NATIONAL INSTITUTE OF OCEANOGRAPHY

WORMLEY, GODALMING, SURREY

# N. I. O. Computer Programs 2

by

DIANA CATTON

N. I. O. Internal Report No. N 2

SEPTEMBER 1961

# N.I.O. COMPUTER PROGRAMS II

by

Diana Catton

National Institute of Oceanography

# N.I.O. Programs II

13 Tide Prediction. Harmonic constituents.

14 Conversion of Loran C to latitude and longitude.

15 Reverse series.

16 Matrix tape to cards conversion.

17 Tape to cards conversion for 4-digit integers.

18 Prepare harmonic constants.

19 Pitch and roll buoy azimuth correction.

20 Scale data.

21 Separate two series.

22 Cross band spectra.

23 Tide prediction.

24 Predict harmonic constituents.

## Introduction

This report contains a collection of programs written for DEUCE Mark I between July 1960 and September 1961. It is a continuation of N.I.O. Internal Report N1, and all the general comments at the beginning of that report apply here.

All the programs except 14 and 22 are connected with the N.I.O. Tide Elimination and Prediction Scheme (see N.I.O. Internal Report N4). Program 23 is also available in a slightly modified form which works on Deuce Mark II A.

N.I.O. PROGRAM 13

Title Tide Prediction. Harmonic Constituents.

Code Deuce basic.

Purpose To obtain modified harmonic components from those calculated by N.I.O. Program 3.

Order of Cards Program (cards 0-112 and constants cards; see Parameters), basic matrices, extreme z card, $S^-$ harmonics, $S^+$ harmonics.

Parameter Cards (1) Basic matrices. Two sets of these constants are available, for N = 8856 (cards 0-275) and N = 8520 (cards 0-277), in standard Scheme B binary form. For their calculation see N.I.O. Programs 33 and 16.

(2) Extreme z card.
Columns 18-21 $z_{-n}$ + c; sign in column 17
23-26 $z_{+n}$ + c; 22
28-31 c ; 27
c must be punched even if it is zero.

Data (1) $S^-$ harmonics. This is the output from N.I.O. Program 3, with series $S^-$ as data. See Note (1).

(2) $S^+$ harmonics. As (1) but with series $S^+$ as data.

End Card None.

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

(2) Run in the parameter cards and data.
When one calculation is complete the program hoots.
After clearing the hooter operation (2) may be repeated.

Output (1) 63 cards.
Columns 17-19 i ; sign in column 17
20-26 $X_i$; 20
28-34 $Y_i$; 28
36-42 $A_i$; 36
44-48 $\phi_i$; 44

(2) 1 card.
Columns 17-26 $\alpha_o$; sign in column 17.

$X_i$, $Y_i$, $A_i$ and $\alpha_o$ are given to 3D.
$\phi_i$ is given to 2D, and $-180° \leqslant \phi_i < +180°$.

Parameters (1) If another N is used, and there are q values of u instead of 82, then (program) card 4 row 9$\alpha$ must be altered from 81 $P_{17}$ to (q-1) $P_{17}$.

(2) There are 12 "constants cards", numbered 0-11, to be inserted between cards 108 and 109 of the program. These exist complete for N = 8520 and N = 8856.

(i) Cards 0-8 (tracks 15/10-15/12). The 82 (or q) values of u are punched consecutively, in ascending order of magnitude except that u = 0 is last, as binary integers x $P_1$ in the $\alpha$ field.

(ii) Cards 9-11 (track 15/13) contain several constants which must be copied from the cards for one of the above values of N. In addition, the following parameters are necessary:-

$\omega_0$ $P_{17}$ on card 9, row 3α
$\omega_1$ $P_{17}$ 9, 5α
$\omega_2$ $P_{17}$ 9, 7α
$\omega_3$ $P_{17}$ 9, 9α
$\omega_4$ $P_{17}$ 10, 1α
$\omega_5$ $P_{17}$ 10, 3α
$\omega_6$ $P_{17}$ 10, 5α
$\omega_7$ $P_{17}$ 10, 7α
$\omega_8$ $P_{17}$ 10, 9α

$(10^3 2^{15}/N)$ $P_1$ on card 11, row 6α.

Tracks are punched in triads, the first value being punched on row 2α of the first card. The Yα row of the first card for track a/b contains

$$P_{15} + b.P_{17} + a.P_{21}$$

Restrictions
(1) $n \leqslant 4960$; that is, $N \leqslant 9920$.
(2) $1 \leqslant \omega_i \leqslant 32$, $i = 0(1)8$. There must be 9 groups.
(3) The basic matrices may vary in size but may occupy not more than 82 tracks in all.
An i x j matric occupies (ij + 4) m.c., and a new matrix starts on a fresh track.
(4) $1 \leqslant q \leqslant 96$.

Failures
(1) 4, 1-31x. Wrong matrix parameter card. This occurs if the first basic matrix card has been forgotten.
(2) There are many other failures but they all indicate machine failure. The whole program should be restarted.

Time About 10 minutes.

Method The calculation is in two stages (see Note (2)).

(1) Two sets of harmonic components are available, calculated by N.I.O. Program 3 for the two half-years. The size of DEUCE makes this division necessary, and it may also be necessary to scale the original data $z_r$, $r = -n(1) + n$, by the subtraction of a constant c. This constant, and the end values of $z_r$, (which were not used by N.I.O. Program 3) are required (see parameter card (2)).

The first stage is merely to assemble the harmonic components for the whole year. We call these

$$\alpha_u \text{ and } \beta_u, \quad \text{for } u \geqslant 0$$

where u takes the values specified in N.I.O. Program 3. For both N = 8520 and N = 8856 there are 82 values of u, including u = 0. Note that $\alpha_0$ is effectively the mean over the whole year.

