CALCULATION OF MEAN SEASONAL SEA SURFACE ELEVATION FIELDS OVER THE NORTH ATLANTIC CONTINENTAL SHELF AND SLOPE USING HISTORICAL HYDROSTATION DATA

by

James H. Churchill

February 1980

TECHNICAL REPORT

Prepared for the Department of Energy under Grant DE-AC02-78-ER0005.
CALCULATION OF MEAN SEASONAL SEA SURFACE ELEVATION FIELDS OVER THE NORTH ATLANTIC CONTINENTAL SHELF AND SLOPE USING HISTORICAL HYDROSTATION DATA

by

James H. Churchill

WOODS HOLE OCEANOGRAPHIC INSTITUTION
Woods Hole, Massachusetts 02543

February 1980

TECHNICAL REPORT

Prepared for the Department of Energy under Grant DE-AC02-76-10048.

Reproduction in whole or in part is permitted for any purpose of the United States Government. This report should be cited as: Woods Hole Oceanographic Institution Technical Report WHOI-80-9.

Approved for Distribution

Valentine Worthington, Chairman
Department of Physical Oceanography
CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract</td>
<td>1</td>
</tr>
<tr>
<td>1. Introduction</td>
<td>2</td>
</tr>
<tr>
<td>2. Theoretical Background</td>
<td>2</td>
</tr>
<tr>
<td>3. Data Base</td>
<td>4</td>
</tr>
<tr>
<td>4. Selection of Seasonal Periods</td>
<td>5</td>
</tr>
<tr>
<td>5. Data Analysis</td>
<td>5</td>
</tr>
<tr>
<td>a. Initial Data Reduction</td>
<td>6</td>
</tr>
<tr>
<td>b. Averaging Sigma-t's on a Geographic Grid</td>
<td>7</td>
</tr>
<tr>
<td>c. Calculation of the Surface Elevation Field</td>
<td>9</td>
</tr>
<tr>
<td>6. Results</td>
<td>10</td>
</tr>
<tr>
<td>7. Validity of Specifying $\zeta_0$ and $\zeta_H$ as Functions of Bottom Depth</td>
<td>13</td>
</tr>
<tr>
<td>Acknowledgements</td>
<td>15</td>
</tr>
<tr>
<td>References</td>
<td>16</td>
</tr>
<tr>
<td>Table 1</td>
<td>17</td>
</tr>
<tr>
<td>Figures</td>
<td>18-48</td>
</tr>
<tr>
<td>Appendix I Revision of NODC Data Processing Program OCCOMP</td>
<td>49</td>
</tr>
<tr>
<td>Appendix II Software</td>
<td>53</td>
</tr>
<tr>
<td>OCCOMP</td>
<td>53</td>
</tr>
<tr>
<td>PICKOUT</td>
<td>73</td>
</tr>
<tr>
<td>GRID</td>
<td>75</td>
</tr>
<tr>
<td>STD</td>
<td>78</td>
</tr>
<tr>
<td>INTG</td>
<td>80</td>
</tr>
<tr>
<td>TYGRID</td>
<td>82</td>
</tr>
<tr>
<td>GRPRS</td>
<td>85</td>
</tr>
<tr>
<td>TYGRS</td>
<td>88</td>
</tr>
</tbody>
</table>
Abstract

This report details the calculation and results of a simple coastal elevation field model which was presented in a recent paper (Csanady, 1979). Seasonal mean sea surface elevation fields were calculated over the shelf-slope region off the east coast of North America north of Cape Hatteras by applying the model to a grid of averaged sigma-t's at standard depths. The sigma-t grid was calculated utilizing a data base of approximately 30,000 hydrographic stations from NODC archives. Difficulties involved with the numerical implementation of the model and economizing adaptations to the WHOI NODC data processing program, OCCOMP, are discussed.
1. Introduction

In a recent paper (Csanady, 1979, hereafter referred to as C79) a simple theoretical model was given for the approximate calculation of the density related pressure field over a shelf-slope region. The model was applied to the region off the east coast of North America north of Cape Hatteras. The details of how the calculation was actually carried out were not given in the paper and are described in this report.

2. Theoretical Background

Csanady bases his derivation in C79 on the depth integrated equations of motion and continuity written in the following form:

\[
- \frac{fV}{gH} = - \frac{\partial z}{\partial x} - \oint_{-H} \frac{(1+z/H) \frac{\partial c}{\partial x}}{\rho_o g H} \, dz + \frac{\tau_{xs} - \tau_{xb}}{\rho_o g H}
\]

\[
\frac{fU}{gH} = - \frac{\partial z}{\partial y} - \oint_{-H} \frac{(1+z/H) \frac{\partial c}{\partial y}}{\rho_o g H} \, dz + \frac{\tau_{ys} - \tau_{yb}}{\rho_o g H}
\]

(1)

\[
\frac{\partial U}{\partial x} + \frac{\partial V}{\partial y} = 0
\]

where:

\( U \) and \( V \) are depth integrated velocity components
\( f \) is the Coriolis parameter
\( g \) is the gravitational acceleration
\( H \) is the bottom depth
\( \tau_{ys}, \tau_{xs}, \tau_{xb}, \tau_{yb} \) are the surface and bottom stress components respectively
\( \zeta \) is the vertical displacement from the equilibrium position of the sea surface \((z = 0)\)
\( \epsilon \) is the density excess.

The density at any level is given by

\[
\rho = \rho_o (1 + \epsilon(x,y,z)) \tag{2}
\]

where \( \rho_o \) is the density of the deepest water, the abyssal density.

For this application surface stress is not considered and the bottom stress is parameterized according to two schemes. In scheme 1 the stress is considered proportional to the bottom geostrophic velocity. In scheme 2 the stress is proportional to the depth averaged velocity.

A local coordinate system is used with \( y \) orientated parallel to the isobaths, \( x \) perpendicular to the isobaths, and \( z \) positive downward. For this system the \( x \) component of bottom stress is considered zero. The idealization of zero long-isobath density variation is adopted, and the boundary conditions of zero transport at the coast and zero elevation field at the slope boundary (\( \zeta = 0 \) at the outer edge) are applied.

For stress scheme 1 the following equation describing the coastal elevation field results:

\[
\zeta = \zeta_d + \zeta_b \tag{3}
\]

where:
\[ \zeta_d = - \int_{-H}^{0} \epsilon_d dz \]
\[ \zeta_b = - \int_{H}^{H_m} \epsilon_b dH \]

\( \epsilon_b \) is the bottom density excess, a function of depth alone since the long-isobath density variation is considered zero. \( H_m \) is the maximum depth of the region considered.

For stress scheme 2 the resulting equation is

\[ \zeta = \zeta_d - \zeta_g + \zeta_H \]  \hspace{1cm} (4)

where

\[ \zeta_g = \int_{-H}^{0} \epsilon \frac{z}{H} dz \]
\[ \zeta_H = \int_{H}^{H_m} \frac{\zeta_g}{H} dH \]

The method and approximations required for computing the above elevations numerically will be discussed.

3. Data Base

The data base consists of temperature, salinity and depth station data (binary format) from NODC archives for the area outlined in Figure 1. This area encompasses the continental shelf and slope, from the shore to approximately
4,000 m depth. All historical data up to and including 1977, approximately 30,000 stations in all, have been included. The data has strong seasonal and regional variability as indicated by the seasonal station plots of Figures 1-4.

4. Selection of Seasonal Periods

The data density was insufficient to permit a useful calculation of the monthly mean elevation field, necessitating seasonal mean calculations. The surface temperature information presented in Table 1 was used to select the time periods of the seasons. At all locations listed there are two periods of relative temperature stability (Jan.-March and July-Sept.) and two periods of transition (Apr.-June and Oct.-Dec.). These periods were chosen as seasons and will hereafter be referred to as follows:

<table>
<thead>
<tr>
<th>Season</th>
<th>Period</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winter</td>
<td>Jan.-March</td>
</tr>
<tr>
<td>Spring</td>
<td>April-June</td>
</tr>
<tr>
<td>Summer</td>
<td>July-Sept.</td>
</tr>
<tr>
<td>Fall</td>
<td>Oct.-Dec.</td>
</tr>
</tbody>
</table>

5. Data Analysis

The method for computing the elevation field from the NODC data involves calculating sigma-t's at standard levels and averaging these sigma-t's within elements of a geographic grid. For each grid element, the integrals \( \zeta_d \) and \( \zeta_g \) of eqns. (3) and (4) are computed by trapezoid integration. \( \zeta_H \) and \( \zeta_b \) are specified as piecewise linear functions of depth whose values are obtained by integration of \( \zeta \) and \( \zeta_g \) along a representative transect from the deepest water to the shore.
A detailed step by step description of this method follows:

5a. Initial Data Reduction

The first data processing step is to read the NODC tapes, calculate the sigma-t's at standard depths and sort the data into seasonal and regional groups.

W.H.O.I. maintains a program, OCCOMP, which processes NODC data. For this specific application OCCOMP proved too costly and cumbersome to be feasible (the estimated cost of processing all 30,000 stations was over $10,000). It was therefore modified extensively, as described in Appendix I, to suit this particular project. The basic features of the modified version of OCCOMP are described below.

— The data is read and processed from the input tape station by station. The salinity and temperature at each observation depth are checked for data quality. If a station contains less than 3 good observations it is disregarded. The salinity and temperature are interpolated to standard levels. The standard levels are (in meters): 1, 10, 20, 30, 40, 50, 60, 75, 100, 125, 150, 200, 250, 300, 400, 500, 600, 700, 800, 900, 1000, 1200, 1500, 1750, 2000, 2500, 3000, 4000. The sigma-t and specific volume anomaly are calculated at each standard level.

— For each station two binary records are output to tape or disk. The first record contains general station information. The second has the values of salinity, temperature, sigma-t and specific volume anomaly at the standard depths.

— The processed data was output to 4 tapes, 1 per season. The tapes were divided into files according to Marsden Square.
— The sigma-t's were extracted from these tapes and written onto disk files by program PICKOUT. This program rejects sigma-t's which have associated temperature of greater than 40°C or less than 2°C, or which have associated salinity of greater than 39‰ or less than 26‰. The sigma-t's which pass are written to a binary disk file. A printout is generated which lists, for each station, the number of sigma-t's recorded and the number rejected. Typically less than 0.02% of the sigma-t's were rejected.

5b. Averaging Sigma-t's on a Geographic Grid

For each season the binary sigma-t files were combined into two regional files whose geographic bounds are shown in Figure 1. Region 1 extends from Cape Hatteras to Nova Scotia, encompassing Georges Bank; Region 2 stretches north to Newfoundland, including the Grand Banks. In each region the sigma-t's were averaged within elements of a geographic grid by program GRID.

The grid of each region is specified by a card-imaged file containing the center locations and dimensions of each grid element together with the bottom depth, generally taken as the depth at the element's center. Program GRID reads the regional sigma-t file station by station and calculates the average sigma-t and standard deviation at each standard level of each grid element. The data file is then reread and the sigma-t's are filtered based on the deviations from the mean.

The filtering procedure is similar to one employed by Levitus and Oort (1977). For each sigma-t the affected mean and standard deviation are calculated with that sigma-t absent. For convenience let \( \langle \sigma_t \rangle \), STD and \( \langle \sigma'_t \rangle \), STD' represent the mean and standard deviation with and without the sigma-t. If the sigma-t is greater than 5 STD' from
\( <\sigma_t> \)' , then STD' and \( <\sigma_t> \)' replace \( <\sigma_t> \) and STD. Filtering is not performed when the standard deviation is less than .02 or when the number of observations is less than 4.

Density inversions were occasionally encountered in the averaged sigma-t field. Various techniques, including smoothing methods were experimented with to make the sigma-t's monotonically increase with depth. In the end it was decided that the technique which had the least effect on the uninverted portion of the field should be used. This is simply to raise the value of the inverted sigma-t to that of the standard depth immediately above. All sigma-t's changed are flagged as negative numbers.

Program GRID optionally calculates \( \zeta_d \) , \( \zeta_g \) and \( \epsilon_b \) . The values obtained are not the final values used as will be discussed in the next section.

For each grid element a printout is generated of the average sigma-t, standard deviation, and number of observations at each level. The magnitude of any density inversion is also printed.

A binary file is created which contains the grid's dimensions and depths, the average sigma-t's, standard deviations and numbers of observations of each element, as well as \( \zeta_d \) , \( \zeta_g \) and \( \epsilon_b \) if calculated.

Program TYGRID reads this file and prints out sigma-t's, standard deviations, or number of observations at any standard depth desired in the form of the geographic grid.
5c. Calculation of the Surface Elevation Field

For some grid elements the contributions to the surface elevation fields, $\zeta_d$ and $\zeta_g$ of (3) and (4), cannot be calculated from the averaged sigma-t field because the bottom depth exceeds the depth of the deepest available data. In such cases the missing data is filled in with values representative of the particular region for that season. For deep water this is a justifiable procedure because the horizontal density variation is small, as will be shown.

The integrals $\zeta_d$ and $\zeta_g$ are calculated by trapezoid integration with the value of abyssal density, $\rho_o$, always 1.028 (other integration schemes were tried but all gave insignificantly differing results).

$\zeta_H$ and $\zeta_B$ are specified as piecewise linear functions of depth, different functions for each region and season. These functions are obtained using a transect of grid elements which cuts perpendicular to the isobaths at approximately the center of the particular region. Starting with the grid element of greatest bottom depth $\zeta_b$ and $\zeta_H$ are computed by trapezoid integration as follows:

$$\zeta_{bn} = \sum_{i=2}^{n} \left( H_i - H_{i-1} \right) \left( \epsilon_{b_i} + \epsilon_{b_{i-1}} \right) / 2$$

$$\zeta_{Hn} = \sum_{i=2}^{n} \left( H_i - H_{i-1} \right) \left( \epsilon_{g_i} + \epsilon_{g_{i-1}} \right) / 2$$

where:

$\zeta_{bn}$ and $\zeta_{Hn}$ are the values of $\zeta_b$ and $\zeta_H$ at bottom
depth \( H_n \); \( H_i \), \( i = 1, 2, 3 \ldots \) are the bottom depths in decreasing order, \( H_1 \) = maximum bottom depth = \( H_m \); \( \zeta_{bi} \) and \( \zeta_g \) are the associated values of bottom density excess and \( \zeta_g \).

