NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY

N.I.O. Computer Programs 5

N.I.O. INTERNAL REPORT No. N 5

NOVEMBER 1963

E R R A T A

Please amend N.I.O. Program 44 as follows:-

Page 11. Title .... Written for Job 495.70 by Miss D. Raynor and Mr. J.B.J. Thorpe of the Mathematics Department at R.A.E., Farnborough

Page 12. Programmers Miss D. Raynor and Mr. J.B.J. Thorpe

N.I.O. COMPUTER PROGRAMS V

edited by

Pamela Edwards

Internal Report N.5

National Institute of Oceanography

N.5

N.I.O. Programs V

37 Check data

38 Data smoothing

39 Add or subtract series

40 Tide prediction (for DEUCE Mark II A)

41 Analysis of tidal currents at La Chapelle Bank, Nov. 1961, part I

42 " " " " " " " " " " part II

43 Density of sea water

44 Two-dimensional atmospheric pressure analysis

45 Polynomial residuals

46 Digitiser decoding

47, 47A } Least squares fitting of tracks

48, 48A } Tidal analysis and prediction

N.I.O. PROGRAM 37

Title Check data (cards).

Code Deuce basic. Machine DEUCE.

Purpose To find large mistakes in data punched as 8 signed 4-digit integers per card.

Order of cards Program (cards 0-37), parameter card, data.

Parameter card Columns 17-20 n, the number of terms.
21-24 M, the maximum difference expected.

Data n integers, punched as 8 signed 4-digit numbers per card in the α-field.

End card None.

Operation (1) Read the program with the initial input key.

(2) Run in the parameter card and data.
When the results have been punched, the program hoots and stops on 1,1-1. A single shot clears the hooter.
Operation (2) may then be repeated.

Output (1) 1 card.
Columns 18-27 $\sum_{0}^{n-1} x_i$; sign in column 17.

(2) Fault cards.
This is a list of the numbers of the terms causing faults, punched as 8 signed 4-digit integers per card in the α-field. The number of cards punched must be a multiple of 4, and any card (or part of a card) occurring after the true end of the results is filled with zeros.

Parameters None.

Restrictions (1) $1 \leq n \leq 3776$.
(2) $0 \leq N \leq 3744$, where N is the number of faults.

Failures If N > 3744, the program will be over-written, and so will apparently either fall out, stop, or go into a loop.

Time About (8n + 2N)/200 seconds.

Method The data is a series of n terms, denoted by x_i, $i = 0(1)n-1$.

The program first computes and punches $\sum_{0}^{n-1} x_i$, and then punches a list of the i for which

$$|x_{i+1} - x_i| \geq M$$

In general i = n - 1 will appear.

Programmer Miss Diana Catton.

N.I.O. PROGRAM 38

Title Data smoothing.

Code Deuce basic.

Machine DEUCE.

Purpose To filter or smooth data.

Order of cards Program (cards 0-67), parameter card 1, parameter card 2, weights card, parameter card 3, production data.

Parameter cards (1) Smoothing specification.

Columns	
17-20	blank
21-24	h
25-28	R
29-32	0001
33-36	0 for Y or y 1 for z
37-40	0 if 8 numbers per card 1 if 1 number per card
41-44	0 for y or z 1 for Y

(2) Weighting specification.

Y_α row: a one in each digit (or sign) position except the last digit of a number (see Data (1)).

Columns 17-20 R.

(3) Data specification.

Y_α row: leave this blank if there are 8 4-digit numbers per card; otherwise, punch a one in each digit (or sign) position except the last digit of a number (see Data (2)).

Columns 17-20 n.

Data (1) Weights.

These are punched as R integers in the layout specified by parameter card (2). The first weight must start in column 17 of the first weights card, but the weights need not occupy the whole of the last card; the weights are completely general.

(2) Production data.

This must be punched in such a way as to be consistent with parameter cards (1) (columns 37-40) and (3); the n numbers are punched as either:

(i) 8 signed 4-digit integers per card in the α-field,
or (ii) 1 signed single length integer per card in the α-field; the single integer may start anywhere on the card (in the α-field) but column 17 is preferable.

All signs (only negative signs need to be punched) are overpunched.

End Card None.

Operation

(1) Read in the program with the initial input key; it stops on 1, 7-12.

(2) Run in the parameter cards, weights card, and production data in the correct order and give a single shot. When all the data has been read results are punched and the program hoots and stops on 1, 6-24.

(3) Give a single shot. This clears the hooter and the program returns to 1,7-12; operation (2) may now be repeated.

Output

(1) N integers, in a layout corresponding to that of the production data, i.e. either:

(i) 8 signed 4-digit integers per card in the α-field; the number of cards punched out will be a multiple of 4, and any card (or part of a card) occurring after the end of the data will be punched with zero terms.

or (ii) 1 signed integer per card (N cards), having: columns 17-26 Y, y or z; sign in column 17. This output layout applies to any input layout so long as the input was given as 1 number per card.

(2) 1 card, containing means:

Columns 17-26 $\bar{x}$; sign in column 17.
27-36 $\bar{Y}$, $\bar{y}$ or $\bar{z}$; sign in column 27.

