D. Crease

NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY

N. I. O. Computer Programs 3

by

DIANA CATTON

N. I. O. Internal Report No. N 3

SEPTEMBER 1961

P.E. Anomaly Ψ

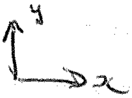
$$\Psi(x,p_1) = \int_0^{p_1} \alpha\, p\, dp$$

$$\left(\frac{\partial \Psi}{\partial x}\right)_{p} = \left\{\frac{\partial}{\partial x}\left[\int_0^{p_1} \alpha p\, dp\right]\right\}_{p_1}$$

$$= \left\{\frac{\partial}{\partial x}\left[\int_{z_0}^{z_1} \alpha p \left(\frac{g\, dz}{\alpha}\right)\right]\right\}_{z_1} - \left(\frac{\partial p}{\partial x}\right)_{z_1}\left\{\frac{\partial}{\partial p}\left[\int_0^{p_1} \alpha p\, dp\right]\right\}_x$$

$$= \int_{z_0}^{z_1} g\left(\frac{\partial p}{\partial x}\right)_{z_1} dz - \left[\alpha\left(\frac{\partial p}{\partial x}\right)_{z_1} \cdot p\right]_{p=p_1}$$

But $fv = +\alpha \frac{\partial p}{\partial x}$



$$\frac{1}{fg}\left(\frac{\partial \Psi(p,x)}{\partial x}\right)_{p=p_1} = \int_{z_0}^{z_1} \rho v\, dz - \left[\frac{p v}{g}\right]_{p=p_1}$$

$$= \int_{z_0}^{z_1} \rho v\, dz - v(p_1)\int_{z_0}^{z_1} \rho\, dz$$

$$= \int_{z_0}^{z_1} \rho\left(v(z) - v(z_1)\right) dz.$$

Thus $\frac{1}{fg}\frac{\partial \Psi}{\partial x} = \Phi$ is the mass transport relative to a depth z_1 between that depth & surface.

To find mass transport between two layers rel. to level 1

$$\Phi(p_1) - \Phi(p_2) = \int_{z_0}^{z_1} \rho\left(v(z) - v(z_1)\right) dz - \int_{z_0}^{z_2} \rho\left(v(z) - v(z_2)\right) dz$$

$$= \int_{z_2}^{z_1} \rho\left(v(z) - v(z_1)\right) dz + \int_{z_0}^{z_2} \rho\left(V(z_2) - v(z_1)\right) dz$$

Thus the mass transport up to level 2 relative to level 1 is given by

$$\frac{1}{fg}\left[\frac{\partial \Psi}{\partial x}(p_1) - \frac{\partial \Psi}{\partial x}(p_2)\right] - \int_{z_0}^{z_2} \rho\,(v(z_2) - v(z_1))\,dz$$

but the last integral is just ...

$$-\frac{p_2}{fg}\left[\frac{\partial \Delta D}{\partial x}(p_2) - \frac{\partial}{\partial x}\Delta D(p_1)\right]$$

where ΔD is the dynamic height anomaly $\int_0^p \delta\,dp$

so the mass transport relative to level 1 is

$$\frac{1}{fg}\left[\frac{\partial}{\partial x}\left(\Psi(p_1) - \Psi(p_2)\right) - p_2\frac{\partial}{\partial x}\left(\Delta D(p_1) - \Delta D(p_2)\right)\right]$$

Explanation of lists

Each station has two tables -

a) At observed pressures:

Column	Units	Decimal Places	Observation	Remarks
1	-	-	serial number	
2	decibars	0	pressure	Observed or interpolated from unprotected thermometer readings
3	metres	0	depth	Calculated from pressure and spec. volume
4	‰	3	salinity	Electrical conductivity
5	°C.	2	temperature	
6	°C.	2	potential temp.	Computed from Ekman's compressibility formula (1908), and Cox & Smith's (1959) values of specific heat.
7		3	σ_t	
8	ml/gm	5	sp. volume	
9	ml/gm	6	sp. volume anomaly	

b) At standard pressures:

Column	Units	Decimal Places	Observation	Remarks
1	decibars	0	pressure	
2	dyn. metres	3	dyn. height anomaly	By interpolation and integration of sp. vol. anomalies
4	dyn. metres decibars	1	potential energy anomaly	By integration of product of pressure and sp. vol. anomaly
6	m/sec.	1	sound velocity	By W.D.Wilson's formula (1960)
8	m/sec.	1	sounding velocity	Average sound velocity from surface to pressure p

Notes:

1. The interpolation to standard pressures is 3 point Lagrangian. Two interpolations are made at each standard pressure, firstly with two pivotal points above and secondly with two below the required pressure. The results in columns 2, 4, 6 and 8 are the mean of the two interpolations and the errors in columns 3, 5, 7 and 9 are half the difference.

2. Largish errors are usually due either to
   a) the impossibility of using 3 point Lagrangian interpolation at the end points; or
   b) the use of 2 pivotal points very close together (e.g., at changeover from one cast of bottles to another). This can usually be resolved by inspection.

3. All integrations are with respect to pressure starting with the lowest pressure. When this is not at the surface the following starting procedures are used:
   a) first depth - quadratic function of pressure determined as closely as possible from earlier data in same area;
   b) first dyn. height and potential energy anomaly - zero at first observed pressure in table (a);
   c) first sounding velocity - sound velocity at first observed pressure.

4. 0·16 m/sec. should be added to all sound and sounding velocities as formula used refers to absolute pressure and not pressure above atmospheric used elsewhere.

N.I.O. COMPUTER PROGRAMS III

by

Diana Catton

National Institute of Oceanography

N.3

CONTENTS

Introduction

This report contains a collection of oceanographic programs written for Mercury between January 1960 and December 1961. Most of them are special purpose programs and no attempt has been made to generalise them, because if variations are required it is usually easier to write a new Autocode program.

The descriptions are intended for the operator, but also contain comments on the programming and mathematics. The following general remarks apply to most programs, and for convenience are given under the usual headings.

Title Each program tape is headed by at least three titles.

(1) N.I.O. Program number.
(2) Programmer's name and RAE Job number.
(3) Name of Program, which is given at the beginning of the description.
(4) Column headings, if required.

Code This is either Autocode or Pig II.

Parameter tapes For the method of punching, see Data. Usually a parameter tape is short, so for ease of handling it is best to stick it either to the end of the program or to the beginning of the data tape.
Particular care must be taken if the data tape is the output from another program, as then all titles must be removed before the parameter tape is attached.

Data Numbers must be punched in the standard Mercury code; that is, each number must be followed by SpSp or CRLF.

End indication In general, all program and data tapes should end with

---> CRLF

unless otherwise specified. It is also useful to punch 6 erases on the very end of a tape.

Operation On the RAE Mercury, it is convenient to leave the actual machine operation to the operators provided. The information given in the program descriptions should be copied onto the briefing sheet.

Output Blank tape is used liberally to separate groups of results, and some tapes end with the conventional 6 erases. All results start with a copy of the titles punched at the beginning of the program tape.

Parameters These are not generally used.

Time In general this excludes reading the program tape. The approximate times given are intended only as a guide.

Method This gives a summary of the mathematics involved, and defines symbols used elsewhere in the description. The notation does not always correspond with the Autocode variables used within the program.

Note This whole report presupposes some knowledge of Autocode. If there is any doubt about punching data etc. the operator should consult an Autocode manual.

N.I.O. Program 25

Title Digitiser Conversion.

Code Mercury Pig II.

Purpose To convert paper tape punched in the N.I.O. digitiser code to paper tape punched in Mercury teleprinter code.

Order of tapes Program, parameter tape, data.

Parameter tape M, in Mercury teleprinter code. This may be punched by hand at the beginning of the data tape.

Data A tape in N.I.O. digitiser code. (See Method). The first block to be converted must be preceded by a comma; no commas are allowed before this, as the first comma on the data tape is the signal to begin conversion. Everything (except M - see parameter tape - if punched at the beginning) before the first comma is ignored.

End indication At least eleven consecutive rows of erases must be punched at the end of the data tape, and may begin anywhere in a block. No blank tape may be left between the data and the erases.

Operation (1) Read the program tape. It comes to a 99 stop.

(2) Put the parameter tape in the reader and prepulse. When M has been read the program punches 40 rows of blank tape and comes to a 99 stop.

(3) Put the data tape in the reader and prepulse. The program then reads a block and punches a block alternately until the erases are reached. It then punches the fault list and comes to a 99 stop. (See Failure (1)).

Operations (2) and (3) may then be repeated.

Output (1) Title.