In practice a number of elements close to the transect are added to ensure a sufficient number of values for meaningful integration. The need for and the validity of using this method to calculate \( \zeta_b \) and \( \zeta_H \) will be discussed in a later section.

Program GRPRS calculates the seasonal surface elevation fields. This program reads the binary file of averaged sigma-t's created by program GRID. The library of deepwater densities and the values of \( \zeta_b \) and \( \zeta_H \) at specified depths are input from cards. The program prints out the surface elevations according to eqns. (3) and (4) together with \( \zeta_d \), \( \zeta_b \), \( \zeta_g \) and \( \zeta_H \) for each grid element. The above values are also printed in the form of the geographic grid.

6. Results

The grid networks used to represent Regions 1 and 2 both consist of overlapping 1 degree squares separated by 1/2 degree. A 1/2 degree square grid was tried but the results were excessively erratic.

Grid depths of Region 1 are shown in Figure 5. The elements enclosed in circles are those used for calculating \( \zeta_b \) and \( \zeta_H \) of equations (5) and (6).

The sigma-t field at 10 m during summer is presented in Figure 6. The sigma-t's are written as the differences from 20, times 100. 450, for example, represents a sigma-t of 24.50. The standard deviations times 100 and the number of observations per element are displayed in Figures 7 and 8.
Note that the standard deviations are approximately 1 in the area between Cape Hatteras and Long Island and about 1/2 over the rest of the region. This pattern is characteristic of all seasons.

Contoured sigma-t fields, drawn from the digital printouts, of all seasons at 10 m are shown in Figures 9-12. These seasonal patterns are similar; all show lower sigma-t's in the region of Cape Hatteras and the northern Nova Scotian shelf and higher sigma-t's in the area of Georges Bank and the Gulf of Maine.

A printout of the sigma-t field at 3000 m during summer is shown in Figure 13. Note that the horizontal density variation at this depth is very slight. This is characteristic of all seasons.

The sea surface elevation field of summer according to bottom stress parameterization scheme 1 (equation 3) is shown in Figure 14 (See also C79, Fig.7). A few questionable points (due to scarcity of reliable observations) have been deleted. The main feature is a level change of about 10 cm approximately coincident with the 1000 m isobath. As indicated in C79 this would produce a geostrophic surface current to the southwest of magnitude 20 cm/sec, flowing parallel to the isobaths and of width in the order of 100 km. There is also a definite high associated with the Gulf of Maine. The scattered high values at the eastern edge of the region are presumably due to Gulf Stream influence.

Figure 15 show the sea surface elevation field according to bottom stress parameterization scheme 2. The predominant features are the same as in Figure 13 except that the level differences are diminished in magnitude by about 1/2. Since these two schemes yield such similar results further discussion will be limited to results using scheme 1, where the signal to noise ratio is better.
Sea surface elevations for the remaining seasons are displayed in Figures 16-18. All seasonal fields possess the same basic pattern. The level change at the shelf break appears slightly higher in the winter and spring than in the autumn.

The bottom depths of grid elements modeling Region 2 are shown in Figure 19. The circled elements are again those used in determining \( \zeta_b \) and \( \zeta_H \). The standard deviations times 100 and the number of observations at 10 m during spring are shown in Figures 20 and 21.

Contoured sigma-t fields at 10 m for spring and summer are displayed in Figures 22-23. (Autumn and winter were not processed due to very low density of observations). For both seasons there is a density minimum associated with the St. Lawrence outflow and the density steadily increases in the eastward direction.

Figures 24-25 present the sea surface elevation fields for spring and summer. Both seasonal fields display a high in the area of the St. Lawrence outflow with the elevation steadily decreasing in the eastward direction over the Grand Banks region. As in the seasonal fields of Region 1 there is a well defined level change nearly coincident with the 1000 m isobath.

The implications of these elevation fields are discussed in detail in C79. The basic conclusion is that the mean pressure field is partly caused by fresh water influx (most noticeably from the St. Lawrence), but the major contribution is from the deep ocean.
7. Validity of Specifying $\zeta_b$ and $\zeta_H$ as Functions of Bottom Depth

For one of the early attempts at calculating the elevation fields of equations (3) and (4) the integrals $\zeta_b$ and $\zeta_H$ were computed for each transect of grid elements which formed nearly a right angle with the isobaths. Unfortunately, few transects had sufficient values for meaningful integration. This combined with the scatter of $\epsilon_b$ and $\zeta_g$ caused the resulting elevation fields to be excessively erratic, prohibiting useful interpretation. The approach of calculating $\zeta_b$ and $\zeta_H$ from $\epsilon_b$ and $\zeta_g$ values in the vicinity of one transect, conveniently chosen at the approximate center of the particular region, and specifying these as functions of bottom depth was finally decided upon. To what extent this approach is valid may be discerned by examining the $\epsilon_b$ as a function of bottom depth.

$\epsilon_b$ vs bottom depth for Region 1 in summer is displayed in Figure 26. Points marked with an X are those used in the $\zeta_b$ calculation (eqn.(5)). The scatter is seen more clearly in Figure 27 which presents only the top 400 m. A similar graph for spring is presented in Figure 28. As one can see the scatter is nontrivial.

One can qualitatively determine from the graphs of both seasons that a polynomial curve fit by least squares to the $\epsilon_b$ values would generally lie above the points chosen for the integration. These curves would, however, have a positive second derivative in the direction of integration. For such curves trapezoid integration computes a high integral value. Graphs for winter and fall have characteristics similar to the spring and summer graphs.
\(\varepsilon_b\) vs bottom depth for Region 2 during spring is shown in Figures 29 and 30. Values from the two transects used for the \(\varepsilon_b\) calculation are indicated by +'s and x's (see Figure 14). The points marked by \(\Delta\)'s are a continuation of the transect denoted by +'s. Together the +'s and \(\Delta\)'s demonstrate a hysteresis which occurs progressing shoreward along the transect from deep water to shallow water and back into deeper water.

Figure 31 shows a similar pattern during summer.

One can observe that each region has a markedly different dependence of \(\varepsilon_b\) with depth. \(\varepsilon_b\) vs depth graphs of various subregions of Regions 1 and 2 were produced in an attempt to determine if the scatter could be reduced by using smaller regions. Within each region the scatter, and thus the \(\varepsilon_b\) vs depth dependence, appeared to be fairly uniform, indicating that our rather arbitrary choice of region boundaries was good.
Acknowledgements

I would like to express deep appreciation to Dr. G.T. Csanady who developed the concept and theory of this project. My thanks also go to Mr. George Heimerdinger who was very helpful in obtaining NODC data.
References


### Table 1

**Monthly Estimates of Sea Surface Temperatures**
**on the North Atlantic Continental Shelf**

<table>
<thead>
<tr>
<th>Month</th>
<th>CH</th>
<th>CM</th>
<th>MV</th>
<th>NS</th>
<th>AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jan</td>
<td>60</td>
<td>42</td>
<td>40</td>
<td>34</td>
<td>32</td>
</tr>
<tr>
<td>Feb</td>
<td>60</td>
<td>20</td>
<td>40</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>Mar</td>
<td>62</td>
<td>42</td>
<td>40</td>
<td>32</td>
<td>30</td>
</tr>
<tr>
<td>Apr</td>
<td>66</td>
<td>46</td>
<td>42</td>
<td>34</td>
<td>33</td>
</tr>
<tr>
<td>May</td>
<td>72</td>
<td>55</td>
<td>48</td>
<td>42</td>
<td>36</td>
</tr>
<tr>
<td>Jun</td>
<td>76</td>
<td>65</td>
<td>56</td>
<td>48</td>
<td>42</td>
</tr>
<tr>
<td>Jul</td>
<td>80</td>
<td>72</td>
<td>64</td>
<td>56</td>
<td>50</td>
</tr>
<tr>
<td>Aug</td>
<td>80</td>
<td>74</td>
<td>66</td>
<td>60</td>
<td>54</td>
</tr>
<tr>
<td>Sep</td>
<td>78</td>
<td>70</td>
<td>66</td>
<td>60</td>
<td>54</td>
</tr>
<tr>
<td>Oct</td>
<td>74</td>
<td>63</td>
<td>57</td>
<td>52</td>
<td>48</td>
</tr>
<tr>
<td>Nov</td>
<td>68</td>
<td>55</td>
<td>50</td>
<td>45</td>
<td>42</td>
</tr>
<tr>
<td>Dec</td>
<td>66</td>
<td>47</td>
<td>44</td>
<td>38</td>
<td>35</td>
</tr>
</tbody>
</table>

**Abbreviations of Nearest Landmarks:**

- CH - Cape Hatteras
- CM - Cape May
- MV - Martha's Vineyard
- NS - Nova Scotia
- AP - Avalon Peninsula (off Newfoundland)

Figure 1. NODC hydrostation locations during Jan.-March.
Figure 2. NODC hydrostation locations during April-June.
Figure 3. NODC hydrostation locations during July-Sept.
Figure 8. Number of observations per grid element at 10 m during summer.
Figure 9. Sigma-t field at 10 m during winter (Jan.-March).
Figure 10. Sigma-t field at 10 m during spring (April-June).
Figure 11. Sigma-t field at 10 m during summer (July-Sept.).
Figure 12. Sigma-t field at 10 m during autumn (Oct.-Dec.).
Figure 13. (Sigma-t-20) x 100 at depth 3000 m during summer. # indicates no data available.
Figure 14. Sea surface elevation (cm) calculated by equation (3) during summer.
Figure 16. Sea surface elevation (cm) during winter.
Figure 17. Sea surface elevation (cm) during spring.
Figure 18. Sea surface elevation (cm) during autumn.
Figure 19. (Bottom depth)/10 of the grid of Region 2. Circled grid elements used in $\zeta_b$ calculation.
Figure 20. (Standard deviation) x 100 at 10 m during spring.
Figure 21. Number of observations per grid element at 10 m during spring.
Figure 22. Sigma-t field at 10 m during spring.
Figure 23. Sigma-t field at 10 m during summer.
Figure 24. Sea surface elevation (cm) during spring.
Figure 25. Sea surface elevation (cm) during summer.
Figure 26. $\zeta_b$ vs bottom depth for Region 1 during summer. Values marked as x are from the circled elements of Figure 5 and were used in the $\zeta_b$ determination.
Figure 27. $\varepsilon_b$ vs bottom depth for Region 1 during summer. Top 400 m only.
Figure 28. $e_b$ vs bottom depth for Region 1 during spring. Top 400 m only.
Figure 29. $\varepsilon$ vs bottom depth for Region 2 during spring. Values marked as x, + and △ are from the grid elements marked in Figure 19.
Figure 30. $e_b$ vs bottom depth for Region 2 during spring. Top 800 m only.
Figure 31. $\varepsilon_b$ vs bottom depth for Region 2 during summer. Top 800 m only.
Appendix I

Revision of NODC Data Processing Program OCCOMP

OCCOMP, as written by the Information Processing Center (IPC) of W.H.O.I., reads and processes NODC data from either the new (binary) or old (card imaged) formats. Execution of the program in its original form proved too costly to be feasible for our particular application. For instance, processing the data from Marsden Square 1209 for months January through March cost approximately $420. These 800 stations represented roughly 1/30 of the total data which required processing.

OCCOMP was modified, as will be described, to significantly reduce our processing cost. The changes, however, were directed to specific needs of this project and do not necessarily represent an improvement for all applications of the program.

Before dealing with the changes a general description of the original version of OCCOMP will be instructive.

The main program reads input options, positions input tapes and prints the header page. If interpolation to standard levels is desired these levels are read from cards. For each station a loop which calls the various processing subroutines is executed, as described below.

The main program calls subroutine STATIN which initializes indicators. STATIN calls either NEWINPUT or OLDINPUT depending on the form of the data (OLDINPUT will be ignored in this discussion). NEWINPUT inputs a binary record into buffer. If it is the first record for the particular station (indicated by a value of 1 or 0 for the first character) subroutine NEWMASTER is called. NEWMASTER decodes station information (location, date and time of sampling, wind speed, etc.) and returns to NEWINPUT. Subroutine NEWDETAIL is called from NEWINPUT once for each sampling depth. NEWDETAIL checks which data is recorded (temperature, salinity, O₂, NO₂, etc.) and
calls decoding entries for each one present. Quality indicators of depth, temperature and salinity are checked. If one of these indicates a bad datum (value of greater than 6) the data from that depth is rejected. After decoding data from all depths NEWINPUT checks to determine if there is an additional record for the station. If so, it continues to decode information for the station until all records have been processed, at which point it returns to STATIN which returns to the main program.

If less than 3 sampling depths of data have been accepted, the station is discarded and the loop begins again for the next station. Otherwise, the main program calls subroutine USELECT, a user supplied subroutine which establishes criteria for station selection. If USELECT determines the station is unwanted, then the computation loop is restarted for the next station.

If the station is wanted BASICOMP is called. BASICOMP calculates pressure, Brunt-Väisälä frequency, sound velocity, sigma-t, potential temperature, sigma-theta, specific volume anomaly, and dynamic depth from the temperature, salinity and depth data. These are output to the printer and stored in arrays for future writing on tape or disk.

BASICOMP returns to the main program which calls, at the user's request, subroutine STDINTRP. This subroutine interpolates temperature, potential temperature, salinity, sigma-theta and specific volume anomaly to standard depths. These values are placed in an array for future tape or disk storage. Subroutine VOLTRANS, which calculates geostrophic transport, can also be called.

The last subroutine to be called is SAVEARRAY which outputs three binary records per station to tape or disk. The first record contains master station information (as decoded by NEWMASTER); the second contains all input data and calculated quantities; and the third has all interpolated values.

In the revised version of OCCOMP only salinity, temperature and depth data are read and checked, all other data are ignored.
Unnecessary computations have been deleted. Certain subroutines have been merged and those dealing with data in the old format have been deleted.

The most important alteration is a change of the location at which the station is either selected or rejected by subroutine USELECT. In the new version USELECT is called from NEWINPUT immediately after the call to NEWMASTER. Rejecting a station at this point eliminates considerable computation time devoted to reading and checking unwanted data. This change can represent a significant savings for our application where as many as 9/10 of the stations may be rejected.

The following describes specifically all major changes made in OCCOMP.

Subroutine STATIN, whose main task is to call NEWINPUT and do some indexing, has been merged with NEWINPUT.

