NATIONAL INSTITUTE OF OCEANOGRAPHY
WORMLEY, GODALMING, SURREY

N.I.O. Computer Programs 6

N.I.O. Internal Report No. N6

JANUARY 1966

N.I.O. COMPUTER PROGRAMS VI

edited by

Pamela Edwards

Internal Report N.6

National Institute of Oceanography

N.6

N.I.O. Programs VI

49	Analysis of tidal currents at La Chapelle Bank, July 1962
50	Directional wave analysis 1
51	Directional wave analysis 2
52) 52A)	Tidal spreading, parts 1 and 2
53	Check by differencing
54	Directional wave analysis 3
55	Statistical moments
56) 56A)	Polynomial fitting to thermistor calibration curves
57A	Integrals of products of associated Legendre Functions
58 Atlas	Station data
Special 58 Atlas	Station data (for DISCOVERY CRUISE 6, 1965)
58 Mercury	Station data
59	Thermometer corrections for deep-sea reversing thermometers
60	Cross spectrum of two series

N.I.O. PROGRAM 49

Title Analysis of Tidal Currents at La Chapelle Bank, July 1962.

Code Mercury autocode. **Machine** MERCURY

Purpose To combine and extend N.I.O. Programs 41 and 42.

Tapes Program tape followed by data tape.

Data

(a) 27 numbers, being the values of t not required in the majority of the matrix calculations (where $w = \frac{1}{2}t - 18{\cdot}0$ and w is the time in half-hours).

(b) Then 17 lines each containing 3 values:

$x_i \quad y_i \quad z_i, \quad i = 0(1)16$

(These values are the tidal parameters σ_i, ϕ_i, R_i.)

(c) Followed by values of

$$\left.\begin{matrix} v_1(t) \\ v_2(t) \\ v_3(t) \\ v_4(t) \end{matrix}\right\} \text{ where } t = 0(1)104 \text{ omitting those values of } t \text{ in (a).}$$

(These values are the North (shallow), North (deep), East (shallow), East (deep) velocity components of currents.)

End Indication -> CRLF

Operation Read in program tape followed by data tape. Only a fifth of the data tape is read in at first. Punching begins. After the third set of results has been punched, the remainder of the data is read and the punching recommences almost immediately and continues to the end when the machine hoots.

Output

(a) 105 blocks of 8 numbers (as output of N.I.O. Program 41)

(b) Matrix [M] (7x7)

(c) Matrix $[M^{-1}]$ (7x7)

(d) Matrix $[v][M^{-1}]$ (7x4)

(e) 4 values of σ^2

(f) Matrix (4x104)

Restrictions See note (1).

Time 7 minutes.

Method

(a) As for N.I.O. Program 41, but the values of t differ.

(b) The following matrix [M] is then computed for values of t (omitting those values at the start of the data tape).

78	Σf_1	Σf_2	Σf_4	Σg_1	Σg_2	Σg_4
Σf_1	Σf_1^2	$\Sigma f_1 f_2$	$\Sigma f_1 f_4$	$\Sigma f_1 g_1$	$\Sigma f_1 g_2$	$\Sigma f_1 g_4$
Σf_2	$\Sigma f_2 f_1$	Σf_2^2	$\Sigma f_2 f_4$	$\Sigma f_2 g_1$	$\Sigma f_2 g_2$	$\Sigma f_2 g_4$
Σf_4	$\Sigma f_4 f_1$	$\Sigma f_4 f_2$	Σf_4^2	$\Sigma f_4 g_1$	$\Sigma f_4 g_2$	$\Sigma f_4 g_4$
Σg_1	$\Sigma g_1 f_1$	$\Sigma g_1 f_2$	$\Sigma g_1 f_4$	Σg_1^2	$\Sigma g_1 g_2$	$\Sigma g_1 g_4$
Σg_2	$\Sigma g_2 f_1$	$\Sigma g_2 f_2$	$\Sigma g_2 f_4$	$\Sigma g_2 g_1$	Σg_2^2	$\Sigma g_2 g_4$
Σg_4	$\Sigma g_4 f_1$	$\Sigma g_4 f_2$	$\Sigma g_4 f_4$	$\Sigma g_4 g_1$	$\Sigma g_4 g_2$	Σg_4^2

(c) is (b) inverted; $[M^{-1}]$.

(d) Values of α and β are computed as follows using values of $v_i(t)$

$$\begin{bmatrix} \alpha_{i0} \\ \alpha_{i1} \\ \alpha_{i2} \\ \alpha_{i4} \\ \beta_{i1} \\ \beta_{i2} \\ \beta_{i4} \end{bmatrix} = [M^{-1}] \begin{bmatrix} \Sigma v_i \\ \Sigma v_i f_1 \\ \Sigma v_i f_2 \\ \Sigma v_i f_4 \\ \Sigma v_i g_1 \\ \Sigma v_i g_2 \\ \Sigma v_i g_4 \end{bmatrix} \quad \text{where } i = 1(1)4$$

(e) The four values of σ_i^2 are then computed (where $i = 1(1)4$).

$$\sigma_i^2 = \frac{1}{78}(\Sigma v_i^2 - \alpha_{i0}\Sigma v_i - \alpha_{i1}\Sigma v_i f_1 - \alpha_{i2}\Sigma v_i f_2 - \alpha_{i4}\Sigma v_i f_4 - \beta_{i1}\Sigma v_i g_1 - \beta_{i2}\Sigma v_i g_2 - \beta_{i4}\Sigma v_i g_4).$$

(f) Then the four series as follows are computed

$\alpha_{i0} + \alpha_{i1}f_1 + \alpha_{i2}f_2 + \alpha_{i4}f_4 + \beta_{i1}g_1 + \beta_{i2}g_2 + \beta_{i4}g_4$.

where $i = 1(1)4$ and for ALL values of t (including those previously omitted).

Notes

(1) This is not a general purpose program, but one specially written for Job no. 1284.

(2) See also N.I.O. Programs 41 and 42.

(3) This program has been used successfully with CHLF 3 and 4.

Programmer MRS. WENDY WILSON

(d) Values of α and β are computed as follows using values of $v_i(t)$

$$\begin{bmatrix} \alpha_{i0} \\ \alpha_{i1} \\ \alpha_{i2} \\ \alpha_{i4} \\ \beta_{i1} \\ \beta_{i2} \\ \beta_{i4} \end{bmatrix} = [M^{-1}] \begin{bmatrix} \Sigma v_i \\ \Sigma v_i f_1 \\ \Sigma v_i f_2 \\ \Sigma v_i f_4 \\ \Sigma v_i g_1 \\ \Sigma v_i g_2 \\ \Sigma v_i g_4 \end{bmatrix} \quad \text{where } i = 1(1)4$$

(e) The four values of σ_i^2 are then computed (where $i = 1(1)4$).

$$\sigma_i^2 = \frac{1}{78}(\Sigma v_i^2 - \alpha_{i0}\Sigma v_i - \alpha_{i1}\Sigma v_i f_1 - \alpha_{i2}\Sigma v_i f_2 - \alpha_{i4}\Sigma v_i f_4 - \beta_{i1}\Sigma v_i g_1 - \beta_{i2}\Sigma v_i g_2 - \beta_{i4}\Sigma v_i g_4).$$

(f) Then the four series as follows are computed

$$\alpha_{i0} + \alpha_{i1}f_1 + \alpha_{i2}f_2 + \alpha_{i4}f_4 + \beta_{i1}g_1 + \beta_{i2}g_2 + \beta_{i4}g_4.$$

where $i = 1(1)4$ and for ALL values of t (including those previously omitted).

Notes

(1) This is not a general purpose program, but one specially written for Job no. 1284.

(2) See also N.I.O. Programs 41 and 42.

(3) This program has been used successfully with CHLF 3 and 4.

Programmer MRS. WENDY WILSON

N.I.O. PROGRAM 50

Title Directional Wave Analysis 1.

Code Mercury autocode. Machine MERCURY

Purpose Given the cosine and sine components of the Fourier analyses of 8 wave signals, over a range of frequencies, to form the co- and quad-spectrum estimates between various pairs of signals (including auto-spectrum estimates) with a triangular smoothing filter of arbitrary width.

Tapes Program tape, parameter tape 1, data tape, parameter tape 2.

Parameter tape 1
- n' no. of terms in each of the 2×8 sets of values
- m smoothing factor
- ℓ' first value of s of A_s and B_s values
- i', j', k' } $n = i'(j')k'$ where the n^{th} terms of the smoothed values are required for N.I.O. Program 51 input
- h a constant.

Data 8 blocks of numbers, each block containing n' pairs of numbers, A_s and B_s.

(A_s and B_s are obtained by Fourier analysis.)

