MODEL

30

HEWLETT-PACKARD CALCULATOR

# Table of Contents

# Introduction

The Model 9830 is a unique instrument in the field of digital calculation in that it combines the best features of programmable calculators and larger computers. The Special Function Keys allow extreme ease in program operation, and we have tried to take full advantage of the operating flexibility offered by these keys in the programs in this package. As a result, these programs are more complex than most "stand-alone" programs. For example, in the program "Function Analysis", the user may define a function in which he is interested; and then, by use of the single-stroke Special Function Keys, he may cause the program to search for the minimums, maximums, and zeros of the function in a specified interval, to calculate the value of the function, its derivative, or its integral at a given point or over a specified range at given increments, and to obtain a table of these values and/or plot the results. Once a program has been mastered, we believe that the user will find his efforts in learning to operate that program amply rewarded by the increased control which he is able to exercise over the program's operations.

The programs in this package were selected as being representative of those mathematical calculations which appear to be most generally requested and used. The last section contains a collection of 28 additional functions which may be used to expand the function set available in the standard BASIC language. These programs, together with the vast selection of published BASIC programs, make the 9830 an extremely versatile calculating tool. We are planning a continuing expansion of the 9830 BASIC program library, and would greatly appreciate any suggestions, program contributions, or comments.

John H. Nairn
Model 30 Applications
Hewlett-Packard Company
Loveland, Colorado

# Commentary

In writing the programs for this package, it has been assumed that the user is familiar with the basic keyboard operations of the 9830, and also with some of the elementary programming concepts such as data entry, use of the Special Function Keys (SFK) and defined function utilization.

There are several general comments which are applicable to most of the programs in this package and which should aid the user in running those programs. These comments have been collected in this section for easy reference, and to reduce the length of the operating instructions for the individual programs. It is recommended that the user read this section before running the various programs in this package.

1. SCRATCHA should be executed prior to loading any of the programs for running. This assures that no extraneous program lines or variables are left in the memory from the previous program, which could cause program malfunction.

2. Many of the programs which operate from the Special Function Keys (SFK) require that RUN, START (where START is one of the pre-defined SFK's) be pressed to initiate program operation. It is the purpose of the START key in these programs to initialize certain variables and to dimension the arrays which the program will use. If START is pressed without previously pressing the RUN key in these programs, ERROR 41 (Array or string has not been initialized) will result.

3. Most of the programs require data input at various points throughout the running. When data entry is required, the display will show some indication of the particular data being requested, followed by a question mark. The user should enter the desired number(s) and press EXECUTE. If more than one number is requested, the user may enter the numbers one at a time, pressing EXECUTE after each entry; or he may enter all the numbers at once, separated by commas, and then press EXECUTE. If an error is made in entering the number(s), press CLEAR and enter the numbers again. The program does not accept the input until EXECUTE is pressed.

When entering lists of data (such as x and y values of data points in the Polynomial Regression program) the program sometimes must do a considerable amount of computation between entries. As a result, the user may enter the points faster than the program can process them. Be sure to wait until the question mark appears in the display before entering the next value.

4. Most of the programs give printed results of the computations that are performed. In many cases the user can control the format of these results by placing the calculator in the appropriate format mode (FIXED N, FLOAT N, or STANDARD). In addition, by placing the calculator in the PRINT ALL mode, a record of the data entered will also be printed. (Note that some programs already print out data entry. For these programs, the PRINT ALL mode should be OFF.) The user may exercise these options at his own discretion to obtain a printout which suits his needs.

Each of the programs is followed by an example case. There are some items printed in these cases (such as which key is to be pressed next) that will not appear on the printout when the program is run. These are intended to aid the user in running the example case, and the fact that the non-numeric portion of the printout obtained does not identically match the example case should not be cause for concern.

5. With the exception of the last section, Additional Functions, all of the programs in this package are written to be run from the Special Function Keys. The overlay which is provided with each of these programs labels each of the special function keys according to three basic types:

a. Program Control Keys - These keys are labeled as to their particular function and are used to run the program.

b. Internal Program Keys - These keys contain various subroutines and defined functions which are used by the program itself and should not be pressed during program execution. These keys are shaded on the SFK overlays for easy identification.

