DECUS PROGRAM LIBRARY

DECUS NO.	FOCAL8-19
TITLE	LEAST SQUARES FIT TO AN EXPONENTIAL
AUTHOR	J. W. Lynn
COMPANY	Georgia Institute of Technology Atlanta, Georgia
DATE	October 22, 1968
SOURCE LANGUAGE	FOCAL

Least Squares Fit to an Exponential

DECUS Program Library Write-up — DECUS No. FOCAL8-19

Abstract

This program is used in conjunction with "FOCAL" to make the best two parameter least squares fit of

Y = A * EXP (ALPHA * X)

To the user's data.

Requirements

Storage: There are four separate programs that can handle different amounts of data

1) The data is read into the computer in two arrays, X and Y. It can handle up to 34 data points.

2) A variation on the above program assumes that the values of X are equally incremented. Only the Y values are inputed, and up to 65 points can be handled.

3) This program utilizes the data storing capabilities of the FNEW function available on some computers. The only practical limitation is run time. At least 660 data points can be stored.

4) This program is the same as program two, except that the FNEW function is again used. At least 1320 points can be handled.

Usage

Loading: "FOCAL" (with the extended functions) must be loaded in the machine and operating before this program can be read in. Refer to the "FOCAL" manual, section 1.2, for the procedure for loading "FOCAL".

Startup: After the least squares program that you have selected has been read in, Type "G" followed by a carriage return. This will initiate the program.

Restrictions (not applicable)

Description

For programs one and three: The computer will ask for the number of data points "N", the number of passes "NOP", and then the X and Y values of the data (see "Methods" for the explanation of "NOP"). For example:

```
"G" "CARRIAGE RETURN"
N "3"
NOP "4"
Y    1:".4946"  X  1:"1"
Y    2:".0532"  X  2:"2"
Y    3:".0057"  X  3:"3"
```

where the user's response are denoted by quotes. The program will then type

```
ALPHA = -0.230000E+01
A = 0.460000E+02
F Test:   0.214569E-04
```

Which are the answers desired and you are through. If more than one run is desired, then simply type "G" "RETURN" and start over again. The following is an example of an actual run with program one. The data was taken from a gamma-ray attenuation experiment.

```
G
N   21
Y       1:3198     X    1:0
Y       2:2617     X    2:2.11
Y       3:2295     X    3:4.22
Y       4:2098     X    4:6.33
Y       5:1765     X    5:8.44
Y       6:1662     X    6:10.55
Y       7:1523     X    7:12.66
Y       8:1275     X    8:14.77
Y       9:1177     X    9:16.88
Y      10:1043     X   10:18.99
Y      11:963      X   11:21.10
Y      12:854      X   12:23.21
Y      13:764      X   13:25.32
Y      14:740      X   14:27.43
Y      15:605      X   15:29.54
Y      16:553      X   16:31.65
Y      17:482      X   17:33.76
Y      18:430      X   18:35.87
Y      19:400      X   19:37.98
Y      20:345      X   20:40.09
Y      21:276      X   21:42.20
  NOP   3
```

```
ALPHA = -0.600006E-01
A = 0.311262E+04
F  TEST:     0.821105E+01
```

A is the coefficient of the exponential, and ALPHA is the exponent. The F TEST is a measure of the quality of the fit. See section 7.3.

For programs two and four: the computer will ask for the number of data points "N", then the data points; then the number of passes "NOP", "XSTART", which is the initial value of X, and finally "DELTAX" which is the increment of X. It then calculates A, ALPHA and the F test as before. For example, the following is a run of the same data with program two:

```
G
N  21
Y       1:3198
Y       2:2617
Y       3:2295
Y       4:2089
Y       5:1765
Y       6:1662
Y       7:1523
Y       8:1275
Y       9:1177
Y      10:1043
Y      11:963
Y      12:854
Y      13:764
Y      14:740
Y      15:605
Y      16:553
Y      17:482
Y      18:430
Y      19:400
Y      20:345
Y      21:276
NOP    4
XSTART  0
DELTAX  2.11
ALPHA = -0.530031E-01
A = 0.330353E+04
F  TEST:        0.247457E+01
```

Further Refinements

If we have calculated A and ALPHA once, and wish to refine the calculation further, type "GOTO 1.2", followed by a "carriage return". The computer will ask for the new value of "NOP" (and also XSTART and deltax for programs two and four). For example, if we have made three passes, (the old NOP was 3) and then wish to make two additional passes, we let NOP be equal to five.

