI=ABS(VMXI(L2)-VMNI(L2))                                       INVR2180
      WRITE(6,30)L2,DSXI,VMXI(L2),STXI(L2),VMXI(L2),SBXI(L2)            INVR2190
      WRITE(6,32)DVMI,VMNI(L2),STNI(L2),VMNI(L2),SBNI(L2)               INVR2200
   80 L2=L2+1                                                           INVR2210
      I2=I2+1                                                           INVR2220
      GO TO 2                                                           INVR2230
C                                                                       INVR2240
C     END OF THE ITERATIVE PROCESS                                      INVR2250
C                                                                       INVR2260
   98 WRITE(6,99)                                                       INVR2270
   99 FORMAT('0',10X,'L=2 : NO INVERSIONS')                             INVR2280
```

```
      GO TO 120                                                         INVR2290
  100 WRITE(6,101)                                                      INVR2300
  101 FORMAT('0',10X,'J IS UNDEFINED')                                  INVR2310
  120 WRITE(6,121)                                                      INVR2320
  121 FORMAT('0',10X,'END OF INVERSION ANALYSIS')                       INVR2330
      RETURN                                                            INVR2340
      END                                                               INVR2350
```

APPENDIX H

FORTRAN IV SUBPROGRAM PROPAR FOR IBM 360/OS

```
C
C     ..................................................................
C
C        SUBROUTINE PROPAR
C
C        PURPOSE
C           TO COMPUTE THE FRICTION VELOCITY,THE ROUGHNESS PARAMETER AND
C           STABILITY LENGTH FROM WIND SPEED AND AIR-SEA TEMPERATURE
C           DIFFERENCE DATA
C
C        USAGE
C           CALL PROPAR(UM,ZM,TD,ZTM,VST,ZO,SLN)
C
C        DESCRIPTION OF PARAMETERS
C           UM     -INPUT VALUE CONTAINING THE WIND SPEED (FT/SEC)
C           ZM     -INPUT VALUE CONTAINING THE ANEMOMETER HEIGHT (FT)
C           TD     -INPUT VALUE CONTAINING THE AIR-SEA TEMPERATURE
C                    DIFFERENCE (CELSIUS)
C           ZTM    -INPUT VALUE CONTAINING THE THERMOMETER HEIGHT (FT)
C           VST    -OUTPUT VALUE CONTAINING THE FRICTION VELOCITY (FT/SEC)
C           ZO     -OUTPUT VALUE CONTAINING THE ROUGHNESS PARAMETER (FT)
C           SLN    -OUTPUT VALUE CONTAINING THE STABILITY LENGTH (FT)
C
C        REMARKS
C           TO NOTICE THE UNITS OF THE PARAMETERS
C
C        SUBROUTINES AND FUNCTIONS REQUIRED
C           PSI(P)
C           SHP(PS)
C           ABS (IBM SCIENTIFIC LIBRARY)
C           LOG (IBM SCIENTIFIC LIBRARY)
C
C        METHOD AND SOURCE
C           CARDONE,V.J.,1969.'SPECIFICATION OF THE WIND DISTRIBUTION IN
C           THE MARINE BOUNDARY LAYER FOR WAVE FORECASTING'.NEW YORK
C           UNIVERSITY,DEPARTMENT OF METEOROLOGY AND OCEANOGRAPHY
C
C     ..................................................................
C

      SUBROUTINE PROPAR(UM,ZM,TD,ZTM,VST,ZO,SLN)
     DATA A/7.3627E-04/B/1.3045E-03/C/-1.4534E-03/CF/54.3478/
      VST=.04*UM
      IF(ABS(TD).GT.1.) GO TO 2000
 1000 VSTN=(.4*UM)/(LOG(ZM/(A/VST+B*VST**2+C)))
      IF(ABS(VSTN-VST).LT..05) GO TO 1400
 1200 VST=VSTN
```

```
C                                                                        PRPR0470
C     FOR NEUTRAL CONDITIONS 'SLN' RETURNS 0                             PRPR0480
C                                                                        PRPR0490
 1400 SLN=0.                                                             PRPR0500
      GO TO 12                                                           PRPR0510
 2000 SLG=VST**2*CF*(LOG(ZTM/(A/VST+B*VST**2+C)))/TD                     PRPR0520
    1 VSTN=(.4*UM)/(LOG(ZM/(A/VST+B*VST**2+C))-PSI(ZM/SLG))             PRPR0530
      IF(ABS(VSTN-VST).LT..05) GO TO 4                                   PRPR0540
    2 VST=VSTN                                                           PRPR0550
      GO TO 1                                                            PRPR0560
    4 SL=SLG                                                             PRPR0570
    5 SLN=CF*VSTN**2*(LOG(ZTM/(A/VSTN+B*VSTN**2+C))-PSI(ZTM/SL))/TD      PRPR0580
      IF(ABS(SLN-SL).LT.1.) GO TO 8                                      PRPR0590
    6 SL=SLN                                                             PRPR0600
      GO TO 5                                                            PRPR0610
    8 IF(ABS(SLN-SLG).LT.1.) GO TO 12                                    PRPR0620
      SLG=SLN                                                            PRPR0630
      GO TO 1                                                            PRPR0640
   12 ZO=A/VST+B*VST**2+C                                                PRPR0650
      RETURN                                                             PRPR0660
      END                                                                PRPR0670
```

APPENDIX I

FORTRAN IV SUBPROGRAM ITRATE FOR IBM 360/OS