In the original version of OCCOMP the main program has a counter which counts all stations that are accepted. When this equals a user specified number the run is terminated. Unfortunately, prior to execution the number of stations accepted cannot be determined. For this reason the revised program counts the number of station encountered instead.

An option was added which allows the user to delete printout of data and calculated values. This option is executed using variable IPROP which is entered in the first column of the second option card. If IPROP = 0 printout is deleted.

As previously mentioned, the revised version calls USELECT from subroutine NEWINPUT immediately after the call to NEWMASTER. If the station is wanted processing continues. If it's unwanted, a check is made to determine if the station has another record. If so, that record is skipped using subroutine SKPREC (WHOI account 3). Control is then returned to the main program which restarts the computation loop.

Unfortunately there is no way of determining how many additional records a rejected station may have. To avoid mistaking a continuation record of a rejected station with the first record
of a subsequent station a check is made on the first record of each station to determine if it's a continuation record. If so, that record is rejected and control returns to the main program which restarts the computation loop.

STDINTRP has taken over the computation chores of BASICOMP and BASICOMP has been deleted. The modified STDINTRP operates in the following manner.

Temperature and salinity are interpolated to standard depths. Pressure, sigma-t, specific volume and specific volume anomaly are calculated from these interpolated values. These are printed if IPROP≠0. The depths, temperatures, salinities, sigma-t's (not sigma-theta as in the original OCCOMP), and specific volume anomalies are put into arrays for future tape or disk output. This procedure differs from that employed by the original version of OCCOMP which calculates all quantities at the observed depths and then interpolates.

Subroutine NEWDETAIL has been changed to decode only depth, temperature and salinity. Quantities not required in this application (O₂, phosphate, ζi, NO₂, NO₃ and pH) are ignored.

Only two records per station are output to tape or disk. The first contains master station information and the second has the interpolated values.

A number of very large arrays (over 6000 words in all) have been deleted.

Processing the same data base with the original and the modified versions of OCCOMP yielded nearly identical results. The differences were due to the different methods of interpolating to standard depths and never exceeded 2%.

These modifications saved a considerable amount of computer time for our application. The modified program processed the winter data from MARSDEN Square 1209 at a cost of roughly $25 compared to $420 required by the original OCCOMP.

Listing of the modified program follows.
Appendix II  Software

PCL
COPY OCCOMP TO LP

C PROGRAM OCCOMP  DEC 22 1978
C
C THIS PROGRAM IS A REVISED OF OCCOMP VERSION 6.0 WRITTEN
C BY MARY HUNT. IT DEALS WITH ONLY NODC DATA IN BINARY FORMAT
C ONLY TEMP, SAL AND PRESSURE DATA IS DECODED, SIGMA-T'S AND
C SPECIFIC VOLUME ANOMALIES ARE CALCULATED AT STANDARD DEPTHS
C FROM INTERPOLATED VALUES OF TEMP AND SAL.

C ******************************************************
C * *
C * COMMON /LOGUNIT/ *
C * *
C ******************************************************
C
C THIS BLOCK CONTAINS LOGICAL UNIT NUMBERS
C OF ALL DEVICES AND FILES USED
C PRINTER, CARD READER, INPUT DEVICE, AND
C FILE FOR SAVING STUFF
C
C THIS BLOCK IS USED BY ALL ROUTINES
C
C COMMON/LOGUNIT/LP,
C * ICR,
C * INUNIT,
C * SAVFIL

C INTEGER LP/108/
C INTEGER ICR/105/
C INTEGER INUNIT/8/
C INTEGER SAVFIL/1/

C ******************************************************
C * *
C * COMMON /OPTIONS/ *
C * *
C ******************************************************
C
C THIS BLOCK CONTAINS INPUT OPTIONS AND
C PROCESSING FLAGS
C
C THIS BLOCK IS USED BY
C NEWINPUT
C OLDINPUT
C BASICOMP
C STINTRP
C VOLTRANS
C
C COMMON/OPTIONS/ COMPOPT, INPOPT, SAVOPT,
C * BLKFALT, EDATA
C COMMON/OPTIONS/ NUMSTAT, MAXOBS
C INTEGER COMPOPT, INPOPT, SAVOPT, BLKFALT
C LOGICAL EDATA
C INTEGER NUMSTAT
C INTEGER MAXOBS/140/
***************
* COMMON /BUFFERS/ *
* *
***************

THIS BLOCK CONTAINS BUFFERS AND THINGS
NEEDED BY THE INPUT ROUTINES

INBUFR FOR COMPLETE PHYSICAL RECORD
NUBUF FOR ONE LOGICAL RECORD ( OLDINPUT )
INDEX IS POSITION IN PHYSICAL RECORD ( NEWINPUT )

THIS BLOCK IS USED BY
OLDINPUT
NEWINPUT

COMMON/BUFFERS/INBUFR,
* NUBUF,
* INDEX

INTEGER INBUFR(1600)
INTEGER NUBUF(20)
INTEGER INDEX

***************
* COMMON /ALPHNUM/ *
* *
***************

THIS BLOCK CONTAINS ALPHANUMERIC CHARACTERS
USED IN DECODING INPUT

THIS BLOCK IS USED BY
OLDINPUT, NEWINPUT
OLDMASTR, NEWMASTR
OLDECOD, NUDECOD

COMMON/ALPHNUM/ IBLNK, ASTRISK, ZEE, CUE, PEA,
+ JAY, ONE, THREE, FOUR,
+ ESS, EEE, ENN, UU
INTEGER IBLNK/ , ASTRISK/**/,
INTEGER ZEE/ZE/, CUE/Q/, PEA/P/
INTEGER JAY/J/, ONE/1/,
INTEGER THREE/3/, FOUR/4/,
INTEGER ESS/S/, EEE/F/,
INTEGER ENN/N/, UU/W/

***************
* COMMON /HEADING/ *
* *
***************

THIS BLOCK CONTAINS INFORMATION FROM
C  STATION HEADER RECORD
C
C  THIS BLOCK IS USED BY
C  OLDINPUT
C  NEWINPUT
C  HEADPAGE
C
C   COMMON/HEADING/COUNTRY,SHIP,
*   LATITUDE,LATNS,
*   LONGITUDE,LONGEW,
*   YEAR,MONTH,DAY,
*   CRUSFNO,STATNO,
*   BOTDEP, DEEPEST,
*   TIMODAY,
*   LATMIN,LONGMIN,
*   HOUR,USIT
C
C   INTEGER COUNTRY,SHIP(2)
C   INTEGER LATITUDE,LATNS
C   INTEGER LONGITUDE,LONGEW
C   INTEGER YEAR,MONTH,DAY
C   INTEGER CRUSFNO,STATNO(3)
C   INTEGER BOTDEP, DEEPEST
C   INTEGER TIMODAY(4)
C   REAL LATMIN,LONGMIN
C   REAL HOUR
C
C
C   INTEGER HEADATA(16)
C
C
C
C********************************************
C  *  COMMON /INKERS/  *
C  *
C********************************************
C
C  THIS BLOCK CONTAINS INDICATORS USED IN
C  DECODING THE INPUT
C
C  THIS BLOCK IS USED BY
C  OLDASTR, NEWMASTR
C  OLDECOD, NUDECOD
C
C   COMMON/INKERS/  DQUAL, DTERM, TPREC, TQUAL,
+   SPREC, SQUAL, OPREC, OQUAL,
+   IPREC, FPREC, SIPREC,
+   NO2PR, NO3PR, PHPR
C
C
C********************************************
C  *  COMMON /VARUSE/  *
C  *
C********************************************
C THIS BLOCK CONTAINS VALUES OF INPUT
C PARAMETERS FOR ONE DEPTH
C
C THIS BLOCK IS USED BY
C OLMAS TR, NEWMAS TR
C OLCDECOD, NUCDECOD
C
C COMMON/VARUSE/ TEMP, SALTY, DPTH, OX, PHOS, TPHOS, SIL,
+ NO2, NO3, PH
REAL DPTH, OX, PHCS, TPHOS, SIL
REAL NO2, NO3, PH
DOUBLE PRECISION TEMP, SALTY
C
C ******************************************************
C *
C COMMON /INPUTVAR/ *
C *
C ******************************************************
C
C THIS BLOCK CONTAINS VALUES OF INPUT VARIABLES
C DEPTH, TEMPERATURE, AND SALINITY
C LATITUDE AND LONGITUDE, IN DEGREES
C NGOOD IS NUMBER OF VALUES IN DEPTH, ETC. ARRAYS
C
C THIS BLOCK IS USED BY
C OLDINPUT
C NFWINPUT
C BASICOMP
C STOINTRP
C
C COMMON/INPUTVAR/DEPTH,
+ TEMPERATURE,
+ SALINITY,
+ XLAT,XLONG,
+ NGOOD
C
C DOUBLE PRECISION DPTH(140)
DOUBLE PRECISION TEMPERATURE(140)
DOUBLE PRECISION SALINITY(140)
DOUBLE PRECISION XLAT,XLONG
INTEGER NGOOD
C
C ******************************************************
C *
C COMMON /OUTPARTI/ *
C *
C ******************************************************
C
C THIS BLOCK CONTAINS THE ARRAY OF FORMATTED PRINT LINES
C INDICATRGS FOR COMPLETE DATA
C THESE ARRAYS ARE USED FOR LOCAL VARIABLES
C IN STOINTRP AND VOLTRANS
C
C THIS BLOCK IS USED BY
C OLDINPUT
C NFWINPUT
C * BASIGMP
C
CCCC COMMON/OUTPART1/PRINTLN,
CCCC *     INDCTR,
CCCC *     NPLIN
C
CCCC INTEGER PRINTLN(14,140)
CCCC INTEGER INDCTR(140)
CCCC INTEGER NPLIN/14/
C
C
C   ************************************************************
C   *
C   * COMMON /DATASAV/ *
C   *
C   ************************************************************
C
C THIS BLOCK CONTAINS ARRAYS WHICH ARE STORED
C ON TAPE OR DISK FOR LATER PROCESSING
C
COMMON/DATASAV/ SAVDAT2(5,140),
+     NDETAIL, NINT,
+     NSAV1, NSAV2
REAL SAVDAT2
INTEGER NDETAIL, NINT
INTEGER NSAV1/19/, NSAV2/5/
C
C
C   ************************************************************
C   *
C   * COMMON /STANDARD/ *
C   *
C   ************************************************************
C
C THIS BLOCK CONTAINS STANDARD DEPTHS AND
C STANDARD PRESSURES AND NUMBER OF EACH
C
C THIS BLOCK IS USED BY
C STOINTRP
C VOLTRANS
C
COMMON/STANDARD/ STANDEP(140), STANPRES(140),
+     NSDEP, NSPRES, IREF, IPROP
DOUBLE PRECISION STANDEP, STANPRES
INTEGER NSDEP, NSPRES, IREF
C
C
C   ************************************************************
C   *
C   * COMMON/PART123/ *
C   *
C   ************************************************************
C
C THIS BLOCK CONTAINS ARRAYS FOUND BY PART 1
C WHICH ARE USED BY PART 2 AND/OR PART 3
C
C THIS BLOCK IS USED BY
C BASIGMP
C STOINTRP
C VOLTRANS
C
C
C
********************************************************************************
*  COMMON/LOCAL/  *
*  *****************
C
C
THIS BLOCK CONTAINS ARRAYS AND VARIABLES
WHICH ARE USED ONLY IN THE ROUTINE AND
DO NOT NEED TO BE KEPT.

COMMON/LOCAL/ RECSKP, MESS, INDSKP,
+    IR, REFLEVL,
+    JUNK(1685)
INTEGER RECSKP, MESS, INDSKP, JUNK
INTEGER IR
DOUBLE PRECISION REFLEVL

********************************************************************************
LOCAL VARIABLES

********************************************************************************
INTEGER TOTSTAT
LOGICAL USIT
INTEGER MOPE/*OPTI*/

********************************************************************************

READS AND CHECKS OPTION CARD
COMOPT=0, FOR BASIC COMPUTATIONS ONLY
   1, TO INTERPOLATE AT STANDARD DEPTHS
   2, TO INTERPOLATE AT STANDARD PRESSURES
      AND FIND VOLUME TRANSPORT
   3, TO DO BOTH STANDARD DEPTHS AND
      PRESSURES
INOPT= 0, FOR CARD INPUT
   -1, FOR NEW FORMAT NODC INPUT
      OTHERWISE, FOR BLOCKED CARD IMAGES
      ON TAPE, SHOULD BE BLOCKING FACTOR.
RECSKP IS NUMBER OF PHYSICAL TAPE RECORDS TO SKIP
SAVOPT SHOULD BE NON-ZERO TO SAVE ARRAYS IN FILE
FOR FURTHER PROCESSING

TOTSTAT IS NUMBER OF STATIONS TO PROCESS
IF LEFT BLANK, WILL PROCESS ALL STATIONS
(SETS TOTSTAT TO 10000)

CHECKS FOR TAPE POSITIONING,
AND DOES IT IF REQUESTED

READS STANDARD DEPTHS AND/OR PRESSURES,
AS REQUIRED

GETS TIME OF DAY AND PRINTS HEADER PAGE

READ AND CHECK OPTION CARD

READ(1CR,1000) MESS, COMPOPT,
  + INPOPT, RECSKP,
    + SAVOPT, TOTSTAT
1000 FORMAT(A4,6X,515)
READ(1CR,1003) IPROP
1003 FORMAT(11)
READ(1CR,1003) NOFTS
IF ( MESS .NE. MOPE ) GO TO 500
IF ( TOTSTAT .EQ. 0 ) TOTSTAT = 10000
OUTPUT 'GOCOMP VERSION MAY 1978'
OUTPUT COMPOPT, INPOPT, RECSKP, SAVOPT, TOTSTAT, IPROP, NOFTS
IF ( COMPOPT .LT. 0 .OR. COMPOPT .GT. 3 ) GO TO 530
BLKFACT = INPOPT
IF ( INPOPT .LE. 0 ) BLKFACT = 1

CHECK FOR TAPE POSITIONING

IF ( (INPOPT .EQ. 0) .OR. (RECSKP .LE. 0) ) GO TO 20
CALL SETSKP ( INDSKP )
CALL SKPRECO ( INUNIT, RECSKP )
IF ( INDSKP .NE. 2 ) GO TO 510

GET STANDARD DEPTHS AND/OR
PRESSURES, AS REQUIRED

20 CONTINUE
IF ( (COMPOPT .EQ. 1) .OR. (COMPOPT .EQ. 3) )
  + CALL READARRAY (STANDEP, NSDEP)
IF ( COMPOPT .LT. 2 ) GO TO 40

READ(1CR, 1010) REFLEVL
1010 FORMAT(G)
CALL READARRAY ( STANPRES, NSPRES )

IREF = 0
DO 30 IR=1, NSPRES
  IF ( STANPRES(IR) .EQ. REFLEVL ) IREF = IR
30 CONTINUE
IF ( IREF .EQ. 0 ) GO TO 520
GET TIME OF DAY AND PRINT FIRST PAGE
SET STATION COUNT TO ZERO

40 CONTINUE
CALL TODAY(TIMODAY)
CALL HEADPAGE
NUMSTAT = 0
ENDATA = .FALSE.
IF(NOFTS.EQ.0) GO TO 41
CALL SKIPFIL(SAVFIL,NOFTS)
41 CALL TIC

**************************************************************************
**                        COMPUTATION SECTION                        **
**************************************************************************

MAIN PROGRAM LOOP
FOR EACH STATION, TAKES THE FOLLOWING STEPS:
BLANKS PRINT BUFFER
SETS THE SAVE DATA ARRAY TO DEFAULT
INPUTS DATA FOR STATION
DOES REQUIRED COMPUTATION AND OUTPUT
STORES DATA ARRAYS IF REQUESTED
STOPS WHEN END OF INPUT FILE IS REACHED,
OR SPECIFIED NUMBER OF STATIONS COMPLETED.