Note: A number may be delimited by a space or a carriage return, whereas a command, such as "GOTO 1.2" must be followed by a carriage return.

As an example of a refinement, take another look at the data run on program one. NOP is equal to 3, and the answers are close, but not good enough. I respond as follows:

```
GOTO 1.2
NOP  4
ALPHA = -0.530014E-01
A = 0.295395E+04
F  TEST :        0.247537E+01
```

These answers are the "REAL" answers, that is, they correspond to what is known about the data.

If A and ALPHA have been calculated previously, and we wish to refine our calculation after having left the computer, or "BOMBED" it, etc., begin as in section 4, and follow the usual procedure as you did before (section 6) as if you are starting entirely over again. But after all the data has been typed in and the computer begins to calculate, hit the "CTRL" and "C" keys simultaneously. The computer will respond by typing a few numbers and sysmbols, followed by a line-feed, carriage return, and then an asterisk, such as

```
?  01.00  @  10.20
*
```

The control of the computer has thus been returned to the teletype. We now set the values of A, ALPHA, and INcrement equal to the values we had previously. We do this by the SET command in FOCAL. "CR" means carriage return. We type (the ASTERISK'S are the responses of the computer, and are not typed by the user).

```
*SET  A = "           "  "CR"
*SET ALPHA = "           "  " CR"
```

Where the previously calculated values of A and ALPHA are typed after the equals. (the quotes are not typed -- they signify the values of A and ALPHA). Then type

```
*SET IN = 1/(10↑NOP) "CR"
```

where NOP is the NOP used in the previous calculations. The computer is now in the situation described in 6.2.1, and we proceed as in that section.

For example, suppose we wish to make the refinement in section 6.2.1 the following day. We have calculated ALPHA = .06, A = 3112 already. Our refinement will look like

```
? 01.00 @ 11.20
*SET ALPHA = -.06
*SET A = 3112
*SET  IN = 1/(10 ↑ 3)
*G 1.2


N 21
Y       1:3198    X    1:0
Y       2:2617    X    2:2.11
Y       3:2295    X    3:4.22
Y       4:2089    X    4:6.33
Y       5:1765    X    5:8.44
Y       6:1662    X    6:10.55
Y       7:1523    X    7:12.66
Y       8:1275    X    8:14.77
Y       9:1177    X    9:16.88
Y      10:1043    X   10:18.99
Y      11:963     X   11:21.10
Y      12:854     X   12:23.21
Y      13:764     X   13:25.32
Y      14:764     X   14:27.43
Y      15:605     X   15:29.54
Y      16:553     X   16:31.65
Y      17:482     X   17:33.76
Y      18:430     X   18:35.87
Y      19:400     X   19:37.98
Y      20:345     X   20:40.09
Y      21:276     X   21:42.20
NOP    4
ALPHA = -0.530014E-01
A  =  0.295395E+04
F  TEST:  0.247537E+01
```

Note: At the very beginning of the program there is an "ERASE" command which sets all variables equal to zero. If your data has already been read into the machine, and you wish to keep the data, do not type "G" "RETURN". This command of course, does not affect FNEW variables, and hence this is one of the advantages of using the FNEW programs.

Discussion

When the standard mathematics of least squares analysis (see section 7.2) are applied to this problem, one finds that it cannot be solved in closed form, and therefore an iterative procedure is required. However, the above mathematical manipulations do enable one to write the coefficient A as a function of ALPHA (at the minimum), and therefore we need to vary only one parameter. A gaussian distribution is assumed, that is, the weighting factor for each Y is $\sqrt{Y}$. The iterative procedure is as follows: Initially ALPHA = 0, and the weighted least squares sum is calculated. Then ALPHA is incremented by -1, and the weighted sum is calculated again. The two sums are then compared, and if we are progressing in the correct direction (i.e., the least square sums are getting smaller), then we continue to increment with the same step size, and in the same direction. If the new sum is larger than the old, the increment is divided by ten, and we increment in the opposite direction. Each time the sums begin to increase, we divide the increment by ten and move in the opposite direction, thus completing a "PASS". "NOP" is the "NUMBER OF PASSES" we wish to make.