Restrictions

(1) $n \leqslant 3392$ (see Note (1)).
(2) $1 \leqslant R \leqslant 31$
(3) $R \leqslant n$
(4) if z is required R must be odd
(5) if W = 0 only Y is possible
(6) Y, y, and z must be single length. If the output is 8 numbers per card, they must also satisfy

$$-9999 \leqslant Y, y, z \leqslant 9999.$$

Failures

(1) 2, 1-1x $n < R$
(2) 2, 4-4x z required for R even
(3) 2, 2-2x $R > 31$

Time About (17R + N/20) seconds.

Method The data (see Data (2)) consists of a series of n terms denoted by x_i, $i = 0(1)n-1$. There are also R weights, denoted by w_j, $j = 0(1)R-1$. We define

n = number of terms in the data (x-series)
N = number of terms in the smoothed series (Y, y or z)
h = smoothing interval
R = number of weights

No special provision is made at the ends of the series, so the smoothed series contains N(<n) terms, where N is the integral part of

$$(n - R + h) / h.$$

The form of the output is specified by parameter card (1), columns 33-36 and 41-44. There are three possibilities, namely Y, y or z. The definitions are

$$Y_i = \sum_{j=0}^{R-1} w_j \, x_{i+j},$$

$$y_i = Y_i/W,$$

and $z_i = x_{i+r-1} - y_i$, where $W = \sum_{j=0}^{R-1} w_j$ and $i = 0(h)N-1$.

Usually $h = 1$ or $h = R$. Clearly z_i is only possible when R is odd and $R = 2r + 1$, and neither y_i nor z_i should be considered when $W = 0$ or is small. The case $W = 0$ arises in differencing.

<u>Notes</u> (1) The program has not been tested for large amounts of data.

(2) For output (1) (i) list output with board labelled "8 4-digit numbers";
for output (1)(ii) list with specially plugged board labelled "NIO 38 output".

<u>Programmer</u> Miss Diana Catton.

N.I.O. PROGRAM 39

Title Add or subtract series.

Code Deuce basic.

Machine DEUCE.

Purpose To add or subtract corresponding terms of two series.

Order of cards Program (cards 0-37),
parameter card 1,
series 1,
parameter card 2,
series 2.

Parameter cards (1) Columns 17-20 n
21 blank or - (x row)

(2) Columns 17-20 n
21 blank or - (x row)

Data (1) x-series. n integers punched as 8 signed 4-digit numbers per card in the α-field.

(2) y-series. n integers punched as in (1).

End card None.

Operation (1) Read the program with the initial input key; it stops on 6, 6-12.

(2) Run in the parameter cards and data in the correct order. The program is still on 6, 6-12.

(3) Give a single shot. When all the data has been read the results are punched and the program hoots and stops on 1, 6-24.

(4) A single shot clears the hooter; operation (2) may then be repeated.

Output n terms of the z-series, punched as 8 signed 4-digit integers per card in the α-field. The number of cards punched out will be a multiple of 4, and any card (or part of a card) occurring after the end of the data will be punched with zero terms.

Restrictions (1) $1 \leq n \leq 3744$
(2) $|z_i| = |\pm x_i \pm y_i| \leq 9999$ for $i = 0(1)n-1$.

Failures 6, 1-1 x with hooter if the number of terms in the two series is not the same. A single shot returns the program to read a new parameter card (2).

Time About n/300 minutes.

Method The program combines corresponding terms x_i and y_i to form the term z_i of a new series. The formulae are specified by the two parameter cards thus:

	Parameter card 1 (column 21)	Parameter card 2 (column 21)	Formula
(i)	Blank	Blank	$z_i = +x_i + y_i$
(ii)	Blank	- (minus)	$z_i = +x_i - y_i$
(iii)	- (minus)	Blank	$z_i = -x_i + y_i$
(iv)	- (minus)	- (minus)	$z_i = -x_i - y_i$

In all cases $i = 0(1)n-1$.

 Programmer Miss Diana Catton.

N.I.O. PROGRAM 40

Title Tide prediction (for DEUCE Mark II A).

Code Deuce basic.

Machine This program should work on both DEUCE Mark I and DEUCE Mark II A.

Purpose This program has exactly the same purpose as N.I.O. Program 23 but the latter will work only on a DEUCE Mark I.

Notes When N.I.O. Program 40 is run on a DEUCE Mark II A the input data must be read in in separate groups of cards; the first three groups must be read in at 200 cards/minute, and the last two at 100 cards/minute.

The groups are as follows:

at 200 cards/minute
(1) program (cards 0-61)
(2) X, Y data (N.I.O. 13 output)
(3) range card

at 100 cards/minute
(4) S^- (with parameter card)
(5) S^+ (" " ")

The program has been run at the Ministry of Agriculture, Fisheries and Food, Guildford in 1961, 1962, and early 1963, at which time the personal contact was with MR. LONGWORTH.

Programmer Miss Diana Catton

N.I.O. PROGRAM 41

Title Predicted Tides at Brest, Nov. 1961. (See Notes (2) and (3))

Code Mercury autocode. **Machine** MERCURY

Purpose The prediction of separate tidal species of the tides at Brest.

Tapes Program and data on one tape.

Data 17 lines each containing 3 values:

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

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

End Indication -> CRLF

Operation Read in the program and data. When all the data has been read the output is punched continuously, finishing in an end hoot.

Output 87 blocks of 8 numbers

t

$f_1(t) \qquad g_1(t)$

$f_2(t) \qquad g_2(t)$

$f_4(t) \qquad g_4(t)$

$f_1(t) + f_2(t) + f_4(t).$

Restrictions See Note (1).

Time 5 minutes approximately.