The second stage is to reduce the number of harmonic components $\alpha_u$ and $\beta_u$ to the 63 values corresponding to known tidal components. The basic matrices (see Parameter card (1)) do this, by the method of Least Squares, and the final result is a table of 63 components

$$X_i,\ Y_i,\ A_i \text{ and } \phi_i,$$

where $A_i = (X_i^2 + Y_i^2)^{\frac{1}{2}}$

and $\phi_i = \arctan Y_i / X_i$, $(-180° \leqslant \phi_i < 180°)$,

for $i = 0(1)62$.

Notes (1) The following symbols refer to N.I.O. Program 3.
The parameters required are:-

$N + 1 = 83$
$n = 4428$ or $4260$
$k = 2$

The values of N and n depend on the original number of terms; the values of s are the same as u in N.I.O. Program 33, and parameter cards and s-data cards exist for the two cases already considered.

(2) A more comprehensive account of the theory is given in the N.I.O. Tide Elimination Scheme description.

N.I.O. PROGRAM 14

Title Conversion of Loran C to Latitude and Longitude.

Code Deuce basic.

Purpose Given oblique co-ordinates of a geographical position, the program finds its Latitude (North) and Longitude (West) from a table of the co-ordinate intersections with the degrees of latitude.

Order of Cards Program (cards 0-71), parameter card (1), basic data, parameter card (3), data.

Parameter Cards (1) columns 17-20 N(J)
21-24 h(J)
25-28 $\ell_o$(J)
29-32 N(M)
33-36 h(M)
37-40 $\ell_o$(M)

(2) columns 17-20 $\ell_j$ or $\ell_m$ (degrees)
21-28 $J_o$ or $M_o$ } integers
29-36 $J_{\nu-1}$ or $M_{\nu-1}$ }

(3) columns 17-20 n }
21-24 $\alpha$ } degrees
25-28 $\beta$ }

Data (1) Basic data. This is in two main groups, the J-data and the M-data. The two groups are punched in exactly the same way, so only the J-data group is described.

The J-data is divided into N(J) groups, (N ≡ N(J) for the rest of this section), each group corresponding to a degree of longitude $\ell_j$, j = 0(1)N-1, where always $\ell_j = \ell_{j+1} + 1$. Each of these N groups is preceded by a parameter card (2). The data within a group is punched two values to a card, in degrees and minutes to 3D (latitude $\lambda$), in the order which ensures that $J_o > J_{\nu-1}$, where $\nu$ is the number of terms in the group.

The layout is:-

columns 17-19 $\lambda_i$ (degrees)
20-24 $\lambda_i$ (minutes to 3D)
25-27 $\lambda_{i+1}$ (degrees)
28-32 $\lambda_{i+1}$ (minutes to 3D)

If there is an odd number of values of $\lambda$ in a group, columns 25-32 on the last card may be left blank.

The order of cards is thus

parameter card (1)
parameter card (2) for group j } j = 0(1)N(J)-1
J-data for group j }
parameter card (2) for group m } m = 0(1)N(M)-1
M-data for group m }

(2) Production data.

columns 17-24 X
25-32 Y

Both X and Y are punched to 1D.

End Card None.

**Operation**

(1) Read the program with the initial input key. It stops on 1, 2-12.
(2) Run in the basic data, with the appropriate parameter cards. The program is still on 1, 2-12.
(3) Give a single shot. The program reads the basic data and stops on 7, 7-13.
(4) Run in parameter card (3), followed by the production data. The program punches the results, hoots and stops on 1, 3-3. A single shot clears the hooter and causes the program to return to 7, 7-13. Operation (4) may then be repeated.

**Output**

columns 17-19 $\lambda$ (degrees)
20-24 $\lambda$ (minutes to 3D)
25-27 $\ell$ (degrees)
28-32 $\ell$ (minutes to 3D)

**Parameters**

None.

**Restrictions**

(1) $1 \leqslant n \leqslant 1024$.
(2) $\alpha + \epsilon \geqslant \ell \geqslant \beta - \epsilon$, where $\epsilon$ is small. See Method.
(3) $\sum_{0}^{N(J)-1} \nu_j \leqslant 1024$, $\sum_{0}^{N(M)-1} \nu_m \leqslant 1024$.
(4) $(\delta^4\lambda) \leqslant 20$ in all basic data.

**Failures**

None.

**Time**

15 values/minute, + initial data input + program. (1 or 2 minutes)

**Method Definitions**

$N(J)$ = number of groups of J-data.
$h(J)$ = interval in J; that is, $h = J_i - J_{i+1}$ (see Note (1)).
$\ell_0(J)$ = longitude connected with the first group of J-data.
The corresponding terms for the M-data are $N(M)$, $h(M)$ and $\ell_0(M)$.

In a typical group, for example a J-group
$\nu$ = number of values of J-data.
$J_0$ = first value of J.
$J_{\nu-1}$ = last value of J.

Also
n = number of pairs of terms in the production data.
$\alpha$, $\beta$ = values of longitude (in degrees) between which a given point P is expected to lie.
X, Y = Loran C co-ordinates of P.
$\ell$ = longitude of P.
$\lambda$ = latitude of P.

We are given a point P with co-ordinates X, Y in the Loran C system, and we require the latitude $\lambda$ and longitude $\ell$ of P. It is known that

$$\alpha + \epsilon \geqslant \ell \geqslant \beta - \epsilon, \quad \text{where } \epsilon \text{ is small and } \alpha = \beta + 1 \text{ (in degrees)}$$

The basic data consists of tables of latitude (by now converted to minutes) against J and latitude against M for each required degree of longitude. For any specified longitude the latitude corresponding to X may be obtained by direct interpolation in the table with argument J. Similarly the latitude corresponding to Y may be obtained from the M-tables. The arguments J and M correspond to the two sets of reference lines in the Loran C co-ordinate system.

We now consider the interpolation in detail. It falls

into 2 stages.

(1) Direct interpolation.
(2) Inverse interpolation.