One should keep in mind that since the coefficient A is approximated by an expression which is valid only at the least squares minimum, the convergence is stronger than at first thought, since as ALPHA "IMPROVES", A "IMPROVES" simultaneously.

Least Squares Analysis

$F(X_n) = A * \exp(\alpha X_n)$

We want to minimize $\Delta = \sum_n [Y_n - F(X_n)]^2$

Where Y_n is the observed value at X_n.

$$\Delta = \sum_n [Y_n^2 - 2Y_n F(X_n) + F(X_n)^2]$$

$$= \sum_n [Y_n^2 - 2Y_n A\exp(\alpha X_n) + A^2 \exp(2\alpha X_n)]$$

$$\frac{\partial \Delta}{\partial A} = \sum_n [-2Y_n \exp(\alpha X_n) + 2A \exp(2\alpha X_n)] = 0$$

(1) $$= \sum_n [\exp(\alpha X_n)(A \exp(\alpha X_n) - Y_n)] = 0$$

$$\frac{\partial \Delta}{\partial \alpha} = \sum_n [-2Y_n A X_n \exp(\alpha X_n) + 2A^2 X_n \exp(2\alpha X_n)] = 0$$

$$= \sum_n (A\alpha X_n \exp(2\alpha X_n) - X_n \alpha Y_n \exp(\alpha X_n)] = 0$$

2) $$= \sum_n [A X_n \exp(2\alpha X_n) - X_n Y_n \exp(\alpha X_n)] = 0$$

$$A \sum_n X_n \exp(2\alpha X_n) = \sum_n X_n Y_n \exp(\alpha X_n)$$

From 2) $$A = \frac{\sum_n X_n Y_n \exp(\alpha X_n)}{\sum_n X_n \exp(2\alpha X_n)};$$

From 1) $$A \sum_n \exp(2\alpha X_n) = \sum_n Y_n \exp(\alpha X_n)$$

$$A = \frac{\sum_n Y_n \exp(\alpha X_n)}{\sum_n \exp(2\alpha X_n)};$$

Accuracy

The standard F TEST is made on the data. It is

SUM [Y (OBSERVED) - Y (CALCULATED)]^2/[Y(OBSERVED) (N-2)]^2

In brief, if the F test is less than one, then the assumed statistical deviation is greater than the real deviation, and if it is greater than one, the assumed deviation is less than the real deviation. Ideally the F test should turn out to be one. See references, Section 12.4

Format (See "Discussion", section 6)

Execution Time

The execution time depends, of course, on the number of data points being used, and the number of passes "NOP". Keep in mind that each time we make a "PASS" at the least squares minimum, we are dividing the increment by ten, so that it decreases quite rapidly. Also keep in mind that the convergence is rapid (see section 7.1). For the examples in section six with 21 points and NOP = 4 each of the programs took from three to four minutes to run, with the ones utilizing the FNEW function being slightly faster.

Program

Program Listings

The four programs are listed below:

C-FOCAL , 8/68

```
01.01    C      PROGRAM ONE -- LEAST SQUARES FIT TO AN EXPONENTIAL
01.02    E
01.10    S      IN=10; A ?N ?, !;F  J=1,N;D 3.1
01.20    A      ?NOP ?, !;S NO=1/(10↑NOP)
01.3P    S      T=0;D  10;F  J=1,N;D 3.2
01.40    I      (T1-T) 6.2 ,6.7,6.7

03.10    T      %2.0,"Y",J,":";A Y(J); T "X",J,":", ;A X(J), !
03.20    D      5;S T=T+((Y(J)-Y1) ↑ 2)/Y(J)

05.10    S      Y1=FEXP(AL*X(J))
05.20    S      Y1-A*Y1

06.20    S      IN=-IN/10
06.70    I      (FABS(IN)-NO)9.1,9.1,6.8
06.80    S      AL=AL+IN;S  T1=T;G  1.3

09.10    T      %,"ALPHA =", AL, !;D  10; T "A=", A, !;S T=0;F J=1,1,N;D 3.2
09.20    T      "F TEST:  ",T/(N-2), !!
09.30    QUIT

10.05    S      A=0;F  J=1,1,N;D  11.1
10.07    S      A2=0; S  AL=AL*2;F  J=1,N;D  11.2
10.10    S      AL=AL/2;S A=A/A2

11.10    D      5.1;S  A=A+Y1*Y(J)
11.20    D      5.1;S A2=A2+Y1
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
01.01    C      PROGRAM TWO--LEAST SQUARES FIT TO AN EXPONENTIAL
01.02    E
01.10    S      IN=10; A ?N  ?, !;F  J=1,N;D  3.1
01.20    A      ?NOP  ?, !;S  NO=1/(10↑NOP); A  ?XSTART  ?, !? DELTAX ?, !
01.30    S      T=0;D  10;F  J=1,N;D  3.2
01.40    I      (T1-T)6.2,6.7,6.7

03.10    T      %2.0,"Y",J,":";A  Y (J), !
03.20    D      5;S  T=T+((Y(J)=Y1) ↑ 2)/Y(J)

05.10    S      Y1=FEXP (AL*(XS+J*DE))
05.20    S      Y1=A*Y1

06.20    S      IN=-IN/10
06.70    I      (FABS(IN)-NO)9.1,9.1,6.8
06.80    S      AL=AL+IN;S  T1=T;G  1.3

09.10    T      %,"ALPHA =",AL, !;D  10;T "A=",A, !;S  T=0; F  J=1,1,N;D 3.2
09.20    T      "F  TEST: ",T/(N-2), !!
09.30    QUIT

10.05    S      A=0;F  J=1,1,N;D  11.1
10.07    S      A2=0;S  AL=AL *2;F  J=1,N;D  11.2
10.10    S      AL=AL/2;S A=A/A2

11.10    D      5.1;S  A=A+Y1*Y(J)
11.20    D      5.1;S  A2=A2+Y1
*
```

```
01.01    C      PROGRAM THREE--LEAST SQUARES FIT TO AN EXPONENTIAL
01.02    E
01.10    S      IN=10;A  ?N  ?, !;F  J=1,N;D 2
01.R0    A      ?NOP ?, !;S  NO=1/(10↑NOP)
01.30    S      T=0;D  10;  F  J=1, N;D  3.2
01.40    I      (T1-T) 6.2,6.7,6.7

02.10    T      %2.0, "Y",J,":";A  M;T "X",J,":";A L, !
02.20    S      L=FNEW(J,L);S  M=FNEW(N+J,M)

03.20    D      5;S  T=T+((FNEW(N+J)-Y1) ↑ 2)/FNEW(N+J)

05.10    S      Y1=FEXP(AL*FNEW(J))
05.20    S      Y1=A*Y1

06.20    S      IN=-IN/10
06.70    I      (FABS(IN)-NO)9.1,9.1,6.8
06.80    S      AL=AL+IN;S  T1=T;G 1.3

09.10    T      %,"ALPHA =",AL, !;D  10;T "A=",A, !;S  T=0;F  J=1,1,N;D  3.2
09.20    T      "F TEST: ",T/(N-2), !!
09.30    QUIT

10.05    S      A=0;F  J=1,1,N;D  11.1
10.07    S      A2=0;S  AL=AL*2;F  J=1,N;D  11.2
10.10    S      AL=AL/2;S A=A/A2

11.10    D      5.1;S A=A+Y1*FNEW(N+J)
11.20    D      5.1;S A2=A2+Y1
*@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
*
```

```
01.01    C       PROGRAM FOUR--LEAST SQUARES FIT TO AN EXPONENTIAL
01.02    E
01.10    S       IN=10;A ?N ?, !;F  J=1, N;D  3.1
01.20    A       ?NOP  ?, !;S NO=1/(10 ↑ NOP); A ?XSTART  ?, !?DELTAX ?, !
01.30    S       T=0;D  10;F  J=1, N;D  3.2
01.40    I       (T1-T) 6.2, 6.7, 6.7

03.10    T       %2.0, "Y", J, ":";A M, !;S  M=FNEW(J, M)
03.20    D       5;S  T=T+((FNEW(J)-Y1) ↑ 2)/FNEW(J)

05.10    S       Y1=FEXP(AL*(XS+J*DE))
05.20    S       Y1=A*Y1

06.20    S       IN=-IN/10
06.70    I       (FABS(IN)-NO)9.1, 9.1, 6.8
06.80    S       AL=AL+IN;S  T1=T;G  1.3

09.10    T       %, "ALPHA =", AL, !;D  10;T "A=", A, !;S  T=0;F  J=1, 1, N;D  3.2
09.20    T       "F  TEST:  ", T/(N-2), !!
09.30    QUIT

10.05    S       A=0;F  J=1, 1, N;D  11.1
10:07    S       A2=0;S  AL=AL*2;F  J=1, N;D  11.2
10.10    S       AL=AL/2;S  A=A/A2

11.10    D       5.1; S  A=A+Y1*FNEW(N+J)
11.20    D       5.1; S A2=A2+Y1
```