Method

$$\left.\begin{matrix} f_1(t) \\ g_1(t) \end{matrix}\right\} = \sum_{i=0}^{4} z_i \begin{matrix} \sin \\ \cos \end{matrix} (y_i + tx_i)$$

$$\left.\begin{matrix} f_2(t) \\ g_2(t) \end{matrix}\right\} = \sum_{i=5}^{13} z_i \begin{matrix} \sin \\ \cos \end{matrix} (y_i + tx_i)$$

$$\left.\begin{matrix} f_4(t) \\ g_4(t) \end{matrix}\right\} = \sum_{i=14}^{16} z_i \begin{matrix} \sin \\ \cos \end{matrix} (y_i + tx_i)$$

$f_1(t) + f_2(t) + f_4(t)$

where $t = -3(0\cdot5)40$ and is the time in half-hours.

Notes

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

(2) The above title should have been a sub-title to "Analysis of Tidal Currents at La Chapelle Bank, Nov. 1961".

(3) See also N.I.O. Programs 42 and 49.

Programmer Mrs. Wendy Wilson

N.I.O. PROGRAM 42

Title Analysis of Tidal Currents at La Chapelle Bank, Nov. 1961.

Sub-title Matrix Multiplication.

Code Mercury autocode. Machine MERCURY

Purpose To use output from N.I.O. Program 41 to calculate the estimated relationship between the tidal constants of currents at La Chapelle Bank and tidal height at Brest.

Tapes Program and data on one tape.

Data N.I.O. Program 41 output repunched in list form

$f_1(t)$ for t = -3(0·5)40 (except those mentioned below)

$f_2(t)$ for t as for $f_1(t)$

$f_4(t)$ " " " " "

$g_1(t)$ " " " " "

$g_2(t)$ " " " " "

$g_4(t)$ " " " " "

followed by values of u(t) and v(t) for the same ranges of t, being the North and East components of currents.

[Values of t excluded are -0·5, 0·5, 3, 3·5, 13, 17, 17·5, 18, 24, 27·5, 29·5, 30, 33·5, 36, 36·5, 37, 38·5].

End Indication -> CRLF

Operation Read in the program and data. When all the data has been read the results are punched out at intervals finishing in an end hoot.

Output 3 matrices (6 x 8, 6 x 6, 6 x 2) followed by 70 pairs of numbers (u* and v*).

Restrictions See Note (1).

Time 10 minutes approximately.

Method
(1) The program first multiplies the matrix containing values of f_1, f_2, f_4, g_1, g_2, g_4 by the transpose of itself with values of u and v added. The resulting matrix = [M].

(2) Then $[M^{-1}]$ is calculated.

(3) The last two rows of [M] are multiplied by $[M^{-1}]$. The resulting matrix = [N].

(4) Then the program multiplies [N] by the original matrix (excluding the values of u and v). These last two columns of results are known as u* and v*.

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

(2) See also N.I.O. Programs 41 and 49.

Programmer Mrs. Wendy Wilson

N.I.O. PROGRAM 43

Title Density of sea water.

Code Mercury autocode.

Machine MERCURY.

Purpose This is a special purpose program to determine the equation of state of sea water given as input:

(1) the density at atmospheric pressure as a function of salinity and temperature,

(2) the specific heat at atmospheric pressure,

(3) sound velocity as a polynomial function of salinity, temperature and pressure.

For results cf. DEEP SEA RESEARCH 1962, 9, pp. 209-213.

For further details see J. CREASE.

Programmer JAMES CREASE.

N.I.O. PROGRAM 44

Title Two-dimensional Atmospheric Pressure Analysis. Written for Job 495.70 by Miss D. Raynor of the Mathematics Department at R.A.E., Farnborough.

Code Deuce Basic. **Machine** DEUCE

Purpose To compute the Atmospheric Pressure Gradients and Mean Pressure given the Atmospheric Pressures at three places.

Order of Cards Program (cards 0-56), weight cards, parameter card, data.

Weight Cards 3 cards punched with 3 9-digit numbers per card

	card 1	card 2	card 3
col. 17	sign	sign	sign
cols. 18-26	α_1	β_1	γ_1
col. 27	sign	sign	sign
cols. 28-36	α_2	β_2	γ_2
col. 37	sign	sign	sign
cols. 38-46	α_3	β_3	γ_3

(The decimal places lie between cols. 18/19, 28/29, 38/39.)

Parameter Card Y-row N (in binary) number of cards in each series
X-row k (in binary) usually 500

Data 3 series of n terms each, punched as 8 4-digit numbers per card. All numbers are integers.

End Card None.

Operation
(1) Read in the program with the initial input key.
(2) Run in the weight cards and the parameter card.
(3) Run in the data in order; series x followed by series y then series z. The data is read in 4 cards at a time. The output commences punching during the reading of series z.
(4) One output series is punched. The program stops on 1, 7-7. Single shot.
(5) (4) is repeated for the three output series.
(6) Operations (2) to (6) may then be repeated with more data.

Output 3 packs of cards, series x′, series y′ and series z′. Each card contains 8 signed 4-digit integers (0-999).

Parameters On card 25 row 8 instruction 14-14 may be altered to 23-14 to obtain half the sum of the products before adding k,

i.e. $k + \frac{1}{2}(\alpha_1 x + \beta_1 y + \gamma_1 z)$ etc.

Restrictions
(1) $n \leqslant 2496$
(2) k must be such that x′, y′, z′ lie within the range 0-999.