(2) 40 rows blank tape.
Converted data.
40 rows blank tape.
6 rows erases.

The results are punched so that four numbers (that is, one block) are printed on one line. There are two extra line feeds and 20 rows of blank tape after every 32 numbers; that is, after 8 lines.

(3) 40 rows blank tape.
M
Fault list.
40 rows blank tape.
6 rows erases.
40 rows blank tape.

The faults are punched so that the block number, followed by the fault number, are printed on one line. Note that the first block number is zero.

Layout

(1) Results. The list ends with CR LF LF.
One line:
FS CR LF CR SP 3 digits SPSP SP 3 digits SPSP SP 3digits SPSP.

(2) Faults. The list ends with CR LF LF.
One line:
FS CR LF CR SP (Block no.) SPSP SP(Fault no.) SPSP.

Parameters None.

Restrictions (1) $0 \leq M \leq 999$.
(2) $0 \leq F \leq 255$.

Failures (1) If $F > 255$, the program refuses to read any more data, and punches out the fault list. If this happens it may be assumed that the data tape is too faulty to make further conversion worthwhile.

Time About $(n + F/2)$ seconds.

Method (1) Notation.

A block is a unit of data consisting of 13 rows of tape. Each of the first 12 rows represents a decimal digit, and the last row is known as a comma.

n = number of blocks to be converted.
F = number of faults.
R = number of digit rows in a block; normally $R = 12$.
M = a constant chosen so that a fault is indicated if $|\nabla N_j| \geq M$.

For a block with $R = 12$ we define

d_i' = a decimal digit represented by one row of tape,

d_i = the true decimal digit corresponding to d_i',

N_j' = a positive 3 digit integer in Watts reflected decimal code,

N_j = the positive 3 digit integer in true decimal corresponding to N_j',

where $i = 0(1)11$ and $j = 0(1)3$.

The d_i and the N_j are related as follows:-

$$N_0 = 100d_0 + 10d_1 + d_2$$
$$N_1 = 100d_3 + 10d_4 + d_5$$
$$N_2 = 100d_6 + 10d_7 + d_8$$
$$N_3 = 100d_9 + 10d_{10} + d_{11}$$

with similar expressions relating the d_i' and the N_j'.

(2) Conversion

We consider a block which contains no mistakes except possibly Fault (4). The program reads one block and converts the d_i', (which appear on the tape in the N.I.O. digitiser code), to pure binary, using a dictionary method. It then calculates the new digits d_i (still in binary) so that N_j' would become N_j. Each N_j' is treated separately.

The rule here is:-

If the true digit (d_i) on the immediate left of d_i' is even, then $d_i = d_i'$.

If the true digit (d_i) on the immediate left of d_i' is odd, then $d_i = 9 - d_i'$.

This conversion proceeds from left to right, so the "hundreds" digit is always unchanged. The N_j are calculated from the d_i by direct multiplication and then differenced and punched. The program then proceeds to read the next block.

(3) Faults.

The program can detect five distinct faults in the data tape, and when one of these is found the current block number and the fault number are added to the fault list. If there are no mistakes, the fault list only contains one entry, namely, fault (0) for block 0.

Fault (1) $0 \leq R \leq 11$.

Fault (2) Either $13 \leq R \leq 24$ or $R \geq 26$.
When either (1) or (2) is encountered the program replaces the R digits by 12 zeros.

Fault (3) If one or more digits belonging to N'_j are not punched in the N.I.O. digitiser code, then all 3 digits belonging to N_j are replaced by zeros.

Fault (4) If $|\nabla N_j| \geq M$, fault (4) is recorded for the block to which N_j belongs, but no other action is taken. Note that $\nabla N_j = N_j - N_{j-1}$. This fault is usually unavoidable for the first number in the first block. Also Fault (3) can produce Fault (4) twice.

Fault (5) $R = 25$. This means that a single comma has been mispunched. Conversion proceeds normally for the two blocks involved, and fault (5) is recorded for the block terminating with the mispunched comma.

Notes (1) The N.I.O. digitiser tape code is:-

Code	Meaning
100·00	0
000·01	1
100·11	2
000·10	3
101·10	4
011·10	5
110·10	6
010·11	7
110·01	8
010·00	9
011·11	Comma

A "1" represents a hole on the tape.

NIO 46 Fault nos.

1) as above
2) $19 \leq R \leq 23$, $26 \leq R$
3) as above
4) as above
5) $R = 25$
6) $R = 24$
7) $13 \leq R \leq 18$

N.I.O. Program 26

Title Allometric relationships of squid beaks.

Code Mercury autocode.

Purpose To determine the relation between pairs of series by fitting a logarithmic least squares straight line.

Order of tapes Program, parameter tape, data.

Parameter tape f (family number)
t

Data

p	(x-group label)
n	
$100x_i$	for $i = 0(1)n-1$
q	(y-group label)
m	
$100y_i$	for $i = 0(1)n-1$

End indication None.

Operation Read the program, parameter tape and data in that order, and leave until results are punched and more data is called for.

Output

s	j	p	q
a	b	σ	
x_0	y_0	y_0^+	y_0^-
x_1	y_1	y_1^+	y_1^-
x_2	y_2	y_2^+	y_2^-

Parameters None.

Restrictions (1) $1 \leqslant n,m \leqslant 225$.
(2) $m = n$ for any one pair of series.
(3) All x_i and y_i must have 2D.

Failures (1) $m \neq n$. The program comes to a 99 stop. A prepulse causes a new family number (f) to be read.

Time This is difficult to estimate, but one group of 1000 terms would take about 5 minutes.

Method The symbols used are

f = family number
t = number of runs in family f.
In the output $j = 1(1)t$.
p = x-group label.
q = y-group label.
n = number of terms in x-group.
m(=n) = number of terms in y-group.

The other output symbols will be described as they arise.

Given two sets of numbers x_i and y_i, $i = 0(1)n-1$, we require two constants a and b such that

$$v_i = au_i + b, \quad i = 0(1)n-1,$$

where $v_i = \log_{10} y_i$ and $u_i = \log_{10} x_i$.

The best straight line, which minimises the sum of squares of the deviations of v_i from the line, is given by

$$a = \frac{n \sum_{0}^{n-1} u_i v_i - \sum_{0}^{n-1} u_i \sum_{0}^{n-1} v_i}{n \sum_{0}^{n-1} u_i^2 - \left(\sum_{0}^{n-1} u_i\right)^2}$$

and $b = \frac{1}{n}\left[\sum_{0}^{n-1} v_i - a\sum_{0}^{n-1} u_i\right]$

We also require the standard deviation σ, defined by

$$\sigma^2 = \frac{1}{n}\sum_{0}^{n-1} R_i^2,$$

where the residual R_i is given by

$R_i = v_i - au_i - b.$

The results are to be plotted on log-log paper, and we require enough information for three lines; namely, the best fit, and a line either side to give some measure of the accuracy. Three points are calculated on each line, for arguments x_0, x_1 and x_2, as follows:-

y_0, y_1 and y_2 correspond to $v = au + b$, (the best fit)
y_0+, y_1+ and y_2+ correspond to $v = au + b + 2\sigma$,
and y_0-, y_1- and y_2- correspond to $v = au + b - 2\sigma$.

Notes

(1) After this program was in production it was found that σ is not the best estimate of the error.

N.I.O. Program 27

Title Salinity Tables.

Code Mercury Autocode.

Purpose To prepare a table of Salinity against Relative Conductivity for use with the N.I.O. Salinity Meter.

Order of tapes Program, data.

Parameter tape None.

Data n followed by b_i, i = 0(1)n (See Note (3)).

End indication None.

Operation Read the program, followed by the data.

Output A table of salinity S(3D) against relative conductivity L(4D) for

for L = 1.0000(.0001) 1.1299.

For this range S approximately satisfies

$$35 \leq S < 40.$$

The output needs some modification (see Note (2)).

Parameters None.

Restrictions (1) $0 \leq n \leq 20$.

Failures If any $|\Delta S| \geq .005$ a Fault List is punched after the table.

Time About 5 - 10 minutes.

Method The salinity is computed from the "least squares" formula

$$S = \sum_{i=0}^{n} b_i L^i \qquad \text{(See Note (3))};$$

Notes (1) This is a special purpose program, but the range could be altered by simple modifications to the program.

(2) Spaces are punched instead of non-significant zeros, and these spaces should be replaced by zeros if the table is meant for general publication.