**************************************************************************

REPEAT 100, WHILE .NOT. ENDTA .AND. NUMSTAT.LT. TOTSTAT
USIT = .TRUE.
NUMSTAT = NUMSTAT+1
CALL NEWINPUT
IF(.NOT.USIT) GO TO 100
IF(NGOOD.LT.3) OUTPUT * STATION DELETED; <3GOOD DEPTHS>; GO TO 100

100 CONTINUE

CALL TOC(SECTP)
WRITE(LP,3000) SECTP
IF (SAVOPT.NE.0) END FILE SAVFIL
CALL EXIT
C
C ******************************************************
C ERROR ROUTINES
C ******************************************************
C
C 500 CONTINUE
WRITE(LP,2010)
2010 FORMAT(' OCCOMP OPTION CARD MISSING')
CALL ABORT
C
C 510 CONTINUE
WRITE(LP,2020) INDSKP
2020 FORMAT(' SKPREC ERROR. IND = ',I3)
CALL ABORT
C
C 520 CONTINUE
WRITE(LP,2030) REFLEVEL
2030 FORMAT(' REFERENCE LEVEL ',G13.5,' NOT AT STANDARD PRESSURE')
CALL ABORT
C
C 530 CONTINUE
WRITE(LP,2040) COMPOPT
2040 FORMAT(' INVALID COMPOPT ',I5)
CALL ABORT
C
C 3000 FORMAT(F10.3,' SECONDS TO PROCESS DATA')
C END
C
C  HEADCARD.1   APRIL 27, 1978
C SUBROUTINE HEADCARD
C
C PART OF PROGRAM OCCOMP
C HEADCARD PRINTS INTRODUCTORY PAGE
C ENTRY PRNHEAD CARDS PAGE HEADINGS FOR STATION
C ALL COMMUNICATION BY LABELLED COMMON
C
C PROGRAMMER:  MARY HUNT
C DATE:  FEBRUARY, 1977
C
C******************************************************************************
C COMMON/LOGUNIT/ LP, ICR, INUNIT, SAVFIL
INTEGER LP, ICR, INUNIT, SAVFIL
C******************************************************************************
C COMMON/HEADING/ COUNTRY, SHIP(2), LATITUDE, LATNS,
+     LONGUDE, LONGEW,
+     YEAR, MONTH, DAY, CRUISENO,
+     STATNO(3), BOTDEP, DEEPEST, TIMODAY(4),
+     LATMIN, LONGMIN, HOUR, USIT
INTEGER COUNTRY, SHIP, LATITUDE, LATNS, LONGUDE, LONGEW
INTEGER YEAR, MONTH, DAY, CRUISENO
INTEGER STATNO, BOTDEP, TIMODAY
INTEGER DEEPEST
REAL LATMIN, LONGMIN, HOUR
C
C**********************************************************************
C
INTEGER ROMAN(12)
DATA ROMAN/ 'I', 'II', 'III', 'IV', 'V', 'VI',
  'VII', 'VIII', 'IX', 'X', 'XI', 'XII'/**

C**********************************************************************
C
WRITE(LP,2030)
2030 FORMAT('1PROGRAM OCCOMP',
  * T23,'REVISION APRIL, 1978///')
WRITE(LP,2040)
2040 FORMAT(T4,'THE UNITS OF QUANTITIES USED IN THE',
  * T4,'FOLLOWING CALCULATIONS ARE', ///
  2 T4,'DEPTH',T4,'METERS',//
  3 T4,'PRESSURE',T4,'DECIBARS',//
  4 T4,'TEMPERATURE',T4,'DEGREES CELSIUS',//
  5 T4,'SALINITY',T4,'PARTS PER THOUSAND',//
  6 T4,'OXYGEN',T4,'MILLILITERS PER LITER',//
  7 T4,'INORGANIC PHOSPHATE (PO4)',//
  8 T4,'MICROGRAM-ATOMS PER LITER',//
  9 T4,'TOTAL PHOSPHORUS (PHOS)',//
  A T4,'MICROGRAM-ATOMS PER LITER',//
  B T4,'SILICATE-SILICON',T4,'MICROGRAM-ATOMS PER LITER',//
  C T4,'NITRATES (NO3)',T4,'MICROGRAM-ATOMS PER LITER',//
  D T4,'ANOMALY OF DENSITY AT ATMOSPHERIC',
  E T4,'GRAMS PER LITER',//
  F T4,'PRESSURE (SIGMA-THETA)',//
  G T4,'POTENTIAL TEMPERATURE (THETA)',
  H T4,'DEGREES CELSIUS',//
  I T4,'SPECIFIC VOLUME ANOMALY (DELTA)',,
  J T4,'10 MILLILITERS PER GRAM',//
  K T65,'2',T77,'2'/
  L T4,'DYNAMIC HEIGHT',,
  M T4,'DYNAMIC METERS (10 METERS PER SECOND)',//
  N T4,'SOUND VELOCITY',T4,'METERS PER SECOND',//
  O T4,'B8',T63,'2/T4,'POTENTIAL ENERGY',,
  P T4,'10 ERGS PER CENTIMETER',//
  Q T4,'BRUNT-VAISALA FREQUENCY',,
  R T4,'CYCLES PER HOUR',//
  S T4,'BRUNT-VAISALA PERIOD',T4,'MINUTES',//
  T T4,'VELOCITY',T4,'CENTIMETERS PER SECOND',//
  U T4,'6/T4,'VOLUME TRANSPORT',,
  V T4,'10 CUBIC METERS PER SECOND')

RETURN

C**************************************************************
C
ENTRY PRNHEAD

C**************************************************************
C
WRITE(LP,2000) TIMODAY
2000 FORMAT( T1,'ICOUNTRY',T11,'SHIP',,
  * T18,'CRUISE',T27,'STATION',,
  * T40,'DATE',T51,'GMT',,
  * T58,'LATITUDE',T70,'LONGITUDE',,
  * T82,'DEPTH',T104,'4A4')
WRITE(LP,2010) COUNTRY,SHIP,
* CRUISENO,STATNO,
* DAY,Roman(MONTH),YEAR,HOUR,
* LATITUDE,LATMIN,LATNS,
* LONGITUDE,LONMIN,LONGE,
* "B"UTOORE
2010 FORMAT(5,A3,T12,A4,A2,
* T19,A3,T27,A3,A4,A2,
* T37,T2,T45,A4,14,12,T50,F4.1,
* T57,T2,F5.1,1X,A1,
* T69,I3,F5.1,1X,A1,
* T82,15/1)
C
RETURN
C
END
C
NEWINPUT.MAY 1, 1978
SUBROUTINE NEWINPUT
C
THIS SUBROUTINE IS PART OF PROGRAM OCCOMP
IT INPUTS DATA FOR ONE STATION IN 1974 NOOC FORMAT
C
PROGRAMMER: MARY HUNT
DATE: MARCH, 1977
C
***********************************************************************************
C
COMMON/LOGUNIT/ LP, ICR, INUNIT, SAVFIL
INTEGER LP, ICR, INUNIT, SAVFIL
C
***********************************************************************************
C
COMMON/OPTIONS/ COMPOPT, INPOPT, SAVOPT,
* BLKFACT, ENDATA
COMMON/OPTIONS/ NUMSTAT, MAXOBS
INTEGER COMPOPT, INPOPT, SAVOPT, BLKFACT,
LOGICAL ENDATA
INTEGER NUMSTAT
INTEGER MAXOBS
C
***********************************************************************************
C
COMMON/BUFFERS/ INBUFR(1600), NUBUFR(20), INDEX
INTEGER INBUFR, NUBUFR, INDEX
C
***********************************************************************************
C
COMMON/ALPHNUM/ IBLNK, ASTRISK, ZEE, CUE, PEA,
* JAY, ONE, THREE, FOUR,
* ESS, EEE, ENN, UU
INTEGER IBLNK, ASTRISK, ZEE, CUE, PEA
INTEGER JAY, ONE, THREE, FOUR
INTEGER ESS, EEE, ENN, UU
C
COMMON/INPUTVAR/ DEPTH(140), TEMPERATURE(140), SALINITY(140),
& XLAT,XLONG,NGOOD
C
DOUBLE PRECISION DEPTH, TEMPERATURE, SALINITY, XLAT, XLONG
INTEGER NGOOD

C*****************************************************************************
C COMMON/DATA$AV/
   * SAVDAT2(5,140),
     * NDETAIL, NINT,
     * NSAV1, NSAV2
REAL SAVDAT2
INTEGER NDETAIL, NINT
INTEGER NSAV1, NSAV2

C*****************************************************************************
C COMMON/LOCAL/
   * LCHAR, INKER, NW, ITST,
     * JUNK(1687)
INTEGER LCHAR, INKER, NW, ITST
INTEGER JUNK

C*****************************************************************************
C LOCAL VARIABLES. MUST NOT BE CHANGED BY
C OTHER ROUTINES.
C
INTEGER ND, NDET
INTEGER HEADATA(16)
LOGICAL USIT
INTEGER CONIND
COMMON/HEADING/
   COUNTRY, SHIP(2), LATITUDE, LATNS,
   * LONGITUDE, LONGEW,
   * YEAR, MONTH, DAY, CRUISENO,
   * STATNO(3), BOTDEP, DEEPEST, TIMEDAY(4),
   * LATMIN, LONGMIN, HOUR, USIT
EQUIVALENCE (HEADATA(1), COUNTRY), (HEADATA(2), SHIP)
EQUIVALENCE (HEADATA(4), LATITUDE), (HEADATA(5), LATNS)
EQUIVALENCE (HEADATA(6), LONGITUDE), (HEADATA(7), LONGEW)
EQUIVALENCE (HEADATA(8), YEAR), (HEADATA(9), MONTH)
EQUIVALENCE (HEADATA(10), DAY), (HEADATA(11), CRUISENO)
EQUIVALENCE (HEADATA(12), STATNO), (HEADATA(15), BOTDEP)
EQUIVALENCE (HEADATA(16), DEEPEST )
C
   INTO = 0
NGOOD = 0
NDETAIL = 0
10 CONTINUE
   CALL BUFFERIN( INUNIT, 0, INBUFR, 1600, INKER, NW, ITST )
C
   IF ( INKER .EQ. 3 ) ENDATA = .TRUE.; RETURN
   IF ( INKER .NE. 4 ) GO TO 20
C
   IF ( NUMSTAT .EQ. 0 )
      WRITE(LP,2000) ITST ;
      CALL ABORT
2000 FORMAT ( ' READ ERROR ON FIRST RECORD. CODE IS ',Z2 )
C
   WRITE(LP,2010) ITST
2010 FORMAT ( ' READ ERROR. CODE IS ',Z2 )
C
C*****************************************************************************
C*****************************************************************************
C*****************************************************************************
GO TO 10

HAVE A RECORD IN CORE NOW.
SET INDEX AND DECODE STATION HEADER INFO.