Diagrams

A Flow chart of program is attached. The other programs are essentially the same.

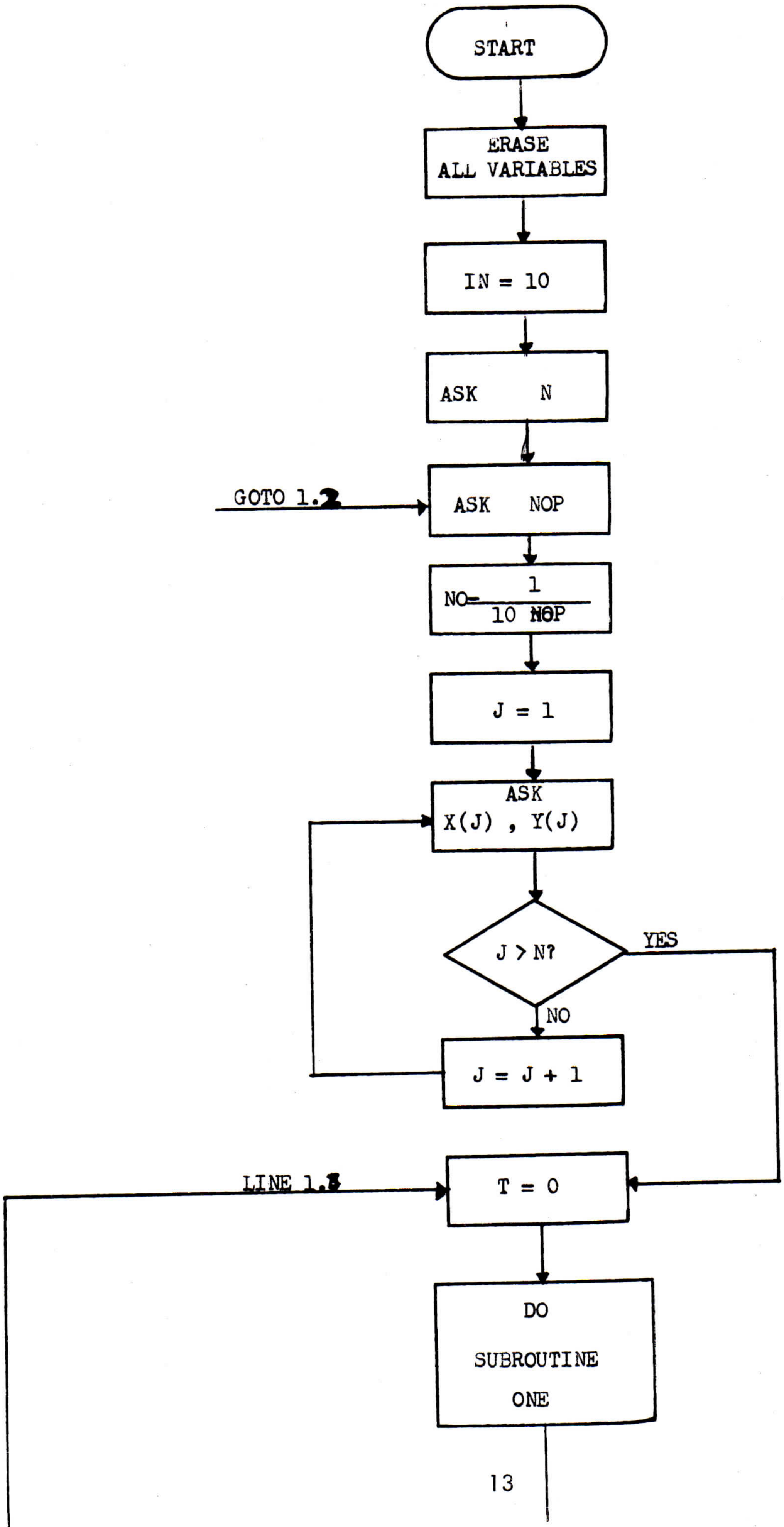