Stage (1)

We first select a longitude $\phi_j$ say. This determines the group of J-data to be used. From this group we select 4 consecutive argument values $J_{i-1}$, $J_i$, $J_{i+1}$, $J_{i+2}$ so that

$$J_i \geq X > J_{i+1}$$

Lagrange 4-point interpolation is adequate provided $|\delta^2\lambda| \leq 20$, so we find the latitude $\lambda_j$ corresponding to X by the formula

$$\lambda_j = A_{-1}(q)\lambda_{i-1} + A_0(q)\lambda_i + A_1(q)\lambda_{i+1} + A_2(q)\lambda_{i+2}$$

where $\lambda_i$ corresponds to $J_i$ and so on,

and $q = \dfrac{10J_i - X}{10h_j}$ .

In general, if $\alpha$ and $\beta$ are properly selected,

$$0 \leq q < 1.$$

The A(q) are the usual Lagrange coefficients, and we assume that enough basic data is given to avoid end of table complications.

This procedure is repeated for $\phi_{j-1}$, $\phi_{j+1}$ and $\phi_{j+2}$, giving 4 values in all.

Then the whole calculation is repeated for Y, using the M-data, for the same 4 values of longitude.

Stage (2)

We have now two tables, of $\lambda_j$ and $\lambda_m$ respectively, both with the same argument; namely

| | | |
|---|---|---|
| $\phi_{j-1}$ $(= \phi_{m-1})$ | $\lambda_{j-1}$ | $\lambda_{m-1}$ |
| $\phi_j$ $(= \phi_m)$ | $\lambda_j$ | $\lambda_m$ |
| $\phi_{j+1}$ $(= \phi_{m+1})$ | $\lambda_{j+1}$ | $\lambda_{m+1}$ |
| $\phi_{j+2}$ $(= \phi_{m+2})$ | $\lambda_{j+2}$ | $\lambda_{m+2}$ |

We require p, in order to calculate $\phi$ and $\lambda$, where

$$\phi = 60(\phi_j - p) \qquad (1)$$

$$\text{and } \lambda = \lambda_j + p\Delta\lambda_j + B_2(p)(\Delta^2\lambda_{j-1} + \Delta^2\lambda_j) + \ldots\ldots\ldots \qquad (2)$$

Both $\phi$ and $\lambda$ are in minutes. Similar relations hold with j replaced by m, and p may be obtained to sufficient accuracy from the recursion

$$p_{i+1} = a + b\,B_2(p_i)$$

where $a = (\lambda_m - \lambda_j)/\gamma$,

$$b = (\Delta^2\lambda_{m-1} + \Delta^2\lambda_m - \Delta^2\lambda_{j-1} - \Delta^2\lambda_j)/\gamma,$$

and $\gamma = \Delta\lambda_j - \Delta\lambda_m$.

Differences above the second are negligible. $B_2(p)$ is the Bessel interpolation coefficient. Substitution in (1) and (2) gives $\phi$ and $\lambda$, which may then be converted back to degrees and minutes.

Notes (1) This is a special purpose program, but it may work (with or without small modifications) for:-

(i) $\phi_{j+1} - \phi_j \neq 1$, $\beta - \alpha \neq 1$.

(ii) East longitudes and/or South latitudes.

(iii) Latitude and longitude interchanged.

N.I.O. PROGRAM 15

Title Reverse Series.

Code Deuce basic.

Purpose To reverse a series term by term.

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

Parameter Card Columns 17-20 -n; sign in column 17.

Data 8 signed 4-digit numbers per card in the $\alpha$-field.

End Card None.

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

Output 8 signed 4-digit numbers per card in the $\alpha$-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 $1 \leqslant n \leqslant 7520$.

Failures None.

Time About $n/500$ minutes.

Method A series $R_1$ is converted into a series $R_2$, each with n terms, where

$$R_1 = x_0, x_1, x_2, \ldots\ldots\ldots\ldots x_{n-2}, x_{n-1}$$

$$R_2 = x_{n-1}, x_{n-2}, \ldots\ldots\ldots\ldots x_2, x_1, x_0$$

Notes (1) This program is used to prepare data for N.I.O. Programs 13 and 23; that is, the Tide Elimination Scheme.
(2) $x_0$ must be at the beginning of a card.

N.I.O. PROGRAM 16

Title — Matrix tape to cards conversion.

Code — Deuce basic.

Purpose — To convert a decimal matrix, given on tape in Mercury code, to standard Scheme B binary form on cards.

Order of input — Program (cards 0-25), parameter card, data tape.

Parameter card — Columns 17-20 m; 21-24 n; 25-28 p; 29-32 k; 33-36 d

Data — An m x n matrix, punched row by row, to d decimal places. Decimal points are ignored. Each element is preceded by

FS CR LF CR

and followed by

Sp Sp,

except that the last element in a row is followed by

Sp Sp LF.

The elements must be single length and in fixed point form. Tape in this form can be produced by the Mercury Autocode function $\phi 8$. (See Note (1))

End card — None.

Operation
(1) Read the program with the initial input key. It stops on 3, 4-16.
(2) Run in the parameter card.
(3) Stimulate the tape reader.
(4) Give a single shot. When one matrix is converted, the program stops on 3, 4-16 and operations (2), (3) and (4) may be repeated.

Output
(1) Scheme B parameter card.

$Y_\alpha$ row: $m.P_{17}$
$X_\alpha$ row: $n.P_{17}$
$0_\alpha$ row: $p.P_{17}$
$1_\alpha$ row: $k.P_{17}$

(2) Scheme B binary matrix.
The matrix is punched row by row in the $\alpha$-field, to p binary places, each row starting on the Y row of a fresh card. The last of the n elements in a row is followed by the row sum, shifted down k binary places.

Parameters — None.

Restrictions
(1) $2^{31-3d}\,|x| \leqslant 1{\cdot}07 \times 10^9$, where x is any element, regarded as an integer.
(2) $1 \leqslant m \leqslant 32$
(3) $1 \leqslant n \leqslant 31$
(4) $0 \leqslant p \leqslant 31$
(5) $0 \leqslant k \leqslant 31$
(6) $0 \leqslant d \leqslant 9$
(7) $(0 \leqslant p - 3d \leqslant 31$ if $0 \leqslant d \leqslant 3$
$(1 \leqslant p - 3d \leqslant 32$ if $4 \leqslant d \leqslant 9$

Failures — (1) 3, 4-16x with hooter. $d \geqslant 10$.