```
C                                                                       ITRT0010
C     ......................................................................ITRT0020
C                                                                       ITRT0030
C     SUBROUTINE ITRATE                                                 ITRT0040
C                                                                       ITRT0050
C     PURPOSE                                                           ITRT0070
C        TO COMPUTE THE FRICTION VELOCITY,THE INFLOW ANGLE,THE          ITRT0070
C        ROUGHNESS PARAMETER AND THE STABILITY LENGTH FROM THE LARGE    ITRT0080
C        SCALE SYNOPTIC PARAMETERS                                      ITRT0090
C                                                                       ITRT0100
C     USAGE                                                             ITRT0110
C        CALL ITRATE(G,F,TD,TH,ETAH,VST,PHI,ZO,PARAM)                   ITRT0120
C                                                                       ITRT0130
C     DESCRIPTION OF PARAMETERS                                         ITRT0140
C        G     -INPUT VALUE CONTAINING THE GEOSTROPHIC WIND SPEED (FT/  ITRT0150
C               /SEC)                                                   ITRT0160
C        F     -INPUT VALUE CONTAINING THE CORIOLIS PARAMETER (1/SEC)   ITRT0170
C        TD    -INPUT VALUE CONTAINING THE AIR-SEA TEMPERATURE          ITRT0180
C               DIFFERENCE (CELSIUS)                                    ITRT0190
C        TH    -INPUT VALUE CONTAINING THE DIMENSIONLESS THERMAL WIND   ITRT0200
C               MAGNITUDE                                               ITRT0210
C        ETAH  -INPUT VALUE CONTAINING THE ANGLE BETWEEN THE            ITRT0220
C               GEOSTROPHIC WIND AND THE THERMAL WIND                   ITRT0230
C        VST   -OUTPUT VALUE CONTAINING THE FRICTION VELOCITY (FT/SEC)  ITRT0240
C        PHI   -OUTPUT VALUE CONTAINING THE INFLOW ANGLE (RADIANS)      ITRT0250
C        ZO    -OUTPUT VALUE CONTAINING THE ROUGHNESS PARAMETER (FT)    ITRT0260
C        PARAM -OUTPUT VALUE CONTAINING THE STABILITY LENGTH (FT)       ITRT0270
C                                                                       ITRT0280
C     REMARKS                                                           ITRT0290
C        TO NOTICE THE UNITS OF THE PARAMETERS                          ITRT0300
C                                                                       ITRT0310
C     SUBROUTINES AND FUNCTIONS REQUIRED                                ITRT0320
C        PSI(P)                                                         ITRT0330
C        SHR(PS)                                                        ITRT0340
C        ABS (IBM SCIENTIFIC LIBRARY)                                   ITRT0350
C        LOG (IBM SCIENTIFIC LIBRARY)                                   ITRT0360
C        SQRT (IBM SCIENTIFIC LIBRARY)                                  ITRT0370
C        ASIN (IBM SCIENTIFIC LIBRARY)                                  ITRT0380
C        ATAN (IBM SCIENTIFIC LIBRARY)                                  ITRT0390
C        SIN (IBM SCIENTIFIC LIBRARY)                                   ITRT0400
C        COS (IBM SCIENTIFIC LIBRARY)                                   ITRT0410
C                                                                       ITRT0420
C     METHOD AND SOURCE                                                 ITRT0430
C        CARDONE,V.J.,1969.'SPECIFICATION OF THE WIND DISTRIBUTION IN   ITRT0440
C        THE MARINE BOUNDARY LAYER FOR WAVE FORECASTING'.NEW YORK       ITRT0450
C        UNIVERSITY,DEPARTMENT OF METEOROLOGY AND OCEANOGRAPHY          ITRT0460
C                                                                       ITRT0470
C     ......................................................................ITRT0480
```