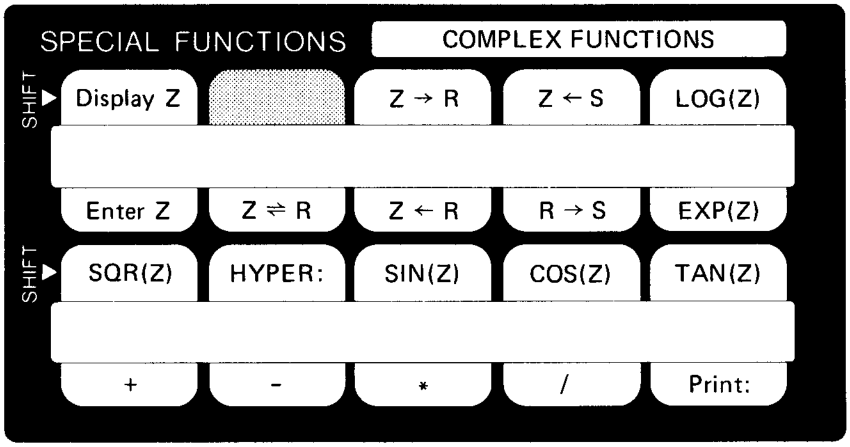
c. User Available Keys - These keys (which are blank on the overlay) are not used by the program and are available for additional user programs if memory space allows.

For the user who wishes to combine the programs in this package with programs of his own, or to make additions to these programs, there is a list of the variables used at the end of each program listing. Since the values of these variables will be altered during the running of the programs, the user should avoid their use in his own programs, if they are to be run simultaneously.

For two reasons, the user should attempt program modification with great care. Since as much as possible has been packed into each program, little or no memory space is available when running on the basic machine. Also, these programs are intricately interwoven; and unless the user fully understands the structure of the program, actual modification of the programs themselves is discouraged.

# COMPLEX FUNCTIONS

| | |
|---|---|
| DESCRIPTION: | This program transforms the special function keys into a complex calculator, allowing the user to add, subtract, multiply, and divide complex numbers. In addition, exponential and trigonometric functions of complex variables may be evaluated. |
| METHODS: | All calculations are carried out by the straight-forward solution of the defining equations (see References). |
| SPECIAL CONSIDERATIONS: | All of the trigonometric functions require that the argument of the function be given in radians, and that the calculator is in the RAD mode.<br>The logarithm function is a multivalued function. When this function is used, the principal value of the function is the calculated result. Thus, the function LOG(EXP(Z)) will not necessarily return the value Z. |
| ACKNOWLEDGMENTS: | John H. Nairn, Hewlett-Packard |
| REFERENCES: | Abramowitz, M., and Stegun, I. A., Handbook of Mathematical Functions (Washington D.C., U.S. Government Printing Office, NBS Applied Mathematics Series #55, 1964), Section 4. |
| SYSTEM SPECIFICATIONS: | 9830 (2K R/W)<br>9866 Printer or 9861 Typewriter (optional) |




**OPERATION:**

1. Type LOADKEY 1, and press EXECUTE.
2. Any combination of the available operations described below may now be used.

Note: There are three "registers" used in this program called Z, R, and S. These registers do not correspond to the BASIC variables Z, R, and S, but are merely the names of three locations where the real and imaginary parts of complex numbers may be stored.

Z: The Z-register is used to input complex numbers and is the argument for the complex functions.

R: The results of all arithmetic and functional operations are placed in the R-register.

S: The S-register may be used for the temporary storage of intermediate complex results.

**SPECIAL FUNCTIONS:**

<u>ENTER Z</u> [SF Key 0]:
When ENTER Z is pressed, the program will ask for the real part of the number to be input. Enter this number and press EXCUTE. The program will then ask for the imaginary part of the number. Enter this number and press EXECUTE.

<u>DISPLAY Z</u> [SF Key 10]:
Pressing DISPLAY Z will cause the current contents of the Z-register to be shown in the display as "X = ........... Y = ..........." where X and Y are the real and imaginary parts respectively of Z.