Parameter tape 2 Sets of 3 values at a time:

p q o

Ending with a number 9.

Operation Read the program followed by parameter tape 1 and the data tape. Then read in parameter tape 2. The machine reads a set of p, q and o values and punches a block alternately until the number 9 is read when punching becomes continuous until an end hoot.

Output The output is in two parts.

Part 1 consists of blocks of numbers, as many blocks as there are sets of parameters on parameter tape 2; each block is headed by the values

p q o

and followed by columns of either

(a) i $C_i'(pq)$ $D_i'(pq)$ when $o = 3,6$

(b) i $C_i'(pq)$ only " $o = 1,4$

(c) i $D_i'(pq)$ only " $o = 2,5$

(d) i $-C_i'(pq)$ $D_i'(pq)$ " $o = 8,10$

or (e) i $-C_i'(pq)$ only " $o = 7,9$

where $i = m(1)n' - m + 1$.

Part 2 is arranged specifically for Job no. 1501 and may be found to be too complicated to rearrange for any other job. It is made up of n blocks of 4 sets of 6 or 7 numbers. These are the 4 sets of numbers:

C_n'' (A)

C_n'' (B)

C_n'' (C)

C_n'' (D)

C_n'' (E)

C_n'' (F)

D_n'''' (N)
D_n'' (A)
D_n'' (B)
D_n'' (C)
D_n'' (D)
D_n'' (E)
D_n'' (F)

C_n''' (G)
C_n''' (H)
C_n''' (I)
C_n''' (J)
C_n''' (K)
C_n''' (L)

D_n''' (M)
D_n''' (G)
D_n''' (H)
D_n''' (I)
D_n''' (J)
D_n''' (K)
D_n''' (L)

where $n = i'(j')k'$ and A, B, C ... N represent different pairs of values of p and q.

Parameters — None.

Restrictions

$0 < n' \leq 120$
$1 < n \leq n' - 2(m-1)$, where $n = i'(j')k'$
$1 \leq p,q \leq 8$
$1 \leq o \leq 10$

Time — This is difficult to estimate, but when

$n' = 119$
$m = 15$
$n = 1(10)91$

and with 25 sets of p, q and o the time taken was 25 minutes.

Method — The symbols used are

$p = A_s$ block label
$q = B_s$ block label

For o see o coding below.

Given 8 sets of values of A_s:

$A_{11}, A_{12}, A_{13} \ldots\ldots A_{1n'}$
$A_{21}, A_{22}, A_{23} \ldots\ldots A_{2n'}$
$A_{31}, A_{32}, A_{33} \ldots\ldots A_{3n'}$
....
$A_{81}, A_{82}, A_{83} \ldots\ldots A_{8n'}$

and a similar 8 sets of values of B_s, values of $C_i(pq)$ and $D_i(pq)$ can be calculated thus:

$$\left.\begin{aligned} C_i(pq) &= A_{pi} A_{qi} + B_{pi} B_{qi} \\ D_i(pq) &= A_{pi} B_{qi} - B_{pi} A_{qi} \end{aligned}\right\} i = 1(1)n'$$

(Also $C_i(pp)$ may be calculated, but $D_i(pp) = 0$)

From these answers the following are calculated and the results punched. (This is Part 1 of the output.)

$$C_i'(pq) = \frac{1}{m^2}\left(\sum_{r=-m}^{+m} (m - |r|)\, C_{r+i}(pq)\right)$$

$$D_i'(pq) = \frac{1}{m^2}\left(\sum_{r=-m}^{+m} (m - |r|)\, D_{r+i}(pq)\right)$$

where $i = m(1)n' - m + 1$.

Part 2 of the output, which is calculated and punched after the machine has read the "9" at the end of the parameter tape 2, consists of the following values.

$$C_n''(pq) = \frac{C_n'(pq)}{\sqrt{C_n'(pp)C_n'(qq)}} \quad , \quad D_n''(pq) = \frac{D_n'(pq)}{\sqrt{C_n'(pp)C_n'(qq)}}$$

$$C_n'''(pq) = \frac{h\ C_n'(pq)}{C_n'(pp)} \quad , \quad D_n'''(pq) = \frac{h\ D_n'(pq)}{C_n'(pp)}$$

$$D_n''''(pq) = \frac{h\ D_n'(pq)}{C_n'(qq)}$$

where $n = i'(j')k'$.

o Coding

o = 1 Print C′ and store
2 " D′ and store
3 " C′ and D′ and store
4 " C′
5 " D′
6 " C′ and D′
7 " -C′ and store
8 " -C′ and +D′ and store
9 " -C′
10 " -C′ and +D′

Where values are stored it is for use in Chapter 0 and re-arrangement for N.I.O. Program 51 input.

Remarks

(1) Highest Chapter no. is 2.

(2) Auxiliary variables range from 0 to 10,000.

(3) This program is for use with CHLF 3 and 4.

Notes

If Part 2 of the output is not required then the "9" at the end of parameter tape 2 may be omitted and the program will end asking for more data.

Programmer MRS. WENDY WILSON

6

N.I.O. PROGRAM 51 (Also 51.0, 51.1, 51.2)

Title Directional wave analysis - 2.

Code Mercury Autocode Machine MERCURY

Purpose To compute Bessel Functions $J_n(q_r)$, $J_n(q'_r)$, $n = 0(1)N$, for given series of arguments q_r, q'_r, and to solve certain sets of simultaneous linear equations with the Bessel Functions as coefficients.

51.0 computes only the Bessel Functions.

51.1 and 51.2 are similar to 51, but can solve a smaller variety of sets of equations.

Tapes Program. Basic data. Parameters and data.

Parameters and data
(a) Integers L, M, N, P.
(b) L numbers q_r ($r = 1(1)L$).
(c) M numbers q'_r ($r = 1(1)M$), (none if M = 0).
(d) If P = 3, repeat from (a) (Program 51.0 returns to (a) for all P) otherwise
(e) Integers i, ℓ (except in Program 51.0).
(f) Sets of data listed in Table 1, according to the value of $i > 0$. (Program 51.1 allows only $i \leqslant 6$, Program 51.2 allows only $i \leqslant 12$). If i = 0 or any negative integer, repeat from (a), otherwise
(g) Repeat from (e).

N.B. If $i \leqslant 0$, it must still be followed by a (dummy) integer ℓ.
i = 11 or 13 is normally preceded by a sequence (e-f) headed by i = 1
i = 12 or 14 " " " " " " " " " i = 8
i = 15 " " " " " " " " " i = 10
i = 16 " " " " " " " " " i = 7

Operation After the program tape, after the basic data tape, and again after any given data tape, (terminated by an arrow), the program asks for more data. It never stops itself, unless given invalid data (see restrictions), in which case it indicates a fault. Output is produced after every input sequence (a-d), unless P = 0, and after every input sequence (e-f).

Output
(a) Title
(b) If P = 0, prints $q_r(2,4)$, $r=1(1)L$, then $q'_r(2,4)$, $r=1(1)M$.
If P = 1, prints $q_r(0,8)$, $J_s(q_r)(1,8)$, $s=0(1)N$, for $r=1(1)L$.
If P = 2 or 3, prints $q_r(0,8)$, $q'_r(0,8)$, $J_s(q_r)$, $J_s(q'_r)(1,8)$, $s = 0(1)N$, for $r=1(1)L$.
(c) Prints i, ℓ.
(d) If $i > 0$, prints numbers listed in Table 1.
Sequences (b, c, d) are repeated as the corresponding input sequences are repeated.

Restrictions
$1 \leqslant \ell \leqslant L \leqslant 8$
$0 \leqslant M \leqslant 8$; if $M \geqslant 1$, then $1 \leqslant m \leqslant M$.
If i = 14 or 16, then $m \geqslant 2$
$0 \leqslant N \leqslant 20$
$0 \leqslant P \leqslant 3$
$0 \leqslant q_r \leqslant 24{\cdot}5$, $0 \leqslant q'_r \leqslant 24{\cdot}5$, all r.
$i \leqslant 16$, (12 for Program 51.2, 6 for Program 51.1).
$K \neq 0$.
If i = 11, then $y_r \neq 0$

The value of N should not be less than the highest order of Bessel Function required in any set of equations following it.

Method The basic data consists of 10 d.p. values of $J_n(x)$ for $n = 0(1)29$, $x = 1(1)24$, taken from Table II of Gray et al[1], with corrections as listed in Fletcher et al[2] vol. 2, p.824. Given input data (a, b, c), the program computes and stores all values of $J_n(q_r)$, $J_n(q'_r)$, using the formulae:

$$J_n(x+a) = J_n(x) + \sum_{p=1}^{\infty} (a/2)^p (p!)^{-1} \Delta_p[J_n(x)],$$

where $\Delta_1[J_n(x)] = J_{n-1}(x) - J_{n+1}(x)$,

$$\Delta_p[J_n(x)] = \Delta_{p-1}[J_{n-1}(x)] - \Delta_{p-1}[J_{n+1}(x)],$$

and $J_{-n}(x) = (-1)^n J_n(x)$.

