NATIONAL INSTITUTE OF OCEANOGRAPHY
WORMLEY, GODALMING, SURREY

# N.I.O. Computer Programs 12

N.I.O. Internal Report No. N.12

DECEMBER 1968

# N.I.O. COMPUTER PROGRAMS 12

edited

by

B. J. Hinde

N.I.O. Internal Report No. N.12

National Institute of Oceanography

# N.I.O. PROGRAMS 12

## 1) PROGRAMS

| | |
|---|---|
| 110 | Paper tape duplication |
| 111 | Current Meter Analysis |

## 2) SUB-PROGRAMS

| | |
|---|---|
| -6 | Tape Security System: Find a file |
| -7 | Bit setting routine |
| -8 | Three-point lineprinter plotting |
| -9 | Variable position integer printing |
| -10 | EBCDIC to BCDIC conversion table |
| -11 | Current Meter Conversion |
| -12 | 1442 Stacker select |
| -13 | BCDIC to EBCDI conversion table |
| -14 | Arctangent |

N.I.O. PROGRAM 110

Title Paper tape duplication

Name PADUP

Machine 1800

Language 1800 Assembler

Purpose To duplicate 5, 6, 7 or 8 track paper tape.

Input // JOB

// * Project No./Name/Job Title

// XEQ PADUP

*CCEND

Operation Place the paper tape to be duplicated in the reader with either runout or the first required character over the reading head. Set the punch track selector and tape guides as required. Load in the job cards (or tape) in the usual way and after the program is brought into core from disk, duplication will begin and continue to the physical end of the tape. At any time during execution, duplication may be stopped by pressing console STOP and restarted using console START without loss of characters. However, if it is required to edit two tapes together (for example) this method should not be used since, on restarting, the characters (up to 4) remaining in the buffer areas will be punched first. Instead, stop duplication by setting data switch 14 ON. Execution will cease with all characters that have been read correctly punched. To restart, set data switch 14 OFF and press console START.

Method The program uses two buffer areas in order to achieve maximum punching speed. The data switches are sensed after every four characters.

Execution Speed Slightly less than 150 characters/second.

Programmer Brian Hinde

N.I.O. PROGRAM 111

Title Current Meter Analysis

Name CRANO

Language 1800 Fortran IV

Purpose To read, calibrate and plot data from Bergen or Plessey current meters. Data is read from cards, two complete sets of readings (e.g. reading, number, current speed, direction and temperature) being punched on each card. There may be up to six readings per set and any one of the readings may be designated as rotor count, the difference between consecutive readings then being used by the program as the basis of current speed. The first reading in each set is normally the reading number, and calibrations of the form y = ax + b are applied to all the other readings. Provision is made for a dead space on the speed rotor and up to three graphs may be plotted as output (e.g. speed, direction and temperature). All results are also output to magnetic tape.

Inputs Job and data

Job Description

```
// JOB
// *Job No./Name/Title
// XEQ CRANO
*FILES(201,MK,0) where K is the magnetic tape number to
                 be used.
*LOCAL(NEXTO,STORO,FINDO),INECB
*CCEND
```

Data The data is in two sections, immediately following *CCEND:-

Part a), seven cards, is for program control.

Part b) is raw data.

a) CARD 1 Normally a blank card. (For exceptions see operators instructions).

CARD 2

```
              CC 14      22        30        38        46        54
A(1)bTObA(6)bX.XXXXXbY.YYYYYbV.VVVVVbW.WWWWWbQ.QQQQQbP.PPPPP
```

X.XXXXX a number representing calibration constant A(1) (See method)

Y.YYYYY similarly for A(2)

: :

P.PPPPP A(6)

CARD 3

```
              CC14       22        30        38        46        54
B(1)bTObB(6)bX.XXXXXbY.YYYYYbV.VVVVVbW.WWWWWbQ.QQQQQbP.PPPPP
```

identical to card 2, except that the numbers represent calibration constants B(1 to6)

CARD 4

```
          CC18    26      34      42      50      58
RNG(1)bTObRNG(6)bXXXX.XXbYYYY.YYbZZZZ.ZZbWWWW.WWbQQQQ.QQbPPPP.PP
```

XXXX.XX a positive number, representing the minimum expected value of the first calibrated result to be plotted.