```
C
      SUBROUTINE ITRATE(G,F,TD,TH,ETAH,VST,PHI,ZO,PARAM)
      DATA C1/7.3627E-04/C2/1.3045E-03/C3/-1.4534E-03/B/3.E-04/TA/280./
      VST=.0245*G
      T=0.
      ETA=0.
      INDEX=1
      IF(ABS(TD).GT.1.) GO TO 4
    3 ZONL=0.
      PARAM=0.
      FI=0.
      S=1.
      GO TO 1110
    4 PARAM=TA*VST**2/(.16*32.2*.1*TD)
    6 ZONL=((B*G)/F)/PARAM
      FI=PSI(ZONL)
      S=SHR(ZONL)
 1110 CPA=(*G/(F*S))*(LOG(B*G/(F*(C1/VST+C2*VST**2+C3)))-FI)
      BET=SQRT(S*F**2/(2*VST*.4*B*G))
      P=SQRT(1.+2.*CPA**2*BET**2+2.*CPA*BET)
      R=SQRT(1.+(F**2*T**2*CPA**2)/G**2-(2.*F*T*CPA*COS(ETA))/G)
      ALPH=ASIN(CPA*BET/P)
      ARGUM=(F*T*CPA*SIN(ETA)/(R*G))
      GAMMA=ASIN(ARGUM)
 1114 PHI=ATAN(SIN(ALPH+GAMMA)/(P/R-COS(ALPH+GAMMA)))
      VSTN=((.4*G*R*SIN(ALPH+GAMMA))/(P*SIN(PHI)))/(LOG(B*G/F*(C1/VST+
     *C2*VST**2+C3)))-FI)
      TEST=ABS(VSTN-VST)
      IF(TEST-.05)1130,1130,1120
 1120 VST=VSTN
      GO TO 1110
 1130 IF(ZONL-0.)1140,1150,1140
 1140 FE=PSI(33./PARAM)
      PARMNU=(VST**2TA*(LOG(33./(C1/VST+C2*VST**2+C3))-FE))/(5.15*TD)
      IF(ABS(PARMNU-PARAM).LT.5) GO TO 1150
 1142 PARAM=PARAMNU
      GO TO 6
 1150 IF(INDEX-2)1152,1155,1152
 1152 T=TH
      ETA=ETAH
      INDEX=2
      GO TO 1110
 1155 ZO=C1/VST+C2*VST**2+C3
      RETURN
      END
C
```

```
C     ..................................................................PSI 0020
C                                                                       PSI 0030
C     FUNCTION PSI(P)                                                   PSI 0040
C                                                                       PSI 0050
C     ..................................................................PSI 0060
C                                                                       PSI 0070

      FUNCTION PSI(P)                                                   PSI 0080
      IF(P.GT.0.) GO TO 40                                              PSI 0090
    5 S=SHR(P)                                                          PSI 0100
      PSI=1.-S-3.*LOG(S)+2.*LOG((1.+S)/2)+2.*ATAN(S)-1.5708+            PSI 0110
     *LOG((1.+S**2)/2.)                                                 PSI 0120
      GO TO 50                                                          PSI 0130
   40 PSI=-7.*P                                                         PSI 0140
   50 RETURN                                                            PSI 0150
      END                                                               PSI 0160
C                                                                       SHR 0010
C     ..................................................................SHR 0020
C                                                                       SHR 0030
C     FUNCTION SHR(PS)                                                  SHR 0040
C                                                                       SHR 0050
C     ..................................................................SHR 0060
C                                                                       SHR 0070

      FUNCTION SHR(PS)                                                  SHR 0080
      IF(PS.GT.0.) GO TO 40                                             SHR 0090
    5 RI=PS                                                             SHR 0100
   10 RINEW=PS*(1.-18.*RI)**(1./4.)                                     SHR 0110
      IF(ABS(RINEW-RI).LT..001) GO TO 30                                SHR 0120
   20 RI=RINEW                                                          SHR 0130
      GO TO 10                                                          SHR 0140
   30 SHR=1./(1.-18.*RINEW)**(1./4.)                                    SHR 0150
      GO TO 50                                                          SHR 0160
   40 SHR=1.+7.*PS                                                      SHR 0170
   50 RETURN                                                            SHR 0180
      END                                                               SHR 0190
```

APPENDIX J

FORTRAN IV SUBPROGRAM DRAW FOR IBM 360/OS