(3) The coefficients b_i were obtained using the least squares method, with RAE 109/A. The data was:-

Relative Conductivity	Salinity
0.74670	25.311
0.87836	30.264
0.92281	32.000
0.94869	33.000
0.97441	34.000
0.99329	34.735
1.00000	35.000
1.02545	36.000
1.05082	37.000
1.07608	38.000
1.10140	39.000
1.12690	40.000

The final table is given in "Conversion tables; Bridge reading to salinity", prepared for use with the thermostat salinity meter (Design No. NIO/4777). It replaces the incorrect table previously given in the Handbook.

N.I.O. PROGRAM 28

Title Properties of Sea Water. I.

Code Mercury autocode.

Purpose Various properties of sea water are calculated from the sets of readings of pressure, temperature and salinity taken at a station. Some results are given at observed, and some at standard, pressures.

Order of tapes Program, parameter tape, data.

Parameter tape T = ambient temperature; that is, temperature reading of salinometer.

m ($m \leq 19$)

a_i listed for i = 0(1)m-1.

Do not end with an -> indication.

The standard values of pressure required in Calculation (3) (see Method) are defined by

$$P = a_0(a_1)a_2(a_3)a_4 \;\ldots\ldots\; a_{m-3}(a_{m-2})a_{m-1}$$

where m must be odd. Do not punch brackets.

T must be punched. There are 3 alternatives:-

(i) T = 15
(ii) T = 20
(iii) Any other value. For convenience punch T = 0.

Data The data for one station consists of:-

(1) A title. This may begin with CR, but may not contain another CR before the final CR LF. Any other symbols may be present, provided the number does not exceed one teleprinter line; that is, 68 printed symbols. The title must end with CR LF.
(2) n = the number of samples to follow.
(3) P_i $100t_i$ $1000S_i$, listed for i = 0(1)n-1.

These are integers. Do not end with an -> indication. A sample consists of a line of 3 readings; that is, readings for one value of i. It should be punched on a line by itself. The pressure (P_i) is in 1/10 kg/sq cm, the temperature (t_i,2D) in °C, and the salinity (S_i,3D) in ‰ as usual.

If t_i is missing, punch $100t_i$ = 9999.

If S_i is missing, punch $1000S_i$ = 99999.

For any one i, either or both of t_i and S_i may be missing.

P_i must be present. If P_0 is small (P_0 = 1, say) then punch P_0 = 0. See Note (3).

End indication None.

Operation

(1) Read the program tape in the usual way. When the program has been read it stops on an -> indication.
(2) Put the Single-Continuous key to Single.
(3) Put the parameter tape in the reader, return the Single-Continuous key to Continuous, and depress Key 9. When the last character has been read, the program comes to a halt.

(4) Put the data tape in the reader and depress Key 9. When one station has been dealt with the program comes to a halt.

Operation (4) may then be repeated, but never touch Key 9 until the new data tape is in the reader.

If a new parameter tape is required, then the program must either be restarted or read in again. See Note (4).

Output After the general program titles, the output for each station consists of:-

(1) Station title.
(2) Results at observed pressures.
(3) Results at standard pressures.

All the sections are clearly separated by blank tape, and the tape ends with 6 erases. The listing for a typical station is attached.
In section (2), if either or both temperature and salinity were missing in the data, an asterisk (*) is punched in place of each missing value. The pressure and any remaining data are punched for the relevant line, but no other results appear on that line.
The quantities punched are:-

Results (2)
- Sample number
- Pressure (decibars, integer)
- Depth (metres, integer)
- Corrected salinity (‰, 3D)
- Temperature (°C, 2D)
- Potential temperature (°C, 2D)
- Sigma t (3D)
- Specific volume (5D)
- Specific volume anomaly (1D)

Results (3)
- Pressure (decibars, integer)
- Dynamic height anomaly (dynamic metres, 3D)
- Potential energy anomaly (2D)
- Sound velocity (metres/sec, 1D)
- Sounding velocity (metres/sec, 1D)

All values (except pressure) are followed by the interpolation error.

Unspecified units are irrelevant.

Parameters None.

Restrictions (1) $1 \leqslant n \leqslant 50$.
(2) $m \leqslant 19$, and m is odd.

Failures None.

Time About $(5 \cdot 5m + 2N)$ seconds, where N is the number of standard pressures.

Method Introduction

The complete calculation for one station will be described. A station consists of n sets of data, where one set of data (or one sample) consists of three numbers - pressure, temperature and salinity. Before any further calculation is attempted, the pressure is converted to decibars (if not already in these units) and the salinity is corrected if necessary. See Note (5).
Suppose the corrected and original salinities are S and s respectively, and the ambient temperature is T. If

$$s \leqslant 35, \text{ then } S = s \text{ for any } T,$$

and if

$$\begin{aligned} s > 35, \text{ then } S &= s + c_1(s - 35) \text{ if } T = 15°C, \\ \text{and } S &= s + c_2(s - 35) \text{ if } T = 20°C, \\ \text{and } S &= s \quad \text{ if } T \neq 15°C \text{ and } T \neq 20°C. \end{aligned}$$

Missing data

If the pressure is missing, the whole sample must be omitted. It either or both of the temperature and salinity are missing, but the pressure is present, the sample is included in the station, but it is not used in any calculation except possibly a salinity correction.

The program

This is in 4 chapters, 3 for computation and one mainly for organisation (Chapter 0). The calculations fall into three distinct sectionsas follows:-

(1) Quantities depending on only one sample (chapter 1).
(2) Quantities depending on several adjacent samples (chapter 2).
(3) Interpolation to standard pressures (chapter 3).

The ith sample consists of 3 values, pressure (P_i), temperature (t_i) and salinity (S_i), where i = 0(1)n-1, and the corrected values, not the original readings, are considered in the following description. Other symbols will be defined as they arise.

Calculation (1)

For a sample i, there are 5 quantities which depend on P_i, t_i and S_i only. They are

σ_t = sigma t

α = specific volume
δ = specific volume anomaly
θ = potential temperature
V = sound velocity

and all are defined as polynomials in P, t and S. In general, the polynomial coefficients b_j, and the other constants c_j, have been obtained by the method of Least Squares.

We first define

$$\sigma_0 = \sum_0^3 b_r S^r \tag{1}$$

and then

$$\sigma_t = \frac{\sum_{r=0}^{3} b_{r+4} t^r + \sum_{r=0}^{3} b_{r+4} t^r + \sigma_0 \sum_{r=0}^{3} b_{r+8} t^r + \sigma_0^2 t \sum_{r=0}^{3} b_{r+12} t^r}{c_3 + t} . \tag{2}$$

To define $\alpha \equiv \alpha(S,t,P)$ and $\delta \equiv \delta(S,t,P)$, we require

$$\alpha(S,t,o) = (1 + c_4 \sigma_t)^{-1} \tag{3}$$

$$\text{and } \alpha(35,0,P) = \frac{\sum_{r=0}^{3} b_{r+34} P^r}{c_8 + c_9 P} \tag{4}$$

whence

$$\frac{\alpha(S,t,P)}{\alpha(S,t,0)} = b_{15} + P\sum_{r=0}^{3} b_{r+16} t^r + P\sigma_0 \sum_{r=0}^{2} b_{r+20} t^r$$

$$+ P\sigma_0^2 \sum_{r=0}^{1} b_{r+23} t^r + P^2 \sum_{r=0}^{2} b_{r+25} t^r + P^2 \sigma_0 \sum_{r=0}^{2} b_{r+28} t^r$$

$$+ P^2\sigma_o^2 \sum_{r=o}^{1} b_{r+31} t^r + b_{33} P^3 t - \frac{c_5 P}{c_6 + c_7 P} \qquad (5)$$

and $\delta(S,t,P) = \alpha(S,t,P) - \alpha(35,0,P)$ (6)

All the constants in equations (1) to (5) come from [1].

A new formula for the potential temperature has been derived, by fitting a least squares polynomial to the results from N.I.O. Program 32. The various steps of the calculation can be followed from the Method descriptions of N.I.O. Programs 31, 34 and 32, in that order. The least squares program used was RAE 195/A. (See also Note (1)).

The polynomial is

$$\theta - 10 = \sum_{\ell=o}^{4} \sum_{m=o}^{4} \sum_{n=o}^{4} b_{16\ell+4m+n+59} (P-3000)^{\ell} (t-10)^m (S-35)^n \qquad (7)$$

where $\ell+m+n \leq 4$.