The summation is taken as far as $p = 9$. Since $2^{-p}\Delta_p < 1$ for all p, and it is arranged that x is always taken as the integer nearest to q_r, so that $|a| \leqslant 0{\cdot}5$, the error due to truncation of the series is always less than $0{\cdot}3 \times 10^{-9}$.

The parameter P merely determines whether or not the Bessel Functions are printed, and whether there is an immediate return to input (a).

The parameter i determines which of 16 different sets of instructions are followed. In each set, the stored Bessel Functions are used (without being destroyed) to form and solve the equations for the unknowns a_s or b_s detailed in Table II, given values of x_r or y_r as right hand sides. Apart from the trivial single-variable equations shown, cases i = 1 to 10 solve ℓ equations for ℓ unknowns. Cases i = 13 to 16 solve the trivial equations, if any, then m equations for m unknowns, and include in their right hand sides ℓ or $\ell+1$ values a_r or b_r, which are stored as solutions from the previous case i. To make physical sense, this previous value of i must be as indicated in Table II. Cases i = 11, 12 merely form m estimates of a value K which is the ratio of a given number y_r to an expression involving the Bessel Functions and the stored solutions of the previous case i as indicated.

After printing the solutions, the program requests another parameter i with associated data. Any number of sequences headed by i may be taken, or repeated with different data, using the same set of Bessel Functions. Only i = 0 or any negative integer $\geqslant -511$, followed by an integer ℓ, returns the program to the opening sequence to compute a new set of Bessel Functions, which replace the previous set.

Physical interpretation

The program is intended for determining the angular harmonics a_s, b_s of a directional wave spectrum $E_f(\theta)$,

$$a_s + ib_s = \int_0^{2\pi} E_f(\theta) e^{is\theta} d\theta,$$

with s from 0 up to a certain order, given the components of the cross spectra x_r or y_r (as functions of frequency f), between wave detectors at fixed distances along the line from which θ is measured. Detectors of horizontal motion (or slope) as well as of vertical wave motion are assumed in most of the equations. The cross spectra may be obtained in the required form by means of N.I.O. Program 50. The arguments q_r of the Bessel Functions are essentially

2π x distances between detectors/wavelength,

and must be calculated separately, according to the frequency and the physical nature of the waves. Equations i = 1 to 10 are sufficient for ocean surface waves, but all equations are relevant to seismic waves (microseisms). In practice a complete set of harmonics (up to certain order) may be obtained from a limited selection of the equations available.

The theory for ocean waves may be found in papers by N.F. Barber, e.g. ref.3. The theory for seismic waves is given in an unpublished manuscript by M.S. Longuet-Higgins, ref.4. (N.B. There are some misprints in the equations on the 8th and 9th pages of the latter manuscript.)

Remarks If the program is to be used for computation of Bessel Functions only, it is quicker to use the abbreviated form 51.0, which omits the 3 chapters containing the i instructions. Similarly, for ocean wave analysis, program 51.2 is sufficient and quicker than 51. Program 51.1 is quicker than 51.2, but is less comprehensive.

References

1. Gray, Matthews, and Macrobert. A treatise on Bessel Functions. Macmillan, London, 2nd edn. 1952.

2. Fletcher, Miller, Rosenhead, and Comrie. An index of mathematical tables. Blackwell Sci. Pubs., Oxford, 2nd edn. 1962.

3. Barber, N.F. A proposed method of surveying the wave state of the ocean. N.Z.J. Sci. 2, 99-108 (1959).

4. Longuet-Higgins, M.S. Theory of noise measurement by seismic arrays. (Unpublished manuscript).

Programmer DAVID CARTWRIGHT

N.6

TABLE I

i	Input	Printed output *
1	$\ell+1$ numbers x_r, $r=0(1)\ell$.	0, a_0, x_r, $(r=1(1)\ell)$, s, a_s, $(s=2(2)2\ell)$.
2	$\ell+2$ numbers x_0, x', x_r, $r=1(1)\ell$.	0, a_0, 2, a_2, x_r, $(r=1(1)\ell)$, s, a_s, $(s=4(2)2\ell+2)$.
3	Same as 1	1, a_1, x_r, $(r=1(1)\ell)$, s, a_s, $(s=3(2)2\ell+1)$.
4	ℓ numbers y_r, $r=1(1)\ell$.	x_r $(r=1(1)\ell)$, s, a_s, $(s=1(2)2\ell-1)$.
5	$\ell+1$ numbers y_r, $r=0(1)\ell$.	Same as 1.
6	$\ell+2$ numbers y_0, y', y_r, $r=1(1)\ell$.	Same as 2.
7	Same as 1.	1, b_1, x_r, $(r=1(1)\ell)$, s, b_s, $(s=3(2)2\ell+1)$.
8	Same as 1.	Same as 3.
9	Same as 1.	2, b_2, x_r $(r=1(1)\ell)$, s, b_s, $(s=4(2)2\ell+2)$.
10	ℓ numbers x_r, $r=1(1)\ell$.	x_r, $(r=1(1)\ell)$, s, b_s, $(s=2(2)2\ell)$.
11	Integer m, then m numbers y_r, $r=1(1)m$.	y_r, z_r, z_r/y_r, $(r=1(1)m)$. z_r is l.h.s. of equation (Table II).
12	Same as 11.	y_r, z_r, y_r/z_r, $(r=1(1)m)$. " " 2 x l.h.s. " " "
13	Integer m, number K, then m + 2 numbers y_0, y', y_r, $r=1(1)m$.	m, K, 0, a'_0, y_0, 2, a'_2, y', y_r, $(r=1(1)m)$, s, a'_s, $(s=4(2)2m+2)$.
14	" , " , then m numbers y_r, $r=1(1)m$.	m, K, y_r, $(r=1(1)m)$, s, a'_s, $(s=1(2)2m-1)$.
15	" , " , then m + 1 numbers y', y_r, $r=1(1)m$.	m, K, 2, b'_2, y', y_r, $(r=1(1)m)$, s, b'_s, $(s=4(2)2m+2)$.
16	Same as 14.	m, K, y_r $(r=1(1)m)$, s, b'_s, $(s=1(2)2m-1)$.

* These printed outputs follow those detailed under Output (b) and (c).