```
C                                                                       DRAW0010
C     ..................................................................DRAW0020
C                                                                       DRAW0030
C     SUBROUTINE DRAW                                                   DRAW0040
C                                                                       DRAW0050
C     PURPOSE                                                           DRAW0060
C        TO NEUTRALIZE THE 'IBM SUBROUTINE DRAW'                        DRAW0070
C                                                                       DRAW0080
C     USAGE                                                             DRAW0090
C        CALL DRAW(I,A,B,J,K,LABEL,ITITLE,D,E,K1,K2,L1,L2,L3,L4,L5,     DRAW0100
C        LAST)                                                          DRAW0110
C                                                                       DRAW0120
C     REMARKS                                                           DRAW0130
C        (1)IF THIS SUBROUTINE IS INCLUDED IN A GIVEN PROGRAM, THIS     DRAW0140
C        WON'T HAVE OUTPUT PLOTS                                        DRAW0150
C        (2)IF THIS SUBROUTINE IS INCLUDED IN A GIVEN PROGRAM, THE      DRAW0160
C        OUTPUT VALUE OF 'LAST' WILL BE 9                               DRAW0170
C                                                                       DRAW0180
C     ..................................................................DRAW0190
C                                                                       DRAW0200
C
      SUBROUTINE DRAW(I,A,B,J,K,LABEL,ITITLE,D,E,K1,K2,L1,L2,L3,L4,L5,  DRAW0210
     *LAST)                                                             DRAW0220
      LAST=9                                                            DRAW0230
      RETURN                                                            DRAW0240
      END                                                               DRAW0250
```

MODIFICATION TO THE FORTRAN IV BASIC COMPUTER PROGRAM (APPENDIX F) TO INCLUDE THE SUBPROGRAM PROPAR (APPENDIX I)

```
C                                                                               MAIN0010
C     ..........................................................................MAIN0020
C                                                                               MAIN0030
C                                                                               MAIN0040
C     MAIN PROGRAM                                                              MAIN0050
C                                                                               MAIN0060
C                                                                               MAIN0070
C     .......
C
      COMMON/WORKH/W00(20,12),W01(20,12),W02(20,12),W03(20,12),                 MAIN4681
     *W05(20,12),W10(20,12),ZM00(20,12),ZM01(20,12),ZM02(20,12),                MAIN4682
     *ZM03(20,12),ZM05(20,12),ZM10(20,12),TD00(20,12),TD01(20,12),              MAIN4683
     *TD02(20,12),TD03(20,12),TD05(20,12),TD10(20,12),ZTM00(20,12),             MAIN4684
     *ZTM01(20,12),ZTM02(20,12),ZTM03(20,12),ZTM05(20,12),ZTM10(20,12)          MAIN4685
     *,Z000(20,12),Z001(20,12),Z002(20,12),Z003(20,12),Z005(20,12),             MAIN4686
     *Z010(20,12),SLN00(20,12),SLN01(20,12),SLN02(20,12),SLN03(20,12),          MAIN4687
     *SLN05(20,12),SLN10(20,12)                                                 MAIN4688
C
C     .......
C
C                                                                               MAIN5680
C     PROCESSING THE WIND FIELD                                                 MAIN5690
C                                                                               MAIN5700
      READ(5,310) XGRID,YGRID,KAMMA1,KKK                                        MAIN5710
  310 FORMAT(2F10.0,2I3)                                                        MAIN5720
      READ(5,312) W00(NORDER,ISTA),KANG00(NORDER,ISTA),                         MAIN5730
     *ZM00(NORDER,ISTA),TD00(NORDER,ISTA),ZTM00(NORDER,ISTA)                    MAIN5740
  312 FORMAT(F10.0,I3,3F10.0)                                                   MAIN5750
      CALL PROPAR(W00(NORDER,ISTA),ZM00(NORDER,ISTA),TD00(NORDER,ISTA)          MAIN5760
     *,ZTM00(NORDER,ISTA),R00(NORDER,ISTA),Z000(NORDER,ISTA),                   MAIN5770
     *SLN00(NORDER,ISTA))                                                       MAIN5775
      CALL WIND00(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,          MAIN5780
     *N10)                                                                      MAIN5790
      READ(5,312) W01(NORDER,ISTA),KANG01(NORDER,ISTA),                         MAIN5791
     *ZM01(NORDER,ISTA),TD01(NORDER,ISTA),ZTM01(NORDER,ISTA)                    MAIN5792
      CALL PROPAR(W01(NORDER,ISTA),ZM01(NORDER,ISTA),TD01(NORDER,ISTA)          MAIN5793
     *,ZTM01(NORDER,ISTA),R01(NORDER,ISTA),Z001(NORDER,ISTA),                   MAIN5794
     *SLN01(NORDER,ISTA))                                                       MAIN5795
      CALL WIND01(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,          MAIN5800
     *N10)                                                                      MAIN5810
      READ(5,312) W02(NORDER,ISTA),KANG02(NORDER,ISTA),                         MAIN5811
     *ZM02(NORDER,ISTA),TD02(NORDER,ISTA),ZTM02(NORDER,ISTA)                    MAIN5812
      CALL PROPAR(W02(NORDER,ISTA),ZM02(NORDER,ISTA),TD02(NORDER,ISTA)          MAIN5813
     *,ZTM02(NORDER,ISTA),R02(NORDER,ISTA),Z002(NORDER,ISTA),                   MAIN5814
     *SLN02(NORDER,ISTA))                                                       MAIN5815
      CALL WIND02(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,          MAIN5820
     *N10)                                                                      MAIN5830
      READ(5,312) W03(NORDER,ISTA),KANG03(NORDER,ISTA),                         MAIN5831
```