20 CONTINUE
INDEX = 1
NDET = (NW/20) - 2
DECODE(4, 1010, INBUF) CONIND
1010 FORMAT (II)

IF (CONIND.GT.1.AND.INTO.EQ.0) WRITE(108, 801) USIT = .FALSE.;
& RETURN
801 FORMAT('FIRST RECORD IN STATION IS A CONTINUATION RECORD,
& STATION DELETED')
IF (CONIND.GT.1) GO TO 25
INTO = 1

CALL NEWMASTR(INBUF)
CALL USELECT(HEADATA, USIT)
IF (USIT) GO TO 25
IF (CONIND.GT.0).AND.(CONIND.LT.9) CALL SKPREC(INUNIT, 1)
RETURN

NOW DECODE DETAIL OBSERVATIONS
FROM THIS RECORD

25 CONTINUE
INDEX = INDEX + 40
DO 40 ND=1, NDET
DECODE (80, 1000, INBUF(INDEX)) LCHAR
IF (LCHAR .NE. THREE) .AND. (LCHAR .NE. FOUR) GO TO 30
NDETAIL = NDETAIL + 1
IF (NDETAIL.GT.MAX_OBS) GO TO 50
CALL NEWDETAIL (INBUF(INDEX))
30 CONTINUE
INDEX = INDEX + 20
40 CONTINUE
1000 FORMAT (180, A1)

CHECK IF ANOTHER RECORD FOR
THIS STATION

IF (CONIND.GT.0).AND.(CONIND.LT.9) GO TO 10

50 CONTINUE
DEEPEST = DEPTH(NGOOD)
RETURN

END

NEWMASTR, 3
MAY 1, 1978
SUBROUTINE NEWMASTR (BUFR)

INTEGER BUFR (1)

DECODES STATION HEADER INFORMATION
NODC 1974 FORMAT

PROGRAMMER: MARY HUNT

DATE: MARCH, 1977
C***************************************************************************
C
C              COMMON/ALPHNUM/ IBLNK, ASTRISK, ZEE, CUE, PEA,
C               JAY, ONE, THREE, FOUR,
C               ESS, EEE, ENN, UU
C              INTEGER IBLNK, ASTRISK, ZEE, CUE, PEA
C              INTEGER JAY, ONE, THREE, FOUR
C              INTEGER ESS, EEE, ENN, UU
C
C***************************************************************************
C
C              COMMON/HEADING/ COUNTRY, SHIP(2), LATITUDE, LATNS,
C               LONGITUDE, LONGEW,
C               YEAR, MONTH, DAY, CRUISENO,
C               STATNO(3), BOTDEP, DEEPEST, TIMODAY(4),
C               LATMIN, LONGMIN, HOUR, USIT
C              INTEGER COUNTRY, SHIP, LATITUDE, LATNS, LONGITUDE, LONGEW
C              INTEGER YEAR, MONTH, DAY, CRUISENO
C              INTEGER STATNO, BOTDEP, TIMODAY
C              INTEGER DEEPEST
C              REAL LATMIN, LONGMIN, HOUR
C
C***************************************************************************
C
C              COMMON/VARUSE/ TEMP, SALTY, DPTH, CX, PHOS, TPHOS, SIL,
C               NO2, NO3, PH
C              DOUBLE PRECISION TEMP, SALTY
C              REAL DPTH, CX, PHCS, TPHOS, SIL
C              REAL NO2, NO3, PH
C
C***************************************************************************
C
C              COMMON/INPUTVAR/ CEPH(140), TEMPTURE(140), SALINITY(140),
C               XLAT, XLONG, NGOOD
C              DOUBLE PRECISION DEPTH, TEMPTURE, SALINITY, XLAT, XLONG
C              INTEGER NGOOD
C
C***************************************************************************
C
C              COMMON/DATAINV/ SAVDAT2(5,140),
C               NDETAIL, NINT,
C               NSAV1, NSAV2
C              REAL SAVDAT2
C              INTEGER NDETAIL, NINT
C              INTEGER NSAV1, NSAV2
C
C***************************************************************************
C
C              COMMON/INKERS/ DQUAL, DTHERM, TPREC, TQUAL,
C               SPREC, SQUAL, OPREC, OQUAL,
C               IPREC, FPREC, SIPREC,
C               NO2PR, NO3PR, PHPR
C              INTEGER DQUAL, DTHERM, TPREC, TQUAL, SPREC, SQUAL
C              INTEGER OPREC, OQUAL, IPREC, FPREC, SIPREC
C              INTEGER NO2PR, NO3PR, PHPR
**C*********************************************************************

C

DECODE(160,1000,BUFR)
C    COUNTRY, LATNS, LATITUDE, LATMIN,
C    LONGW, LONGDE, LONGMIN,
C    YEAR, MONTH, DAY, HOUR, SHIP(1), SHIP(2),
C    BTODEP, CRUISENO, STATNO(1), STATNO(2), STATNO(3),
C    NEXT
C
1000 FORMAT( T2,A3,  T27,A1,12,F3.1,
C             + T33,A1,13,F3.1,
C             + T41,312,F3.1, T50,A4,A2,
C             + T56,15, T95,A3,A3,A4,A2,
C             + T159,I1 )
C
C
C    XLAT = LATITUDE + LATMIN/60.0
C    IF ( LATNS .EQ. ESS ) XLAT = -XLAT
C    XLONG = LONGDE + LONGMIN/60.0
C    IF ( LONGW .NE. EEE ) XLONG = -XLONG
C
C RETURN
C
C
C*********************************************************************

C

ENTRY NEWDETAIL ( BUFR )
C
C
C HANDLES OBSERVED DEPTH INFORMATION FOR ONE DEPTH
C IN NEW NODC FORMAT
C
C
C DECODE CONTENTS OF RECORD
C
C DECODE(80,1010,BUFR) DPTH, OQUAL, DTHERM,
C    TEMP, TPREQ, TQUAL,
C    SALTY, SPREC, SQHAL
C
1010 FORMAT( I5,I1,A1,
C            + T8,F5.3,21I1,
C            + T15,F5.3,21I1 )
C
C
C    CHECK FOR VALIDITY OF TEMP.
C    AND SAL AT THIS DEPTH
C
C IF ( ( DQUAL .GT. 6 ) .OR.
C    + ( TQUAL .GT. 6 ) .OR.
C    + ( SQUAL .GT. 6 ) ) RETURN
C
C IF( ( TPREQ .EQ. 9 ) .OR. ( SPREC .EQ. 9 ) )
C    RETURN
C
C THIS OBSERVATION IS A KEEPER
C
C NGOOD = NGOOD + 1
C DEPTH(NGOOD) = DPTH
C TEMPERATURE(NGOOD) = TEMP
C SALINITY(NGOOD) = SALTY

C*********************************************************************
INDCR(NGOOD) = NDETAIL
RETURN
END

MAY 1, 1978

PART OF PROGRAM OCCOMP
INTERPOLATES TEMPERATURE, THETA, SALINITY, SIGMA-THETA,
AND DELTA AT STANDARD DEPTHS.

PROGRAMMER: MARY HUNT
DATE: FEBRUARY, 1977

COMM/LOGUNIT/ LP, ICR, INUNIT, SAVFIL
INTEGER LP, ICK, INUNIT, SAVFIL

COMM/OPTIONS/ COMPOPT, INOPPT, SAVOPT,
* BLKFAC, ENDATA
COMM/OPTIONS/ NUMSTAT, MAXOBS
INTEGER COMPOPT, INOPPT, SAVOPT, BLKFAC
LOGICAL ENDATA
INTEGER NUMSTAT, MAXOBS

COMM/INPUTVAR/ DEPTH(140), TEMPERATURE(140), SALINITY(140),
+ DOUBLE PRECISION DEPTH, TEMPERATURE, SALINITY, XLAT, XLONG
INTEGER NGOOD

DIMENSION IK1(140), IK3(140)

COMMON/STANDARD/ STANDEP(140), STANPRES(140),
+ NSDEP, NSPRES, IREF, IPROP
DOUBLE PRECISION STANDEP, STANPRES
INTEGER NSDEP, NSPRES, IREF

COMMON/DATASAV/
+ NDETAIL, NINT,
+ NSAV1, NSAV2
REAL SAVDAT2
INTEGER NDETAIL, NINT
INTEGER NSAV1, NSAV2
**C**

**C********************************************************************
C**
COMMUN/LCL/ XTM(140), XTHL(140), XSAL(140),
  + XSGTH(140), XDEL(140), XTRA(140),
  + FRST, LST, INX, JUNK(8)
DOUBLE PRECISION XTEM, XTHT, XSAL
DOUBLE PRECISION XSGTH, XDEL, XTRA
DOUBLE PRECISION XXX, ZED
INTEGER FRST, LST, INX, JUNK
DOUBLE PRECISION PRESX(140), ALPHAIL, SNTEMP
C
**C********************************************************************
C**

**C**
CHECK END POINTS OF OBSERVED DEPTHS
FIND LAST STANDARD DEPTH WITHIN
RANGE OF OBSERVED DEPTHS
"LST" WILL BE INDEX

**C**

DEPTH(1) = IDINT ( DEPTH(1) + 0.5DO )

**C**
DO 10 LST = NSDEP, 1, -1
   IF ( STANDEP(LST) .LE. DEPTH(NGOOD) ) GO TO 20
10 CONTINUE

**C**
FIND FIRST STANDARD DEPTH WITHIN
RANGE OF OBSERVED DEPTHS
"FRST" WILL BE INDEX

**C**
20 CONTINUE
DO 30 FRST = 1, NSDEP
   IF ( STANDEP(FRST) .GE. DEPTH(1) ) GO TO 40
30 CONTINUE

**C**
COMPUTE NUMBER OF STANDARD DEPTHS
FOR INTERPOLATION.
IF LESS THAN 3, SKIP INTERPOLATION

**C**
40 CONTINUE
NINT = LST - FRST + 1
   IF ( ( NINT .LT. 3 ) .OR. ( NGOOD .LT. 3 ) )
   + WRITE(LP,2000) ;
   + NINT = 0 ;
   + RETURN
2000 FORMAT(' NOT ENOUGH POINTS FOR VALID INTERPOLATION.')
C
**C**
NOW DO THE INTERPOLATIONS
C
CALL ZLAINT ( DEPTH, TEMPERATURE, NGOOD,
   + STANDEP(FRST), XTEM(FRST), IK1(FRST),
   + NINT, 0.5DO )
C
CALL ZLAINT ( DEPTH, SALINITY, NGOOD,
   + STANDEP(FRST), XSAL(FRST), IK3(FRST),
   + NINT, 0.100 )
   CALL DPRESS(XLAT, STANDEP(FRST), XTEM(FRST), XSAL(FRST), PRESX(FRST), NINT)
INTERPOLATIONS COMPLETED.
PRINT PAGE AND COLUMN HEADINGS.

IF (IPROP.EQ.0) GO TO 78
CALL PRNTHED
WRITE(LP,2010)
2010 FORMAT(T8,'STANDARD DEPTH', T26,'TEMP', T36,'TETRA',+
T47,'SAL', T55,'SIGMA-', T66,'DELTA' /+
T55,'THETA',T75,'PRESS' //)

PRINT INTERPOLATED ARRAYS AND
SAVE FOR OUTPUT TO TAPE OR DISK.

78 INX = 1
DO 80 IN=FRST,LST
CALL DSGMT(XTEM(IN),XSAL(IN),SNTEMP,XSGTH(IN))
CALL DSPVOL(PRESX(IN),XTEM(IN),SNTEMP,XSGTH(IN),ALPHA1)
CALL DSVAMM(PRESX(IN),ALPHA1,XDEL(IN))
IF (IPROP.EQ.0) GO TO 79
WRITE(LP,2020) STANDEP(IN),XTEM(IN),IK1(IN),XSAL(IN),+
& IK3(IN),XSGTH(IN),XDEL(IN),PRESX(IN)
& F6.1,T75,F6.1)
79 IF (SAVEPT.EQ.0) GO TO 80
SAVDAT2(1,INX) = STANDEP(IN)
SAVDAT2(2,INX) = XTEM(IN)
SAVDAT2(3,INX) = XSAL(IN)
SAVDAT2(4,INX) = XSGTH(IN)
SAVDAT2(5,INX) = XDEL(IN)

INX = INX + 1

C

RETURN
END
C
SAVARAY.1 APRIL 27, 1978
SUBROUTINE SAVARAY
C
PART OF PROGRAM OCCOMP
STORES INFORMATION TO BE SAVED AS A BINARY FILE
ON LOGICAL UNIT # 1, USING BUFFEROUT
C
PROGRAMMER: MARY HUNT
DATE: FEBRUARY, 1977

******************************************************************************
COMMON/LOGUNIT/ LP, ICR, INUNIT, SAVFIL
INTEGER LP, ICR, INUNIT, SAVFIL
******************************************************************************
COMMON/HEADING/ COUNTRY, SHIP(2), LATITUDE, LATNS,
& LONGITUDE, LONGEW,
YEAR, MONTH, DAY, CRUISENO,
+ STATNO(3), BOTDEP, DEEPEST, TIMODAY(4),
+ LATMIN, LONGMIN, HOUR, USIT
INTEGER COUNTRY, SHIP, LATITUDE, LATNS, LONGITUDE, LONGEW
INTEGER YEAR, MONTH, DAY, CRUISENO
INTEGER STATNO, BOTDEP, TIMODAY
INTEGER DEEPEST
REAL LATMIN, LONGMIN, HOUR
C
C******************************************************************************
C
C******************************************************************************
C
COMMON/INPUTVAR/ CDEPTH(140), TEMPERATURE(140), SALINITY(140),
+ DOUBLE PRECISION DEPTH, TEMPERATURE, SALINITY, XLAT, XLONG
INTEGER NGOOD
C
C******************************************************************************
C
COMMON/DATA1AV/ SAVDAT2(5,140),
+ NDETAIL, NINT,
+ NSAV1, NSAV2
REAL SAVDAT2
INTEGER NDETAIL, NINT
INTEGER NSAV1, NSAV2
C
C******************************************************************************
C
COMMON/LOCAL/ BLAT, BLON,
+ NOUT1, NOUT2, JUNK(1687)
REAL BLAT, BLON
INTEGER NOUT1, NOUT2, JUNK
REAL RJUNK(1687)
EQUIVALENCE (JUNK, RJUNK)
C
C******************************************************************************
C
C
BLAT = XLAT
BLON = XLONG
C
JUNK(1) = COUNTRY
JUNK(2) = SHIP(1)
JUNK(3) = SHIP(2)
JUNK(4) = CRUISENO
JUNK(5) = STATNO(1)
JUNK(6) = STATNO(2)
JUNK(7) = STATNO(3)
JUNK(8) = DAY
JUNK(9) = MONTH
JUNK(10) = YEAR
RJUNK(11) = HOUR
RJUNK(12) = BLAT
RJUNK(13) = BLON
JUNK(14) = BOTDEP
JUNK(15) = NDETAIL
JUNK(16) = NINT
CALL BUFFEROUT (SAVFIL, 1, JUNK, 16, INKER)
C CCCCC IF ( INKER .NE. 2 ) GO TO 100
CCCCC NOUT1 = NSAV1 * NDETAIL
CCCCC CALL BUFFEROUT ( SAVFIL, 1, NOUT1, INKER )
C IF ( INKER .NE. 2 ) GO TO 100

C NOUT2 = NSAV2 * NINT
C IF ( NINT .GT. 0 )
C + CALL BUFFEROUT ( SAVFIL, 1, SAVDAT2, NOUT2, INKER )
C IF ( INKER .NE. 2 ) GO TO 100
C
RETURN
C
C 100 CONTINUE
C
C WRITE(LP,1000) INKER
C 1000 FORMAT(' OUTPUT ERROR ON SAVE FILE. IND. = ',I2)
STOP
C
C END
C
SUBROUTINE READARAY ( SD, NSD )
C
DOUBLE PRECISION SD(1)
C
NSD = 1
C 10 CONTINUE
C READ(105,1000,END=50) SD(NSD)
C 1000 FORMAT(I)
C NSD = NSD + 1
C GO TO 10
C
C 50 CONTINUE
C NSD = NSD - 1
C RETURN
C
C END
C
COPY CCGCOMP TO CP
PCL PROCESSING TERMINATED
COPY PICKOUT TC LP