Finally, the sound velocity is defined by

$$V = b_{38} + \Delta V_t + \Delta V_p + \Delta V_s + \Delta V_{stp} \qquad (8)$$

where

$$\Delta V_t = t \sum_{r=o}^{3} b_{r+39} t^r, \qquad (9)$$

$$\Delta V_p = P \sum_{r=o}^{3} b_{r+43} P^r, \qquad (10)$$

$$\Delta V_s = b_{47}(S-35) + b_{48}(S-35)^2 \qquad (11)$$

and $\Delta V_{stp} = (S-35)(b_{49} t + b_{50} P + b_{51} P^2 + b_{52} Pt)$

$$+ Pt \sum_{r=o}^{2} b_{r+53} t^r + P^2 t \sum_{r=o}^{1} b_{56+r} t + b_{58} P^3 t \qquad (12)$$

The coefficients used in equations (8) to (12) come from [2] The sound velocity is the only case in which the pressure is in kg/sq cm and not in decibars.

Calculation (2)

There are 4 quantities obtained by integration, and these depend on several values of i; that is, on several samples. They are

D = depth
ΔD = dynamic height anomaly
χ = potential energy anomaly
V̄ = sounding velocity

The general formal definitions are

$$D_i = C_{10}\int_{P_o}^{P_i} \alpha(S,t,p)dp \tag{13}$$

$$\Delta D_i = C_{11}\int_{P_o}^{P_i} \delta(S,t,p)dp \tag{14}$$

$$\chi_i = C_{10}\int_{P_o}^{P_i} p\delta(S,t,p)dp \tag{15}$$

$$\bar{V}_i = \frac{1}{P_i}\int_{o}^{P_i} Vdp \tag{16}$$

The constants ensure that P may be given in decibars.

Since P is given at unequal, but fairly close, intervals, integration by the trapezium rule is convenient and adequate. The formulae used in the calculation follow, and in all cases we define for convenience

$$2h_i = P_i - P_{i-1},$$

$$\alpha_i = \alpha_i(S,t,P),$$

$$\text{and } \delta_i = \delta_i(S,t,P). \tag{17}$$

Thus we have

$$D_i = D_{i-1} + C_{10}h_i(\alpha_i + \alpha_{i-1}) \text{ for } i \geq 1, \tag{18}$$

$$\text{and } D_o = C_9(C_{10}P_o) + C_8(C_{10}P_o)^2. \tag{19}$$

Similarly

$$\Delta D_i = \Delta D_{i-1} + C_{11}h_i(\delta_i + \delta_{i-1}) \text{ for } i \geq 1 \tag{20}$$

and $\Delta D_o = 0$. The potential energy anomaly becomes

$$\chi_i = \chi_{i-1} + C_{10}h_i(P_i\delta_i + P_{i-1}\delta_{i-1}) \text{ for } i \geq 1 \tag{21}$$

and $\chi_o = 0$.

Finally we have

$$P_i\bar{V}_i = P_{i-1}\bar{V}_{i-1} + h_i(V_i + V_{i-1}) \tag{22}$$

and $\bar{V}_o = V_o$

When Calculations (1) and (2) are complete, the first table of results is punched. No interpolation has yet been attempted.

Calculation (3)

Some of the quantities calculated in Sections (1) and (2) are required at standard pressures, and these are

ΔD = Dynamic height anomaly
χ = Potential energy anomaly
V = Sound velocity
$\bar{V}$ = Sounding velocity

The argument is the pressure P.

The interpolation for each variable is carried out in an exactly similar way, so only one variable, y say, will be considered.

We start with n values of P_i, $i = 0(1)n-1$, and there is a y_i corresponding to each P_i. Then y is required for certain specified values of P denoted by P_j, $j = 0(1)M-1$. Generally P_j is specified at equal intervals, but the interval may vary over the range of P. (See Parameter tape). We now consider one value P for which $y \equiv y(P)$ is required.

There are 5 possible cases.

(1) $P < P_0$
(2) $P_0 \leq P < P_1$
(3) $P_i \leq P < P_{i+1}$ for $i = 1(1)n-3$
(4) $P_{n-2} \leq P \leq P_{n-1}$
(5) $P > P_{n-1}$

In cases (1) and (5) interpolation is impossible. In the other cases two values (Y_1 and Y_2) of y are computed, using different formulae or slightly different arguments, and y is taken as the mean of the results. The difference e gives an indication of the accuracy. Thus we calculate

$$y = \tfrac{1}{2}(Y_1 + Y_2)$$
$$\text{and } e = \tfrac{1}{2}(Y_1 - Y_2).$$

In general the Lagrange 3 point interpolation formula is used. If the arguments are P_{r-1}, P_r and P_{r+1}, then

$$Y_1 \text{ (or } Y_2) = A_{r-1}y_{r-1} + A_r y_r + A_{r+1}y_{r+1},$$

$$\text{where } A_{r-1} = \frac{(P - P_r)(P - P_{r+1})}{(P_{r-1} - P_r)(P_{r-1} - P_{r+1})}$$

$$A_r = \frac{(P - P_{r-1})(P - P_{r+1})}{(P_r - P_{r-1})(P_r - P_{r+1})}$$

$$\text{and } A_{r+1} = \frac{(P - P_{r-1})(P - P_r)}{(P_{r+1} - P_{r-1})(P_{r+1} - P_r)}$$

Near the end of the range linear interpolation is necessary.

We can now list the formulae used in each case.

Case (1) Interpolation is impossible.

Case (2) $Y_1 = y_0 + \dfrac{y_1 - y_0}{P_1 - P_0}(P - P_0)$

$$Y_2 = A_0 y_0 + A_1 y_1 + A_2 y_2$$

Case (3) $Y_1 = A_{i-1}y_{i-1} + A_i y_i + A_{i+1}y_{i+1}$

$$Y_2 = A_i y_i + A_{i+1}y_{i+1} + A_{i+2}y_{i+2}$$

Case (4) $Y_1 = A_{n-3}y_{n-3} + A_{n-2}y_{n-2} + A_{n-1}y_{n-1}$

$$Y_2 = y_{n-2} + \frac{y_{n-1} - y_{n-2}}{P_{n-1} - P_{n-2}} (P - P_{n-2})$$

Case (5) Interpolation is impossible.

During interpolation each line is punched as soon as the calculation is finished. Any specified pressures which fall outside the range of data given are ignored completely.

Note (1) The least squares trivariate polynomial for $\theta(P,S,t)$ was obtained in 6 stages.

(i) $\Gamma(P,S,t)$ was calculated for specified values of P, S and t. (N.I.O. Program 31).

(ii) Results from (i) were scaled and rearranged. (N.I.O. Program 34).

(iii) A least squares polynomial was fitted to $\Gamma(P,S,t)$. (R.A.E. 195/A).

(iv) $\theta(P,S,t)$ was obtained from an integral equation, using the polynomial to calculate $\Gamma(P,S,t)$. (N.I.O. Program 32).

(v) Results from (iv) were scaled and rearranged. (N.I.O. Program 34).

(vi) A least squares polynomial was fitted to $\theta(P,S,t)$. (R.A.E. 195/A).

(2) The polynomial coefficients and other constants are listed at the end of the program tape.

(3) If several stations are punched on one tape, there must be at least five rows of blank tape - and nothing except blank tape - between the final SpSp or CRLF of one station and the first CR of the next title. The final number of a station must be followed by two characters only.

(4) If the program is restarted (Keys 1 and 9) the last two blocks on the program tape must be reread; that is, the constants.

(5) If the pressure is not given in $^1/_{10}$ Kg/sq cm, see N.I.O. Program 36.

References

[1] Fisheries Research Board of Canada. M.R.27. By N.P. Fofonoff and C. Froese.

[2] W.D. Wilson. Speed of sound in sea water as a function of salinity, temperature and pressure. J. Acoustical Soc. America, 32, 6, p.641 (1960).

N.I.O. Program 29

Title Wave Prediction.

Code Mercury autocode.

Purpose To predict the r.m.s. height, dominant direction, and dominant period of waves in the North Atlantic. The wind speeds and directions are given at the time for which the prediction is required, and also 12, 24, 36 and 48 hours previously.

Order of tapes Program, parameter tape, data.

Parameter tape m n

m = number of rows; $i = 0(1)m-1$
n = number of columns; $j = 0(1)n-1$.

Data There are 5 groups of data, one for each time. A group is headed by the time u_{ℓ}. This is followed by lines

$i \quad j \quad w_{ij} \quad v_{ij}$

for as many i, j as have a non-zero w_{ij} (for that u_{ℓ}). The whole group ends with -1. The full possible range is $i = 0(1)m-1$, $j = 0(1)n-1$, but in practice many entries will be zero. There must be at least one entry for $i = m-1$, even if this means punching $w_{ij} = 0$.