```
     *ZM03(NORDER,ISTA),TD03(NORDER,ISTA),ZTM03(NORDER,ISTA)          MAIN5832
      CALL PROPAR(W03(NORDER,ISTA),ZM03(NORDER,ISTA),TD03(NORDER,ISTA) MAIN5833
     *,ZTM03(NORDER,ISTA),R03(NORDER,ISTA),Z003(NORDER,ISTA),         MAIN5834
     *SLN03(NORDER,ISTA))                                             MAIN5835
      CALL WIND03(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1, MAIN5840
     *N10)                                                            MAIN5850
      READ(5,312) W05(NORDER,ISTA),KANG05(NORDER,ISTA),               MAIN5851
     *ZM05(NORDER,ISTA),TD05(NORDER,ISTA),ZTM05(NORDER,ISTA)          MAIN5852
      CALL PROPAR(W05(NORDER,ISTA),ZM05(NORDER,ISTA),TD05(NORDER,ISTA) MAIN5853
     *,ZTM05(NORDER,ISTA),R05(NORDER,ISTA),Z005(NORDER,ISTA),         MAIN5854
     *SLN05(NORDER,ISTA))                                             MAIN5855
      CALL WIND05(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1, MAIN5860
     *N10)                                                            MAIN5870
      READ(5,312) W10(NORDER,ISTA),KANG10(NORDER,ISTA),               MAIN5871
     *ZM10(NORDER,ISTA),TD10(NORDER,ISTA),ZTM10(NORDER,ISTA)          MAIN5872
      CALL PROPAR(W10(NORDER,ISTA),ZM10(NORDER,ISTA),TD10(NORDER,ISTA) MAIN5873
     *,ZTM10(NORDER,ISTA),R10(NORDER,ISTA),Z010(NORDER,ISTA),         MAIN5874
     *SLN10(NORDER,ISTA))                                             MAIN5875
      CALL WIND10(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1, MAIN5880
     *N10)                                                            MAIN5890
  450 CONTINUE                                                        MAIN5900
C
C       .......
C
```

APPENDIX L

MODIFICATION TO THE FORTRAN IV BASIC COMPUTER PROGRAM (APPENDIX F) TO INCLUDE THE SUBPROGRAM ITRATE (APPENDIX J)

```
C                                                                         MAIN0010
C     ......................................................................MAIN0020
C                                                                         MAIN0030
C                                                                         MAIN0040
C     MAIN PROGRAM                                                        MAIN0050
C                                                                         MAIN0060
C                                                                         MAIN0070
C
C     .......
C
      COMMON/WORKG/W00(20,12),W01(20,12),W02(20,12),W03(20,12)            MAIN4681
     *W05(20,12),W10(20,12),TD00(20,12),TD01(20,12),TD02(20,12),          MAIN4682
     *TD03(20,12),TD05(20,12),TD10(20,12),TH00(20,12),TH01(20,12),        MAIN4683
     *TH02(20,12),TH03(20,12),TH05(20,12),TH10(20,12),ETAH00(20,12),      MAIN4684
     *ETAH01(20,12),ETAH02(20,12),ETAH03(20,12),ETAH05(20,12),            MAIN4685
     *ETAH10(20,12),PHI00(20,12),PHI01(20,12),PHI02(20,12),PHI03(20,12)   MAIN4686
     *,PHI05(20,12),PHI10(20,12),Z000(20,12),Z001(20,12),Z002(20,12),     MAIN4687
     *Z003(20,12),Z005(20,12),Z010(20,12),PARM00(20,12),PARM01(20,12),    MAIN4688
     *PARM02(20,12),PARM03(20,12),PARM05(20,12),PARM10(20,12),F(12)       MAIN4689
C
C     .......
C
C                                                                         MAIN5680
C     PROCESSING THE WIND FIELD                                           MAIN5690
C                                                                         MAIN5700
      READ(5,310) XGRID,YGRID,KAMMA1,KKK,F(ISTA)                          MAIN5710
  310 FORMAT(2F10.0,2I3,F10.0)                                            MAIN5720
      READ(5,312) W00(NORDER,ISTA),KANG00(NORDER,ISTA),                   MAIN5730
     *TD00(NORDER,ISTA),TH00(NORDER,ISTA),ETAH00(NORDER,ISTA)             MAIN5740
  312 FORMAT(F10.0,I3,3F10.0)                                             MAIN5750
      CALL ITRATE(W00(NORDER,ISTA),F(ISTA),TD00(NORDER,ISTA),             MAIN5760
     *TH00(NORDER,ISTA),ETAH00(NORDER,ISTA),R00(NORDER,ISTA),             MAIN5770
     *PHI00(NORDER,ISTA),Z000(NORDER,ISTA),PARM00(NORDER,ISTA))           MAIN5775
      CALL WIND00(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,    MAIN5780
     *N10)                                                                MAIN5790
      READ(5,312) W01(NORDER,ISTA),KANG01(NORDER,ISTA),                   MAIN5791
     *TD01(NORDER,ISTA),TH01(NORDER,ISTA),ETAH01(NORDER,ISTA)             MAIN5792
      CALL ITRATE(W01(NORDER,ISTA),F(ISTA),TD01(NORDER,ISTA),             MAIN5793
     *TH01(NORDER,ISTA),ETAH01(NORDER,ISTA),R01(NORDER,ISTA),             MAIN5794
     *PHI01(NORDER,ISTA),Z001(NORDER,ISTA),PARM01(NORDER,ISTA))           MAIN5795
      CALL WIND01(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,    MAIN5800
     *N10)                                                                MAIN5810
      READ(5,312) W02(NORDER,ISTA),KANG02(NORDER,ISTA),                   MAIN5811
     *TD02(NORDER,ISTA),TH02(NORDER,ISTA),ETAH02(NORDER,ISTA)             MAIN5812
      CALL ITRATE(W02(NORDER,ISTA),F(ISTA),TD02(NORDER,ISTA),             MAIN5813
     *TH02(NORDER,ISTA),ETAH02(NORDER,ISTA),R02(NORDER,ISTA),             MAIN5814
     *PHI02(NORDER,ISTA),Z002(NORDER,ISTA),PARM02(NORDER,ISTA))           MAIN5815
      CALL WIND02(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,    MAIN5820
     *N10)                                                                MAIN5830
```