(2) 2, 9-24x. Non-numeric character read. Move the tape back to the beginning of the number and give a single shot. The program will try to read the number again.

Time — This is effectively Reading + Punching time.

Method — The following symbols are used:-

m = number of rows,
n = number of columns,
p = number of binary places in the output.
k = row sum shift. If the true row sum is S, then $2^{-k}S$ is punched.
d = number of decimal places in the input.

The program first reads the complete matrix and then punches it, preceded by a parameter card.

Notes

(1) If the decimal matrix is regarded as being punched column by column, for listing purposes, this program will produce the transpose on cards.

N.I.O. PROGRAM 17

Title — Tape to cards conversion for 4-digit integers.

Code — Deuce basic.

Purpose — To transfer decimal integers from tape to cards, punching 8 signed 4-digit integers per card.

Order of input — Program (cards 0-40), parameter card, data tape.

Parameter card — Columns 17-20 n.

Data — n signed decimal integers of not more than 4 digits, punched on paper tape in the standard Mercury code. (See Note (1)).

End card — None.

Operation
(1) Read the program with the initial input key. It stops on 2, 1-13x.
(2) Run in the parameter card.
(3) Stimulate the tape reader.
(4) Give a single shot. When one run is complete operations (2), (3) and (4) may be repeated.

Output
(1) n signed 4-digit integers, punched 8 to a card in the $\alpha$-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.
(2) Sum: Columns 18-36 $\Sigma$ ; sign in column 17,
where $\Sigma$ is the sum of the n terms.

Parameters — None.

Restrictions — $1 \leqslant n \leqslant 7424$.

Failures
(1) 1, 6-6x with hooter. $n > 7424$. Clear the hooter and give a single shot. The program returns to 2, 1-13x and a new parameter card may be read.
(2) 2, 9-24x. Non-numeric character read. Move the tape back to the beginning of the number and give a single shot. The program will try to read the number again.
(3) 1, 7-24.) Sum check failure. This is a machine fault, and the whole program should be read in again.

Time — About $\frac{n}{10}$ seconds.

Method — The program reads the data in groups of 32 numbers (the last group may contain less), calculates the sum and writes them on to the drum. It then reads the track down again, recalculates the sum and checks that it is the same as before. If it is not Failure (3) is shown. If all is well the current sum is added to the sum of all the previously read numbers. In this way the whole tape is read.
The n numbers are then punched as described in Output, followed by a single card containing the sum.

Notes
(1) This program is suitable for use with the output from NIO Program 25.

N.I.O. PROGRAM 18

Title — Prepare harmonic constants.

Code — Deuce basic.

Purpose — To prepare constants to be used to modify the harmonic components of a specified year so that they will apply to another specified year. (See N.I.O. Tide Elimination Scheme)

Order of cards — Program (cards 0-97), parameter card, data. Note that there may be no data.

Parameter card — Columns 17-21 s ; sign in column 17
22-26 h ; 22
27-31 p ; 27
32-36 N ; 32
37 - or blank.

s, h, p, N are given to 2D and are in degrees. Column 37 contains - (punching in the X row) if there is no data to follow, otherwise it must be left blank.

Data — Columns 17-19 i ; sign in column 17
20-26 $X_i$(3D) ; 20
28-34 $Y_i$(3D) ; 28
36-42 $A_i$(3D) ; 36
44-48 $\phi_i$(2D) ; 44

for i = 0(1)62. This will generally be the first 63 cards of N.I.O. Program 13 output.

End card — None.

Operation — (1) Read the program with the initial input key.
(2) Run in the parameter card and data (if any). The program alternately reads a card and punches a card, and when 63 cards have been punched it hoots and stops on 1, 1-1. A single shot clears the hooter. Operation (2) may then be repeated.

Output — Columns 17-18 i ; sign in column 17
19-25 $H_i$(3D) ; 19
26-30 $g_i$(2D) ; 26
31-36 $f_i$(5D) ; 31
37-41 $u_i$(2D) ; 37
42-46 $V_i$(2D) ; 42

for i = 0(1)62. The angles (that is, g, u and V) are in degrees and lie between 0° and 360°.

Parameters — None.

Restrictions — (1) $-999{\cdot}99° \leq s, h, p, N \leq +999{\cdot}99°$.

Failures — None.

Time — About 5 minutes.

Method — For each i, we have $f_i$, $u_i$ and $V_i$ defined separately in terms of s, h, p and N. A list of the definitions is available. (See Note (1)).
If there are no harmonic components, that is, there is no data, then

$$H_i = 0, \quad g_i = u_i + V_i \text{ for } i = 0(1)62.$$

If data is available, then

$$H_i = A_i/f_i,\ g_i = \phi_i + u_i + V_i \text{ for } i = 0(1)62$$

As usual, we have

$$A_i = \sqrt{X_i^2 + Y_i^2}\ ,\ \phi_i = \arctan Y_i/X_i, \text{ for } i = 0(1)62$$

where $X_i$ and $Y_i$ are defined in N.I.O. Program 13.

Notes (1) A list of the f, u, and V definitions may be found in:-
A.T. Doodson. The Analysis of Tidal Observations.
Phil. Trans. Roy. Soc. Series A 227 p.274.

N.I.O. PROGRAM 19

Title Pitch and roll buoy azimuth correction.

Code Deuce basic.

Purpose To obtain true pitch and roll components from the observations taken with variable axes.

Order of cards Program (cards 0-58), $\phi$ parameter card, $\phi$ data, P parameter card, P data, R parameter card, R data.

Parameter cards (1) $\phi$ parameter card.

| | | | |
|---|---|---|---|
| Columns | 17-20 | N | |
| | 21-24 | h | |
| | 25-28 | $10^3\alpha$ ; | sign in column 25 |
| | 29-32 | $10^3\beta$ ; | 29 |
| | 33-36 | $\gamma$ ; | 33 |

(2) P parameter card.