YYYY.YY a positive number, of two decimal digits, representing the maximum expected value of the first calibrated result to be plotted.

VVVV.VV similar to XXXX.XX but for the second value to be plotted.

WWWW.WW identical to YYYY.YY but for the second value to be plotted.

QQQQ.QQ min. for the third value to be plotted.

PPPP.PP max. for the third value to be plotted.

N.B. When no graph is required, dummy values of 0 min. and and 100 max. should be used.

CARD 5

```
          CC 18    24    30    36    42    48
XNI(1)bTObXNI(6)bXXXXXbYYYYYbVVVVVbWWWWWbQQQQQbPPPPP
```

Each of the above numbers represent card columns on the raw data cards. They may take the value 00001 only if the columns contain data, else their values must be -1000.

| | | Series Number |
|---|---|---|
| XXXXX | raw data card columns 1 to 6 and 41 to 46 | 1 |
| YYYYY | " " " " 7 to 12 and 47 to 52 | 2 |
| VVVVV | " " " " 13 to 18 and 53 to 58 | 3 |
| WWWWW | " " " " 19 to 24 and 59 to 64 | 4 |
| QQQQQ | " " " " 25 to 30 and 65 to 70 | 5 |
| PPPPP | " " " " 31 to 36 and 71 to 76 | 6 |

CARD 6

```
CC7          20           33
INTNObXXXXXXbISENSbYYYYYYbLSTNObZZZZZZ
```

XXXXXX is the meter number and should be right justified

YYYYYY signifies the direction of meter rotation. If clockwise the EBCDIC character code for + (24640) should be used. If anti-clockwise, - (20032). [Right justified].

ZZZZZZ This is the last meter reading of the set of data. It will be found, on the last raw data card, in columns 1 to 6 or 41 to 46, depending on whether the last data card contains one or two sets of data (see section b) [Right justified].

CARD 7

```
CC3  8   12      19      26      33      41          53 55 57 59 61 63
DbXXbNbYbKbZbL(1)bWbL(2)bObL(3)bPbISRUSbQbISR(IT06)bRbSbTbUbVbW
```

XX a two digit number signifying the meters dead space. [Right justified].

Y the number of graphs required from 0 to 3.

Z may take values 1, 2, ...., 6. These are the series numbers as described in b). Any such series specified will have the similarly referenced value from the previous set of raw data subtracted from it before its value is analysed.

W,O,P may take values 0, 1, 2, ...., 6. They may only be identical for o's. Any series number used will give a plotted position for that column of analysed data. The lowest of W, O or P will be given the first two ranges as denoted on CARD 4. The second lowest, the next two etc.... The 0 is used when a graph is not required.

Q The number of columns of raw data per set (from 0 to 6). See data section b).

R,S,T,U,V,W, The series numbers of columns without data on the data on the raw data card. (Range 1 to 6)

b) The raw data cards follow CARD 7 immediately:- Each card may contain two sets of data the first in cols. 1 to 36 and the second in 41 to 76. Each set may contain a maximum of 6 digit numbers in the serial number positions 1 to 6 as described under CARD 5. Any of the cols. can be blank.

EXAMPLE The following is a typical raw data card

```
CC 1     67    1213  1819  2425  3031  36  40
   bb2473b39182bbb101bbbbbb666569bbbbbbbbbb  -  SET 1
CC 41    4647  5253  5859  6465  7071  76  80
   bb2474bb2193bb1010bbbbbb667666bbbbbbbbbb  -  SET 2
   1     2     3     4     5     6
Serial Direct- Speed  Tempera-
No.    ion            ture
```