TABLE II

i	Equations solved for a_s, b_s, a'_s, or b'_s
1	$a_0 = x_0$; $\frac{1}{2}a_0J_0(q_r) - a_2J_2 + \ldots + (-1)^{\ell}a_{2\ell}J_{2\ell}(q_r) = \frac{1}{2}x_r$.
2	$a_0 = x_0$; $a_2 = x'$; $\frac{1}{2}a_0J_0(q_r) - a_2J_2 + \ldots - (-1)^{\ell}a_{2\ell+2}J_{2\ell+2}(q_r) = \frac{1}{2}x_r$.
3	$a_1 = x_0$; $a_1J_1(q_r) - a_3J_3 + \ldots + (-1)^{\ell}a_{2\ell+1}J_{2\ell+1}(q_r) = \frac{1}{2}x_r$.
4	$a_1J_1(q'_r) - a_3J_3(q'_r) + \ldots - (-1)^{\ell}a_{2\ell-1}J_{2\ell-1}(q'_r) = \frac{1}{2}x_r$.
5	Similar to $i = 1$, but with argument q'_r instead of q_r.
6	" " $i = 2$, " " " " " " "
7	$b_1 = x_0$; $b_1(J_0(q_r) + J_2(q_r)) - b_3(J_2 + J_4) + \ldots + (-1)^{\ell}b_{2\ell+1}(J_{2\ell} + J_{2\ell+2}) = y_r$.
8	$a_1 = x_0$; $a_1(J_0(q_r) - J_2(q_r)) - a_3(J_2 - J_4) + \ldots + (-1)^{\ell}a_{2\ell+1}(J_{2\ell} - J_{2\ell+2}) = x_r$.
9	$b_2 = x_0$; $b_2(J_1(q_r) + J_3(q_r)) - b_4(J_3 + J_5) + \ldots + (-1)^{\ell}b_{2\ell+2}(J_{2\ell+1} + J_{2\ell+3}) = x_r$.
10	$b_2(J_1(q_r) + J_3(q_r)) - b_4(J_3 + J_5) + \ldots - (-1)^{\ell}b_{2\ell}(J_{2\ell-1} + J_{2\ell+1}) = x_r$.
11	Solutions from $i = 1$ used to give m estimates of K from $a_0J_{-1}(q_r) - a_2(J_1 - J_3) + a_4(J_3 - J_5) - \ldots + (-1)^{\ell}a_{2\ell}(J_{2\ell-1} - J_{2\ell+1}) = Ky_r$.
12	Solutions from $i = 8$ used to compute m estimates of K from $a_1J_1(q_r) - a_3J_3(q_r) + \ldots + (-1)^{\ell}a_{2\ell+1}J_{2\ell+1}(q_r) = \frac{1}{2}K^{-1}y_r$.
13*	$a'_0 = y_0 - a_0$; $a'_2 = y' + a_2$; $a'_0J_2(q'_r) - a'_2(J_0 + J_4) + a'_4(J_2 + J_6) - \ldots - (-1)^{m}a'_{2m+2}(J_{2m} + J_{2m+4}) =$ $y_r + [a_0J_2(q_r) - a_2(J_0 + J_4) + \ldots + (-1)^{\ell}a_{2\ell}(J_{2\ell-2} + J_{2\ell+2})]$, where the a_s are solutions from $i = 1$, divided by K^2.
14*	$a'_1(J_{-1}(q'_r) + J_3(q'_r)) - a'_3(J_1 + J_5) + \ldots - (-1)^{m}a'_{2m-1}(J_{2m-3} + J_{2m+1}) =$ $y_r + [a_1(J_{-1}(q_r) + J_3(q_r)) - \ldots + (-1)^{\ell}a_{2\ell+1}(J_{2\ell-1} + J_{2\ell+3})]$, where the a_s are solutions from $i = 8$, divided by K.
15*	$b'_2 = y' + b_2$; $b'_2(J_0(q'_r) - J_4(q'_r)) - b'_4(J_2 - J_6) + \ldots + (-1)^{m}b'_{2m+2}(J_{2m} - J_{2m+4}) =$ $y_r + [b_2(J_0(q_r) - J_4(q_r)) - b_4(J_2 - J_6) + \ldots - (-1)^{\ell}b_{2\ell}(J_{2\ell-2} - J_{2\ell+2})]$, where the b_s are solutions from $i = 10$, divided by K.
16*	$b'_1(J_{-1}(q'_r) - J_3(q'_r)) - b'_3(J_1 - J_5) + \ldots - (-1)^{m}b'_{2m-1}(J_{2m-3} - J_{2m+1}) =$ $y_r + [b_1(J_{-1}(q_r) - J_3(q_r)) - \ldots + (-1)^{\ell}b_{2\ell+1}(J_{2\ell-1} - J_{2\ell+3})]$, where the b_s are solutions from $i = 7$, divided by K.

For $i = 1 - 10$, the suffix r takes the values $1(1)\ell$,
For $i = 13 - 16$, " " " " " " $1(1)m$.
*N.B. For $i = 13 - 16$, the argument on the left is q'_r; on the right q_r.

N.I.O. PROGRAM 52 and 52A.

Title Tidal spreading, parts 1 and 2 being Programs 52 and 52A.

Language CHLF 3/4

Machine Mercury

Purpose N.I.O. Program 52 attempts to fit the product of a low pass filter and tidal frequencies assuming that the input is the amplitude spectrum of the tides.

N.I.O. Program 52A attempts the same thing but also allows the velocity spectrum to enter.

These are special purpose programs. Details of the computations are kept in a mathematics library file.

Programmer JAMES CREASE

N.I.O. PROGRAM 53

Title Check by Differencing.

Code Mercury autocode. Machine MERCURY

Purpose To find large mistakes in data punched on 5-hole Mercury tape.

Tapes Program tape, parameter tape, data tape.

Parameter Tape
j, k — where $n = 100j + k$ is the no. of terms.
x — the maximum difference expected.

Data n numbers.

End Indication → CRLF.

Operation
(1) Run in the program tape.
(2) Run in the parameter tape followed by the data. Punching will begin whilst the data is being read. Finishes asking for more data.
(3) (2) may be repeated with a new set of parameters and data.

Output This is a list of the numbers of the terms where the differences exceed the value x.

At the end of the list the sum of all the terms is given.

Parameters None.

Restrictions
$0 \leq j \leq 511$
$0 \leq k \leq 99$

Time $0 \cdot 56n + 0 \cdot 22N$ (where N is the no. of times x is exceeded)

Method The data is a series of n terms, denoted by a_i, where $i = 0(1)n-1$. The program first punches a list of the values of i for which

$$|a_i - a_{i-1}| > x$$

and then punches the value

$$\sum_0^{n-1} a_i$$

Remarks
(1) The highest Chapter no. is 0.
(2) Matrix routines are not required.
(3) This program makes no use of any CHLF 3 or 4 facility.

Note This program was written under Job no. 1605.

Programmer MRS. WENDY WILSON

N.I.O. PROGRAM 54

Title Directional wave analysis - 3.

Code Mercury Autocode. **Machine** MERCURY

Purpose Given the amplitudes of the Fourier harmonics up to order N of an unknown function, to compute two reasonable approximations to the function at given increments of angle from 0 to 360°.

(Designed primarily for use with output of Program 51)

Tapes Program. Parameters and data.

Parameters and data 5 integers:

I = Identification parameter (such as frequency)
J = (a) any integer < 0; results divided by π
(b) any integer ≥ 0; results not divided by π
K = Reference angle in whole degrees
L = Incremental angle in whole degrees
N = Highest order harmonic

followed by:

N pairs of numbers s (integer), A_s (fixed decimal number), where s covers the numbers 1 to N in any order, then another set of N pairs s, B_s, arranged similarly.

(It is assumed that $A_0 = 1$, so that in practice these harmonics are in effect A_s/A_0, B_s/A_0)

Restrictions

$0 \leq K \leq 360$
$1 \leq L \leq 90$
$1 \leq N \leq 64$

Operation The computer takes the program, then parameters and data tape, prints output, then returns to ask for next set of parameters and data, starting with I.

Output

(a) 4 integers I, J, K, N, from input
(b) Integer s, followed by 2 numbers $E_1(s)$, $E_2(s)$, where s takes the values 0(L)M, M being the largest multiple of L which is ≤ 360.

Method If $I \geq 0$, so that E_1 and E_2 are not divided by π, the quantities computed are

$$E_1(s) = \tfrac{1}{2} + \sum_{r=1}^{N} [A_r \cos r(s-K) + B_r \sin r(s-K)],$$

$$E_2(s) = \tfrac{1}{2} + \sum_{r=1}^{N} Q_r(s)[A_r \cos r(s-K) + B_r \sin r(s-K)],$$

$$\text{where } Q_r(s) = \frac{(N!)^2}{(N+r)!(N-r)!}$$

Remarks The quantities E_1, E_2, defined above, can be shown to be expressible in the form

$$E_p(s) = \int_{-\pi}^{\pi} E(\phi - K)\, f_p(\phi - s)\, d\phi,$$

where $E(\phi) = \frac{1}{2} + \sum_{r=1}^{\infty} (A_r \cos r\phi + B_r \sin r\phi)$,

$$f_1(\phi) = \frac{1}{2\pi} \frac{\sin(N+\frac{1}{2})\phi}{\sin \frac{1}{2}\phi} ,$$

$$f_2(\phi) = \frac{2^{2N}}{2\pi} \frac{(N!)^2}{(2N)!} \cos^{2N} \tfrac{1}{2}\phi .$$

$f_1(\phi)$ and $f_2(\phi)$ are "filter functions" of total integral unity, peaked in the direction $\phi = 0$, the more sharply the larger the value of N.

For a general function $F(\phi)$ we have also

$$E(\phi) = \pi F(\phi) \Big/ \int_{-\pi}^{\pi} F(\phi)\, d\phi$$

$$\text{and } A_r + iB_r = (1/\pi) \int_{-\pi}^{\pi} E(\phi)\, e^{ir\phi}\, d\phi ,$$

<u>Programmer</u> DAVID CARTWRIGHT

N.I.O. PROGRAM 55

Title Statistical Moments.

Code Mercury Autocode. **Machine** MERCURY

Purpose To compute the mean, moments, and cumulants, up to order 4, of a given set of numbers, and also the co-moments and co-cumulants when two parallel sets of numbers are given.

Tapes Program, parameters, data (possibly data tape 1 and data tape 2).

Parameters
(1) Integer $i = 1$ or 2.
(2) Integers j, k, where $(100j + k) = n$ is the number of terms in each series.

Data If $i = 1$, a single set of n numbers.
If $i = 2$, the first set, followed by the second set, each containing n numbers.
The numbers in the data need not be integers, or positive.