Columns 17-20 n

(3) R parameter card.

Columns 17-20 n

Data (1) $\phi$ data: 8 signed 4-digit numbers per card in the $\alpha$-field. $\phi_w$ is given to the nearest degree.

(2) P data: 8 positive 4-digit numbers per card in the $\alpha$-field.
(3) R data: 8 positive 4-digit numbers per card in the $\alpha$-field.

End card None.

Operation
(1) Read the program with the initial input key.
(2) Run in the parameter cards and data. The program punches P* and stops on 8, 8-21.
(3) Give a single shot. The program punches R*, hoots and stops on 8, 3-3.
(4) A single shot clears the hooter. Operations (2), (3) and (4) may then be repeated.

Output
(1) $P_r^*$, (r = 0(1)n-1), punched as 8 signed 4-digit numbers (see Note (3)) per card in the $\alpha$-field.
(2) $R_r^*$ (r = 0(1)n-1), punched like $P_r^*$. In each case 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) $2 \leq N \leq 480$.
(2) $|\phi_w| \leq 360$, w = 0(1)n-1.
(3) $1 \leq h \leq 364$ (see Note (1)).
(4) $1 \leq n \leq 3328$.
(5) $0 \leq P_r, R_r \leq 999$, r = 0(1)n-1.
(6) $|P_r^*|, |R_r^*| \leq 9999$ (see Note (2)).
(7) $|y| \leq 9999$.
(8) $|\alpha|, |\beta| < 2$.

Failures None.

Time About n/100 minutes.

Method For r = 0(1)n-1 we define

$P_r$, $R_r$ = observations of pitch and roll respectively.
$P_r^*$, $R_r^*$ = true values of pitch and roll.

To obtain $P_r^*$ and $R_r^*$ from $P_r$ and $R_r$ the axes must be rotated through $\phi_r^0$.

Thus $P_r^* = \gamma + \alpha(P_r - \bar{P}) \cos \phi_r - \beta(R_r - \bar{R}) \sin \phi_r$

$R_r^* = \gamma + \alpha(P_r - \bar{P}) \sin \phi_r + \beta(R_r - \bar{R}) \cos \phi_r$

where $\alpha$, $\beta$, $\gamma$ are constants depending partly on the instruments used and partly on the limitations of DEUCE, and $\bar{P}$, $\bar{R}$ are means defined in the usual way.

To obtain $\phi_r$, we define

N = number of values of $\phi$.
h = interval between consecutive values of $\phi$, assuming the interval between consecutive values of $P_r$ (or $R_r$) to be unity.

Thus $\phi_r$ is given for r = 0(h)Nh - h, and intermediate values are found from

$$\phi_r = (1 - p)\phi_{kh} + p\phi_{kh + h},$$

where hp = r - kh and k is a positive integer.

To ensure that

$$kh \leqslant r < kh + h,$$

it is best to choose N so that

$$n \leqslant hN.$$

If r > hN a value for $\phi_r$ is found by extrapolation, but it may not be reliable.

Notes (1) The division subroutine used to calculate $1/h$ is very slow for small h. It may be speeded up as follows:-

$12_{20}$ Set $2^s$ so that $2^s/h$ has 3 s.f. (integer)

$6_{28}$ Shift up (27-s) places.

At present s = 17. Time = $2^s 10^{-3}/h$ seconds

| h | $2^{17}/h$ | Time (secs) |
|---|---|---|
| 1 | 131072 | 131 = 2·2 mins |
| 50 | 2621·44 | 2·6 |
| 100 | 1310·72 | 1·3 |
| 150 | 873·81 | 0·9 |
| 200 | 655·36 | 0·7 |
| 300 | 436·90 | 0·4 |
| 364 | 360·09 | 0·4 |

(2) If $P_r^*$ and $R_r^*$ are required for input to RAE174 (correlation) then they must also satisfy

$$0 \leqslant P_r^*, R_r^* \leqslant 999.$$

(3) $P_r^*$ and $R_r^*$ have a maximum error of ±0·51 in units of the last decimal place.

N.I.O. PROGRAM 20

Title Scale data.

Code Deuce basic.

Purpose To scale 4-digit signed integers.

Order of cards Program (cards 0-34), parameter card, data.

Parameter card Columns 17-20 n
21-24 $10^4$a ; sign in column 21
25-28 c ; 25

If a = 1, columns 21-24 must be left blank.
If c = 0, columns 25-28 may be left blank.

Data 8 signed 4-digit numbers in the $\alpha$-field.

End card None.

Operation (1) Read the program with the initial input key. It stops on 1, 2-12.
(2) Run in the parameter card, followed by the data.
(3) Give a single shot. When a run is complete the program hoots and returns to 1, 2-12. The hooter may then be cleared and operations (2) and (3) repeated.

Output 8 signed 4-digit integers in the $\alpha$-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) $0 < n < 7584$.
(2) $-9999 \leq 10^4a \leq 9998$, unless a = 1.
(3) $-9999 \leq c \leq 9998$.
(4) $|ax_i + c| \leq 9999$, for all i.

Failures None.

Time About $n/8$ seconds.

Method Given the n integers $x_i$, i = 0(1)n-1, the program computes

$$ax_i + c, \quad i = 0(1)n-1,$$

where a and c are constants.

N.I.O. PROGRAM 21

Title Separate two series.

Code Deuce basic.

Purpose To separate two series of 4-digit numbers punched alternately, possibly omitting terms at the beginning and/or at the end.

Order of cards Program (cards 0-34), parameter card, data.

Parameter card Columns 17-20 n
21-24 s
25-28 t

Data 8 signed 4-digit numbers per card in the α-field. (Series x)

End card None.

Operation
(1) Read the program with the initial input key. It stops on 3, 16-1.
(2) Run in the parameter card, followed by the data. The program punches Series y and stops on 1, 4-21.
(3) Give a single shot. The program punches Series z, hoots and stops on 4, 7-7.
(4) Give a single shot. The program returns to 3, 16-1 and the hooter is cleared. Operations (2), (3) and (4) may then be repeated.