```
      READ(5,312) W03(NORDER,ISTA),KANG03(NORDER,ISTA),                 MAIN5831
     *TD03(NORDER,ISTA),TH03(NORDER,ISTA),ETAH03(NORDER,ISTA)           MAIN5832
      CALL ITRATE(W03(NORDER,ISTA),F(ISTA),TD03(NORDER,ISTA),           MAIN5833
     *TH03(NORDER,ISTA),ETAH03(NORDER,ISTA),R03(NORDER,ISTA),           MAIN5834
     *PHI03(NORDER,ISTA),Z003(NORDER,ISTA),PARM03(NORDER,ISTA))         MAIN5835
      CALL WIND03(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,  MAIN5840
     *N10)                                                              MAIN5850
      READ(5,312) W05(NORDER,ISTA),KANG05(NORDER,ISTA),                 MAIN5851
     *TD05(NORDER,ISTA),TH05(NORDER,ISTA),ETAH05(NORDER,ISTA)           MAIN5852
      CALL ITRATE(W05(NORDER,ISTA),F(ISTA),TD05(NORDER,ISTA),           MAIN5853
     *TH05(NORDER,ISTA),ETAH05(NORDER,ISTA),R05(NORDER,ISTA),           MAIN5854
     *PHI05(NORDER,ISTA),Z005(NORDER,ISTA),PARM05(NORDER,ISTA))         MAIN5855
      CALL WIND05(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,  MAIN5860
     *N10)                                                              MAIN5870
      READ(5,312) W10(NORDER,ISTA),KANG10(NORDER,ISTA),                 MAIN5871
     *TD10(NORDER,ISTA),TH10(NORDER,ISTA),ETAH10(NORDER,ISTA)           MAIN5872
      CALL ITRATE(W10(NORDER,ISTA),F(ISTA),TD10(NORDER,ISTA),           MAIN5873
     *TH10(NORDER,ISTA),ETAH10(NORDER,ISTA),R10(NORDER,ISTA),           MAIN5874
     *PHI10(NORDER,ISTA),Z010(NORDER,ISTA),PARM10(NORDER,ISTA))         MAIN5875
      CALL WIND10(NORDER,ISTA,XGRID,YGRID,KKK,N10,LABEL,ITITL4,KAMMA1,  MAIN5880
     *N10)                                                              MAIN5890
  450 CONTINUE                                                          MAIN5900
C
C     .......
C
```

APPENDIX M

MODIFICATION TO THE FORTRAN IV BASIC
COMPUTER PROGRAM (APPENDIX F) TO INCLUDE
THE SUBPROGRAM TREND (APPENDIX G)