Failures If any of x′, y′, z′ lies outside the range 0-999 the machine stops on 6, 1-1 x. (This occurs during the reading of the last series.) O.S. lights show in binary which of the 32 numbers of the last 4 cards read in is associated with the failure. T.S.14 indicates whether the number computed is <0 or ⩾1000. Drum position indicator (middle row of lights) shows the failure to be in the x′, y′, or z′ position.

x′ uses 1, 2 ... 5
y′ uses 6, ...10
z′ uses 11, ...15

Failure may occur in all three.
The program will continue after a single shot.

Time $(\frac{457}{100}N + 25)$ seconds.

Method Given 3 series x, y and z each containing n terms denoted by

$$x_0, x_1, x_2 \ldots\ldots x_{n-1}$$
$$y_0, y_1, y_2 \ldots\ldots y_{n-1}$$
$$z_0, z_1, z_2 \ldots\ldots z_{n-1}$$

calculate the Pressure Gradients N-S(x'), E-W(y'), and Mean Pressure (z').

$$x_i' = k + \alpha_1 x_i + \beta_1 x_i + \gamma_1 x_i$$
$$y_i' = k + \alpha_2 y_i + \beta_2 y_i + \gamma_2 y_i$$
$$z_i' = k + \alpha_3 z_i + \beta_3 z_i + \gamma_3 z_i$$

where $\alpha_1 + \beta_1 + \gamma_1 = 0$

$\alpha_2 + \beta_2 + \gamma_2 = 0$

$\alpha_3 + \beta_3 + \gamma_3 = 1$ (usually $\alpha_3 = \beta_3 = \gamma_3 = 1/3$).

Note For Job 495.70 values of Atmospheric Pressure (to 1 decimal place) minus 900.0 were used for data, where pressure was in decibars

Note on restarting If a restart is required for any reason (e.g. machine failure, wrong data etc.) the complete set of data, preceded by the last two program cards may be used with the INITIAL INPUT key.

Programmer Miss Diana Raynor

N.I.O. PROGRAM 45

Title Polynomial residuals.

Code Mercury autocode.

Machine MERCURY.

Purpose To compute a polynomial at given data points and subtract the computed values from independent observational values at the same points.

Tapes
(1) Program,
(2) parameter tape,
(3) data tape.

Parameter tape M = degree of polynomial

K = 0 if computed values as well as residuals (observed-computed) are required.
= 1 if residuals only are required.

a_i = M + 1 coefficients in ascending powers of polynomial

$$\bar{y}_j = \sum_{i=0}^{m} a_i x_j^i \; ;$$ these may be punched in floating point decimal if desired.

The parameters may be punched at the start of the data tape; if not, end the parameter tape -> CR.LF.

Data tape N = number of points at which the polynomial is to be computed.

x_j, y_j; x_j are the points at which the polynomial is to be evaluated;
y_j are the independent observed values to be compared with the computed values of $\bar{y}_j$ above $[j = 1(1)N]$;
x_j and y_j should be separated by Sp.Sp. and pairs of x_j, y_j may be followed by Sp.Sp. or CR.LF.

The data tape should end with the usual end of tape indication -> CR.LF. followed by erases, unless being immediately followed by a new series of parameters and data.

Operation Read in the program tape, followed by the parameter and data tapes.

Output $j \quad \bar{y}_j \quad y_j - \bar{y}_j$ ($\bar{y}_j$ is not punched if k = 1)

Three consecutive blocks are printed across the page, e.g.

$j \;\; \bar{y}j \;\; yj-\bar{y}j$, $j+1 \;\; \bar{y}j+1 \;\; yj+1 - \bar{y}j+1$, $j+2 \;\; \bar{y}j+2 \;\; yj+2 - \bar{y}j+2$

After the last block $Z = \Sigma(y_j - \bar{y}_j)^2$ is printed.

Restrictions
(1) $M \leq 10$
(2) $N < 512$

Time Time to read program + 0·4N seconds for M = 6.

Remarks $\bar{y}_j$ and $y_j - \bar{y}_j$ are punched to 1 decimal place; $\bar{y}_j$ has up to 5 figures before the decimal point and $y_j - \bar{y}_j$ has up to 3.

The program is specifically intended for integer values of $y_j < 10^5$ with residuals $< 10^3$, although higher numbers will usually be punched but with some loss of tidiness.

Z is punched in floating point decimal to 5 significant figures.

Programmer JAMES CREASE.

N.I.O. PROGRAM 46

Title Conversion of N.I.O. digitiser code* to Mercury teleprinter code.

Code Mercury PIG F.

Machine MERCURY.

Purpose To convert paper tape punched in the N.I.O. digitiser code to paper tape punched in Mercury teleprinter code, and to detect faults in the digitiser output.

Tapes (1) Program
(2) Parameter and data tape.

Parameter and data tape This must start with the parameter M (see Method) in Mercury teleprinter code, followed immediately by CR.LF and several rows of blank tape; the parameter may be punched by hand on to the beginning of the data tape.

The data follows; it will be in the N.I.O. digitiser code (see Method). Any data before the first comma is ignored, i.e. only data following the first comma is converted; therefore there must not be any spurious commas before the first relevant comma.

End indication The last row of data must be immediately followed by at least eleven consecutive erases; these may begin anywhere in a block, but as any data following the last comma will be ignored, these rows of data may be made into erases if desired. There must not be any blank tape between the data and the erases.