Output
(1) $y_p$, (p = 0(1)N-1) punched as 8 signed 4-digit numbers per card in the α-field.
(2) $z_p$, (p = 0(1)N-1) punched in the same way as $y_p$. In each case 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) $n \leqslant 5056$.
(2) $N = \frac{1}{2}(n-t-s) > 0$ and integral.

Failures None.

Time About n/6 seconds.

Method We are given a series of n terms $x_r$, r = 0(1)n-1, of which in general alternate terms belong to Series y and the rest belong to Series z. There may be terms at the beginning and/or end of Series x which belong to neither Series y nor Series z.

n = number of terms in Series x.
s = number of unwanted terms at the beginning of Series x.
t = number of unwanted terms at the end of Series x.
N = number of terms in Series y (and in Series z).

Clearly $N = \frac{1}{2}(n-t-s)$, where t and s must be chosen so that N is an integer.

$y_p$ and $z_p$ are defined by

$$y_p = x_{s+p},$$

$$z_p = x_{s+p+1}, \text{ where } p = 0(1)N-1.$$

N.I.O. PROGRAM 22

Title Cross band spectra.

Code Deuce basic.

Purpose To condense and combine two related sets of data.

Order of cards Program (0-76), parameter card, data series 1, data series 2, group cards (in the required order), end card.

Parameter cards (1) Parameter card.

$Y_\alpha$ row: data layout. A one in each digit (or sign) position except the last digit of a number, and zeros everywhere else. (See Data)
Columns 17-20 n.

(2) Group card(s).

Columns 17-20 N
21-24 r
25-28 $s_0$

Data 2 single length signed integers per card in the $\alpha$-field. The signs must be overpunched and the first number may begin in any of columns 18 to 47. It must not begin in column 17.

End card Columns 17-20 zeros.

Operation (1) Read the program with the initial input key. It stops on 1,9-12.
(2) Give a single shot.
(3) Run in the parameter card, followed by Series 1 and then Series 2.
(4) Run in the group card(s) followed by the end card.
When the end card has been read the program punches results as follows:-

Output (1). After punching the program stops on $[P_1]$,1,11-1.
Output (2). " " " " " " $[P_1]$,1,11-1.
Output (3). " " " " " " $[P_1]$,1,11-1.
Output (4). " " " " " " $[P_1]$,6,11-1.
Output (5). " " " " " " 1,9-12.

In the first 4 cases a single shot enables the next set of output to be punched. After Output (5) operations (2) - (4) may be repeated.
Outputs (2), (3) and (4) are punched continuously, and in outputs (1) and (5) the punch pauses between each card.

Output There are 5 sets of output.
(1) Columns 18-22 N ; sign in column 18
26-30 $\rho$ ; 26
34-38 $s_0$ or zero ; 34.

Note that $\rho = 1(1)r$, and that $s_0$ is replaced by zero in all except the first of the r groups of N terms.

(5) Columns 18-24 arctan y; sign in column 18.
This is given in radians to 5D.

(2), (3), (4) Columns 17 $b_1$ ; sign in column 17
18-24 $a_1$ ; 18
25 $b_2$ ; 25
26-32 $a_2$ ; 26

| | | |
|---|---|---|
| Columns 33 | $b_3$ ; | sign in column 33 |
| 34-40 | $a_3$ ; | 34 |
| 41 | $b_4$ ; | 41 |
| 42-48 | $a_4$ ; | 42 |

where a floating point number $c_i$ is represented by

$$c_i = a_i \, 10^{bi},$$

where $a_i$ is given to 6D and $b_i$ is an integer (see Note (2)).
The actual numbers punched are

(2) $\Sigma A_p$, $\Sigma B_p$, $\Sigma \alpha_p$, $\Sigma \beta_p$.

(3) $10^{-9}\Sigma A_p\alpha_p$, $10^{-9}\Sigma B_p\beta_p$, $10^{-9}\Sigma A_p\beta_p$, $10^{-9}\Sigma B_p\alpha_p$.

(4) $\Sigma I_p$, $10^{-9}\Sigma I^2{}_p$, $\Sigma i_p$, $10^{-9}\Sigma i^2{}_p$.

The same plugboard will list outputs (1) - (5).

**Parameters** None.

**Restrictions**

(1) $0 < n \leq 1024$
(2) $0 < N \leq 32$
(3) $\Sigma N \leq n$
(4) $0 < m \leq 84$
(5) $1 \leq 2M^2N \leq 10^{18}$, where x is any of A, B, $\alpha$, $\beta$ and $|x| \leq M$ (see Note (2)).

**Failures**

(1) 1, 5-5 $n > 1024$ Give a single shot to read a new parameter card.
(2) 7, 6-6 $N > 32$
(3) 7, 7-7 $\Sigma N > n$
(4) 7, 8-8 $m > 84$
If any of (2), (3) or (4) occur, give a single shot to read a new group card.
(5) 1, 3-3
(6) 1, 4-4
Failures (5) and (6) mean that the program has been asked to store data on the drum in a non existent mc (> 8191). They should never occur, but if they do, a machine fault is indicated, and the program should be restarted.
All these failures are accompanied by the hooter.
(7) Overflow. This may occur in Outputs (2), (3), (4) if Restriction (5) is violated. (See Note (2)).

**Time** About $\left(\frac{n}{30} + \frac{N}{10}\right)$ minutes, assuming $n \doteqdot rN$, + time to reset card numbers etc. between groups.

**Method**

n = number of data cards in each series.
N = number of terms in a group.
m = number of groups.
r = number of (consecutive) times the group of N terms is to be repeated.
$s_o$ = the label of the first term of the first group specified.

$A_p$, $B_p$, p = 0(1)n-1 are the terms of Series 1. $A_p$ and $B_p$ are punched on one card.
Similarly $\alpha_p$, $\beta_p$ are the terms of Series 2.