The rows must be punched for decreasing i, which means reading down the sheet in the usual way (see Method). The values for one row (one i) must be consecutive, and within the row the values of j must be punched with increasing magnitude. The 5 values of u_{ℓ} are

0000, 1200, 2400, 3600, 4800

and the groups must be punched in this order. Always punch 4 digits for u_{ℓ}. Note that the prediction is required at $u_{\ell} = 4800$.

w_{ij} is the wind speed in knots to the nearest integer, and v_{ij} is the wind direction in degrees to the nearest 45°, and $0 \leq v_{ij} \leq 315°$. The usual wind convention is followed; that is, winds from the North, East, South and West have directions 0, 90°, 180° and 270° respectively.

End indication -1 is punched immediately after the last v_{ij} in a group.

Both the parameter and data tapes end with --> CRLF.

Operation Read the program, followed by the parameter tape and data tape. All the data is read before the calculation starts. When the results have been punched, more data is called for. A new parameter tape, followed by new data, may then be read.

Output After the titles, m and n are punched. The results appear in 5 columns, headed Row, Column, Height, Direction and Period. One line corresponds to a square, and the squares are given in the same order as the data. The 3 predicted numbers are:-

Height (r.m.s.)(feet, 1D)
Dominant direction (to the nearest 45°; current convention).
Dominant period (to the nearest 3 seconds).

The output ends with blank tape and erases.

Parameters d, the length of square side. Here d = 200, and no other values have been tried.

Restrictions (1) $mn \leq 180$
(2) $n \leq 16$.

Failures None.

Time Some test case times are:

m	n	time (mins)
3	5	5
3	9	11
3	12	15
9	12	60+

Method (1) Introduction

The North Atlantic is divided into squares, and the average wind speed and direction in each square are given at times 0000 hrs, 1200 hrs, 2400 hrs, 3600 hrs and 4800 hrs. The program predicts the r.m.s. height, dominant direction and dominant period of waves resulting from the wind data. Results are given for each square at 4800 hrs. All constants are listed in the program.

(2) Notation

- d = side of square (miles)
- m = number of rows
- n = number of columns
- (i,j) = the co-ordinates of a square, or, where more convenient, the centre of the square. The range is i = 0(1)m-1, j = 0(1)n-1, and the numbering starts at the SW corner, so that i increases from S to N and j increases from W to E.
- a = period (seconds, to the nearest 3 seconds) a = 3(3)18
- f = frequency ($f = a^{-1}$)
- w_{ij} = wind speed in (i,j), (knots) (probably ⅔ geostrophic)
- v_{ij} = wind direction in (i,j), (nearest 45°) (See Data) The program replaces v by v ± 180, where the sign is chosen to make 0 ≤ v ± 180 < 360, and works with the new value throughout. This new value is called v_{ij} in the following description.
- u_ℓ = time. Here u_0 = 0000, u_1 = 1200, ... u_4 = 4800
- h_{ij} = r.m.s. wave height in (i,j)
- b_{ij} = dominant direction of waves in (i,j)
- z_{ij} = dominant period of waves in (i,j)
- g_{ij} = initial energy in (i,j)
- $e_{ij}^{(s)}$ = energy in (i,j) due to the influence of the wind in (i,j) and other squares. There is one value in each direction, s = 0(1)7, where s = 0 implies N and so on. There is one $e_{ij}^{(s)}$ for each period.
- x_{ij} = energy total over all directions in (i,j) Thus

$$x_{ij} = \sum_{s=0}^{7} e_{ij}^{(s)} \qquad (1)$$

- $y_{ij}^{(s)}$ = total energy in one direction in (i,j), taken over all periods. Thus

$$y_{ij}^{(s)} = \sum_{a=3}^{18} e_{ij}^{(s)} \qquad (2)$$

where a = 3(3)18.

Where no confusion is likely to result, we write y_s and e_s instead of $y_{ij}^{(s)}$ and $e_{ij}^{(s)}$.
In the description of energy calculation all angles are made to satisfy

$$-180° \leqslant \text{angle} < 180°, \qquad (3)$$

unless otherwise stated. The constants include conversion to radians where necessary.

(3) Initial Energy

We define
$$g_{ij} = 3f^2 c_0 w_{ij}^4 \exp - \left[\frac{(f - f_0)^2}{c_1(f - f_0 + c_2)}\right]^{\frac{1}{2}} \qquad (4)$$

where $f_0 = c_3 w_{ij}^{-\frac{1}{2}}$, (5)

and $g_{ij} = 0$ if $c_1(f - f_0 + c_2) < 0$. (6)

(4) Type of energy spread

There is an angle ψ, depending on wind speed and period only, which influences the amount of energy to travel in each direction. It is defined by

$$\psi = \cos^{-1}(3a/w_{ij}) \quad \text{if } w_{ij} \geqslant 3a \qquad (7)$$

$$\text{and } \psi = 0 \qquad \text{if } w_{ij} < 3a. \qquad (8)$$

(5) General time loop

For a fixed period (a), there is a separate calculation for each u_{ℓ}. The general case deals with u_{ℓ} = 0000, 1200, 2400 and 3600 hours, and we define

$$u = 4800 - u_{\ell} > 0. \qquad (9)$$

Consider a square (p,q). We can calculate the magnitude and direction of the energy (if any) contributed to (i,j) by (p,q).

Let R = distance of (p,q) from (i,j) (centres).
and θ = bearing of (i,j) from (p,q); that is, the direction of travel from (p,q) to (i,j).

Then $R = d[(p - i)^2 + (q - j)^2]^{\frac{1}{2}}$ (10)

and $\theta = \arctan\left(\frac{j - q}{i - p}\right)$, $0 \leqslant \theta < 360°$. (11)

The choice of numbering squares (see definition of (i,j)) ensures that θ lies in the correct quadrant.

Let d_2 = distance moved by energy of period a in time u.

Then $d_2 = auc_6/100$ (12)

There is a contribution from (p,q) if two conditions are satisfied; namely

$$g_{pq} > 0 \qquad (13)$$

and $d_2 - \frac{1}{2}d \leqslant R < d_2 + \frac{1}{2}d$, (14)

unless $d_2 - \frac{1}{2}d < 0$, in which case we replace (14) by

.3

$$1 \leqslant R < d_2 + \tfrac{1}{2}d \tag{15}$$

The magnitude of the energy is

$$e = \int_\theta E(\theta)d\theta, \tag{16}$$

where $E(\theta) = gc_5 \left[\exp \left\{ -c_4(\theta - v + \psi)^2 \right\} \right.$

$$\left. + \exp \left\{ -c_4(\theta - v - \psi)^2 \right\} \right], \tag{17}$$

and where $c_5 = 1/2\Delta\sqrt{(2\pi)}$, (18)

$c_4 = 1/2\Delta^2$, (19)

and $\Delta = \pi/6$, (20)

assuming all angles to be radians. The values g, v and ψ refer to (p,q).

$E(\theta)$ is integrated over the angle subtended by square (p,q) at the centre of (i,j). When R >> d, a sufficiently good approximation is

$$e = \frac{d}{R} E(\theta) \tag{21}$$

When R = d, e may be evaluated by means of (21), and then replaced by $c_{10}e$. This constant c_{10} is estimated graphically.

The direction V is calculated to the nearest 45°, by

$V = v + \phi$ for $\phi - 22\tfrac{1}{2}° \leqslant \theta - v < \phi + 22\tfrac{1}{2}°$,

where $\phi = 0, \pm 45°, \pm 90°, \pm 135°$, (22)

and $V = v - 180$ for $-180° \leqslant \theta - v < -157\tfrac{1}{2}°$ (23)

and $V = v + 180$ for $157\tfrac{1}{2}° \leqslant \theta - v < 180°$. (24)

Taking (i,j) as fixed for the moment, the above process is repeated for p = 0(1)m-1 and q = 0(1)n-1. For one time u_ℓ, the total energy contributed to (i,j) is added according to direction. Thus for each u_ℓ, each (i,j) has 8 entries e_s, s = 0(1)7.

(6) Final time loop

For the same fixed period (a) as above, we still have to deal with u_ℓ = 4800 hours. This gives u = 0. Clearly (p,q) and (i,j) coincide, so R = 0 and formula (21) is not valid. Instead we must integrate (17) about the selected directions

$$V = 0, \pm 45°, \pm 90°, \pm 135°, -180° \tag{25}$$

This involves the error function, difined by

$$\mathrm{Erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x \exp(-t^2)\,dt, \tag{26}$$

and which may be evaluated using a well-known Chebyshev polynomial.
Consider the first exponential in (17).

Let $\Phi = c_8(\theta - v + \psi)$ (27)

where $c_8 = 1/\Delta\sqrt{2} = \sqrt{c_4}$. (28)

Note that c_8 in the program includes conversion to radians.