C PROGRAM PICKOUT READS THE FILE OF NCDC DATA CREATED BY
C PROGRAM GCCOMP, FILTERS OUT SIGMA-TS WHICH HAVE BAD TEMP
C, SAL OR WHOSE DEPTHS DON'T MATCH STANDARD DEPTHS.
C THE SIGMA-TS ARE RECORDED ON TAPE OR DISK FILE

C PROGRAMMER: JAMES CHURCHILL W.H.O.I. 12/19/78

C COMMON CC, SCL, SC2, CN, SN1, SN2, SN3, IDA, IMG, IYR, GMT, FLAT, FLONG,
C & WD, ND2, ND3, DATA(180)
C COMMON CCP, SCP, SC2P, CNP, SN1P, SN2P, SN3P, IDAP, IMGP, IYRP
C & FLATP, FLCP, WDP, ND3P, ST(30)
C DIMENSION DM(28)
C INTEGER CC, SCL, SC2, CN, SN1, SN2, SN3, SCP, SC2P, CNP,
C & SN1P, SN2P, SN3P, CCP, WD, WDP
C DATA(CM(J),J) 1, 28 /1, 1C, 2C, 30, 40, 50, 60, 75, 100, /
C 125, 150, 200, 250, 300, 400, 500, 600, 700, 800, 900, /
C 1000, 1500, 1900, 1750, 2000, 2500, 3000, 4000, /
C
C INPUT NUMBER OF FILES ON INPUT TAPE TO SKIP AND NUMBER
C TO READ

C READ(105, 10) NOFTS, NTD00
C IF(NOFTS.GT.0) CALL SKPFIL(2,NOFTS)
C NDCE = 0
C FORMAT(212)
C
C READ MASTER RECORD
C
C CALL BUFFER IN (2,1,CC,16, ISTAT,NIN,MCNERR)
C IF(ISTAT.EQ.3) GC TO 1000
C IF(MCNERR.NE.0) OUTPUT,NIN,MCNERR
C CCP=CC
C SCP = SCL
C SC2P = SC2
C CNP = CN
C SN1P = SN1
C SN2P = SN2
C SN3P = SN3
C IMCP = IMG
C IDAP = IDA
C IYRP = IYR
C FLATP = FLAT
C FLCP = FLONG
C WDP = WD
C ND3P = ND3
C
C READ RECORD OF INTERPOLATED VALUES
C
C IF(ND3.EQ.0) WRITE(108,90): GO TO 1
C IF(ND3.GT.28) WRITE(108,91): GO TO 1
C CALL BUFFER IN (2,1, CATA, ND3*5, ISTAT)
C IF(ISTAT.EQ.3) GC TO 1000
C IF(ISTAT.EQ.4) WRITE(108,70)
C
C EXTRACT SIGMA-TS, SET TO ZERO SIGMA-TS WHICH HAVE BAD
C TEMP, SAL, OR DEPTH
NSK=0
DE1=CATA(1)
IF(DM(1).EQ.DL1) GO TO 45
ST(1)=0.
NSK=1
DO 55 I=2,28
IF(DM(I).EQ.DE1) GO TO 45
ST(I) = 0.
NSK= 1
59 CONTINUE
WRITE(108,92)DE1
GO TO 1
45 NDL = C
DO 100 I=1,N3
ST(I+NSK)=DATA(I-1)*5+4
DE=DATA(I-1)*5+1
TEMP=DATA(I-1)*5+2
SAL=DATA(I-1)*5+3
IF(CG.NE.DM(I+NSK)) WRITE(108,80)DM(I+NSK),DE ; GO TO 95
IF(TEMP.GT.40.,JR,TEMP.LT.-2.) GO TO 55
IF(SAL.GT.39.,OK,SAL.LT.26.) GO TO 55
GO TO 100
95 NDL = NDL+1
ST(I+NSK)=0.
100 CONTINUE
NU3P = ND3P+NSK
C
C WRITE MASTER RECORD AND SIGMA-TS ON DEVICE 1
C
CALL BUFFER OUT(1,1,CMP,14,ISTAT,N)
CALL BUFFER OUT(1,1,ST,ND3P,ISTAT2,N2)
NCC=ND3-NDL
IF(ISTAT.EQ.2,ANC,ISTAT2.EQ.2)
& WRITE(108,50)SCPF,SCP2,CNP,SN1P,SN2P,SN3P,IMOP,1DAP,
& 1YRP,FLATP,FLCNP,NDC,NDL
OUTPUT WD,WD
GO TO 1
50 FORMAT(' CRUISE: ',3A4,' STATION: ',3A4,3I3,3X,
& 'LAT: ',F7.2,' LNG: ',F7.2,3X,14,' SIGMATS LOADED',14,' REJECTED')
70 FORMAT(' INPUT ERROR')
80 FORMAT(' DEPTH',F9.1,' MISTAKENLY RECORDED AS',F9.1)
90 FORMAT(' NO INTERPOLATED VALUES, STATION SKIPPED')
91 FORMAT(' TOO MANY DEPTHS, STATION SKIPPED')
92 FORMAT(' FIRST DEPTH',F10.2,' CANNOT BE FOUND, STATION SKIPPED')
1000 NDCNE = NDCNE+1
IF(NDONE.LT.NTODC) GO TO 1
END FILE 1
CALL EXIT
END

WEOF LP
COPY GRID TO LP.

PROGRAM GRID
PROGRAM GRID AVERAGES SIGMATS AT STANDARD LEVELS
WITHIN A GEOGRAPHIC ARRAY OF GRID BOXES. THE DATA IS
FROM NODC, PROCESSED BY PROGRAMS OCCOMP AND PICKOUT.
DENSITY INVERSIONS IN THE AVERAGED SIGMATS ARE FLAGGED
AND CORRECTED. THERE IS AN OPTION TO FILTER DATA > 5 STD
FROM THE AVERAGE. ANOTHER OPTION CALLS FOR THE
CALCULATION OF THE FOLLOWING INTEGRALS:
FIE = INTEGRAL (ETA*U2) ; LIMITS 0 TO H
FIZ = INTEGRAL (ETA*Z/H/2) ; LIMITS 0 TO H
WHERE: H = BOTTOM DEPTH
ETA = (JEN-50)/50 ; SO = ABYSMALT DENSITY
THE AVERAGE, STANDARD DEVIATION, AND # OF OBSERVATIONS
FOR EACH GRID ELEMENT ARE OUTPUT TO BINARY FILE.
SUBROUTINES REQUIRED: STD, INTG

PROGRAMMER: JAMES CHURCHILL, W.H.O.I. 12/15/78

COMMON /IN/CC,SC1,SC2,CN,SN1,SN2,SN3,NID,NID,IMC,IMR,FLAT,FLONG,
& WD,NDS,ST(30)
COMMON NGRID, ILOPT, FLATC(270), FLONC(270), BOTE(270),
& ST(28,270), NST(28,270), SUST(28,270), NMAX(270),
& BODEP(270), FIE(270), FIZ(270)
DIMENSION OM(2J), DLIB(4), ELIB(4)
INTEGER CC, SC1, SC2, CN, SN1, SN2, SN3, WD
DATA(OM(J),J=1,281)/1.,10.,20.,30.,40.,50.,60.,70.,75.,80.,85.,
& 100.,125.,150.,200.,250.,300.,400.,500.,600.,700.,800.,900.,1000.,
& 1200.,1500.,1750.,2000.,2500.,3000.,4000./

INPUT PROCESSING OPTIONS:
IF ILOPT.NE.0 DJ NOT DO DEPTH INTEGRATIONS
IF IGPSDL.NE.0 DELETE DATA > 5 STD FROM THE AVERAGE

READ(105,6) ILOPT, ILOPTD
6 FORMAT(2I2)

READ ABYSMAL DENSITY AND LIBRARY SIGMATS TO FILL IN
MISSING DATA.
CALCULATE ETAS FROM THE SIGMATS

READ(105,7) NL1J,SO
READ(105,8)(DLIB(I),ELIB(I),I=1,NLIB)
7 FORMAT(2F7.4)
8 FORMAT(F6.0,F6.2)
WRITE(108,790) SC
DO 10 I=1,NLIB
WRITE(108,795) JLIB(I),ELIB(I)
10 DENI = 1. + ELIB(I)/1000.
ELIB(I) = (DENI-SC)/SO
CONTINUE

READ DIMENSIONS OF GRID
READ(15,11) NGRID, DLAT, DLON
11 FORMAT(32F5.2)
READ(15,20)(FLATC(I), FLONC(I), BOTE(I), I=1,NGRID)
20 FORMAT(2F10.2,F10.0)
INIALIZE

DO 75 IGR = 1, NGRID
NDMAX(IGR) = 0
DC 5C I=1,28
AST(I,IGR) = SDST(I,IGR)=0.
NST(I,IGR)=0
50 CONTINUE
75 CONTINUE

READ IN STATION DATA STATION BY STATION

5 CALL BUFFER IN(1,1,CC,14,ISTAT)
IF(ISTAT.EQ.3) GO TO 200
FLCNG = -FLCNG
CALL BUFFER IN(1,1,ST,ND,ISTAT)
IF(ISTAT.EQ.3) GO TO 200

ACCUMULATE DATA WITHIN GRID BOXES

DO 100 IGR = 1, NGRID
IF(FLAT.LT.(FLATC(IGR)-DLAT).OR.FLAT.GT.(FLATC(IGR)
& +DLAT)) GO TO 100
IF(FLCNG.LT.(FLUNC(IGR)-DLUN).OR.FLCNG.GT.
& (FLUNC(IGR)+DLGN)) GO TO 100
IF(NC.GT.NDMAX(IGR)) NDMAX(IGR)=ND
DO 150 I=1,ND
IF(ST(I).LE.0) GO TO 150
AST(I,IGR) = AST(I,IGR) + ST(I)
NST(I,IGR) = NST(I,IGR) + 1
SDST(I,IGR) = SDST(I,IGR) + ST(I)*ST(I)
150 CONTINUE
100 CONTINUE
GO TO 5

CALCULATE AVERAGES AND STANDARD DEVIATIONS WITHIN EACH GRID BOX

200 DC 300 IGR = 1, NGRID
DO 250 I=1, NDMAX(IGR)
IF(NST(I,IGR).EQ.0) GO TO 250
FN = NST(I,IGR)
AST(I,IGR) = AST(I,IGR)/FN
SUN = SDST(I,IGR)-FN*AST(I,IGR)*AST(I,IGR)
IF(SLN.LE.0) SDST(I,IGR) = 0.; GO TO 250
SDST(I,IGR) = SQRT(SUN/(FN-1.)
250 CONTINUE
300 CONTINUE

IF ICPTSD>0 DELETE DATA GREATER THAN 5 STD FROM THE AVERAGE
(BOTH THE STD AND THE AVERAGE ARE WITHOUT THE DATUM BEING CHECKED)

IF(ICPTSD.GT.0) WRITE(108,780);CALL SDAT(AST,NST,SDST,FLUNC,
& FLATC,DM,DLGN,DLAT,NGRID)

FOR EACH GRID OUTPUT THE AVERAGE FOR ALL LEVELS

DO 400 IGR = 1, NGRID
WRITE(108,800) FLNC(I,GRK),FLATC(I,GRK),DLCN,DLAT
WRITE(108,810)
DO 350 I=1,NNDMAX(IGR)
IF(NSI(I,IGR).EQ.0) GO TO 350
IF(I.EQ.1) GO TO 225

C CHECK FOR DENSITY INVERSION. IF PRESENT SET INVERTED
C VALUE EQUAL TO THE VALUE ABOVE AND FLAG WITH A -SIGN.
C
DIFF = ASI(I,IGR) - ABS(AST(I-1,IGR))
IF(DIFF.LT.0.0) WRITE(108,830)-DIFF;AST(I,IGR)=-AST(I-1,IGR)
WRITE(108,820)DM(I,NSI(I,IGR),ABS(AST(I,IGR)),SDST(I,IGR))
350 CONTINUE
C
DO DEPTH INTEGRATION IF IOPT = 0
C
IF(IOPT.GT.0) GO TO 400
CALL INTG(BOTDEP(I,GRK),NMAX(IGR),AST(I,IGR),
6 FIE(I,GRK),FIZ(I,GRK),BOTDEP(I,GRK),SDS(I,GRK),
WRITE(108,840) -FIE(I,GRK),-BOTDEP(I,GRK),FIZ(I,GRK),-
WRITE(108,850) BOTDEP(I,GRK)
400 CONTINUE

780 FORMAT(//40X'VALUES EXCEEDING 5STD DEV OF THE AVERAGES'//
, & 4X,'LAT',3X,'LONG',3X,'DEPTH',6X,'CRUISE',5X,
, & 'VALUE',4X,'CLD VALUES',4X,'NEW VALUES',40X,'ELL',3X,
, & 'NUM',3X,'AVE',3X,'STDEV',2X,'NUM',3X,'AVE',4X,'STDEV'/)
790 FORMAT(1A8B8YAL DENSITY =',F8.4//'
, & ' LIBRARY DENSITIES FOR GRIDS DEEPER THAN AVAILABLE DATA,' & //7X,'DEPTH','5X,'DEN SITY')
795 FORMAT(F12.0,F12.2)
800 FORMAT(//,' AVERAGED SIGMA-TS FOR GRID',F6.2,'DEG W',
, & F6.2,'DEG N/20X','SIZE:',F5.2,'DEG LONG X ',F5.2,
, & 'DEG LAT')
810 FORMAT(1D9.1,5X,'NUMBER',8X,'AVE',6X,'ST DEV'/11X,
, & 'OF CBS',7X,'S1GT')
820 FORMAT(F6.0,F19,F13.2,F11.2)
830 FORMAT(1D9.1,' DENSITY INVERSION OF',F7.2)
840 FORMAT(17X,'0','/ -INTEGRAL ETA*Z','0',/16X,'F6.0/21X','0',/16X,'F6.0/21X','0','INTEGRAL ETA*Z/H*Z')
850 FORMAT(1D9.1)' TOPCM ETA =',F8.3)
C
C OUTPUT DATA TO DISK OR TAPE
C
CALL BUFFER OUT(2,1,NGRID,2,1,ISTAT)
CALL BUFFER OUT(2,1,FLATC,NGRID,ISTAT)
CALL BUFFER OUT(2,1,FLONG,NGRID,ISTAT)
CALL BUFFER OUT(2,1,BOTDEP,NGRID,ISTAT)
CALL BUFFER OUT(2,1,AST,NGRID*28,ISTAT)
CALL BUFFER OUT(2,1,NST,NGRID*28,ISTAT)
CALL BUFFER OUT(2,1,SDST,NGRID*28,ISTAT)
IF(IOPT.GT.0) GO TO 1000
CALL BUFFER OUT(2,1,FIE,NGRID,ISTAT)
CALL BUFFER OUT(2,1,FIZ,NGRID,ISTAT)
CALL BUFFER OUT(2,1,BOTDEP,NGRID,ISTAT)
1000 END FILE 2
CALL EXIT
END
COPY STD TO LP

SUBROUTINE STD(AST,NST,SDST,FLGCC,FLATC,DM,DLCN,DLAT,NGRID)

SUBROUTINE STD RECEIVES FORM THE CALLING PROGRAM

AVERAGE SIGMA-TS, STANDARD DEVIATIONS AND # OF OBSERVATIONS

FOR EACH ELEMENT OF A GEOGRAPHIC GRID. THE DEVICE

CONTAINING THE STATION DATA IS REWIND AND REREAD.