<u>SIN(Z)</u> [SF Key 17]:
<u>COS(Z)</u> [SF Key 18]:
<u>TAN(Z)</u> [SF Key 19]:
<u>LOG(Z)</u> [SF Key 14]:
<u>EXP(Z)</u> [SF Key 4]:
<u>SQR(Z)</u> [SF Key 15]:

These keys compute the given function of Z and place the result in the R-register. The result is also displayed, and the Z-register remains unchanged.

HYPER: [SF Key 16]:
The HYPER: key is a prefix key. When pressed before any of the three trigonometric functions, the corresponding hyperbolic trigonometric function of Z is calculated.

Z TO R [SF Key 12]:
R TO Z [SF Key 2]:
R TO S [SF Key 3]:
S TO Z [SF Key 13]:
These keys copy the results of one register into another, thus allowing operations to be chained together. For example, pressing Z to R [SF Key 12] causes the contents of the Z-register to be copied into the R-register, and Z remains unchanged.

EXCHANGE Z, R [SF Key 1]:
This key exchanges the contents of the Z- and the R-registers.

ADD [SF Key 5]:
SUBTRACT [SF Key 6]:
MULTIPLY [SF Key 7]:
DIVIDE [SF Key 8]:
When one of these keys is pressed, one of the following operations is performed:

| | | |
|---|---|---|
| ADD | R + Z | R |
| SUBTRACT | R - Z | R |
| MULTIPLY | R * Z | R |
| DIVIDE | R / Z | R |

The Z-register is unchanged.

PRINT: [SF Key 9]:
The PRINT: key is a prefix key. When pressed before any of the arithmetic or functional keys, the result of the operation is printed as well as displayed.

[SF Key 11]:
This key is used by the program and is not available to the user.

Note: The user can control the format of the printouts and displays through the use of the FIXED, FLOAT, and STANDARD keys. For general use, FIXED 5 is best for display of results and STANDARD is best for printed results.

## EXAMPLE

```
VERIFY THE IDENTIITY:  SIN(Z)↑2 + COS(Z)↑2 = 1.


ENTER Z
X(REAL)=?3
Y(IMAG)=?4

SIN(Z):     X= 3        Y= 4             R= 3.853738038        I=-27.01681326
SHIFT R TO Z
MULTIPLY
        R=-715.0569017        I=-208.2314418
SHIFT R TO S

ENTER Z
X(REAL)=?3
Y(IMAG)=?4

COS(Z):     X= 3        Y= 4             R=-27.0349456         I=-3.851153335
SHIFT R TO Z
MULTIPLY
        R= 716.0569017        I= 208.2314418

SHIFT S TO Z
ADD
        R= 1        I= 0
```

ENTER Z

```
5 RAD
10 P=H=X=Y=0
20 DISP "X(REAL)=";
30 INPUT X
40 DISP "Y(IMAG)=";
50 INPUT Y
60 END
```

DISPLAY Z

```
10 DISP "X=";X;"Y=";Y
20 END
```

Z,R EXCHANGE

```
10 X1=X
20 Y1=Y
30 X=R
40 Y=I
50 R=X1
60 I=Y1
70 END
```

SHIFT Z TO R

```
10 R=X
20 I=Y
30 END
```

SHIFT R TO Z

```
10 X=R
20 Y=I
30 END
```

SHIFT R TO S

```
10 A=R
20 B=I
30 END
```

SHIFT S TO Z

```
10 X=A
20 Y=B
30 END
```

EXP(Z)

```
10 R=EXPX*COSY
20 I=EXPX*SINY
30 IF P=0 THEN 50
40 PRINT "EXP(Z): ";
50 P=FNZ(2)
60 END
```

LOG(Z)

```
10 R=LOGSQR(X*X+Y*Y)
20 I=ATN(Y/X)
30 IF P=0 THEN 50
40 PRINT "LOG(Z): ";
50 P=FNZ(2)
60 END
```

SQR(Z)

```
10 S=SQR(X*X+Y*Y)
20 R=SQR((S+X)/2)
30 I=SQR((S-X)/2)
40 IF P=0 THEN 60
50 PRINT "SQR(Z): ";
60 P=FNZ(2)
70 END
```