References

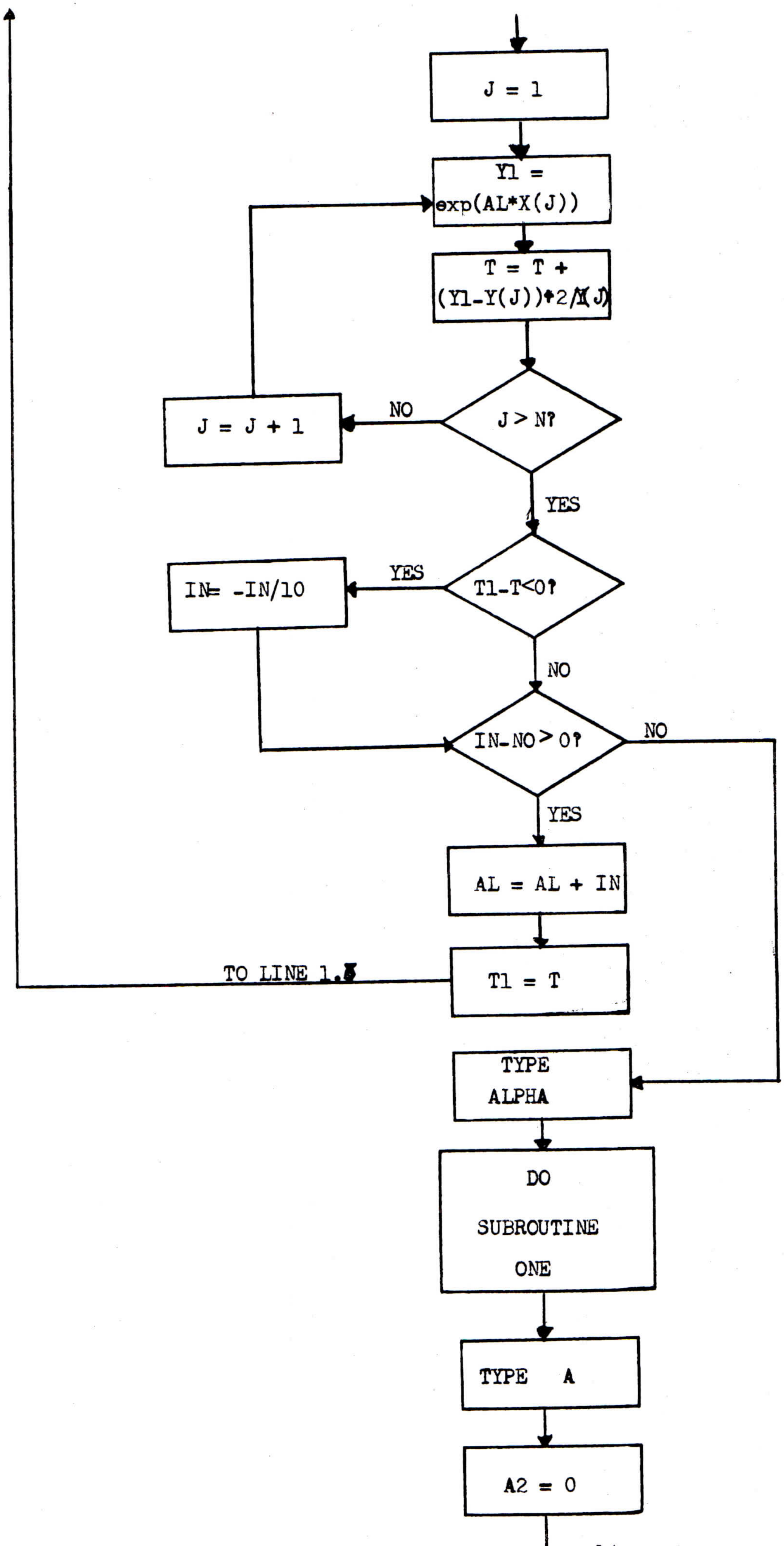
See for Example:
"Data Reduction and Error Analysis for Physicists" by Philip R. Bevington