FOR EACH ELEMENT ALL VALUES GREATER THAN 5 STANDARD

DEVIATIONS FROM THE AVERAGE ARE DELETED. THE CORRECTED

AVERAGE, STANDARD DEVIATION AND # OF OBSERVATIONS ARE

RETURNED.

PROGRAMMER: J. CHURCHILL, W. H. O. I. 12/22/78

COMMON /IN/CC,SC1,SC2,CN,SN1,SN2,SN3,IDA,IMC,RY,FLAT,FLCNG,
&page 1
WD,NC,ST(30)

DIMENSION AST(28,270),SDST(28,270),NST(28,270),FLCNG(270),
&page 2
FLATC(270),DM(28)

INTEGER CC,SC1,SC2,CN,SN1,SN2,SN3,WD

REWIND 1

INPUT DATA STATION BY STATION

CALL BUFFER IN(1,1,CC,14,ISTAT)

IF(ISTAT.EQ.3) GC TO 200

FLCNG = -FLONG

CALL BUFFER IN(1,1,ST,ND,ISTAT)

IF(ISTAT.EQ.3) GC TO 200

FIND THE GRIDS IN WHICH THE CONSIDERED STATION LIES

DO 100 IGR = 1,NGRID

IF(FLATC(IGR).LT.FLATC(I)=.AND.OR.FLATC(IGR).GT.(FLATC(I)=
               & CLAT)) GO TO 100

IF(FLONG.LT.(FLONG(IGR).LT.0).AND.OR.FLONG.GT.
               & (FLONG(IGR)).GT.0)) 100

100 CONTINUE

CONSIDER EACH LEVEL EXCEPT IF THE # OF OBSERVATIONS IS

+ LE.3 OR IF THE ST. DEV. IS + LE0.02

DO 150 I=1,ND

IF(ISST(I,IGR).LE.*02) GO TO 15G

IF(NST(I,IGR).LE.3) 150

150 CONTINUE

FN = NST(I,IGR)

CALCULATE THE AVERAGE AND THE STD. DEV. WITH THE CONSIDERED

STATION DELETED (ANEW,SDNEW).

ANEW = (AST(I,IGR)*FN-ST(I))/((FN-1.)

SDNEW = (FN-1.)*SDST(I,IGR)*SDST(I,IGR)-ANEW*ANEW

SDNEW = SDNEW + FN*AST(I,IGR)*AST(I,IGR)-ST(I)*ST(I)

IF(SDNEW.1,0.4) SDNEW = .001

SDNEW = SQRT(SDNEW/((FN-2.))

IF THE CONSIDERED SIGMA-T IS 5*SDNEW FROM ANEW, REPLACE

THE AVERAGE BY ANEW AND THE STANDARD DEVIATION BY SDNEW

AND DECREASE THE # OF OBS. BY 1
C
IF(ST(I).LT.ANEW-5.*SDNEW.OR.ST(I).GT.ANEW+5.*SDNEW)GO TO 130
GO TO 150
130 WRITE(108,800)FLATC(IGR),FLONC(IGR),DM(I),SC1,SC2,CN,ST(I),
& NST(I,IGR),AST(I,IGR),SDDST(I,IGR),NST(I,IGR)-1,ANEW,SDNEW
SDDST(I,IGR) = SDNEW
AST(I,IGR) = ANEW
NST(I,IGR) = NST(I,IGR)-1
150 CONTINUE
100 CONTINUE
GO TO 5
800 FORMAT(2F7.2,F8.0,3X,3A4,F7.2,15,2F7.2,15,2F7.2)
200 RETURN
FNC
WEOF LP
COPY INTG TO LP

SUBROUTINE INTG

SUBROUTINE INTG RECEIVES SIGMATS AT STANDARD LEVELS FROM
THE CALLING PROGRAM. THE FOLLOWING INTEGRALS ARE
RETURNED:

\[ \text{FIE} = \text{INTEGRAL}(\text{ETA} \times DZ) \text{ WITH LIMITS 0 TO H} \]
\[ \text{FIZ} = \text{INTEGRAL}(\text{ETA} \times Z/H \times CZ) \text{ WITH LIMITS 0 TO H} \]
WHERE: \( \text{ETA} = (\text{UEN-SO})/\text{SO} \); \( \text{SO} = \text{ABYSSAL DENSITY} \)
\( \text{H} = \text{BOTTOM DEPTH} \)

THE VALUE OF ETA AT THE BOTTOM IS ALSO RETURNED.
IF THE AVAILABLE DATA DOES NOT EXTEND TO THE
BOTTOM, THEN A LIBRARY OF STANDARD VALUES IS USED.

PROGRAMMER: JAMES CHURCHILL, H.H.G.I. 12/17/78

SUBROUTINE INTG(BOTDEP, NDMAX, AST, FIE, FIZ, BOTE, SO, DLIB, ELIB)
DIMENSION DM(28), AST(28), Z(28), D1(28), E(28), ELIB(4), DLIB(4)
DATA(DMI,J,J=1,28), 0.0, 10.0, 20.0, 30.0, 40.0, 50.0, 60.0, 75.0, 100.0, 125.0,
& 150.0, 200.0, 250.0, 300.0, 400.0, 500.0, 600.0, 700.0, 800.0, 900.0, 1000.0,
& 1200.0, 1500.0, 1750.0, 2000.0, 2500.0, 3000.0, 4000.0,
NDMA = NDMAX

CALCULATE THE ETA AND ETA*DEPTH FROM THE SIGMATS

DO 100 I=1,NDMA
 IF(AST(I).EQ.0.0) NDMAX = I-1; GO TO 101
 UEN1 = 1.0 + ABS(AST(I))/1000.
 E(I) = (UEN1-SO)/SO
 IF(DM(I).GE.BOTDEP) D2 = DM(I); GO TO 200
 Z(I) = E(I)*DM(I)
 D1(I) = DM(I)
 100 CONTINUE

IF THE BOTTOM DEPTH IS DEEPER THAN AVAILABLE DATA USE
THE LIBRARY VALUES

101 WRITE(108,800)BOTDEP,DM(NDMAX)
 I=NDMAX+1
 DO 150 J=1,4
 IF(CLIB(J).LT.BOTDEP) GO TO 175
 E(I) = ELIB(J)
 D2 = DLIB(J)
 GO TO 200
175 IF(CLIB(J).LE.DM(NDMAX)) GO TO 150
 E(I) = ELIB(J)
 D1(I) = CLIB(J)
 Z(I) = E(I)*D1(I)
 I=I+1
 150 CONTINUE

INTERPOLATE ETA TO THE BOTTOM DEPTH

200 D1(I) = BOTDEP
 D1 = D1(I-1)
 E(I) = E(I-1)+(Z(I(I)-D1)*(E(I)-E(I-1))/(D2-D1))
 BOTE = E(I)
 Z(I) = E(I)*D1(I)
 NBOD = I
C
C       CALCULATE TRAPEZOIDAL INTEGRALS
C
DO 360 I=2,NBOU
EA = (E(I)+E(I-1))/2.
ZA = (Z(I)+Z(I-1))/2.
FIE = FIE + EA*(DI(I)-DI(I-1))
FIZ = FIZ + ZA*(CI(I)-DI(I-1))
300 CONTINUE
   FIZ = -FIZ/BOTDEP
800 FORMAT('BOTTOM DEPTH OF ',F6.0,' IS GREATER THAN DEEPEST DATA OF ',
       'F6.0')
RETURN
END
COPY TICGRID TO LP

PROGRAM TICGRID TYPES OUT SIGMATS, STANDARD DEVIATIONS OR
NUMBER OF OBSERVATIONS ON A GEOGRAPHIC GRID

PROGRAMMER: J. CHURCHILL W.H.C.I. 12/22/78

COMMON NGRID, IUPF, FLRT, FLONC(270), AST(28, 270), NST(28, 270)
& SDST(28, 270), FLT(270), FIZ(270), BCT(270), BCT(28, 270)
DIMENSION IAA(270), IAP(270), DM(28), JTITLE(20)
DATA DM(J, J) = 1, 28, 1, 1, 125, 125, 125, 125, 100, 100, 100,
& 100, 150, 200, 250, 300, 400, 500, 600, 700, 800, 900, 1000,
& 1200, 1500, 1750, 2000, 2500, 3000, 4000,

INPUT DIMENSIONS AND INCREMENTS OF OUTPUT GRID

READ(105, 5) FLC, MIN, FLONMAX, DLON
READ(105, 5) FLATMIN, FLATMAX, DLAT
5 FORMAT(3F6.2)
NLAT = (FLATMAX-FLATMIN)/DLAT + 1
READ(105, 6) (JTITLE(I), I = 1, 20)
6 FORMAT(20A4)
NLON = (FLONMAX-FLONMIN)/CLAT + 1

INPUT GRID OF INPUT DATA AND DATA

CALL BUFFER IN(2, 1, NGRID, 2, ISTAT)
CALL BUFFER IN(2, 1, FLATC, NGRID, ISTAT)
CALL BUFFER IN(2, 1, FLATNC, NGRID, ISTAT)
CALL BUFFER IN(2, 1, BOTDEP, NGRID, ISTAT)
CALL BUFFER IN(2, 1, AST, NGRID*28, ISTAT)
CALL BUFFER IN(2, 1, NST, NGRID*28, ISTAT)
CALL BUFFER IN(2, 1, SDST, NGRID*28, ISTAT)
IF (IUPF GT 0) GO TO 7
CALL BUFFER IN(2, 1, FIE, NGRID, ISTAT)
CALL BUFFER IN(2, 1, FIZ, NGRID, ISTAT)
CALL BUFFER IN(2, 1, NOTE, NGRID, ISTAT)

INPUT DESIRED LEVEL AND, AND DESIRED OUTPUT
NWant = 1 FOR SIGMATS, 2 FOR ST DEVS, 3 FOR # OF OBS
4 FOR INTEGRAL ETA, 5 FOR INTEGRAL ETA*Z, 6 FOR BOTTOM DEPTH, 7 FOR GRID NUMBERS, 8 FOR BOTTOM DENSITY

7 NDDEP = 0
8 READ(105, 10, END = 1000) NDDEP, NWANT
10 FORMAT(2I2)
IF (NDDEP GT 28) GO TO 1000

TRANSFORM DATA TO DESIRED FORM FOR .PRINTOUT
AND PRINT HEADING

GO TO (100, 110, 12C, 160, 18C, 195, 198, 201) NWANT
DO 130 I = 1, NGRID
110 WRITE(108, 800) (JTITLE(I), I = 1, 20)
GO TO 200
120 CONTINUE

130 WRITE(108, 800) (JTITLE(I), I = 1, 20)
GO TO 200
IAA(I) = SDST(NDEP,I)*100. + .5
CONTINUE
WRITE(108,810) (JTITLE(I), I=1,20)
GO TO 200
DO 150 I=1,NGRID
IAA(I) = NST(NDEP,I)
CONTINUE
WRITE(108,820) (JTITLE(I), I=1,20)
GO TO 200
DO 170 I=1,NGRID
IAA(I) = -FIE(I)*100. + .5
CONTINUE
WRITE(108,822) (JTITLE(I), I=1,20)
GO TO 210
DO 180 I=1,NGRID
IAA(I) = -FIZ(I)*100. + 0.50
CONTINUE
WRITE(108,824) (JTITLE(I), I=1,20)
GO TO 210
DO 197 I=1,NGRID
IAA(I) = BCTDEP(I)/10.
CONTINUE
WRITE(108,826) (JTITLE(I), I=1,20)
GO TO 210
DO 198 I=1,NGRID
IAA(I) = 1
CONTINUE
WRITE(108,827) (JTITLE(I), I=1,20)
GO TO 210
DO 201 I=1,NGRID
ST = ((BCTE(I)+1.)*1.028-1.)*1000.
IAA(I) = (ST-20.)*100. + .50
CONTINUE
WRITE(108,828) (JTITLE(I), I=1,20)
GO TO 210
WRITE(108,830) DI(NDEP)
WRITE(108,840) FLNMAX,FLATMAX
C
C TYPE OUT DATA LINE BY LINE
C
FLAT = FLATMAX
DO 400 J=1,NLAT
FLCA = FLCNMAX
DO 360 I=1,NLGN
DO 250 II=1,NGRIC
IF(FLAT.EQ.FLAC(II).AND.FLCN.EQ.FLNC(II)) GC TO 270
CONTINUE
IAP(I,J) = 0
GO TO 280
CONTINUE
IAP(I,J) = IAA(I)
FLCN = FLCA-DLCN
CONTINUE
IF(FLAT.LE.FLATMIN) WRITE(108,855) NLCN,(IAP(IP), IP=1,NLGN)
& : GO TO 400
WRITE(108,850) NLCN,(IAP(IP), IP=1,NLGN)
FLAT = FLAT-DLAT
CONTINUE
WRITE(108,860) NLCN*4-2,NLCN*4-12,FLCAML,FLATMIN
GO TO 3
FORMAT(*1*20A4/5C*,*SIGMA'S*)
810 FORMAT(*'1',20A4/48X,*'STD.DEVIATIONS')
820 FORMAT(*'1',20A4/45X,*'NO OF OBSERVATIONS')
822 FORMAT(*'1',20A4/41X,*'INTEGRAL(ETA*Z/CZ) -CM')
824 FORMAT(*'1',20A4/40X,*'INTEGRAL(ETA*Z/H *CZ) -CM')
826 FORMAT(*'1',20A4/47X,*'BOTTOM DEPTH/10')
827 FORMAT(*'1',20A4/47X,*'GRID NUMBERS')
828 FORMAT(*'1',20A4/44X,*'BOTTOM DENSITIES')
830 FORMAT(43X,*'FJK DEPTH',F6.0)
840 FORMAT(1X,F6.2,1X,F6.2/1)
850 FORMAT(N14/)
855 FORMAT(N14)
860 FORMAT(NX,1X/NX,F6.2,1X,F6.2)
1000 CALL EXIT
END