HYPER:

```
10 H=(H#1)
20 END
```

SIN(Z)

```
10 K=FNZ0
20 IF H=1 THEN 80
30 R=SINX*C2
40 I=COSX*S2
50 IF P=0 THEN 120
60 PRINT "SIN(Z): ";
70 GOTO 120
80 R=COSY*S1
90 I=SINY*C1
100 IF P=0 THEN 120
110 PRINT "SINH(Z):";
120 P=FNZ(2)
130 END
```

COS(Z)

```
10 K=FNZ0
20 IF H=1 THEN 80
30 R=COSX*C2
40 I=-SINX*S2
50 IF P=0 THEN 120
60 PRINT "COS(Z): ";
70 GOTO 120
80 R=COSY*C1
90 I=SINY*S1
100 IF P=0 THEN 120
110 PRINT "COSH(Z):";
120 P=FNZ(2)
130 END
```

TAN(Z)

```
10 K=FNZ1
20 IF H=1 THEN 90
30 K=COS(2*X)+C4
40 R=SIN(2*X)/K
50 I=S4/K
60 IF P=0 THEN 140
70 PRINT "TAN(Z): ";
80 GOTO 140
90 K=COS(2*Y)+C3
100 R=S3/K
110 I=SIN(2*Y)/K
120 IF P=0 THEN 140
130 PRINT "TANH(Z):";
140 P=FNZ(2)
150 END
```

ADD

```
10 R=R+X
20 I=I+Y
30 P=FNZ(3)
40 END
```

SUBTRACT

```
10 R=R-X
20 I=I-Y
30 P=FNZ(3)
40 END
```

MULTIPLY

```
10 T=R
20 R=X*R-I*Y
```

```
30 I=X*I+Y*T
40 P=FNZ(3)
50 END

DIVIDE

10 C=X*X+Y*Y
20 T=R
30 R=(X*R+Y*I)/C
40 I=(X*I-Y*T)/C
50 P=FNZ(3)
60 END

PRINT:

10 P=(P=0)
20 END

(FNZ)

10 DEF FNZ(W)
20 IF W>1 THEN 160
30 IF W=1 THEN 110
40 DEF FNS(U)=(EXPU-1/EXPU)/2
50 DEF FNC(U)=(EXPU+1/EXPU)/2
60 S1=FNSX
70 C1=FNCX
80 S2=FNSY
90 C2=FNCY
100 RETURN 0
110 S3=FNS(2*X)
120 C3=FNC(2*X)
130 S4=FNS(2*Y)
140 C4=FNC(2*Y)
150 RETURN 0
160 H=0
170 IF P=0 THEN 210
180 IF W=3 THEN 200
190 PRINT "    X=";X;"    Y=";Y;
200 PRINT "        R=";R;"    I=";I
210 DISP "REAL=";R;"IMAG=";I;
220 RETURN H


VARIABLES USED: A, B, C, H, I, K, P, T, X, Y,
                X1,Y1,C1,C2,C3,C4,S1,S2,S3,S4
```

MODEL 30

HEWLETT · PACKARD

# LAGRANGIAN INTERPOLATION

**DESCRIPTION:**

This program takes the x and y coordinates of a set of data points, and interpolates the value of y at any given x coordinate.

**METHODS:**

When the data points are entered, the x and y coordinates of these points are stored. The user may then specify some subset of these points to be used in the calculation. When N points are specified for use in the calculation, the Lagrange method fits these points to an (N-1)th order polynomial and calculates the value of y for the given x. In general, the accuracy of the results increases with the number of data points used. The running time also increases, however, and the user should determine the optimum number of points to be used for his desired degree of accuracy vs. running time.

**SPECIAL CONSIDERATIONS:**

The 2K version of this program allows for the input of up to 50 data points. If the program is to be run on a 4K machine and more data points are required, the dimension statement in line 10 of the program on the START key [SF Key 10] may be changed to dimension X, Y, and UI to be up to 250.