Operation After taking the program, the computer may repeat the cycle: parameters-data-output, any number of times, and calls for further parameters and data after each output.

If a set of data A is to be used in conjunction with set B and also with set C, then it must be fed in twice separately, as for example, 2, j, k, A, B; 2, j, k, A, C; and so on. This means that tape A may have to be re-wound rather quickly.

Output
(1) Title.
(2a) If $i = 2$, prints n, m_{10}, m_{01}, then the following array of 36 numbers:

$$\begin{array}{lllll}
\mu_{10} & \mu_{01} & & & \\
\mu_{20} & \mu_{11} & \mu_{02} & & \\
\mu_{30} & \mu_{21} & \mu_{12} & \mu_{03} & \\
\mu_{40} & \mu_{31} & \mu_{22} & \mu_{13} & \mu_{04} \\
\kappa_{40} & \kappa_{31} & \kappa_{22} & \kappa_{13} & \kappa_{04} \\
 & & & & \\
1 & \mu'_{11} & 1 & & \\
\mu'_{30} & \mu'_{21} & \mu'_{12} & \mu'_{03} & \\
\mu'_{40} & \mu'_{31} & \mu'_{22} & \mu'_{13} & \mu'_{04} \\
\kappa'_{40} & \kappa'_{31} & \kappa'_{22} & \kappa'_{13} & \kappa'_{04}
\end{array}$$

Values of μ and κ are printed (0,4); values of μ' and κ' are printed (2,4).

(2b) If $i = 1$, prints n, m_{10}, then 9 numbers, consisting of the first column only of the above array.
(3) Output of the form 2a or 2b is repeated after every input sequence starting with i, j, k.

Restrictions
$i = 1$ or 2 only
$0 \leqslant j \leqslant 215$ if $i = 1$, 107 if $i = 2$
$0 \leqslant k \leqslant 99$; $i(100j + k) \leqslant 21504$

Both j and k must be given, even if one of them is 0.

Method Calling the two series x_r, y_r, $r = 1(1)n$, $n = 100j + k$, (x_r only, if $i = 1$), the program first computes the mean values

$$m_{10} = \frac{1}{n}\Sigma x_r, \quad m_{01} = \frac{1}{n}\Sigma y_r,$$

then, after replacing each value x_r, y_r by

$$X_r = x_r - m_{10}, \quad Y_r = y_r - m_{01},$$

computes the moments μ_{pq}, $(p + q = 1, 2, 3, 4)$,

where $\mu_{pq} = \frac{1}{n} \Sigma X^p Y^q$,

and the 4th order cumulants:

$$\kappa_{40} = \mu_{40} - 3\mu^2_{20}$$

$$\kappa_{31} = \mu_{31} - 3\mu_{11}\mu_{20}$$

$$\kappa_{22} = \mu_{22} - \mu_{20}\mu_{02} - 2\mu^2_{11}$$

$$\kappa_{13} = \mu_{13} - 3\mu_{11}\mu_{02}$$

$$\kappa_{04} = \mu_{04} - 3\mu^2_{02}$$

Finally, the program computes the normalised moments and cumulants:

$$\mu'_{pq} = \mu_{pq}/(\mu_{20})^{p/2}(\mu_{02})^{q/2}, \quad (p + q = 2, 3, 4),$$

$$\kappa'_{pq} = \kappa_{pq}/(\mu_{20})^{p/2}(\mu_{02})^{q/2}, \quad (p + q = 4).$$

Notes

μ_{10} and μ_{01} should ideally be zero, but owing to rounding-off errors they will usually be printed (in floating decimal) as small non zero numbers. The normalised variances μ'_{20}, μ'_{02} are always printed as exactly 1.

μ'_{11} is the correlation coefficient
μ'_{30}, μ'_{03} are the coefficients of "skewness"
μ'_{40}, μ'_{04} are the coefficients of "kurtosis"

The cumulants of order 2 and 3 are identical with the corresponding moments.

Programmer DAVID CARTWRIGHT

N.I.O. PROGRAMS 56 and 56A

Title Polynomial fitting to Thermistor Calibration Curves.

Code Mercury autocode. Machine MERCURY

Purpose To approximate to some measured calibration curves of a series of thermistors (used on Discovery II, August 1962) by polynomials of degree up to 4, fitted by "least squares" to 7 pairs of readings.

Tapes Program and data on one tape.

Parameter tape None.

Data A block of 7 pairs of numbers

x_i y_i , where $i = 1(1)7$

End Indication -> CRLF.

Operation (1) Run in the program.

(2) Run in the data. Punches output.

(3) (2) may be repeated with more blocks of data.

Output A list of values

a_n where $n = 0(1)4$

v_4

a_n where $n = 0(1)3$

v_3

a_n where $n = 0(1)2$

v_2

(In N.I.O. program 56A v_4, v_3, v_2 are replaced by v_4^*, v_3^*, v_2^*.)

Parameters None.

Restrictions $i = \bar{7}$
$n = 4(-1)2$

Failures N.I.O. Program 56 fails when asked to calculate a negative square root; i.e. when v^2 is negative. If this occurs run the data with N.I.O. Program 56A, which prints v^* (which equals v^2) instead of v.

Time The program took 3 minutes to do 5 cases.

Method Given 7 values of x_i and y_i, we use matrix substitution to find the values of a_n in the following simultaneous equations:

$$7a_0 + a_1\Sigma x + a_2\Sigma x^2 \;\ldots\ldots + a_n\Sigma x^n = \Sigma y$$
$$a_0\Sigma x + a_1\Sigma x^2 + a_2\Sigma x^3 \;\ldots\ldots + a_n\Sigma x^{n+1} = \Sigma xy$$
$$\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots$$
$$a_0\Sigma x^n + a_1\Sigma x^{n+1} + a_2\Sigma x^{n+2} \;\ldots\ldots + a_n\Sigma x^{2n} = \Sigma x^n y$$

where $n = 4(-1)2$.

We also require either

(a) the value v_n^* in Program 56A

$$v_n^* = \frac{1}{n}\left\{\Sigma y^2 - a_0\Sigma y - a_1\Sigma xy \;\ldots\ldots - a_n\Sigma x^n y\right\}$$

or (b) the value of v_n in Program 56

$$v_n = \sqrt{v_n^*}$$

Notes

(1) This is not a general purpose program but one written specially for Job no. 1626.

(2) When N.I.O. Program 56A is used an * is printed against the last figure of the value of v_n^*.

Programmer MRS. WENDY WILSON

N.I.O. PROGRAM 57A, Atlas and Mercury

Title: Integrals of products of associated Legendre Functions.

Language: Atlas version / Mercury version — CHLF 3/4

Notes: N.I.O. Program 57A was tested and initially run on Mercury; it was then run on Atlas to obtain more accuracy.

N.I.O. Program 57 was abandoned before completion because the data was incorrect.

These are special purpose programs.

Programmer: JAMES CREASE

N.I.O. PROGRAM 58 Atlas - Station data

There are several versions of this program depending on whether pressure or depth is imput (see note A in method). They are:-

a) NIO 58 Atlas pressure (db) input
b) NIO 58 Atlas pressure (1/10 kg/cm^2) input
c) NIO 58 Atlas pressure (db) input, potl d output
d) NIO 58/1 Atlas, depth input, Mediterranean
e) NIO 58/2 Atlas, depth input, Western North Atlantic
f) NIO 58/3 Atlas, depth input, Eastern North Atlantic.

Language EMA Machine ATLAS 1

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

Tapes Program, parameters and data are all one document but usually consist of two tapes:-

tape 1 job description and program, which ends with the program constants and * * * T

tape 2 parameters and data, ending with * * * Z.

N.B. It is essential that the tapes be read into the computer in the correct order.

Job description and program

This is at the start of the program tape and consists of:-

JOB

S001 + 6 digits, followed by program title

OUTPUT

0 LINE PRINTER 1 LINES

COMPUTING c INSTRUCTIONS

STORE 22 BLOCKS

where l = 200 + 100 (no. of stations)

and c = 1500 + 300 (no. of stations)

The program is headed by:-

COMPILER EMA

Parameters and data

The parameters are:-

1) T (salinity correction, zero usually)

2) m ($m \leq 19$ and must be odd, see a_i below)

3) a_i (standard pressures and intervals) for $i = 0(1)\ m-1$. The standard pressures are defined by

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

The data consists of items 4) to 6):-

4) a title consisting of one line only, i.e. 1 teleprinter line, such as ship and station number, date, etc.; the title must not be omitted and must begin with CR; it may not contain another CR before the final CR LF.

5) n the number of samples for that station ($n \leq 49$).