```
C                                                                          MAIN0010
C     ......................................................................MAIN0020
C                                                                          MAIN0030
C                                                                          MAIN0040
C     MAIN PROGRAM                                                         MAIN0050
C                                                                          MAIN0060
C                                                                          MAIN0070
C     .......
C
      COMMON/WORKF/SEXT(50),VEXT(50),STOP(50),SPOT(50),VISC(50),V1(50),    MAIN4684
     *SMAX(50),VMAX(50),SMIN(50),VMIN(50)                                  MAIN4686
C
C     .......
C
      WRITE(6,43)(I40,TC(I40),I40,ZC(I40),I40=1,NOP)                       MAIN5470
      CALL TREND(NOP)                                                      MAIN5475
      CALL LINTPS(NOP,DTW,M60)                                             MAIN5480
C
C     .......
C
```

APPENDIX N

JOB CONTROL LANGUAGE (JCL) FOR
THE HEAD PART OF THE FORTRAN IV
BASIC COMPUTER PROGRAM (APPENDIX F)

```
a //SAL10079 JOB (0079,0581FT,OP02),'SALDANHA',TIME=(5,)
b //      EXEC FORTCLGP,REGION.FORT=150K,REGION.GO=300K,TIME.GO=(5,)
c //FORT.SYSPRINT DD SYSOUT=A,SPACE=(CYL,(3,1))
d //FORT.SYSIN DD DSNAME=SSP3(MISR),DISP=SHR
e //            DD *
```

<u>APPENDIX O</u>

JOB CONTROL LANGUAGE (JCL) FOR THE LAST PART OF THE FORTRAN IV BASIC COMPUTER PROGRAM (APPENDIX F)

```
a //GO.FTC6F001  DD SPACE=(CYL,(20,3))
b //GO.FTC9F001  DD DSN=S0079.DATA,UNIT=2314,VOL=SER=DUFFY,
c //                DISP=(OLD,KEEP),DCB=(RECFM=VBS,BLKSIZE=3500),
d //                SPACE=(CYL,1),LABEL=EXPDT=71274
e //GO.SYSIN  DD *
```

APPENDIX P

JOB CONTROL LANGUAGE (JCL) FOR THE LAST PART OF THE FORTRAN IV BASIC COMPUTER PROGRAM (APPENDIX F)