Operation With key 9 (bottom row of hand keys on console) up, read in the program tape; it stops(there is no hoot). Put the parameter and data tape in the reader and prepulse; it reads one number (the parameter M) and stops(no hoot); prepulse to read in data. It reads and punches alternately until it reaches the erases at the end of the data, when it punches out a fault list and stops. The next parameter and data tape may now be put into the reader and the sequence repeated.

Output (1) Title

(2) 40 rows of blank tape
converted data
40 rows of blank tape
6 rows of erases.

The output is 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 (i.e. 8 blocks or lines).

(3) 40 rows of blank tape
M
fault list
40 rows of blank tape
6 rows of erases
40 rows of blank tape

The fault list is 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 of output

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

(2) Faults.
One line contains:
FS CR LF CR SP block no. SP SP SP fault no. SP SP
The list ends with CR LF LF.

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 $\left(\frac{\text{no. of terms}}{8} + \frac{F}{2}\right)$ secs.

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 N.I.O. digitiser 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.

The program can detect seven 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 there may still be one entry; either the last entry from the previous data tape may be repeated, or a fault 4 for block 0 may be given, where the first converted number has been taken from either zero or the last number of the previous data tape.

The fault list is as follows:

Fault 1 $0 \leq R \leq 11$

The R digits are replaced by 4 sets of rows of blank tape and spaces, where each set contains: one space, three rows of blank tape, two spaces. The 4 sets make 1 block.

Fault 2 $19 \leq R < 24$ and $R \geq 26$

The R digits are replaced by 8 sets of: one space, three rows of blank tape, two spaces. The 8 sets make 2 blocks. The block number given in the fault list is that of the second block.

Fault 3

If one or more of the digits of a number are not punched in the N.I.O. digitiser code, the three digits comprising the number are replaced by 3 blank rows.

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 $\nabla N_j = N_j - N_{j-1}$. Fault 3 can produce fault 4 twice.

Fault 5 $R = 25$

It is assumed that there is a mis-punched comma between two blocks, and conversion proceeds normally for the two blocks. The block number given in the fault list is that of the first block.

Fault 6 $R = 24$

It is assumed that there is a comma missing between two blocks, and conversion proceeds as for fault 5. The block number given in the fault list is that of the first block.

Fault 7 $13 \leq R \leq 18$

As in fault 1.

NB The three rows of blank tape are to enable a corrected number to be punched on to the blank tape by hand. When printed out (before correcting) the tape will produce an irregular listing if there are any faults of this type.

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

Tape code	Meaning (this is a Watt's Reflected Decimal digit)
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.
For decoding details see "Method".

(2) Fault 1 detects the last comma in a block of thirteen characters, where the last one is not itself a comma, and replaces the characters between this comma and that which terminated the previous block by 1 block of spaces and blank tape. It does not look any further back for commas, and it starts the next block from this comma.

(3) If a comma occurs other than as the last row in any valid block (i.e. thirteen rows of which the last is a comma) it will be treated as a fault 3.

(4) Faults in any blocks after the 1023rd will be given a block number which has been reduced by 1024.

(5) This program replaces N.I.O. Program 25

Programmer JAMES CREASE

* For the purpose of this program, the N.I.O. digitiser code referred to is Watt's Reflected Decimal Code.

N.I.O. PROGRAMS 47 and 47/A

Title Least squares fitting of tracks.

N.I.O. Program 47 was used until May 1963, when it was superseded by N.I.O. Program 47/A, of which the final version was used from 18th July 1963; this final version is the one described here, except for Notes.

Code Mercury CHLF 2/3/4.

Machine MERCURY.

Purpose To take a series of positions specified by two coordinates in a rectangular coordinate system at a series of irregular time intervals and fit a polynomial curve to each coordinate separately as a function of time. The components of velocity and the vector sum of the components are required, together with the radius of curvature of the track and various residuals.

Tapes (1) Program tape, which must end with the usual end of tape marker, CR LF -> CR LF.

(2) Parameter and data tape, with a title preceding each block of parameters and data.

Parameter and data tape Each block of data, representing the data for one float, consists of a title, parameters, and data.

The title must start with CR LF and letter shift; it ends at the next CR, allowing one line of title to be printed; it may contain alphanumeric characters but must not include the character π (which would be confused with CR by the computer).

After the title the three parameters are punched, each parameter being on a separate line. The parameters are:

(1) N; the number of times t_i for which x_i, y_i are available, i.e. the number of sets of data.

(2) K; the degree of polynomial to be fitted to the x data.

(3) O; K + O is the degree of polynomial to be fitted to the y data.

The data immediately follows the parameters. The data t_i (hours) x_i (miles), and y_i (miles) are punched one set per line with two spaces between terms, and in that order. The numbers may be in either fixed or floating point form. After the last number of the data there must not be a CR punched until either (1) the CR immediately preceding the next title, or (2) the CR immediately preceding the LF -> CR LF, i.e. the end of tape marker; there may be characters other than CR (or π) if required, such as LF and space.

New blocks of data, each consisting of title, three parameters, and data, may follow on.

All tapes must finish with the end of tape marker and erases to indicate the end of the tape. New data tapes may follow.

Operation (1) Read in the program tape. The machine asks for data.

(2) Read in the parameter and data tape. The machine reads in one block of title, parameters and data; it computes and punches. It then reads another block, and so on; it asks for more data when it reaches the arrow at the end of the tape, and a new parameter and data tape may be read in.