WEOF LP
COPY GRPRS TO LP

PROGRAM GRPRS
  PROGRAM GRPRS READS A BINARY FILE CONTAINING A GRID
  OF AVERAGED SIGMATS AT STANDARD LEVELS COMPUTED BY
  PROGRAM GRID. THE SEA SURFACE LEVEL CALCULATIONS
  ARE DONE AS FOLLOWS:
  JD = -INTEGRAL(ETA*DZ) ; LIMITS = 0 TO -H
  JM = INTEGRAL(ETA*Z/H*DZ) ; LIMITS = 0 TO -H
  Jh = INTEGRAL(JM/H*DF) ; LIMITS = -H TO HMAX
  JB = -INTEGRAL(BOTETA*UH) ; LIMITS = -H TO HMAX
  WHERE: H = BOTTOM DEPTH ; LTA = (DEN-ABYSALL DEP/ABYSALL DEP
  JB AND JH ARE SPECIFIED AS PIECEWISE LINEAR FUNCTIONS
  OF BOTTOM DEPTH
  ELEV:1 = JD - JM + JH
  ELEV:2 = JD + JB

SUBROUTINES REQUIRED: INTC,TYGR,SKPREC(WHUI.3)

PROGRAMMER: JAMES CHURCHILL, W.H.O. 12/15/78

COMMON NGRID IUPT,FLATC(270),FLONC(270),BOTDEP(270),
& AST(25,270),FIC(270),FIZ(270),HOTECT(270),NDMAX(270)
DIMENSION F1(270),F1M(270),NPT(10),DLIB(4),ELIB(4),
& JTITL(20),JU(25),FJH(25),FJB(25)

INPUT OPTIONS: IOPTI = 0 DO NOT DO INTEGRATIONS;
IOPTY = 0 DO NOT TYPE OUTPUT GRID

READ105,1 IOPTI,IOPTY
  FORMAT(2I2)

INPUT ABYSAL DENSITY AND LIBRARY DENSITIES

READ105,7 NLIB,SO
READ105,8(3LIB(I),ELIB(I),I=1,NLIB)
  FORMAT(2I2,F7.4)
  FORMAT(2F6.2)
WRITE108,790 SO
DO 10 I=1,NLIB
  READ108,795 DLIB(I),ELIB(I)
  WRITE108,799 DLIB(I),ELIB(I)
  DEN1 = 1. + ELIB(I)/1000.
  ELIB(I) = (DEN1-50)/50
  CONTINUE

INPUT DIMENSIONS AND INCREMENTS OF OUTPUT GRID

READ105,5 FLCNMIN,FLCNRAX,MCLON
READ105,5 FLATMIN,FLATMAX,CLAT
  FORMAT(3F6.2)
NLAT = (FLATMAX-FLATMIN)/CLAT + 1
NLON = (FLCNRAX-FLCNMIN)/CLAT + 1

INPUT TITLE FOR OUTPUT GRID

READ105,9(JTITL(I),I=1,20)
  FORMAT(20A4)
INPUT NUMBER OF DEPTHS AT WHICH VALUES OF LINE INTEGRALS ARE GIVEN

READ(105,11) NDEP
11 FORMAT(12)

INPUT VALUES OF DEPTH, JH AND JB IN DECREASING DEPTH ORDER

READ(105,12)(D(J),FJH(J),FJB(J),J=1,NDEP)
12 FORMAT(F5.0,2F10.4)

INPUT GRID OF INPUT DATA AND DATA

CALL BUFFER IN(2,1,NGRID,2,ISTAT)
CALL BUFFER IN(2,1,FLATC,NGRID,ISTAT)
CALL BUFFER IN(2,1,FLONC,NGRID,ISTAT)
CALL BUFFER IN(2,1,BOTDEP,NGRID,ISTAT)
CALL BUFFER IN(2,1,AST,NGRID*28,ISTAT)
IF (ICPT.GT.0) GO TO 100
IF (ICPT.GT.0) GC TO 100
CALL SKPREG(2,2)
CALL BUFFER IN(2,1,FLH,NGRID,ISTAT)
CALL BUFFER IN(2,1,FLI,NGRID,ISTAT)
CALL BUFFER IN(2,1,BOTF,NGRID,ISTAT)

DO 99 I=1,NGRID
99 FIZ(I) = -FIZ(I)

CONTINUE

GO TO 200

IF DESIRED COMPUTE DEPTH INTEGRALS, JG AND JM

DO 150 IGR = 1,NGRID
150 CONTINUE

NDMAX(IGR) = 28

CALL INTG(BOTDEP(IGR),NDMAX(IGR),AST(1,IGR),FLH(IGR),& FIJH(IGR),FJB(IGR),FLONC(IGR),BOTE(IGR),SO,DLIB(1),ELIB(1))

WRITE(108,800)
WRITE(108,805)(D(I),FJH(J)*100.,FJB(J)*100.,J=1,NDEP)
WRITE(108,807)

FOR EACH GRID INTERPOLATE JB AND JH TO THE BOTTOM DEPTH
AND CALCULATE THE SURFACE ELEVATION FUNCTIONS

DO 250 IGR = 1,NGRID
250 CONTINUE

ZH = BCTDEP(IGR)
DO 225 I=2,NDEP
IF(D(I).LE.ZH) GG TO 230
225 CONTINUE

I=NDEP

FLJ+(FJH(I)-FJH(I-1))*(ZH-D(I-1))/(D(I)-D(I-1))
FLJ+(FJB(I)-FJB(I-1))*(ZH-D(I-1))/(D(I)-D(I-1))
FLJ+(IGR) = -FLJ+(IGR) - FIZ(IGR) + FLH
FLJ+(IGR) = -FLJ+(IGR) + FLH
WRITE(108,810) IGR,FLH(IGR),FLATC(IGR),FLONC(IGR),BOTE(IGR),& FIZ(IGR),FJH(IGR),FJB(IGR),FLI(IGR),FIJH(IGR),FJB(IGR)

CONTINUE
IF (DGPT, EQ, 0) GC TO 1000
DO 510 I = 1, NGRID
  F1E(I) = -F1E(I)*100.  
  F1Z(I) = F1Z(I)*100.    
  F1(I) = F1(I)*100.    
  FIEM(I) = FIEM(I)*100.  
  B0TDEP(I) = B0TDEP(I)/10.
510 CONTINUE
WRITE(108, 850) 1JTITLE(I), I = 1, 20
CALL TYGR(IFLATMIN, FLATMAX, FLCNMIN, FLCAMAX, NLAT, NLON, 
   & DLON, DLAT, NGRID, FLATC, FLONC, BOTDEP)
WRITE(108, 860) 1JTITLE(I), I = 1, 20
CALL TYGR(IFLATMIN, FLATMAX, FLCNMIN, FLCAMAX, NLAT, NLON, 
   & DLON, DLAT, NGRID, FLATC, FLONC, F1E)
WRITE(108, 870) 1JTITLE(I), I = 1, 20
CALL TYGR(IFLATMIN, FLATMAX, FLCNMIN, FLCAMAX, NLAT, NLON, 
   & DLON, DLAT, NGRID, FLATC, FLONC, F1Z)
WRITE(108, 890) 1JTITLE(I), I = 1, 20
CALL TYGR(IFLATMIN, FLATMAX, FLCNMIN, FLCAMAX, NLAT, NLON, 
   & DLON, DLAT, NGRID, FLATC, F1)
WRITE(108, 910) 1JTITLE(I), I = 1, 20
CALL TYGR(IFLATMIN, FLATMAX, FLCNMIN, FLCAMAX, NLAT, NLON, 
   & DLON, DLAT, NGRID, FLATC, FIEM)
790 FORMAT(* 1 ABBYSAL DENSITY = "F8.4 /*
   & /7X,'DEPTH' ',5X,'DENSITY')*
795 FORMAT(F12.2)
800 FORMAT(*1',40X,'INTEGRALS'/* KEY:/
   & JD = INTEGRAL(ETA*DZ) ; LIMITS = 0 TO -H' /
   & JM = INTEGRAL(ETA*Z/H) ; LIMITS = 0 TO -H' /
   & JH = INTEGRAL(JM) ; LIMITS = -H TO -HMAX' /
   & JB = INTEGRAL(ETA) ; LIMITS = -H TO -HMAX' /
   & PIECEWISE LINEAR FUNCTIONS WITH ENDPCINTS:"/
   & DEPTH, 'X; '+8X,'JD', '+8X,'JM', '+8X,'JH', '+8X,'JB', '+8X,'CM', '+8X,'CM'
805 FORMAT(F6.0, 2F10.2)
870 FORMAT(*1',7X,'LAT', '+6X,'LONG'; '+3X,'BCTOM DEPTH/10' /
   & JD', '+8X,'JH', '+8X,'JL', '+7X,'JDMIN'
810 FORMAT(14, 3X,F7.2, 3X,F7.2, 4X,F6.0, 2F10.4, 12.3, 2F10.4)
850 FORMAT(*1',20A4/4X,'BCTOM DEPTH/10' /
860 FORMAT(*1',20A4/4X,'INTEGRAL(ETA*Z/H) -CM'
870 FORMAT(*1',20A4/4X,'INTEGRAL(ETA*Z/H) -CM'
900 FORMAT(*1',20A4/3X)
   & INT(JM/H*DH) -INT(ETA*Z/H*DH) -CM'
910 FORMAT(*1',20A4/3X,'INT(BCTOM*Z/H*DH) -INT(ETA*Z/H*DH) -CM'
1000 CALL EXIT
END
COPY TYGRS TO LP

C SUBROUTINE TYGR
C SUBROUTINE TYGR TYPES VALUES FROM THE CALLING PROGRAM "A GEOGRAPHIC GRID"
C PROGRAMMED BY J. CHURCHILL, W.H.O.1. 12/19/78
C
C SUBROUTINE TYGR(FLATMIN, FLATMAX, FLOMIN, FLOMAX, NLAT, NLON, DLAT, DLONG, FNU)
C DIMENSION FLATC(270), FLONC(270), FNU(270), IAA(270), IAP(270)
C DO 10 I = 1, NGRID
C 10 CONTINUE
C WRITE(108,840) FLCNMAX, FLATMAX
C
C TYPE OUT DATA LINE BY LINE
C
C FLAT = FLATMAX
C DO 400 I = 1, NLAT
C FLCN = FLOMAX
C DO 300 J = 1, NLON
C DO 250 II = 1, NGRID
C IF(FLAT.EQ.FLATC(III).AND.FLCN.EQ.FLONC(III)) GC TO 270
C 250 CONTINUE
C IAP(J) = 0
C GO TO 280
C 270 IAP(J) = IAA(II)
C 280 FLCN = FLCN-DLCN
C 300 CONTINUE
C IF(FLAT.LE.FLATMIN) WRITE(108,855) NLCN, (IAP(IP), IP=1, NLCN)
C & GC TO 400
C WRITE(108,850) NLCN, (IAP(IP), IP=1, NLCN)
C FLAT = FLAT-DLAT
C 400 CONTINUE
C WRITE(108,860) NLCA*4-2, NLCN*4-12, FLOMIN, FLATMIN
C 840 FORMAT(1X,F6.2,' ',F6.2/F6.2/' + ')
C 850 FORMAT(N4)
C 855 FORMAT(N4)
C 860 FORMAT(N6/4/NX,'/+ /NX,F6.2,' ',F6.2)
C RETURN
C END

IEOF LP
CALCULATION OF MEAN SEASONAL SEA SURFACE ELEVATION FIELDS OVER THE NORTH ATLANTIC CONTINENTAL SHELF AND SLOPE USING HISTORICAL HYDROSTATION DATA

James H. Churchill
Woods Hole Oceanographic Institution
Woods Hole, MA 02543

Department of Energy

This report details the calculation and results of a simple coastal elevation field model which was presented in a recent paper (Csanady, 1979). Seasonal mean sea surface elevation fields were calculated over the shelf slope region off the east coast of North America north of Cape Hatteras by applying the model to a grid of averaged sigma-t's at standard depths. The sigma-t grid was calculated utilizing a data base of approximately 30,000 hydrographic stations from NODC archives. Difficulties involved with the numerical implementation of the model and economizing adaptations to the WHOI-NODC data processing program, OCCOMP, are discussed.

1. Coastal pressure field calculation
2. Model of coastal pressure field
3. East Coast continental shelf & slope pressure field calculation
1. Coastal pressure field calculation
2. Model of coastal pressure field
3. East Coast continental shelf & slope pressure field calculation
1. Churchill, James H.
11. DE-AC02-79-EV10005

This report details the calculation and results of a single coastal elevation field model which was presented in a recent paper (Csanady, 1979). Seasonal mean sea surface elevation fields were calculated over the shelf slope region off the east coast of North America north of Cape Hatteras by applying the model to a grid of averaged sigma-t's at standard depths. The sigma-t grid was calculated utilizing a data base of approximately 30,000 hydrographic stations from NODC archives. Difficulties involved with the numerical implementation of the model and economizing adaptations to the WHOI-NODC data processing program, OCCOMP, are discussed.

This card is UNCLASSIFIED

1. Coastal pressure field calculation
2. Model of coastal pressure field
3. East Coast continental shelf & slope pressure field calculation
1. Churchill, James H.
11. DE-AC02-79-EV10005

This report details the calculation and results of a single coastal elevation field model which was presented in a recent paper (Csanady, 1979). Seasonal mean sea surface elevation fields were calculated over the shelf slope region off the east coast of North America north of Cape Hatteras by applying the model to a grid of averaged sigma-t's at standard depths. The sigma-t grid was calculated utilizing a data base of approximately 30,000 hydrographic stations from NODC archives. Difficulties involved with the numerical implementation of the model and economizing adaptations to the WHOI-NODC data processing program, OCCOMP, are discussed.

This card is UNCLASSIFIED