The first seven cards would then be as follows, assuming beginning of a run and that

a) serial nos. 2 and 5 were wanted for plotting,
b) that serial no. 3 was current speed,
c) the meter number was 999,
d) the meter rotation was clockwise,
e) the number in cc41 to 46 on the last data card was 8888,
f) the meter deadspace was 77,
g) the expected range of serial no. 2 when calibrated was between 0 and 50 and serial 5 was 11 and 22,
h) calibration constants all of A(2 to 6) = 2.0
" " " " B(2 to 6) = 5.0
(see output to lineprinter)

```
CARD 1.       blank                                   b2.00000
CARD 2.A(1)bTObA(6)b1.00000b2.00000b2.00000b2.00000b2.00000
                                                      b5.00000
CARD 3.B(1)bTObB(6)b0.00000b5.00000b5.00000b5.00000b5.00000
CARD 4.RNG(1)bTObRNG(6)b0000.00b0100.00b0000.00b0050.00b0011.00
                                                      b0022.00
CARD 5.XNI(1)bTObXNI(6)b00001b00001b00001b-1000b00001b-1000
CARD 6.IMTN0b000999bISENSb024640bLSTN0b008888
CARD 7.Db77bNb2bKb3bL(1)b2bL(2)b5bL(3)b0bISRUSb4bISR(1T06)b4b6
```

## Output

### To Lineprinter

The page will be headed with the position where the results will be stored on magnetic tape. Then, the meter number will be printed. Following this, each set of calibrated results will be printed on the left of the continuous stationery, the first value being treated as the meter number and printed as an integer. (Hence, A(1) and B(1)

(CARDS 2 AND 3) should be 1 and 0 respectively). The results will be printed in ascending order of serial number (L to R) - for serial numbers see CARD 5.

Immediately following each set of results will be a graph of up to three of the variables utilising the maximum width of paper left. This graph will be headed with symbols and their ranges. At the foot of the graph the max. and min. of each range will be recorded together with their means and standard deviations. These results are based on actual values while points exceeding the ranges will be plotted on the upper or lower limits of the graph.

The output is terminated with the next available storage location on magnetic tape.

Magnetic Tape

The values as written on the lineprinter will be recorded on magnetic tape, their locations being noted on the lineprinter. The format is as follows:-

Each file is headed with the meter number in I7 format. Every set of readings after this is in I7, 1X, 5(F7·2, 1X) format until the run is terminated by an end-of-file marker being written.

Operator Instructions

Due to the fact that large amounts of information are handled, a facility is included to abort a run in an orderly fashion, by switching on data switch 14. The lineprinter graph will be finished and the magnetic file will remain open (i.e. no end-of-file marker).

To re-start the run, the first data card must be punched with

XNI(7)bXXXXXX

XXXXXX being found at the foot of the graph on the terminated run. The message on the lineprinter being,

RCbXXXXXX.

Errors: Should any errors occur they will be flagged and explained.

Execution time: Approx. 4 minutes per 100 cards (200 sets).

Method: The calibrations are performed with the equation

$Y = Ax + B$ where A and B are calibration constants and x is the raw data value [see N.I.O. subroutine -11].

Programmer: W. T. J. Slade.

N.I.O. SUB-PROGRAM -6

| | |
|---|---|
| Title | Tape Security System: Find a file |
| Name | Subroutine FILE0 |
| Machine | IBM 1800 |
| Language | 1800 Fortran IV |

**Purpose**

An execution of CALL FILE0 (N,L) will position the magnetic tape in use so that a READ or WRITE statement following will access the first record of file N. L is the record length in words and should be set to 74 for formatted Fortran I/O tapes and to 146 for unformatted tapes written on the IBM 1800. The *IOCS record should contain MAGNETIC TAPE, 1443 PRINTER.

**Modifications**

The version of FILE0 stored on disk is set up for reading odd parity tapes, 3 bytes/word, 800 b.p.i. density. By changing card FILE0035 as follows other combinations may be used:

DATA MODE/Z0000/

Always 0

Always 0

0 for odd parity
1 for even parity

0 for 3 bytes/word, 800 bpi
1 for 3 bytes/word, 200 bpi
2 for 3 bytes/word, 556 bpi
4 for 2 bytes/word, 800 bpi
5 for 2 bytes/word, 200 bpi
6 for 2 bytes/word, 556 bpi

**Restrictions**

$L \leq 146$
$N \leq 99$

**Errors**

Error Messages are printed as follows:

'WRONG LENGTH RECORD IN FILE NUMBER n'
'UNCORRECTABLE TAPE ERROR IN FILE NUMBER n'
'READ CHECKS HAVE OCCURRED IN FILE NUMBER n'