6) pressure, temperature and salinity values punched as integers, 1 sample per line, and being:-

p_i $100\ t_i$ $1000\ S_i$ for i = 0(1)n-1

where p is pressure or depth (metres), integer (see first paragraph of program description, a) to f), for form of input);

t is temperature °C, 2D;

S is salinity ‰, 3D.

If t_i is missing punch 9999;

if S_i is missing punch 99999.

For any i, either of t_i or S_i may be missing but p_i must be present.

If p_0 is a surface value (e.g. $p_0 = 1$) punch 0.

The data, items 4) to 6), may be repeated as often as required.

If new parameters are required item 6) should be followed by

runout CR LF

NEW PARAMETERS

* (asterisk)

and parameters 1) to 3) followed by data 4) to 6).

At the end of the data, item 6) should be followed by runout CR LF

NO MORE DATA

>

more than 6" blank tape

* * * Z

Output

The output for each station consists of:-

station title

a) results at observed pressures

b) results at standard pressures.

Sections a) and b) are on separate pages on the line printer output.

In section a), if either temperature or salinity was missing in the data an asterisk (*) is printed in place of each missing value. The pressure and any remaining data are printed for the relevant line, but no other results appear on that line.

The results a) at observed pressures are:-

sample number;
pressure, decibars (see note A in Method), integer; this is observed or interpolated from unprotected thermometer readings;
depth, metres, integer; calculated from pressure and specific volume;
salinity, ‰, 3D; from electrical conductivity;
temperature, °C, 2D;
potential temperature, °C, 2D;
σ_t, 3D; program c) in list in first paragraph of program description has potential density i.e. σ_θ (θ being potential temperature) output here, 3D, instead of σ_t;
specific volume, ml/gm, 5D;
specific volume anomaly, ml/gm, 6D.

The latter four quantities are computed from Ekman's compressibility formula (1908), and Cox and Amith's (1959) values of specific heat.

The results b) at standard pressures are:-

pressure, decibars, integer;
dynamic height anomaly, dynamic metres, 3D;
this is obtained by interpolation and integration of specific volume anomalies;
potential energy anomaly, metres x decibars, 1D;
obtained by integration of product of pressure and specific volume anomaly, divided by gravity;
sound velocity, metres/second, 1D; by W.D. Wilson's formula (1960);
sounding velocity, metres/second, 1D; harmonic mean of sound velocity from surface to depth equivalent to given pressure.

All values except pressure are followed by the interpolation error.

Method

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 or depth (see first paragraph of this program description), temperature and salinity.

Note A Depth inputs are immediately converted to pressure (db) by the best available relation between pressure and depth for the geographical area in question, of the form

$$p = C_{16}d + C_{17}d^2$$

where C_{16} and C_{17} are the last two constants on the program tape. C_8 and C_9 have also been adjusted using formulae

$$C_9 = C_0/C_{16}$$

and

$$C_8 = -C_{17}\ C_9{}^2/C_{16}$$

Missing data

If the pressure is missing, the whole sample must be omitted. If 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.

The program

This is in 4 chapters, 3 for computation and one mainly for organisation (chapter 0). The calculations fall into three distinct sections as 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 coreected 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

$$\sigma_0 = \sum_{0}^{3} b_r S^r \qquad (1)$$

and then

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

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

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

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

whence

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

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

$$+ P^2\sigma_0^2 \sum_{r=0}^{1} b_{r+31} t^r + b_{33} P^3 t - \frac{c_5 P}{c_6 + c_7 P} \qquad (5)$$

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

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

$$\theta - 10 = \sum_{\ell=0}^{4} \sum_{m=0}^{4} \sum_{n=0}^{4} b_{16\ell+4m+n+59} \quad (P-3000)^{\ell} \quad (t-10)^{m} \quad (S-35)^{n} \qquad (7)$$

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

$$\Delta V_t = t \sum_{r=0}^{3} b_{r+39} t^r, \qquad (9)$$

$$\Delta V_p = P \sum_{r=0}^{3} b_{r+43} P^r, \qquad (10)$$

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

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

$$+ Pt \sum_{r=0}^{2} b_{r+53} t^r + P^2 t \sum_{r=0}^{1} b_{r+56} t + b_{58} P^3 t \qquad (12)$$

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
$\bar{V}$ = sounding velocity

The general formal definitions are

$$D_i = C_{10} \int_{P_0}^{P_i} \alpha(S,t,p)dp \qquad (13)$$

$$\Delta D_i = C_{11} \int_{P_0}^{P_i} \delta(s,t,p)dp \qquad (14)$$

$$\chi_i = C_{10} \int_{P_0}^{P_i} p\delta(S,t,p)dp \qquad (15)$$

$$\frac{1}{\bar{V}_i} = \frac{1}{D_i} \int_0^{D_i} \frac{dD}{V} \qquad (16)$$

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

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

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

and $\delta_i = \delta_i(S,t,P)$. (17)

Thus we have

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

and $D_0 = C_9(C_{10}P_0) + C_8(C_{10}P_0)^2$. (19)

Similarly

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

and $\Delta D_0 = 0$. The potential energy anomaly becomes

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

and $\chi_0 = 0$.

Finally we have

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

and $\bar{V}_0 = V_0$

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 specific 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) $$Y_1 = y_0 + \frac{y_1 - y_0}{P_1 - P_0}(P - P_0)$$

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

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 output 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 (b_0 - b_{93} followed by c_0 - c_{17}).

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

(4) Mercury versions of these programs are described separately.

(5) W.D. WILSON'S OCTOBER 1960 formula replaces that of JUNE 1960 which was used in previous versions of the station data program, i.e. N.I.O. Programs 28 and 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)

Programmer JAMES CREASE

Special N.I.O. PROGRAM 58 Atlas, dated 10/11/1965

This is a special version of N.I.O. Program 58 Atlas pressure (db) input, for use only with data from DISCOVERY cruise 6, 1965. It is a modernised version (i.e. some Mercury instructions have been replaced by EMA ones) and has additional instructions to compute and print-out the percentage of Norwegian Sea water at each observed pressure level.

The program description for N.I.O. Program 58 Atlas is applicable to this special program with the following reservations:-

a) Parameters 1) to 3) (i.e. T, m, a_i) must here be followed by two special parameters U and B, being the values of σ_t for North Atlantic water (t = 9°C, S = 35·33‰) and Norwegian Sea water (t = -0·45°C, S = 34·92‰) respectively, where U = 27·399, B = 28·084.

b) The words DATA TITLE must precede the line of title (item 4) of data) in every case; similarly if new parameters are required or the end of the data is reached, the line of title must be preceded by the words DATA TITLE, e.g.

```
DATA TITLE
DI 5684
24
followed by data
   :
DATA TITLE
NEW PARAMETERS
*
followed by parameters T, m, a_i,
27·399
28·084
DATA TITLE
DI 5685
   :
ending data tape with
DATA TITLE
NO MORE DATA
>
* * * Z
```

c) The output has an extra value (unlabelled) for each observed pressure, the value being the percentage of Norwegian Sea water at that pressure (as defined above).

N.I.O. PROGRAM 58 Mercury - Station data

There is a set of programs for Mercury similar to the set for Atlas, i.e. programs a) to f) listed in the NIO Program 58 Atlas description.

The Mercury programs differ from the Atlas ones only as follows:-

1) In the Mercury versions there is no JOB DESCRIPTION or COMPILER heading. Instead there are two lines of title.

2) In the Atlas versions there are two RUN OUT instructions near the end of chapter 0; these are omitted in the Mercury versions.

3) In the Mercury version the program tape ends with —>; the Atlas one ends with * * * T

4) The Mercury programs are in CHLF 3/4, not EMA.

The Mercury data tapes end with —>; the Atlas ones end with * * * Z.

Mercury operation

1) Read in program; it stops on the arrow.

2) Read in parameter and data tape; the program reads and punches for one station at a time until the arrow at the end of the data is reached.

If the program is restarted, the re-enter procedure should be used, and the last three blocks on the program tape must be re-read, i.e. the constants.

Time About (5n + 2N) seconds per station + program reading time, where N is the number of standard pressures obtained.

N.I.O. PROGRAM 59

Title Thermometer corrections for deep-sea reversing thermometers.

Language CHLF 3/4

Machine MERCURY

Purpose In correcting deep-sea reversing thermometer readings, corrections which are dependent on the water temperature, the ambient temperature in the laboratory, and the index errors of the thermometers, must be applied. Manual correction is simplified if the following graphs are prepared with the aid of this program.

a) Protected thermometers.

A graph of observed main thermometer readings against observed auxiliary thermometer readings for a sequence of values of ΔT_p, where ΔT_p is the correction to be added to the observed main reading to give the true in situ temperature.