<u>Output</u> Title, followed by three sections of output:

(1) x data.

(a) coefficients of all the least squares polynomials up to the one specified by parameter K.

(b) (i) the contribution to the total sum of squares from each degree up to K.

(ii) the maximum residual and its serial number for each polynomial up to K.

(c) (i) number of degrees of freedom.

(ii) residual sum of squares.

(iii) check value on (ii).

(iv) mean square, i.e. the variance of the observations about the fitted curve.

(2) y data.

Same display as for x data except that the maximum degree of polynomial is $K + 0$.

(3) data combining (1) and (2).

(a) time (hours).

(b) residual $x_i - \bar{x}_i$ (miles), i.e. observed value - predicted value, where $\bar{x}_i$ is computed from polynomial of degree K at time t_i.

(c) residual $y_i - \bar{y}_i$ (miles).

(d) x-velocity component (cm/sec), computed by differentiating the polynomial for x at time t_i.

(e) standard deviation (cm/sec) of computed x-velocity at time t_i.

(f) similar to (d) for y coordinate.

(g) similar to (e) for y coordinate.

(h) speed (cm/sec) - vector sum of (d) and (f).

(i) direction (degrees) of velocity - assuming x components to be westwards and y components northwards.

(j) radius of curvature of track (miles); this will not be printed if K and $K + 0 \leqslant 1$.

<u>Restrictions</u> (1) $0 \leqslant K,\ K + 0 \leqslant 6$

(2) $N \leqslant 30$

<u>Time</u> About 1 minute to read in program and about $\frac{1}{2}$ minute to read, compute and punch the data for one float.

<u>Method</u> This is basically that of RAE 109/$\bar{A}$ using orthogonal polynomials (cf. FORSYTHE, J.S.I.A.M. 1957 p.74). The only extra point to be noted here is with regard to the standard deviation of the velocity estimates, which is obtained by differentiating the orthogonal polynomials and estimating the standard deviation from this derived series.

Full mathematical notes are kept in Computing Folder 1, with other N.I.O. computing notes. For further details see J. CREASE.

Notes N.I.O. 47, which was used until May 1963, had slightly different versions of Output (3), (b) and (c) as follows:

Output (3)(b) residual $\bar{x}_i - x_i$ (miles), i.e. predicted value - observed value.

Output (3)(c) residual $\bar{y}_i - y_i$ (miles).

Programmer JAMES CREASE.

N.I.O. PROGRAM 48

Title Tidal analysis and prediction.

Code Mercury CHLF 3/4. **Machine** MERCURY

Purpose

(1) The analysis of a year's hourly observed values of tidal height into 63 tidal constituents.

(2) The prediction of values of tidal height for a specified period and specified interval between predictions.

(3) The calculation of residuals (observations - predictions) for a specified period and interval.

The program attempts to deal in succession with data from various ports and years with as little duplication of data and computation as can be arranged.

Tapes

(1) Program tape (a BINAC version of CHLF 3/4 exists).

(2) Channel 1 data.

(3) Channel 2 data; this must have at least 6" of blank tape as leader.

Data

Channel 1 data (basic data)

This is in two parts; part 1 only is required if no analysis of observations is being made, otherwise parts 1 and 2 are required.

Different sets of channel 1 data are required for the analysis of records of different duration. At present channel 1 data exists for observations of 8521 and 8857 hours duration; the data tapes are labelled N = 8520 and N = 8856 respectively, and copies of these tapes are kept at N.I.O. and R.A.E. A list of the items contained on these tapes is included in Notes.

Channel 2 data summary - for details see below.

Channel 2 data consists of all or some of items 1-14 listed below:

ITEM (1) Title. This starts with at least 6" of blank tape, followed by CR.LF.LF.LF., followed by alphanumeric data and terminated by a further CR.LF.

" (2) p - an entry parameter.

" (3) ℓ - an analysis parameter.

" (4) Observations for analysis (either 8521 or 8857 hourly observed values of tidal height).

" (5) The hour, day, month and year of the central date of the observations (or predictions if there are no observations for analysis).

" (6) x_3 - the time spacing of the observations, in hours.

" (7) h_0 - the mean sea level at the port in question.

" (8) h_i, g_i - 63 pairs of values of the tidal constants.

" (9) The hour, day, month and year of the beginning of the predictions and/or residuals.
The hour, day, month and year of the end of the predictions and/or residuals.

" (10) x_5 - the number of observations per prediction.

" (11) m - the number of observations per residual.

" (12) m′, n′ - the first and last numbers of the harmonic constituents to be used in making predictions (see Table 1 in Notes).

" (13) A series of observations from which residuals are to be computed (only required if item 4 is not given).

" (14) End of tape marker.

Notes on channel 2 data

ITEM (1) Title. Between the two CRs there may be any character except another CR, but not more than 120 characters may appear on the tape between the two CRs. This enables one line of title to be included in each set of data.

" (2) Entry parameter p. This is used either for initial entry to the program or for re-entry to the program (after finishing one set of data) to avoid unnecessary calculation.

p may have the following values:

10 - to be used only on initial entry and for any further analyses.
11 - new central date and new harmonic constants - no analysis.
12 - same central date and new harmonic constants - no analysis.
13 - same central date and same harmonic constants - no analysis.
14 - new central date and same harmonic constants - no analysis.