For each group the program computes

$$\Sigma A_p, \ \Sigma B_p, \ \Sigma \alpha_p, \ \Sigma \beta_p$$

$10^{-9}\Sigma A_p\alpha_p$, $10^{-9}\Sigma B_p\beta_p$, $10^{-9}\Sigma A_p\beta_p$, $10^{-9}\Sigma B_p\alpha_p$,

$\Sigma I_p$, $10^{-9}\Sigma I^2{}_p$, $\Sigma i_p$, $10^{-9}\Sigma i^2{}_p$

where all the sums are taken over N terms, and where

$$I_p = (A_p{}^2 + B_p{}^2)^{\frac{1}{2}}$$

and $i_p = (\alpha_p{}^2 + \beta_p{}^2)^{\frac{1}{2}}$.

The program also computes arctan $\gamma$,

where $\gamma = \Sigma(A_p\beta_p - B_p\alpha_p) \,/\, \Sigma(A_p\alpha_p + B_p\beta_p)$.

The first group starts at p = 0.

Notes

(1) This program supersedes N.I.O. Program 2.

(2) Restriction (5)
The punch subroutine (P 26) requires that the floating point number $c_i = a_i 10^{b_i}$ satisfies $-9 \leqslant b_i \leqslant 9$, $-10 < a_i < +10$; that is $10^{-10} < |c_i| < 10^{+10}$ or $c_i = 0$.

Let $|A| \leqslant M$, $|B| \leqslant M$, $|\alpha| \leqslant M$, $|\beta| \leqslant M$ in any one group of n terms. Then for this group

$\Sigma A \leqslant NM$, $10^{-9}\Sigma A\alpha \leqslant 10^{-9}NM^2$

$\Sigma I \leqslant NM\sqrt{2}$, $10^{-9}\Sigma I^2 \leqslant 10^{-9}N.2M^2$

and similarly for the corresponding terms in B, $\alpha$, and $\beta$. The restriction $c_i > 10^{-10}$ is necessarily satisfied (except when $c_i = 0$) since A, B, $\alpha$, $\beta$ are all integers. The restriction $c_i < 10^{10}$ is satisfied if both

$NM\sqrt{2} < 10^{10}$ and $2NM^2 < 10^{19}$ are satisfied. If we assume N = 32 the safe limit is $M < 2{\cdot}2 \times 10^8$. This is not unreasonable as for single length working we already require $M < 2^{31}-1 < 2{\cdot}14 \times 10^9$.

N.I.O. PROGRAM 23

Title Tide Prediction.

Code Deuce basic.

Purpose To predict tidal heights, and to calculate the residuals if observed heights are available.

Order of cards Program (cards 0-61), harmonic components, range card, $S^-$ parameter card, $S^-$, $S^+$ parameter card, $S^+$.

Parameter cards (1) Range card.

Columns 17-26 a
27-36 b
37-46 c

(See restrictions and note (1)).

(2) $S^-$ parameter card.

Columns 17-20 n + 1.

(3) $S^+$ parameter card.

Columns 17-20 n + 1.

Data (1) Harmonic components. There are 64 cards, which are either the complete N.I.O. Program 13 output or the complete N.I.O. Program 24 output (63 cards) followed by a hand punched card containing $\alpha_o$. (See Note (2)).

Columns 17-19 i ; sign in column 17
20-26 $X_i$ ; 20
28-34 $Y_i$ ; 28

for cards 0-62, and

Columns 17-26 $\alpha_o$ ; sign in column 17

for card 63.

(2) $S^-$; $Z_o$, $Z_{-1}$, ..... $Z_{-n}$ punched consecutively as 8 signed 4-digit integers per card in the $\alpha$-field. (See Note (3)).

(3) $S^+$; $Z_o$, $Z_1$, ..... $Z_{+n}$ punched consecutively as 8 signed 4-digit integers per card in the $\alpha$-field. (See Note (3)).

End card None.

Operation (1) Read the program with the initial input key. It stops on 7, 8-12.

(2) Give a single shot. Run in the harmonic components (64 cards) followed by the range card. When $\Sigma_r$ has been calculated and stored for the required range the program stops on 3, 16-1.

(3) Run in the $S^-$ parameter card, followed by $S^-$. $T_r$ and $\Sigma_r$ are punched for r = -c (b) - a and the program stops on 3, 16-1.

(4) Run in the $S^+$ parameter card, followed by $S^+$. $T_r$ and $\Sigma_r$ are punched for r = a + b (b) c. Note that the last 8 columns on the last card punched contain rubbish. (See Note (1)). The program hoots and stops on 1, 6-24.

(5) Give a single shot to clear the hooter and return the

program to 7, 8-12. Operations (2) to (5) may then be repeated.

**Output** 8 signed 4-digit integers per card in the α-field. $T_r$ and $\Sigma_r$ are punched alternately for r = -c (b) - a and r = a + b (b) c. During output:-

before operation (3) set column 11 switch to 9, and
before operation (4) set column 11 switch to 1.

**Parameters** None.

**Restrictions**
(1) $c > a \geqslant 0$.
(2) $m/4$ must be an integer, where $bm = c - a + b$.
(3) $4 \leqslant m \leqslant 1216$.
(4) $1 \leqslant n \leqslant 4448$.
(5) $0 \leqslant v_i \leqslant 180°$, all i ($v_i$ in degrees).
(6) $0 \leqslant r \leqslant 2^{15}-1 = 32767$.
(7) $\alpha_o \geqslant 0$.

**Failures**
(1) 1, 7-7, with hooter. This means $m/4$ is not an integer. Clear the hooter, give a single shot and run in a correct range card.
(2) If m is not an integer, the program goes into a loop. Correct the range card, and restart the whole program (the last program card may be used).

**Time** About (7·7m + 0·034N) seconds, where N = 2n.

**Method** The notation corresponds with the N.I.O. Tide Prediction Scheme. For any r,

$Z_r$ = the observed tidal height.
$\Sigma_r$ = the calculated tidal height.

and

$$T_r = Z_r - \Sigma_r.$$

Now $Z_r$ exists for r = 0(1)n only, so when r > n, $\Sigma_r$ gives a prediction. $T_r$ is the residual.