N.B. Index corrections for both auxiliary and main thermometers are included in this graph.

b) Unprotected thermometers.

A graph of observed main thermometer readings against the difference between the in situ temperature and the true auxiliary temperature, for a sequence of values of ΔT_u, where ΔT_u is the correction to be added to the observed main temperature to give the true in situ reading of the main thermometer.

N.B. Auxiliary thermometers must have index corrections applied before using this graph, but main thermometer index corrections are included (see note at the end of Method).

Tapes

1) Routine 804 (General print, CHLF 3/4)
2) NIO Program 59
3) data (including parameters).

The three items may be combined onto one or two tapes, as required; the tapes should all end with →.

Parameters and data Both protected and unprotected thermometers can be dealt with.

Parameters are as follows:-

1) thermometer correction increment in °C (see Note 1 at end of program description).
2) for protected thermometers, the range of ambient temperatures required, °C; e.g. 5 ɓ ɓ 35;
   for unprotected thermometers, the range of ambient − in situ temperatures required, °C; e.g. −10 ɓ ɓ 35.
3) thermometer number, e.g. 3563
4) thermometer type, 0 for protected, 1 for unprotected
5) month and year of calibration, e.g. 6 ɓ ɓ 1965
6) the value of V_0, the volume of mercury in the ~~sten~~ below 0°C.

Data follows and consists of two blocks of pairs of readings of ambient temperature and the corresponding index correction.

Block 1 T and I for the main thermometer, terminated by * (asterisk) on a new line.

Block 2 t and I_t for the auxiliary thermometer, terminated by / (solidus) on a new line.

Parameters 1 to 6 and the two blocks of data constitute the complete data for one thermometer; any number of sets of data my follow.

Output

This is headed by the thermometer number and type, and the calibration date. Then come the following columns:-

1) headed C, is the thermometer correction (see Method)

2) headed T, is the observed main thermometer reading.

3) a) protected thermometers: t', the observed auxiliary thermometer reading corresponding to C

b) unprotected thermometers: $T_w - t$, the difference between the in situ sea temperature and the true auxiliary thermometer temperature.

4) - 7) are repeats of 3) but with C incremented by the input thermometer correction increment (see Parameters).

The output for any given T of a main thermometer ranges over the prescribed input limits of the auxiliary thermometer (or for unprotected thermometers, auxiliary - in situ). The output range will in fact always slightly exceed the input limits as specified in the parameters.

Time

About 2 minutes to read and compile R 804 and N.I.O. Program 59, and about 10 seconds per set of data.

Method

1 Protected thermometers

Let T' = observed main thermometer reading

V_0 = volume of mercury below zero graduation (in °C)

K = 1/(Thermal Expansion); usually $K = 6100$

I = index correction to main thermometer at temperature T'

t' = observed auxiliary thermometer reading

I_t = index correction to auxiliary thermometer $= t - t'$

T, t = true main and auxiliary thermometer readings.

$C_p = T - T'$

C_p may be expressed closely by

$$C_p \approx \frac{(T' + V_0)(T' - t) + IK}{K - (T' + V_0) - (T' - t)} \qquad 1)$$

in terms of observed and known variables.

We require for a temperature correction graph to plot T' against t' for a series of values of C_p (note that C_p is to include the index corrections for both main and auxiliary thermometers as well as the correction due to the difference between T and t.)

Let $x = T' + V_0$, $y = T' - t$ 2)

From 1) $C_p = \dfrac{xy + IK}{K - (x + y)}$

or $y = \dfrac{C_p (K - x) - IK}{x + C_p}$ 3)

We are given T', V_0, K and I (and therefore x) at a number of temperatures T_i' and we wish to find y and thence t and t' for a sequence of C_p sufficient to cover a prescribed range of t from t_1 to t_2 say.

$\therefore$ $t_1 < t < t_2$

$\therefore$ $y_1 = T' - t_2$, $y_2 = T' - t_1$ are the lower and upper limits of y.

Correspondingly

$C_1 \approx xy_1/K + I$ 4)

$C_2 \approx xy_2/K + I$

are the approximate lower and upper limits of C_p.

Suppose we want a C_p increment of b, then for a given T' we will start with a ΔT of

$$b \left[\text{Integral part } \left(\frac{C_1}{b} + 0{\cdot}5\right) - 1{\cdot}5\right]$$

and increment C_p by b until a ΔT of

$$b \left[\text{Integral part } \left(\frac{C_2}{b} + 0{\cdot}5\right) + 1{\cdot}5\right]$$

is reached. This allows for the calculation of an extra two values of y below and above y_1 and y_2.

Having found y from 3) for each increment of C_p for a given x (and therefore T'), t is found from 2) and finally t' is found by adding the interpolated auxiliary index error I_t to t.

The whole process is then repeated for another value of T'.

2 Unprotected thermometers

We require for this graph to plot T_u' against $T_w - t_u$ for a series of values of C_u, where

T_w = known in situ temperature of the water

T_u' = observed unprotected thermometer reading

t_u = true unprotected auxiliary thermometer reading

$C_u = T_u - T_u'$

In this case

$C_u = \dfrac{(T_u' + V_0)(T_w - t_u) + IK}{K - (T_w - t_u)}$ 5)

Writing $x = T_u' + V_0$, $y = T_w - t_u$

we have

$$y = \frac{K(C_u - I)}{x + C_u} \qquad 6)$$

with approximate lower and upper limits on C_u of

$$C_1 = \frac{xy_1}{K} + I, \quad C_2 = \frac{xy_2}{K} + I$$

where y_1 and $y_2 = T_w - t_1$, $T_w - t_2$

From 6) we can find y (i.e. $T_w - t_u$) as a function of x (and therefore T_u') for each increment of C_u.

Note that in this case the unprotected auxiliary thermometer reading is the true reading and any index corrections to the auxiliary thermometer must be applied before using the graph. This is because t_u only enters the computations in conjunction with T_w. Consequently a given $(T_w - t_u)$ might arise from a variety of T_w and t_u, the t_u all having different index corrections.

Notes

1) If it is desired to know corrections to 0·01°C without interpolation, the increment would be 0·01. For a given main thermometer reading T, the auxiliary thermometer reading would be evaluated at points such that the correction C is ± 0·005, ± 0·015 etc., i.e. it would be centred about zero C so that, for example, any point on the graph between the 0·005 and 0·015 lines for C will represent a correction of 0·01 (to the nearest 0·005).

Such a small increment produces a large output and we have found it best to use a large increment and sub-divide with multi-point dividers. In this case note that if sub-division down to 0·01 is finally required an odd multiple of this must be used, otherwise, for example, an increment in the program of 0·10 would give t at values of C = ± 0·05, ± 0·15 etc., and sub-division into 0·01 steps would give values for t at C of 0, ± 0·01, ± 0·02 etc. This in turn means, for example, that points between lines C = + 0·01, + 0·02 correspond to a correction of + 0·015 ± 0·005, and our correction now involves a third decimal.

In practice, for values of V_0 less than about 90 we use plotting intervals of 0·01 and an increment of 0·09; for larger V_0 we use plotting intervals of 0·02 and an increment of 0·18; then nine points of ten-point dividers are used for sub-dividing in both cases.

2) The values of T and ΔT (main thermometer readings and index corrections) can be taken from the N.P.L. calibration certificates or the N.I.O. calibration curves (the N.I.O. calibrations being made at M.A.F.F., Lowestoft); in the latter case the values are read off whenever the line changes gradient significantly

The values of t and Δt (auxiliary thermometer readings and corrections) are taken from the N.P.L. certificates; in the case of unprotected thermometers they are not actually used in the computations, and a dummy pair of zero corrections may be inserted instead of actual readings.

Programmer: JAMES CREASE

N.I.O. PROGRAM 60

Title Cross spectrum of two series.

Code Mercury CHLF 3/4

Machine MERCURY

Purpose Given two time series, with terms presented alternately, to compute their means, variances, covariance, auto- and cross-covariances with lags 0–L, and from these to compute the auto spectra, phase lead, coherence, co- and quad-spectra for frequencies 0 to M/2L. For the last four quantities the second series is assumed to be $\frac{1}{2}$ a time interval (second) later than the first. (See Notes below).

Tapes (1) Program. (2) Parameters and data.

Parameters (a) 7 integers:-

S = Run number.

T = Pair number.

I = 1 – Prints means, auto and cross spectra etc. only.

or 2 – Prints means and lagged covariances only.

or 3 – Prints means, covariances and auto and cross spectra.

J,K where 100J + K is the number of terms in each series.

L = Largest lag required for covariances (Spectral analysis is in terms of harmonics of the basic frequency 0·5/L).

M = Largest order of harmonic in spectra, frequency 0·5M/L.