```
a //GO.FT06F001 DD DUMMY
b //GO.FT09F001 DD DSN=S0079.DATA,UNIT=2314,VOL=SER=DUFFY,
c //               DISP=(OLD,KEEP),DCB=(RECFM=VBS,BLKSIZE=3500),
d //               SPACE=(CYL,1),LABEL=EXPDT=71274
e //GO.SYSIN  DD *
```

BIBLIOGRAPHY

Boston, N. E. J., 1966. Objective Definition of the Thermocline, Texas A&M University, Dept. of Oceanog., Project 286 - D, Ref. 66 - 19T, p. 1-38.

Cardone, V. J., 1969. Specification of the Wind Distribution in the Marine Boundary Layer for Wave Forecasting, New York University, Dept. of Meteorology and Oceanography.

Francis, S. A., and G. C. Campbell, 1965. "A Low Cost Expendable Bathythermograph", Marine Science Instr., Vol 3, Plenum Press, p. 85-89.

Grosfils, E. F., 1968. Objective Digital Analysis of Bathythermograph Traces, M.S. Thesis, Naval Postgraduate School, Monterey, California.

Hale, D. H., and P. D. Stevens, 1970. Computer Processing of Expendable Bathythermograph Traces, Fleet Numerical Weather Central, Tech. Note 61.

James, R. W., 1966. Ocean Thermal Structure Forecasting, NavOceanO, Sp - 105, ASWEPS Manual, V. 5.

LaFond, E. C., 1951. Processing Oceanographic Data, U. S. Navy Hydrographic Office, H.O. Pub. No. 614, p. 1-187.

Magruder, Jr., P. M., 1970. Some Characteristics of Temperature Microstructure in the Ocean, M.S. Thesis, Naval Postgraduate School, Monterey, California.

Rossby, C. G., and R. B. Montgomery, 1935. "The Layer of Frictional Influence in Wind and Ocean Currents", Papers Phys. Oceanog. Meteorol., 3(3), p. 1-101.

Sauer, C. D., 1964. "Bathythermograph Data on Aperture Cards: a New Approach to an Old Problem", Journal of Fisheries Research Board Canada, 21(3), p. 647-650.

Sauer, C. D., and E. L. V. Hope, 1967. "A System to Digitize Bathythermograph Aperture Cards", Journal of Fisheries Research Board Canada, 24(5), p. 1155-1164.

Spilhaus, A. F., 1938. "A Bathythermograph", Journal of Geophysical Research, v. 1, p. 95-100.

Sverdrup, H. U., Johnson, M. W., and Fleming, R. H., 1942. The Oceans, Prentice-Hall.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22134	2
2.	Library Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Naval Weather Service Command Washington Navy Yard Washington, D. C. 20390	1
4.	Commanding Officer Fleet Numerical Weather Central Monterey, California 93940	2
5.	Office of Naval Research, Code 480D Department of the Navy Washington, D. C. 20360	1
6.	Department of Oceanography, Code 58 Naval Postgraduate School Monterey, California 93940	3
7.	Department of Meteorology, Code 51 Naval Postgraduate School Monterey, California 93940	1
8.	National Oceanographic Data Center Attn: Henry Odom Washington, D. C. 20390	1
9.	Director-Geral Instituto Hidrografico Ministerio da Marinha Lisboa, Portugal	2
10.	Director Instituto Hidrografico Cadiz, Espana	1
11.	Director Instituto de Pesquisas da Marinha Ministerio da Marinha Ilha do Governador Rio de Janeiro, Brasil	1

12. Director — 1
Hidrografia e Navegacao
Ministerio da Marinha
Rio de Janeiro, Brasil

13. Biblioteca Central — 1
Ministerio da Marinha
Lisboa, Portugal

14. Director — 1
Servico Meteorologico Nacional
Lisboa, Portugal

15. Dr. Dale F. Leipper — 1
Department of Oceanography, Code 58Lr
Naval Postgraduate School
Monterey, California 93940

16. Dr. Taivo Laevastu — 1
Fleet Numerical Weather Central
Monterey, California 93940

17. Dr. Noel Boston — 5
Department of Oceanography, Code 58Bb
Naval Postgraduate School
Monterey, California 93940

18. Dr. Kenneth Davidson — 5
Department of Meteorology, Code 51Ds
Naval Postgraduate School
Monterey, California 93940

19. Dr. Robert Andrews — 1
Department of Oceanography, Code 58Ad
Naval Postgraduate School
Monterey, California 93940

20. Dr. Douglas R. McLain — 1
National Marine Fisheries Service
Fleet Numerical Weather Service
Monterey, California 93940

21. CDR Jose Ataide — 1
Instituto Hidrografico
Ministerio da Marinha
Lisboa, Portugal

22. CDR Mauro Pereira — 1
Rua Desembargador Isidro, n. 10, Apt. C01
Tijuca, Rio de Janeiro, Brasil

23. LCDR Almir Silva 1
Directoria de Hidrografia e Navegacao
Ministerio da Marinha
Rio de Janeiro, Brasil

24. LCDR Jaime Sanchez 1
ARC "San Andres"
Base Naval Cartagena
Colombia, South America

25. LCDR Leopoldo Salas 1
9 Ralston Drive
Monterey, California 93940

26. LCDR Roberto Rodrigues 1
777 Newton Street
Monterey, California 93940

27. LT Alexander Maratos 1
Hydrographic Office of Navy
Paparrigopoulou 2
Athens, Greece

27. LT Antonio Souto 1
261 Larkin Street
Monterey, California 93940

29. LT Eric Grosfils 1
P. O. Box 502
Brewster, Massachusetts 02631

30. LT Jose Saldanha 3
Instituto Hidrografico
Ministerio da Marinha
Lisboa, Portugal

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

RIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

EPORT TITLE

An Objective Method to Analyze Oceanic Variclines and Their Statistical Relation to Surface Winds

ESCRIPTIVE NOTES (Type of report and, inclusive dates)

Master's Thesis; September 1971

UTHOR(S) (First name, middle initial, last name)

Jose Manuel Pinto Bastos Saldanha

EPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
September 1971	210	13

CONTRACT OR GRANT NO. PROJECT NO.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)

DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

ABSTRACT

Oceanographic and meteorologic data, namely expendable or mechanical bathythermograph traces and wind reports have been systematically collected for years and every day more observations are made.

To handle such a large amount of information, it is advantageous to set up a computerized model which is: (1) automatic, (2) performs an objective analysis of XBT/BT traces to identify the most significant thermal features, such as thermoclines and inversions, (3) restricts the human participation to an initial stage dealing with organization of data available and selection of proper values for the parameters of the model, and (4) preserves the initial information.

A FORTRAN IV program takes into consideration the above requirements to: (1) perform objective analyses of XBT/BT digital traces, (2) process wind information, (3) study the statistical characteristics of the original and processed data, and (4) plot the generated fields.

The digital model was tested and its application may be extended to the study of any environmental variable continuously distributed along a vertical or horizontal axis.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Security Classification

-31408

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Objective						
Model						
Analyze						
Environmental						
Variclines						
Inversions						
Statistical						
Relation						
Surface						
Winds						

Thesis 130772
S1527 Saldanha
c.1
An objective method to analyze oceanic variclines and their statistical relation to surface winds.