**Programmer**

B. J. Hinde

N.I.O. Sub-Program -7

| | |
|---|---|
| Title | Bit setting routine |
| Name | Subroutine MSKDO (LIST (1), N, MASK) |
| Machine | 1800 |
| Language | 1800 Assembler |
| Purpose | To set a desired bit pattern in an 1800 word. |

The result of a call to the subroutine is to place in MASK a word containing bits in positions listed in the first N elements of LIST. Bits already set in MASK are cleared.

e.g. if LIST (1) = 14, LIST (2) = 0, LIST (3) = 3, LIST (4) = 10 then

DIMENSION LIST (16)

CALL MSKDO (LIST, 4, MASK)

results in MASK having 1001 0000 0010 0010

(i.e. bits 14, 0, 3, 10 have been turned on)

while CALL MSKDO (LIST, 2, MASK)

results in 1000 0000 0000 0010

(i.e. only the first two elements of LIST have been used.)

| | |
|---|---|
| Inputs | LIST (1), N |
| Output | MASK |
| Restrictions | N ≤ 16, LIST (I) ≤ 15. |
| Programmer | J. Crease |

| | |
|---|---|
| Title | Three-point lineprinter plotting |
| Name | Subroutine POTIØ (X, Y, Z MORE) |
| Language | Fortran |
| Machine | 1800 |

**Purpose**

To compute and plot the positions, on the same x-axis, of three or less points using the lineprinter. Additional facilities are the recording of maximum and minimum and the calculation of mean and standard deviation for each set of points.

**Data**

Data is entered by the parameters X, Y, Z and MORE and also with the use of a "COMMON" area.

**a) Arguments**

X, Y and Z are the values to be plotted. They must be positive, real and accurate only to six sig. figs. The substitution of negative values in X, Y and (or) Z will inhibit the printing of the respective points until finally, with three negative values, only the axis marker (I) will be present.

MORE may take four values only 1, 2, 3 or 4.

Plain graph (no headings or means etc.)

"MORE" should be given the value 4 for every "call".

Headings and graph only

For the first "call" "MORE" must be 1 and all other arguments should take dummy values. No positions will be plotted. Subsequently the value 4 should be used.

Graph, headings and means etc.

For the first "call" "MORE" should be 1 and all other arguments should take dummy values. No positions will be plotted and there will be no effect on the means etc. For every other "call" except the last it must be 2. For the final "call" "MORE" should be 3.

**b) Common**

Rng (6), Orgin

Through these 7 variables are entered the following:-

Rng (1) and rng (2) represent respectively, the lower and upper ranges of x. Similarly rng (3) and rng (4) for y and rng (5) and rng (6) for z. All x, y and z values outside these ranges (excluding negatives) will be placed on the maximum or minimum for plotting (only). The variable ORGIN gives the character position at which the origin is to be. It can vary from 1 to 119.

**Output**

The positions are plotted by the characters +, *, O for X, Y and Z respectively. Should any points be unipositional then the + will be positioned correctly while the others will be stepped to the right. The reverse will occur only on the lineprinter 120th position. The axis is denoted by the character I. A new line must be called by the user after POTIØ has been used unless over-printing is required.

b) Headings

In every case, when headings are requested, an origin marker followed by a line of minus signs will be printed. All ranges, for headings only, are truncated to the nearest integer, therefore the heading facility should not be used if rng (1) to rng (6) are non-integer or exceed 32767. They will be shown in the headings together with the representative symbols for X, Y and Z. This is true only if orgin has a value less than 100.

c) Means etc.

These results will be printed the full width of the page. The values used to compute them, omitting negatives, will be the actual values presented and not the corrected values used for plotting.

Error Messages

There are many different error messages - all self explanatory. For example if ORGIN was -3 the warning message ORGIN TOO SMALL -3 would be printed. It should be noted that as this a subroutine, control will then be transferred to the mainline program and the execution not halted. To allow for this, at every error MORE is put equal to 100 and hence a test in the mainline program can stop the output of many unnecessary error messages.

Execution Time

The time for each line of plotted points to be processed and printed is approximately (0·8 second when full facilities are in use).