(b) 4 numbers (fixed decimal point) D–G :-

D = Maximum allowable difference between consecutive values of series 1.

E = Maximum allowable difference between consecutive values of series 2.

F = Calibration factor, physical units per digit, series 1.

G = Calibration factor, physical units per digit, series 2.

Data Series of numbers, positive or negative, with or without decimal point, in the order

$$x_0 \quad y_0 \quad x_1 \quad y_1 \quad x_2 \; \ldots\ldots \; x_{n-1} \quad y_{n-1}$$

where n = 100J + K.

(As present, the form of print-out for the digital means and variance assumes the terms are integers between –1000 and +1000 but this is not essential to the program as a whole.)

Restrictions $-512 < S, T < 512$

I = 1, 2 or 3 only

$J \geq 1 \quad (n \nless 100)$

$0 \leq K \leq 99$

$100J + K + 3L \leq 10{,}238$ with CHLF 4
$6{,}142$ with CHLF 3

$2 \leq L \leq 140$

$1 \leq M \leq L$

Operation (1) Read in program

(2) Read in parameters and data.

As it reads in the data tape the computer tests each difference $(x_r - x_{r-1})$, $(y_r - y_{r-1})$ and prints out the value of r in the form j, k where r = j + k for any difference exceeding the given D or E. Values for the y series are preceded by twelve spaces. All such values of r, if any, are printed and unless there is none in both series the computation will not proceed any further, but returns to ask for another set of parameters and data, starting with S. If there are no "errors" in the data, the computations and print-out follow in due course, according to the parameter I. Operation (2) may then be repeated, starting with parameter S.

Output

(a) Title.

(b) Run number, pair number.

(c) Error numbers r (described above)

(d) If there is nothing in (c):-

(i) Number of terms in each series (100J + K).

(ii) Mean value of series x, mean value of series y (digital units).

(iii) Variance of x, variance of y, covariance xy (digital units).

(iv) Variance of x, variance of y, covariance xy (physical units).

(e) (Only if I = 2 or 3) For s = 0(1)L prints (1 line for each s):-

s

ψ_{11} = Normalised auto-covariance of series x, with s lags.

ψ_{22} = Normalised auto-covariance of series y, with s lags.

ψ_{12} = Normalised cross-covariance, x with s time units later than y.

ψ_{21} = Normalised cross covariance, y with s time units later than x.

(f) (Only if I = 1 or 3)

(i) Frequency increment 0·5/L. (See Notes below)

(ii) For s = 0(1)M prints (1 line for each s):-

s

E_{11} = Energy spectrum (spectrum of variance) of series x (physical units).

E_{22} = Energy spectrum (spectrum of variance) of series y (physical units).

(iii) For s = 0(1)M prints (1 line for each s):-

s

E_{12} = Co-spectrum, corrected for $\frac{1}{2}$ unit time lag of y (physical units).

E_{12}^* = Quad-spectrum, corrected for $\frac{1}{2}$ unit time lag of y (physical units).

ϕ = True phase lag of series x behind series y (degrees). (If ϕ is negative, it means x is leading in phase.)

γ^2 = Coherence of x and y.

The value of γ^2 is left blank for any value for which the product of the normalised values of E_{11} and E_{22} is negative or less than 10^{-8}.

Time 2 minutes to read in the program tape.

The following example is a guide to the computation time:

With I = 3, J = 12, K = 0, L = M = 50 Time = 9 minutes (D, E not exceeded).

Method Mean values: $X = \frac{1}{n}\Sigma x \quad Y = \frac{1}{n}\Sigma y.$

Each value x and y is then replaced by

$$x' = x - X, \quad y' = y - Y$$

and the variances computed:

$$V_{11} = \frac{1}{n-1}\Sigma x'^2 \quad (\times F^2 \text{ for physical units})$$

$$V_{22} = \frac{1}{n-1}\Sigma y'^2 \quad (\times G^2 \text{ for physical units})$$

$$V_{12} = \frac{1}{n-1}\Sigma x'y' \quad (\times FG \text{ for physical units})$$

Normalised auto- and cross-covariances:

$$\psi_{11}(s) = [(n-s-1)\,V_{11}]^{-1}\sum_{r=0}^{n-s-1} x'_{r+s}\,x'_r$$

$$\psi_{22}(s) = [(n-s-1)\,V_{22}]^{-1}\sum_{r=0}^{n-s-1} y'_{r+s}\,y'_r$$

$$\psi_{12}(s) = [(n-s-1)\sqrt{V_{11}V_{22}}]^{-1}\sum_{r=0}^{n-s-1} x'_{r+s}\,y'_r$$

$$\psi_{21}(s) = [(n-s-1)\sqrt{V_{11}V_{22}}]^{-1}\sum_{r=0}^{n-s-1} y'_{r+s}\,x'_r$$

where s is the lag number.

Auto spectra:

$$E_{11}(s) = \tfrac{1}{4}\,e_{11}(s-1) + \tfrac{1}{2}\,e_{11}(s) + \tfrac{1}{4}\,e_{11}(s+1), \quad s \neq 0$$

$$E_{11}(0) = \tfrac{1}{2}\,e_{11}(0) + \tfrac{1}{2}\,e_{11}(1).$$

where $e_{11}(s) = 4F^2V_{11}\sum_{r=0}^{L}{}'' \ \psi_{11}(r)\cos\left(\frac{rs\pi}{L}\right)$

and Σ'' means the sum with the first and last terms halved, and s is the harmonic number.

$E_{22}(s)$ is similarly defined.

Co- and quad-spectra:

(The cross spectrum is usually defined as the complex quantity $E_{12} + iE_{12}^*$).

Firstly the direct cross spectral components are computed:

$$F_{12}(s) = \tfrac{1}{4}\,f_{12}(s-1) + \tfrac{1}{2}\,f_{12}(s) + \tfrac{1}{4}\,f_{12}(s+1), \quad s \neq 0$$

$$F_{12}(0) = \tfrac{1}{2}\,f_{12}(0) + \tfrac{1}{2}\,f_{12}(1)$$

where $f_{12}(s) = 4FG\sqrt{V_{11}V_{22}}\sum_{r=0}^{L}{}'' \ \tfrac{1}{2}[\psi_{12}(r) + \psi_{21}(r)]\cos\left(\frac{rs\pi}{L}\right)$

and $F_{12}^*(s)$ is defined similarly in terms of f_{12}^*,

where $f_{12}^*(s) = 4FG\sqrt{V_{11}V_{22}}\sum_{r=0}^{L}{}'' \ \tfrac{1}{2}[\psi_{12}(r) - \psi_{21}(r)]\sin\left(\frac{rs\pi}{L}\right)$

Then the corrections for the time delay are applied:

$$E_{12}(s) = F_{12}(s)\cos\tfrac{1}{2}ks + F_{12}^*(s)\sin\tfrac{1}{2}ks$$

$$E_{12}^*(s) = F_{12}^*(s)\cos\tfrac{1}{2}ks - F_{12}(s)\sin\tfrac{1}{2}ks$$

where $k = \pi/L$.

From these values the phase lag is computed:

$$\phi = \arctan (E_{12}^*/E_{12})$$

Finally, the coherence is evaluated:

$$\gamma^2 = \frac{E_{12}^2 + E_{12}^{*2}}{E_{11}E_{22}}$$

Notes

The Fourier sums are computed by means of Watt's iteration process (see, for example, Cartwright and Catton, Int. Hyd. Rev., Vol. 11, No. 1, 1963). In the actual computations the last terms corresponding to $r = L$ are omitted for convenience; the final results after $\frac{1}{4} + \frac{1}{2} + \frac{1}{4}$ smoothing are quite unaffected by the omission.

The method of deriving the spectrum is a standard one discussed, for example, in "The Measurement of Power Spectra" by Blackman and Tukey (Dover pubs. 1958). The $\frac{1}{4} + \frac{1}{2} + \frac{1}{4}$ smoothing process is that apparently due to a certain J. von Hann, and is equivalent to multiplying the lagged covariances by $\cos^2(\frac{\pi s}{2L})$.

In contrast, the Bartlett smoothing process, used in some R.A.E. programs, multiplies the lagged covariances by $|1 - (\frac{s}{L})|$ but the resulting spectral filter (or window) has larger side-lobes than in the Hann process.

The program as written assumes that each series is sampled once per second. If a different interval (t seconds) is used the spectral analysis will be in terms of harmonics of a basic frequency $\frac{0.5}{Lt}$ c/s although the frequency increment will still be printed as $\frac{0.5}{L}$ c/s. The largest harmonic will have frequency $\frac{0.5M}{Lt}$ c/s.

Programmer DAVID CARTWRIGHT