Then $d\phi = c_8 d\theta$ (29)

and $$I_1 = \int_\theta \exp\left\{-c_4(\theta - v + \psi)^2\right\} d\theta \quad (30)$$

$$= \frac{1}{c_8}\int_\theta \exp(-\phi^2)\, d\phi \quad (31)$$

$$= \frac{\sqrt{\pi}}{2c_8}\left[\mathrm{Erf}(B) - \mathrm{Erf}(A)\right] \quad (32)$$

where $\left.\begin{matrix}A\\B\end{matrix}\right\} = c_8(V - v + \psi \pm 22\tfrac{1}{2}°)$ (33)

Similarly

$$I_2 = \frac{\sqrt{\pi}}{2c_8}\left[\mathrm{Erf}(D) - \mathrm{Erf}(C)\right] \quad (34)$$

where $\left.\begin{matrix}C\\D\end{matrix}\right\} = c_8(V - v - \psi \pm 22\tfrac{1}{2}°)$. (35)

Thus, from (16), (17), (32) and (34),

$$e = gc_7\left[\mathrm{Erf}(B) - \mathrm{Erf}(A) + \mathrm{Erf}(D) - \mathrm{Erf}(C)\right] \quad (36)$$

where $c_7 = \frac{c_5\sqrt{\pi}}{2c_8} = \frac{1}{4}$ (37)

This process is repeated for all (i,j), giving 8 entries e_s in each square as in the general case.

(7) Frequency loop

For one period or frequency each square has 8 entries e_s for each of the 5 times u_ℓ. Thus a specified direction s has 5 values of e_s, which are not added up, but the maximum value is selected and the others are discarded. This chosen value is known as e_s for the period a.

We now have 8 entries per square for each period, and various combinations of these values give the final results.

(8) Totals

These refer to one square (i,j) only.

Let E = grand total of energies in all directions and over all periods.

Then $E \sum_{a=3}^{18} x_{ij} = \sum_{s=0}^{7} y_{ij}^{(s)}$, where a = 3(3)18, (38)

and the root mean square wave height is given by

$$h_{ij} = c_9\sqrt{E}. \quad (39)$$

The dominant direction b_{ij} is the direction for which $y_{ij}^{(s)}$ is largest. Thus

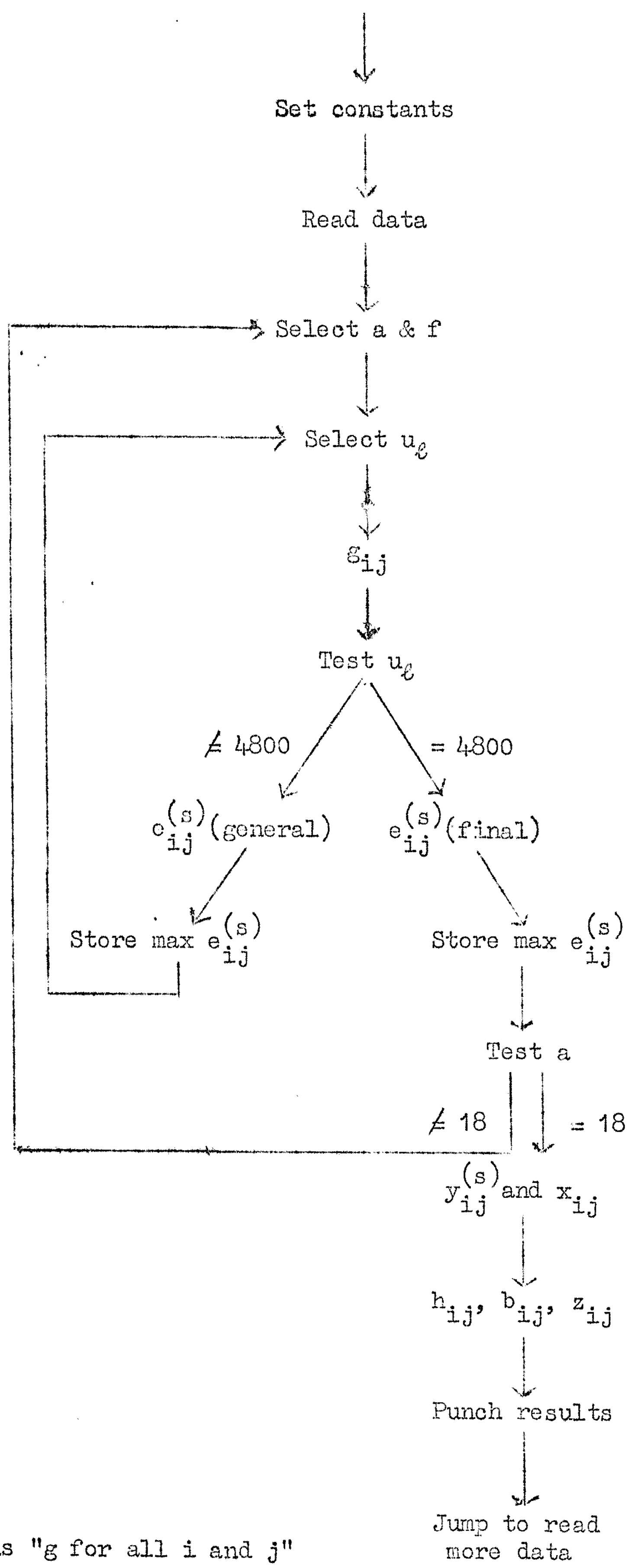
$b_{ij} = 45s$, for the relevant s. (40)

The dominant period z_{ij} is the period for which x_{ij} is largest.

(10) The Program

A block diagram follows. Many of the results described above were obtained by short cut methods. For example, the maximum e_s (see Section (7)) was selected during the time loops, not at the end, and as soon as a new value was computed it was added into the sum.

Block Diagram



Notes

g_{ij} means "g for all i and j"

N.I.O. Program 30

Title Radiant heating of a translucent solid.

Code Mercury Autocode.

Purpose To solve the partial differential equation for various parameters.

Order of tapes Program, data.

Parameter tape None.

Data

p

x_i , listed for i = 0(1)p-1.

q

t_j , listed for j = 0(1)q-1.

r

n_k , listed for k = 0(1)r-1.

End indication → CRLF

Operation Read the program, followed by the data. When one set of results has been punched, more data may be read.

Output $\theta(x_i, t_j, n_k)$ is punched in floating point (8D) and fixed point (8D) for i = 0(1)p-1, j = 0(1)q-1, k = 0(1)r-1.

Parameters None.

Restrictions $1 \leqslant p,n,n \leqslant 51$.

Failures None.

Time About pmn/40 minutes.

Method The equation is

$$\frac{\partial\theta}{\partial t} = \frac{\partial^2\theta}{\partial z^2} + e^{-z}$$

with boundary conditions

$$\theta = 0 \text{ at } t = 0$$

$$\frac{\partial\theta}{\partial z} \to 0 \text{ as } z \to \infty$$

$$\frac{\partial\theta}{\partial z} = n\theta \text{ at } z = 0.$$

The theoretical solution has 3 cases.

(1) Any n except n = 0 and n = 1

$$\theta = -\tfrac{1}{2}F(-1) + \frac{n+1}{n}F(0) - \frac{n+1}{2(n-1)}F(1) + \frac{1}{n(n-1)}F(n) + f_2.$$

(2) n = 0

$$\theta = -\tfrac{1}{2}F(-1) - zF(0) + \tfrac{1}{2}F(1) + f_3 + f_2.$$

(3) n = 1

$$\theta = -\tfrac{1}{2}F(-1) + 2F(0) + f_1F(1) - f_3 + f_2.$$

We define

$$f_1 = 2t + z - \frac{3}{2},$$

$$f_2 = e^{-z}(e^y - 1),$$

$$f_3 = \frac{2\sqrt{t}}{\sqrt{\pi}} \exp(-x^2),$$

and $F(n) = e^{\beta}\mathrm{Erfc}(\alpha)$,

where

$$x = \frac{z}{2\sqrt{t}},$$

$$\beta = nz + n^2 y,$$

$$\alpha = x + n\sqrt{t},$$

and $\mathrm{Erfc}(\alpha) = \frac{2}{\sqrt{\pi}}\int_{\alpha}^{\infty} \exp(-u^2)\, du.$

The calculation of θ is straightforward, except that cancellation takes place and the full 8 decimal places provided by Mercury autocode must be used, to ensure that 3 significant figures are valid in θ, at least for large t.