" (3) Analysis parameter ℓ. This determines the work to be done by the program, as follows:

(i) if analysis is required count 4
(ii) if predictions are required count 2
(iii) if residuals are required count 1

Then add together (i), (ii) and (iii) to find the value of ℓ. In tabular form this gives:

Required \ ℓ	1	2	3	4	5	6	7
analysis				✓	✓	✓	✓
predictions		✓	✓			✓	✓
residuals	✓		✓		✓		✓

" (4) Observations to be analysed. These may be punched in whatever units are convenient bearing in mind the fact that the output (predictions and/or residuals) is punched to only one decimal place of the input units; therefore, e.g. if the output is required in hundredths of a foot the input must be in tenths of a foot or smaller units.

" (5) Central date. This should be the central date of the period for which analysis is required. If no analysis is required, it sould be approximately the central date of the period for which predictions are to be made. The date must be given in full, e.g. 0 17 6 1963; the hour may be integer or half-integer.

" (6) Time spacing of observations, x_3. This must be 1 if an analysis is being made; otherwise it may be $\frac{1}{2}$ or integer.

" (7) and (8) Tidal constants. The h_0 and h_i must be in the same units as any item 13 observations, and the g_i in degrees. They are obtained either from a previous N.I.O. analysis or from the International Hydrographic Bureau (MONACO) Special Publication No.26, which lists the tidal harmonics for various ports. The harmonic constituents are listed in order in Table 1, see Notes.

" (9) Beginning and end of predictions and/or residuals. These dates should be punched in full as for item 5.

" (10) The number of observations per prediction, x_5; this may be integer or $\frac{1}{2}$.

ITEM (11) The number of observations per residual, m; this must be integer.

" (12) m′, n′. The numbers of the harmonic constituents are listed in Table 1 in Notes.

" (13) Observations. The series of observations must start at the time for which the first prediction and/or residual is required. It should finish not earlier than m - 1 observations after the last residual required; there may be more observations on the tape than actually required.

" (14) End of tape marker.
The last valid data should be followed by less than 6" of blank tape, followed by the ordinary end of tape marker (CR.LF -> CR.LF and erases); this will cause the machine to finish on an "end wail". If more than 6" of blank tape precede the end of tape marker the tape will be shot out of the reader.

If a new case is to be read in, it may be punched on the same tape provided that at least 6" of blank tape separates the two cases; it is not necessary to have the CR.LF -> CR.LF and erases after each case, only at the end of the tape.

Operation

(1) Read in the program tape in channel 1 reader. If the BINAC copy is used put up keys 2 and 9 (bottom row of keys on console) before using ITB and 'continuous' keys. When the program has read in, put keys 2 and 9 down (i.e. in the horizontal position). See note (4)
The machine will stop.

(2) Put channel 1 data in channel 1 reader.

(3) Put channel 2 data in channel 2 reader, remembering to leave at least 6" blank tape as leader.

(4) Read in data. (35 mins. to read item (4) on channel 2 for N=8521)
When all the data has read in and punching has finished the machine stops and calls for more data.

Output

Channel 1

(1) Data title	output for		$\ell \neq 1$.
(2) No. of observations for analysis	" "	p = 10	; $\ell \geq 4$.
(3) Central date	" "	p ≠ 12,13;	$\ell \geq 2$.
(4) Harmonic analysis	" "	p = 10	; $\ell \geq 4$.
(5) Starting and finishing date of predictions, followed by time interval between	" "		$\ell = 2,3,6,7$.
(6) Predictions	" "		$\ell = 2,3,6,7$.

Channel 2

(1) Data title	" "		$\ell = 1,3,5,7$.
(3) Central date	" "	p≠12,13;	$\ell = 1,3,5,7$.
(5) Starting and finishing date of residuals, followed by time interval between	" "		$\ell = 1,3,5,7$.
(6) Residuals	" "		$\ell = 1,3,5,7$.

Notes on output

(1) The harmonic constants punched are:

H to 3 d.p.
g to 1 d.p.
f to 5 d.p.
U to 1 d.p.
V to 1 d.p.

Note that H is labelled (FT); this is true only if the observations are in units of a foot, as the H has the same units as the observations.

(2) If the program is re-entered through $p = 12$ or 13 then output item 3 will not be punched but will be the same as for the previous data.

(3) For outputs which have a time spacing of 3 hours or less, values are printed eight per line with seven lines per block; when the time spacing is greater than 3 hours, say n hours, they are punched with the integral part of 24/n readings per line or 1, whichever is the greater.

Restrictions (1) p and ℓ can only appear in certain combinations, as shown in the following table, which also shows the items of channel 2 data required for each combination of p and ℓ.

NB. Items 1, 2 and 3 are always required, **also 14.**

p \ ℓ	1,3	2	4	5, 6, 7
10	5 to 13	5 to 12	4 to 6	4 to 6, 9 to 12
11	5 to 13	5 to 12	NOT ALLOWED	
12	6 to 13	6 to 12	NOT ALLOWED	
13	6, 9 to 13	6, 9 to 12	NOT ALLOWED	6, 9 to 12
14	5, 6, 9 to 13	5, 6, 9 to 12	NOT ALLOWED	

(2) Channel 1 data is required only when $p = 10$, and if $\ell < 4$ part 1 only is required.