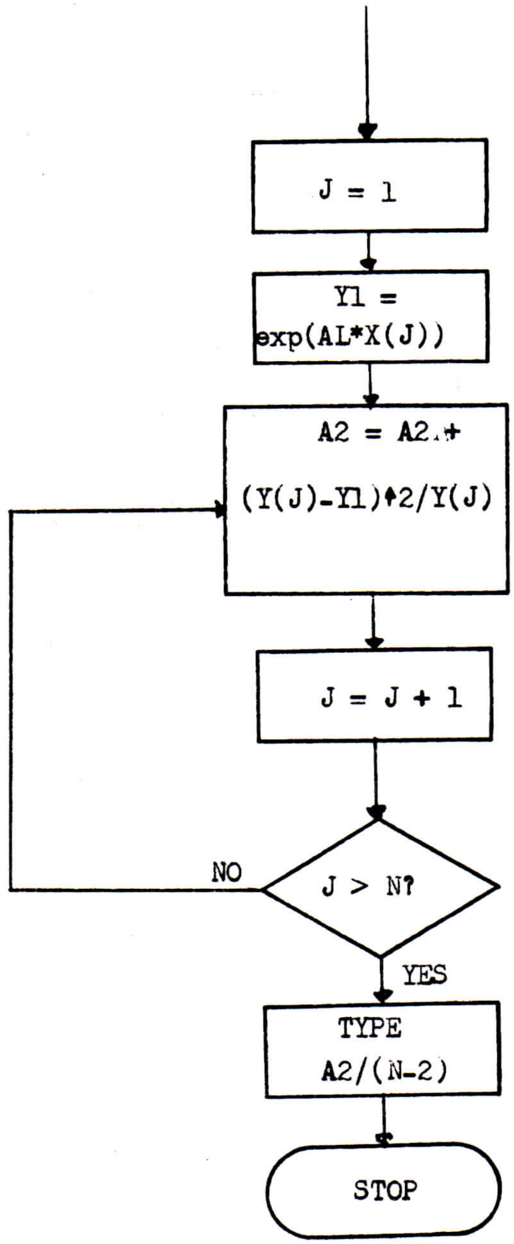
Addenda (not applicable).