F(n) presents the only problem. We may write

$$F(n) = G(\alpha) \exp(-x^2)$$

where $G(\alpha) = \mathrm{Erfc}(\alpha) \exp(\alpha^2)$,

since α, β and x are connected as above.

The calculation of $G(\alpha)$ falls into 3 parts, and the method chosen depends on a constant c. In practice c = 6, as this gives8 significant figures throughout the range.

(1) $\alpha > c$. Here the asymptotic formula for $\mathrm{Erfc}(\alpha)$ is valid, and so

$$G(\alpha) = \frac{1}{\alpha\sqrt{\pi}} \sum_{r=0}^{\infty} a_r (2\alpha^2)^{-r}$$

where $a_r = (-1)^r\, 1.3.5.\ldots\ldots(2r - 1),\ r \geq 1$

and $a_0 = 1$.

(2) $0 \leq \alpha \leq c$.

Since $G(\alpha) = \frac{2}{\sqrt{\pi}} \exp(\alpha^2)\int_{\alpha}^{\infty} \exp(-u^2)\, du,$

it also satisfies

$$\frac{dG}{d\alpha} = 2\alpha G - \frac{2}{\sqrt{\pi}}$$

Differentiating r times and putting

$$\tau_r = \frac{h^r}{r!} \cdot \frac{d^{(r)}G}{d\alpha^r}$$

gives $\tau_r = \frac{2\alpha h}{r}\, \tau_{r-1} + \frac{2h^2}{r}\, \tau_{r-2}$

with $\tau_1 = 2\alpha\tau_0 - \frac{2h}{\sqrt{\pi}}$

and $\tau_0 = 1$.

The recurrence relation converges reasonably quickly for $|h| \leqslant 0 \cdot 5$.
A table of $G(\alpha)$ for $\alpha = 0, 1, 2, \ldots$ c was put into the program, so that any $G(\alpha)$ could be obtained by Taylor Series interpolation, using

$$G(\alpha + h) = \tau_0 + \tau_1 + \tau_2 + \ldots$$

where $-0 \cdot 5 \leqslant h < 0 \cdot 5$.

(3) $\alpha < 0$.

Here $G(\alpha) = 2 \exp(\alpha^2) - G(|\alpha|)$,

where $G(|\alpha|)$ is obtained as in (1) or (2).

Notes None.

N.I.O. Program 31

Title Adiabatic lapse-rate.

Code Mercury autocode.

Purpose To calculate the adiabatic lapse-rate $\Gamma(P,S,t)$ for specified values of pressure (P), temperature (t) and salinity (S).

Order of tapes Program, data. (See Note (1)).

Parameter tape None.

Data This is divided into n groups, each group corresponding to one value of P_i, $i = 0(1)n-1$.

A group is punched as follows:-

P_i (decibars).

q, followed by a list of t_j(°C), $j = 0(1)q-1$.

r, followed by a list of S_k(‰), $k = 0(1)r-1$.

Thus the whole tape contains:-

n, followed by the n groups.

All values punched must be integers.

End indication --> CRLF.

Operation Read the program, followed by the data. The results for one group are punched before the next group is read.

Output For one group, qr lines are punched, each line containing

P t S $\Gamma(P,S,t)$

in that order. Note that in the title Γ is replaced by G. In all cases P, t and S are integers and $\Gamma(P,S,t)$ is in floating point form to 6D.

Parameters None.

Restrictions $1 \leqslant n,\ q,\ r \leqslant 100$ (See Note (1)).

Failures None.

Time About (nqr)/32 minutes.

Method We assume that the program has selected values of pressure (P), temperature (t), and salinity (S), and the calculation of $\Gamma(P,S.t)$ for this typical set of values will be described. The formula for $\Gamma(P,S,t)$ is

$$\Gamma(P,S,t) = \frac{\bar{T}}{100JC_p(S,t,P)} \cdot \frac{\partial\alpha(S,t,P)}{\partial t} \qquad (1)$$

where $\bar{T} = t + 273$. (2)

The definition of $\alpha(S,t,P)$ is exactly as given in N.I.O. Program 23 and so the constants b_i, $i = 0(1)33$ and c_i, $i = 0(1)7$ from that program are also required here. The derivative is given to sufficient accuracy by

$$\frac{\partial\alpha(S.t,P)}{\partial t} = \alpha(S,t+\tfrac{1}{2},P) - \alpha(S,t-\tfrac{1}{2},P) \qquad (3)$$

Now $$JC_p(S,t,P) = JC_p(S,t,0) - 10^{-6}\bar{T}\,\frac{1}{10}\int_0^P \frac{\partial^2\alpha}{\partial t^2}\,dP \qquad (4)$$

N.3

The pressure integral is given by a least squares polynomial, namely

$$\frac{1}{10}\int_0^P \frac{\partial^2\alpha}{\partial t^2}\, dP = P\sum_{r=0}^{3} a_{r+8}t^r + a_{12}PS^2 + PS(a_{13} + a_{14}t)$$

$$+ P^2(a_{15} + a_{16}t) + a_{17}P^2S + P^3(a_{18} + a_{19}t) \qquad (5)$$

The coefficients come from [1].

Also

$$JC_P(S,t,0) = a_4 + a_3(t + a_1S + a_2S^2 + a_0)^2 + a_5S + a_6S^2 \qquad (6)$$

where this formula comes from [2].

A list of the a_i, $i = 0(1)19$, is in the program.

Notes

(1) This is a special purpose program, and the program and data are all on one tape. The results are intended for use with RAE 195/A, which fits a least squares polynomial to $\Gamma(P,S,t)$. This imposes the additional restriction

$$nqr \leqslant 200.$$

(2) The constants used in the calculation of $\Gamma(P,S,t)$ are listed at the end of the program.

References

[1] Fisheries Research Board of Canada. M.R.27. N.P. Fofonoff and C. Froese.

[2] R.A. Cox and N.D. Smith. Specific heat of sea water. P.R.S.A. 252, p.51 (1959).

N.I.O. Program 32

Title Potential Temperature.

Code Mercury autocode.

Purpose To calculate the potential temperature $\theta(P,S,t)$ by solving an integral equation, for specified values of pressure (P), temperature (t) and salinity(S).

Order of tapes Program, parameter tape, data (see Note (1)).

Parameter tape ΔP, the interval of integration.

Data This is divided into n groups, each group corresponding to one value of P_i, $i = 0(1)n-1$.

A group is punched as follows:-

P_i (decibars)

q, followed by a list of t_j(°C), $j = 0(1)q-1$.

r, followed by a list of S_k(‰), $k = 0(1)r-1$.

Thus the whole tape contains:-

n, followed by the n groups.

All values punched must be integers.

End indication ---> CRLF

Operation Read the program followed by the parameter tape and data. The results for one group are punched before the next group is read.

Output For one group, qr lines are punched, each line containing

P t S $\theta(P,S,t)$

in that order. Note that in the title θ is replaced by H. In all cases P, t and S are integers and $\theta(P,S,t)$ is in floating point form to 6D.

Parameters None.

Restrictions (1) $1 \leq n,q,r \leq 50$.
(2) $k \leq 199$, where $P = k.\Delta P$.
k must be an integer.

Failures None.

Time About k/300 minutes, where k is the number of steps.

Method The integral equation to be solved is

$$\theta = t + \int_P^0 \Gamma(\theta,S,p)\,dp, \qquad (1)$$

where θ is the required potential temperature, and the pressure (P), temperature (t) and salinity (S) are known. $\Gamma(\theta,S,P)$ is given by a polynomial in θ,S and P. (See Note (1)).

Suppose the range of integration is divided into a number of pressures P_i, $i = 0(1)n-1$,

where $P_0 = 0$

$P_{n-1} = P$

and $\Delta P = -P_i + P_{i-1}$ and is a positive constant.

Assume that θ_0, θ_1, ... θ_{i-1}, corresponding to P_0, P_1,..... P_{i-1}, are known.
Let ϕ_i be the first estimate of θ_i.
Then $\phi_i = \theta_{i-1} - \Gamma(\theta_{i-1}, S, P_{i-1})\,\Delta P$
and $\theta_i = \theta_{i-1} - [\Gamma(\theta_{i-1}, S, P_{i-1}) + \Gamma(\phi_i, S, P_i)]\,\frac{\Delta P}{2}$
This process is repeated for $i = 1(1)m-1$, starting with $\theta_0 = t$, until finally

$$\theta = \theta_{m-1}.$$

The accuracy can be controlled to a certain extent by suitable choice of ΔP.
The above method is based on that given in [1].

Notes

(1) For the calculation of the polynomial coefficients for $\Gamma(P,S,t)$, see N.I.O. Programs 31 and 34, RAE 195/A and Note (1) of N.I.O. Program 28.