(3) If $p = 13$, and previous data had either $p = 10$ and $\ell \geq 4$, or $p = 13$ and $\ell \geq 5$, then $\ell = 5$, 6 or 7 may be used in place of $\ell = 1$, 2 or 3. This saves time, because with these settings of p and ℓ the observations can be extracted from the auxiliary store (where they have previously been put) instead of being read in under item 13 or the channel 2 data; **in this case, no print - out of the analysis is obtained.**

(4) If $\ell = 1$ or 5 time can be saved by setting $x_5 = m$; similarly, if $\ell = 1$ more time could be saved by setting $x_5 = m = 1$, but this might require a special tape with the observations spaced at the required interval.

(5) When residuals are required the starting time (item 9 of the channel 2 data) must correspond to a time for which an observation is available either from store (after an analysis) or through item 13. In the latter case the first observation of item 13 must correspond to the starting time.

(6) Item 6 of channel 2 data (x_3) must be 1 if $\ell \geq 4$.

(7) Item 11 of channel 2 data (m) must be integer, and item 11/item 10 (m/x_5) must also be integer.

(8) All data may be punched with the usual autocode restrictions, i.e. with numbers separated by sp.sp. or CR. or '. Alphanumeric data is only allowed as item 1 of channel 2 data. It is essential that at least 6" of blank tape precede the initial channel 2 data and all subsequent cases, i.e. the tape must be put into the reader so that at least the first 6" of tape to be read are completely blank.

(9) In analyses there is a restriction on the magnitude of the observations such that

$$N^2M < 2 \cdot 65 \times 10^9 n$$

where N is the number of observations;
M is the maximum absolute value of the observations, expressed as an integer relative to the least significant figure;
n is the degree of fourier harmonic used in the analysis.

This restriction only applies to the harmonics of order 300 and greater, so

$$M < \left[\frac{300 \times 2{\cdot}65 \times 10^9}{8520^2}\right] \approx 10^4$$

This means that four significant figures of data can be dealt with meaningfully (although larger numbers will be accepted).

There is little or no practical restriction on the size of the mean or long period constituents.

Time These times are estimates, and may be very rough estimates.

(1) Reading program: CHLF 3/4 version 6 mins.
BINAC version 2 mins.

(2) Tidal analysis: $(p = 10, \ell \geq 4)$ 40 mins.

(3) Tidal prediction: $\frac{(m + 20)N}{20{,}000}$ mins.

where m = number of harmonics used in predictions,
N = number of predictions required.

(4) Tidal residuals: $\ell > 4$ $\frac{N}{1000}$ mins in addition to item 3.

$\ell < 4$ $\frac{N}{500}$ " " " " " "

where N = number of residuals required.
Thus a full year's analysis, predictions and residuals for hourly observations would take:

$$\left[2 + 40 + \frac{83 \times 8520}{20{,}000} + \frac{8520}{1000}\right] \text{mins} \approx 86 \text{ mins.}$$

A year's predictions and residuals at 3 hourly intervals would take:

$$\left[2 + \frac{83 \times 2840}{20{,}000} + \frac{2840}{500}\right] \text{mins} \approx 20 \text{ mins.}$$

Method This is essentially the same as that described in N.I.O. Internal Report No. N4 but is without those complications and restrictions which were due to the DEUCE computer. N.I.O. Program 48 has less stringent requirements on the magnitude of the observations and is more flexible.

Notes (1) The items on the channel 1 (basic) data tapes are:

Tape 1

- a number of observations - 1 (8520 or 8856).
- σ number of harmonic tidal constants (63).
- ϕ_{10} 63 values of speeds of tidal constituents in degrees/hour.
- d_i 12 values of cumulative monthly totals of days.
- b_i program constants.

Tape 2

- k number of harmonics to be analysed (81).
- s k values of the fourier harmonics.
- t number of matrices used in least squares (for details of matrices see N.I.O. Internal Reports N2, N3, and N4).
- i,j number of rows and columns of each matrix.
- e,f components of cosine and sine inverse matrices, taken alternately and row by row.

(2) This program should be adaptable to ATLAS.

(3) Table 1. Harmonic constituents.

TABLE 1

Number	Symbol	Number	Symbol	Number	Symbol
1	Sa	24	OQ_2	47	MN_4
2	SSa	25	MNS_2	48	M_4
3	Mm	26	$2N_2$	49	SN_4
4	MSf	27	μ_2	50	MS_4
5	Mf	28	N_2	51	MK_4
		29	ν_2	52	S_4
6	$2Q_1$	30	OP_2	53	SK_4
7	σ_1	31	M_2		
8	Q_1	32	MKS_2	54	$2MN_6$
9	ρ_1	33	λ_2	55	M_6
10	O_1	34	L_2	56	MSN_6
11	MP_1	35	T_2	57	$2MS_6$
12	M_1	36	S_2	58	$2MK_6$
13	χ_1	37	R_2	59	$2SM_6$
14	π_1	38	K_2	60	MSK_6
15	P_1	39	MSN_2		
16	S_1	40	KJ_2	61	M_8
17	K_1	41	$2SM_2$	62	M_{10}
18	ψ_1			63	M_{12}
19	ϕ_1	42	MO_3		
20	θ_1	43	M_3		
21	J_1	44	SO_3		
22	SO_1	45	MK_3		
23	OO_1	46	SK_3		

(4) BINAC 4 version requires 4 drums; CHLF 3/4 version may have 3 or 4 drums in use.

(5) N.I.O. Program 48/A is a special purpose program used to rearrange the matrices obtained by Diana Catton into the form required for basic (channel 1) data by N.I.O. Program 48.

Programmer JAMES CREASE