Programmer

W. T. J. Slade

N.I.O. SUB-PROGRAM -9

| | |
|---|---|
| Title | Variable position integer printing |
| Name | Subroutine INECB (INT, IELEM, LIST) |
| Machine | IBM 1800 |
| Language | 1800 Fortran IV |
| Purpose | To convert the integer INT to 6A1 format suitable for printing and to position it in the integer array LIST starting at position IELEM. |
| Data - Inputs | Data is entered through the arguments "INT", "IELEM" and "LIST". "INT" is the integer requiring variable position output, limited only by 1800 restrictions, i.e. $-32767 \leq$ INT $\leq 32767$. It must be of the form xxxxxx, leading blanks being suppressed. "IELEM" specifies the position at which the first character of "INT" is to be placed. N.B. it may be a blank. Thus, the limitations are $-4 \leq$ IELEM $\leq 115$. "LIST" is a 120 element array which will only be over written in the non-blank positions of "INT". |
| Output | Both "INT" and "IELEM" will retain their original values. The results will be given in array "LIST". Following is an example. If the complementary "LIST" array "ISET" contained the following, ............. ISET(22) = N. ISET(23) = O, ISET(24) = . ISET(25) = A, ISET(26) ="=",ISET(27) = *, ISET(28) = Z ISET(29) = ), ISET(30) = blank ISET(31) = * ......... a "CALL INECB" (-30, 24, ISET) would result in ...... ISET(22) = N, ISET(23) = O, ISET(24) = ., ISET(25) = A, ISET(26) = =, ISET(27) = -, ISET(28) = 3, ISET(29) = O, ISET(30) = blank, ISET(31) = * ......................<br><br>N.B. When printing the results, all positions affected by "INECB" should be in A1 format. |
| Restrictions | This subroutine deals only with integers. If real numbers are supplied to the "INT" argument the results will not be a truncated integer, erronious numbers will result. |
| Programmer | W. T. J. Slade |

N.I.O. SUBPROGRAM -10

| | |
|---|---|
| Title | EBCDIC to BCDIC conversion table. |
| Name | EBCDI |
| Machine | IBM 1800 |
| Language | 1800 Assembler |
| Use | The table consists of 256 characters - 128 words with two 8 bit characters per word. The seven low-order bits of the character to be converted (input character) are used as an address. The address designates the position in the table of the corresponding conversion character. The high-order bit (bit 0) of the input character designates which half of the table word is to be used. When bit 0 is 1, the left half of the word is used. When bit 0 is 0, the right half of the word is used. All dummy entries (invalid characters) contain the code for space.<br>To obtain the address of the table entry point<br>CALL EBCDI<br>The table may be used with 1800 ZIPCO<br>The table entries are defined in DPG/P/10. |
| Programmer | B. J. Hinde. |

**Title** Current Meter Conversion

**Name** Subroutine CURNT (XN, X, Y, Z, MORE)

**Machine** IBM 1800

**Language** 1800 Fortran IV

**Purpose** To convert raw current meter data into calibrated output for recording on both line-printer and magnetic tape. This sub-program is compatible with subroutine POTLO, three-point graph plotting (No. -8).

**Data Common** RNG(6), ORGIN,K,A(6),B(6),N,L(3),D,ISENS

**Arguments** XN(a seven element integer array),X,Y,Z,MORE.

**Data Entered**

a) (For analysis of meter readings). Data is entered through the "COMMON" variables A(6), B(6), D, ISENS, K and by the arguments XN(1) to XN(7).

Arrays A and B contain calibration constants for the corresponding speed, direction etc. given by XN, an integer array (see method). Any XN elements not required should be set to -1000. XN(1) is normally regarded as the reading number and printed as an integer, hence, A(1) and B(1) should take the values 1 and 0 respectively, so as to maintain the original value of XN(1). XN(7) is normally controlled by the program itself and contains the previous XN(K) value. For the first "CALL CURNT" however, XN(7) should be loaded with a value, 0, unless continuing a previous set of data. Variable D, an integer, should correspond to the meter's dead space. ISENS denotes clockwise and anti-clockwise meter rotation given respectively by 24640 and 20032 (the EBDIC character codes for + and -). Finally, K pertains to the element of XN from which successive adjacent readings are subtracted (see method).