(2) This is a special purpose program, and the program, parameter tape and data are all on one tape.

References

[1] Fisheries Research Board of Canada, M.R.27. By N.P. Fofonoff and C. Froese.

N.I.O. PROGRAM 33

Title Tide prediction matrices.

Code Mercury autocode.

Purpose To calculate the basic matrices to be used in tide elimination and prediction. They are required by N.I.O. Program 13.

Order of tapes Program, data.

Parameter tapes None.

Data There are 9 groups of data, each punched in the following way.

$$M,\ N,\ u_0, u_1\ \ldots\ldots\ u_{M-1},\ v_0,\ v_1,\ \ldots\ldots\ v_{N-1}.$$

End indication None.

Operation Read the program, followed by the data.

Output For one group of data, six matrices are punched using the Autocode function ϕ8. They are separated by blank tape.

(1) Cosine.
(2) Cosine Inverse.
(3) Unit.
(4) Sine.
(5) Sine Inverse.
(6) Unit.

The elements of Matrices (1), (2), (4), (5) are given to 6D, and each element x should satisfy

$$0 \leqslant |x| < 2.$$

The maximum elements (very near unity) should lie on or near the "diagonal". In general the matrices are not square, so there is no true diagonal, and this means that any row or column might have either one or two large elements.

The maximum error is ± 2 in the 6th decimal place.

The unit matrix (3) is the product of matrices (1) and (2), and should be accurate to about 6D. It is punched to 9D, and is present as a check. Similarly (6) is the product of (4) and (5).

Parameters (1) W ; see Chapter 1, label 11.
(2) W/360 ; see Chapter 1, label 5.

The program tape must be copied with the correct values inserted in Chapter 1.

Restrictions (1) $1 \leqslant M \leqslant 30$.
(2) $1 \leqslant N \leqslant 30$.

Failures None.

Time About MN/18 minutes for one group.

Method We consider one group of data, namely M, N, u_i and v_j, for $i = 0(1)M-1$ and $j = 0(1)N-1$, and a parameter W.
The v_j are constants given to 7D and the u_i are integers depending on W and the v_j.
Suppose the cosine matrix $\underset{\sim}{C}_1$ has elements a_{ij}, and the sine

matrix $\underset{\sim}{D}_1$ has elements b_{ij}.

Then

$$a_{ij} = k_{ij} \sin \pi v_j/180$$

and $\quad b_{ij} = k_{ij} \sin 2\pi u/W$

where $k_{ij} = \dfrac{\sin\pi\alpha}{\pi\alpha} \Big/ \dfrac{\sin\pi\alpha/W}{\pi\alpha/W} \quad \sin\pi(\alpha + 2u_i)/W$

and $\quad \alpha = \alpha_{ij} = Wv_j/360 - u_i$.

The cosine and sine inverses are inverses in the least squares sense unless $M = N$.

The cosine inverse $\underset{\sim}{C}$ is given by

$$\underset{\sim}{C} = (\underset{\sim}{C}_1{}' \; \underset{\sim}{C}_1)^{-1} \; \underset{\sim}{C}_1'.$$

Similarly the sine inverse is

$$\underset{\sim}{D} = (\underset{\sim}{D}_1{}' \; \underset{\sim}{D}_1)^{-1} \; \underset{\sim}{D}_1'.$$

The products $\underset{\sim}{C}_1 \; \underset{\sim}{C}$ and $\underset{\sim}{D}_1 \; \underset{\sim}{D}$ give the unit matrix.

A more comprehensive account of the theory is given in the N.I.O. Tide Elimination scheme description.

N.I.O. Program 34

Title Scale and rearrange.

Code Mercury autocode.

Purpose To put the output from either N.I.O. Program 31 or N.I.O. Program 32 into a form suitable for input to RAE 195/A.

Order of tapes Program, parameter tape, data.

Parameter tape

a_0 b_0
a_1 b_1
a_2 b_2
a_3 b_3
n

Data n lines, each containing 4 values

$x_i \quad y_i \quad z_i \quad w_i$, $i = 0(1)n-1$

End indication --->CRLF

Operation Read the program, followed by the parameter tape and data. When all the data has been read results are punched, and the program calls for a new parameter tape.

Output This consists of 4 blocks, each containing n numbers.

(1) w_i', $i = 0(1)n-1$, punched as 4 floating point (6D) numbers per line.

(2) x_i', $i = 0(1)n-1$, punched as 8 5-digit integers per line.

(3) y_i', $i = 0(1)n-1$.

(4) z_i', $i = 0(1)n-1$.

Both y_i' and z_i' are punched as 12 2-digit integers per line.

Parameters None.

Restrictions $1 \leq n \leq 400$.

Failures None.

Time About n/40 minutes.

Method The calculation has 2 stages.

(1) Scaling. There are 8 scaling parameters, and for each i, $i = 0(1)n-1$, the program calculates

$$x_i' = a_0 x_i + b_0$$
$$y_i' = a_1 y_i + b_1$$
$$z_i' = a_2 z_i + b_2$$
$$w_i' = a_3 w_i + b_3$$

The values of a and b must be chosen so that x_i', y_i' and z_i' are integers which conform with the output layout. The purpose of the scaling is to ensure that the mean values $\bar{x}_i'$, $\bar{y}_i'$ and $\bar{z}_i'$ are all approximately zero. The mean values are defined in the usual way.

(2) Rearranging. The values of x_i', y_i', z_i' and w_i' calculated as above are rearranged so that all the w_i' come first, followed by all the x_i', then the y_i' and finally the z_i'.

The data is now in a form suitable for input to RAE 195/A, which requires data grouped about a mean value near zero.

Notes (1) This is a special purpose program.

N.I.O. Program 35

Title Tables of potential temperature.

Code Mercury autocode.

Purpose To list tables of $t - \theta$, where θ = potential temperature (°C), for selected values of pressure, salinity and temperature.

Order of tapes Program, data.

Parameter tape None.

Data For one table, 9 numbers are punched as follows:-

$$\begin{array}{lll} P_0 & h_1 & P_{n-1} \\ t_0 & h_2 & t_{q-1} \\ S_0 & h_3 & S_{r-1}, \end{array}$$

where $t - \theta$ is required for

$$\begin{aligned} P &= P_0(h_1)P_{p-1} && \text{(db)} \\ t &= t_0(h_2)t_{q-1} && \text{(°C)} \\ \text{and } S &= S_0(h_3)S_{r-1} && (‰) \end{aligned}$$

End indication --> CRLF

Operation Read the program, followed by the data. When one table has been punched, more data may be read.

Output A complete table consists of p groups, one for each value of P_i, and headed by $P = P_i$, $i = 0(1)p-1$. Within a group $t - \theta$ is tabulated against q values of argument t, with one column for each of the r values of s.

Parameters None.

Restrictions (1) $1 \leqslant p,q \leqslant 90$
(2) $1 \leqslant r \leqslant 7$.

Failures (1) If $r \geqslant 8$, the table will not be listed properly, as there will be too many symbols in a teleprinter line.

Time About pqr/100 minutes.

Method For selected values of pressure (P), salinity (S) and temperature (t), the potential temperature is defined by

$$\theta = 10 + \sum_{\ell=0}^{4} \sum_{m=0}^{4} \sum_{n=0}^{4} b_{16\ell + 4m + n} (P - 3000)^{\ell}(t - 10)^{m}(S - 35)^{n}.$$

This is a least squares polynomial. See Note (1) of N.I.O. Program 28.

Notes (1) The least squares polynomial was fitted to the output of N.I.O. Program 32. Values of θ outside the range of P, S and t specified in that program are suspect.

N.I.O. Program 36

Title Properties of sea water, II.

Code Mercury autocode.

Purpose This program is almost identical to N.I.O. Program 28. Only the differences are given below.

Data Instead of pressure (P_i) in 1/10 kg/sq cm, punch depth (d_i) in metres.

Method Before any calculation is attempted, the depth is converted to pressure (db) by

$$P_i = c_{16}d_i + c_{17}d_i^2.$$

Note that the depth (D_i) calculated later may differ slightly from d_i. The more accurate value is D_i. These values of c_{16} and c_{17} are listed with N.I.O. Program 28.

Note If any other form of input is required, modify this program and not N.I.O. Program 28. The constants c_{16} and c_{17} only (c_0 to c_{15} are used elsewhere in the program) are available. The modification must be made in Chapter 0 immediately x_i has been read, ensuring that the new x_i is in decibars. Note that there are only about 30 free locations in Chapter 0.