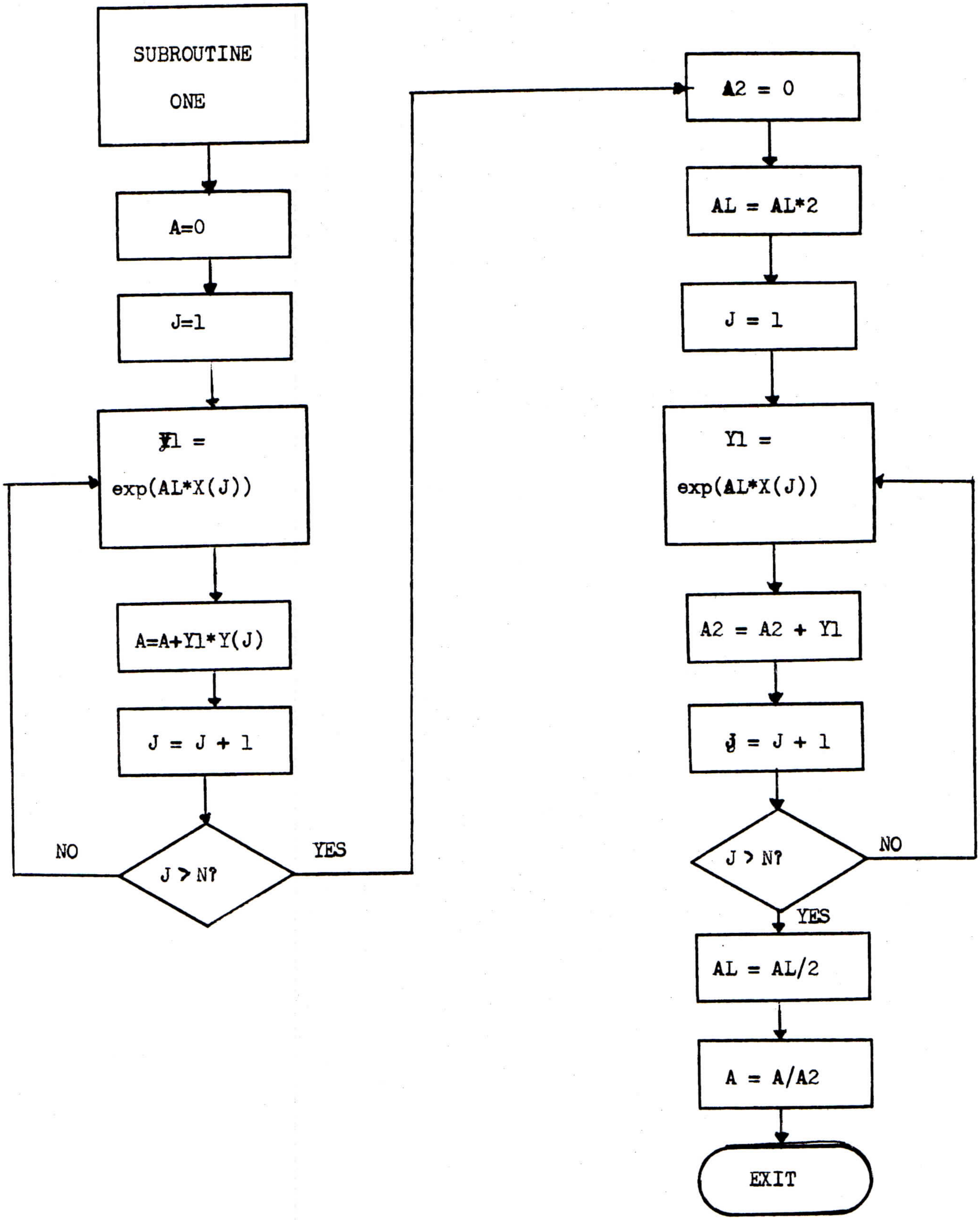
Acknowledgements (not applicable).

Errata as of October 22, 1968.


START
ERASE
ALL VARIABLES
IN = 10
ASK N
GOTO 1.2
ASK NOP
NO= 1 / 10 NOP
J = 1
ASK
X(J) , Y(J)
J > N?
YES
NO
J = J + 1
LINE 1.3
T = 0
DO
SUBROUTINE
ONE


J = 1
Y1 =
exp(AL*X(J))
T = T +
(Y1-Y(J))*2/Y(J)
J > N?
NO
J = J + 1
YES
T1-T<0?
YES
IN= -IN/10
NO
IN-NO > 0?
NO
YES
AL = AL + IN
TO LINE 1.
T1 = T
TYPE
ALPHA
DO
SUBROUTINE
ONE
TYPE A
A2 = 0


J = 1
Y1 =
exp(AL*X(J))
A2 = A2 +
(Y(J)-Y1)↑2/Y(J)
J = J + 1
J > N?
NO
YES
TYPE
A2/(N-2)
STOP


SUBROUTINE
ONE
A=0
J=1
Y1 =
exp(AL*X(J))
A=A+Y1*Y(J)
J = J + 1
NO
J > N?
YES
A2 = 0
AL = AL*2
J = 1
Y1 =
exp(AL*X(J))
A2 = A2 + Y1
J = J + 1
J > N?
NO
YES
AL = AL/2
A = A/A2
EXIT