We define

$$\Sigma_r = \alpha_o + \sum_{i=0}^{62} \left\{ X_i \cos (rv_i)° + Y_i \sin (rv_i)° \right\}$$

where the constants $v_i$ are stored within the program.

**Notes** (1) The N.I.O. Tide Prediction scheme requires $T_r$ and $\Sigma_r$ at 3-hourly intervals, from data punched at hourly intervals, so here b = 3. The two chief cases are:-

N = 8856 ; r = -4429 (3) + 4429
N = 8520 ; r = -4261 (3) + 4261.

Inspection of restriction (3) shows that at least two range cards are necessary. The most satisfactory arrangement at present (taking into account the possible unreliability of DEUCE) is to carry out five runs of about 45 minutes each. The range cards are:-

(i) a = 0, c = 897
(ii) a = 888, c = 1785
(iii) a = 1776, c = 2673
(iv) a = 2664, c = 3561
(v) a = 3552, c = 4449.

In all cases b = 3, n = 300 and $^{m}/_{4}$ = 75. In the $S^{-}$ case the runs overlap by one whole card, and in the $S^{+}$ case there is still an overlap of one card, but the last two numbers on the last card of a pack do not agree with the correct values.

(2) If predicted harmonic components are used, the value of $\alpha_0$ may be assumed to be the same as the $\alpha_0$ for the year from which the prediction was made.

(3) If N.I.O. Program 23 is to be used purely for prediction, and no data ($Z_r$) is available, two artificial series $S^{-}$ and $S^{+}$ must be constructed as follows:-

$S^{-}$ parameter card: put n = 0, thus implying 1 term only in $S^{-}$;
$S^{-}$ itself consists of 1 card, with zeros punched in columns 17-20.
$S^{+}$ is treated in exactly the same way as $S^{-}$.

N.I.O. PROGRAM 24

Title Predict harmonic constituents.

Code Deuce basic.

Purpose To predict the harmonic components for a specified year B by modifying the corresponding harmonic components for another year A.

Order of cards Program (cards 0-40), data (1), data (2).

Parameter card None.

Data (1) i, H, g for year A (63 cards).

| | | | |
|---|---|---|---|
| Columns 17-18 | i | ; | sign in column 17 |
| 19-25 | H(3D) | ; | 19 |
| 26-30 | g(2D) | ; | 26 |

(2) i, f′, u′, V′ for year B (63 cards).

| | | | |
|---|---|---|---|
| Columns 17-18 | i | ; | sign in column 17 |
| 31-36 | f′(5D) | ; | 31 |
| 37-41 | u′(2D) | ; | 37 |
| 42-46 | V′(2D) | ; | 42 |

Either or both sets of data may be produced by N.I.O. Program 18.
H, f′ and i are numbers and g, u′, V′ are in degrees and satisfy

$$0 \leqslant g, u', V' < 360°$$

End card None.

Operation (1) Read the program with the initial input key.
(2) Run in Data (1) followed by Data (2). The reader does not run continuously. When all the data is read, results are punched and the program hoots and stops on 1, 1-1. A single shot clears the hooter, and operation (2) may then be repeated.

Output

| | | | |
|---|---|---|---|
| Columns 17-19 | i | ; | sign in column 17 |
| 20-26 | $x_i$ | ; | 20 |
| 28-34 | $y_i$ | ; | 28 |
| 36-42 | $a_i$ | ; | 36 |
| 44-48 | $\theta_i$ | ; | 44 |

There are 63 cards, for i = 0(1)62, which correspond to the first 63 cards of N.I.O. Program 13 output. See Note (1).

Parameters None.

Restrictions (1) $0 \leqslant g, u', V' < 360°$
(2) $|H| \leqslant 999{\cdot}999$
(3) $|f'| < 3{\cdot}0$.

Failures (1) 1, 2-2x with hooter.
If this occurs during reading, it means that i on the card just read does not agree with the internal count. The cards are probably out of order. A single shot clears the hooter and allows the corrected card to be read, and then the program proceeds normally.

Time About 5 minutes.

Method

The notation used is that of N.I.O. Programs 13 and 18. For any year, N.I.O. Program 18 computes five quantities for each harmonic, namely

$$H_i,\ g_i,\ f_i,\ u_i,\ V_i \qquad \text{for } i = 0(1)62.$$

In some cases $H_i$ and $g_i$ are missing.

We consider two years A and B. Usually B will be the year immediately following A, and we assume that the data for both years comes from the same source.

We require an estimate of $x_i$, $y_i$, $a_i$, $\theta_i$ for year B from the known values $X_i$, $Y_i$, $A_i$, $\phi_i$ for year A, $(i = 0(1)62)$, and various constants. $X_i$, $Y_i$ are the amplitudes for specified harmonics, and

$$A_i = (X_i^2 + Y_i^2)^{\frac{1}{2}}, \qquad \phi_i = \text{arc tan } Y_i / X_i.$$

Similar relations hold for $x_i$, $y_i$, $a_i$, $\theta_i$.

N.I.O. Program 18 provides the absolute constants $H_i$ and $g_i$ for year A, but only provides $f_i$, $u_i$ and $V_i$ (now known as $f'_i$, $u'_i$, $V'_i$) for year B as this is all that can be obtained from the available data.

The results are computed using

$$x_i = f'_i H_i \cos (g_i - u'_i - V'_i)$$

$$y_i = f'_i H_i \sin (g_i - u'_i - V'_i)$$

$$a_i = f'_i H_i$$

$$\theta_i = g_i - u'_i - V'_i \quad (-180 \leqslant \theta_i < 180).$$

Notes

(1) If the output is to be used in the N.I.O. Tide Prediction Scheme, it must be made to correspond exactly with the output from N.I.O. Program 13; that is, a card numbered 63 and containing $\alpha_0$ to 3D must be placed at the end of the output pack.

Columns 17-26 $\alpha_0$ ; sign in column 17.