b) (For plotting a graph - else all variables in this section may be set to 0). This data is entered through the "COMMON" area, RNG(6), N, L(3) and argument MORE. L(1) to L(3) will set the calibrated values derived from XN(L(1)), XN(L(2)) and XN(L(3)) in X, Y and Z. If one of L(N) is 0 then Z is set to -100, if two are zero then Z = Y = -100, etc. (for further details see POTLO). Variable N signifies the number of graphs required, to a maximum of 3. RNG(1) and RNG(2) represent the lower and upper ranges of X. Similarly RNG(3) and RNG(4) for Y and RNG(5) and RNG(6) for Z. If MORE is set to value 1 then conditions are produced for POTLO but no results are written to either lineprinter or magnetic tape (for use with POTLO see range of MORE's values in program description -8).

**Data Output**

Data is output through the "COMMON" variable ORGIN, an integer, and arguments X, Y and Z. (As instructed by b.) ORGIN is automatically set by CURNT for use in POTLO. X, Y and Z take calibrated meter data given by L(1) to L(3). These results are compatible with POTLO.

**Output**

The following will be printed on the lineprinter and identically on magnetic tape:-

$Y_N$ (see method) will be printed on a new line in I7 format, for the reading number, a space and for all readings, not -1000, the results will be printed in (F7·2,1X) format.

**Error Messages**

There are several different error messages, all self-explanatory. Should any fault occur then the argument MORE will be given the value 100. A test for this should be made after every "CALL" as only the mainline program can terminate the run in good order. This is especially true if POTLO is being used, since such a value entered as an argument would cause a disordered abort.

**Warning**

A "COMMON" declaration must be made in any mainline program calling this subroutine from disk. The layout of this is given under "DATA".

**Method**

Calibration is applied in the form,

$$Y_N = A(N)*XN(N) + B(N)$$

One set of raw data, determined by variable K, can have the previous term subtracted as in

$$Y_K = A(K)*(XN(K_M) - XN(K_{M-1})) + B(K)$$

or, with the alternate value of ISENS, (20032)

$$Y_K = A(K)*(XN(K_{M-1}) - XN(K_M)) + B(K).$$

**Programmer**

W. T. J. Slade

| | |
|---|---|
| Title | 1442 Stacker Select |
| Name | Subroutine STACK |
| Machine | IBM 1800 |
| Language | 1800 Assembler |
| Purpose | To stack cards in the second hopper of the 1442 card reader-punch. |
| To Use | Use of the instruction<br>CALL STACK<br>after a reading or punching operation will cause the card to be stacked in the second hopper of the 1442. Successive cards will be stacked in the first hopper. |
| Programmer | W. Slade from a subroutine written by R. K. Louden. |

N.I.O. SUBPROGRAM -13

| | |
|---|---|
| Title | BCDIC to EBCDI conversion table. |
| Name | DICEB |
| Machine | IBM 1800 |
| Language | 1800 Assembler |
| Use | The table consists of 256 characters - 128 words with two 8 bit characters per word. The seven low-order bits of the character to be converted (input character) are used as an address. The address designates the position in the table of the corresponding conversion character. The high-order bit (bit 0) of the input character designates which half of the table word is to be used. When bit 0 is 1, the left half of the word is used. When bit 0 is 0, the right half of the word is used. All dummy entries (invalid characters) contain the code for space. |

To obtain the address of the table entry point

CALL DICEB

The table may be used 1800 ZIPCO.

The table entries are defined in DPG/P/10.

| | |
|---|---|
| Programmer | B. J. Hinde |

N.I.O. SUB-PROGRAM -14

Title Arctangent

Name Function ATAN2 (x, y)

Language 1800 Fortran IV

Machine IBM 1800

Purpose To calculate the arctangent of x/y in the range $-\Pi$ to $+\Pi$; the quadrant is determined as if x and y are proportional to the sine and cosine of the angle respectively. The function is equivalent to the ATAN2 provided in the Fortran V language.

Inputs The two real arguments x and y.

Output The function returns a real result accurate to 9 decimal digits.

e.g. To compute $\tan^{-1} A/B$ and put the result in C:-

```
        .
        .
        .
        .
C = ATAN2 (A, B)
        .
        .
        .
```

Restrictions and failures

None; whenever both arguments are zero, an answer of zero is returned by the function.

Programmer Catherine Clayson