**ACKNOWLEDGMENTS:**

John H. Nairn, Hewlett-Packard

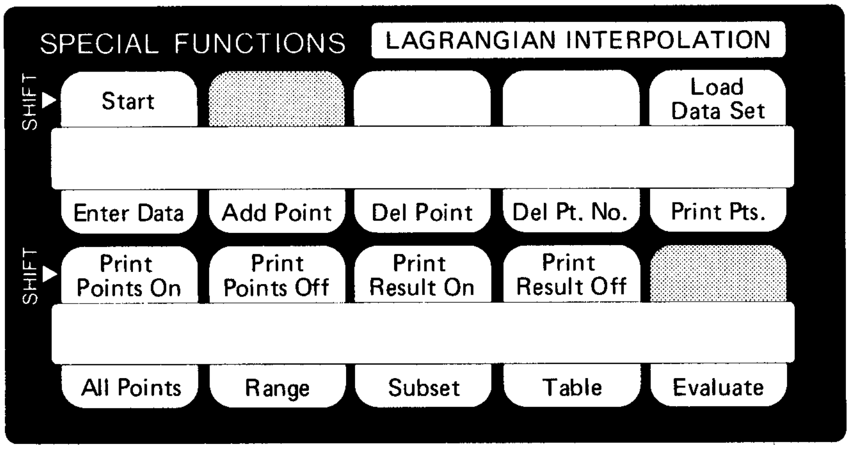
**REFERENCES:**

Abramowitz, M., and Stegun, I. A., Handbook of Mathematical Functions (Washington D.C., U.S. Government Printing Office, NBS Applied Mathematics Series #55, 1964), Section 25.2.1

**SYSTEM SPECIFICATIONS:**

9830 (2K R/W)

9866 Printer or 9861 Typewriter (optional)




OPERATION:

1. Type LOADKEY 2, and press EXECUTE.
2. Press RUN, START [SF Key 10]. Wait for the display to read "READY".
3. Set the desired print options as described below. If the program is being run without a printer or typewriter, leave all print options off.
4. Enter the data points using the special function keys described below for this purpose.
5. Specify the data points to be used as described below, and evaluate the function at any desired point.
6. Pressing the START key will initialize the program for a new set of data points.

SPECIAL FUNCTIONS:

ENTER DATA [SF Key 0]:
When the ENTER DATA key is pressed, the program will ask for the number of data points to be input. Enter this number and press EXECUTE. The program will then ask for the x and y values for each of the data points. The number of the current data point being keyed in is also displayed. For example, when reading the x-value of the third data point, the display will read "X( 3 ) = ? ". Enter each of these values, pressing EXECUTE after each entry. If an error is discovered after pressing EXECUTE, continue entering the points. After entry is complete, the list of data points may be edited using the following three keys.

ADD A POINT [SF Key 1]:
If it is desired to make an addition to the list of data points, this may be done by pressing the ADD POINT key. The program will ask for the x and y values of the point to be added. When EXECUTE is pressed, this point will be added to the list. After each addition, the point numbers will be rearranged so that the points are stored in order of increasing values of x.

DELETE POINT [SF Key 2]:
Any point on the list may be deleted by pressing DELETE POINT. The program will ask for the x-value of the point to be deleted. When EXECUTE is pressed, this point will be deleted from the list and all point numbers will be adjusted. If the point to be deleted is not on the list, the display will show "POINT NOT FOUND".

DELETE POINT NUMBER [SF Key 3]:
It may be more convenient to indicate the number of the point to be deleted than the x-value of the point itself, expecially if the x-value has a large number of digits. This key operates identically to the DELETE POINT key above, except that the point number is specified, rather than the x-value of the point.

PRINT POINTS [SF Key 4]:
At any part in the program, the PRINT POINTS key may be pressed to obtain a printed listing of the points currently stored in the memory. This listing is especially useful when deleting points by point number, and also in specifying the subset of points to be used in the calculation (see below).

LOAD DATA SET [SF Key 14]:
It may happen that the function to be interpolated is given by a rather lengthy list of data points, and that the interpolation program is to be used frequently. In this case, use of the LOAD DATA SET key will avoid the necessity of reentering the data each time the program is to be run.

The user may prepare the data tape by writing a program consisting entirely of DATA statements (using any line numbers less than 9900) with the information given in the following order: N, x(1), y(1), x(2), y(2), ... , x(N), y(N), where N is the number of data points to be read. (If the user is not familiar with the DATA statement, reference to that section of the operating manual may be helpful.)

When the LOAD DATA SET key is pressed, the program will ask for the file on which the data is located. Be sure that the tape containing that file is in the tape drive, enter the file number, and press EXECUTE.

USE ALL POINTS [SF Key 5]:
Unless one of the two options described below is used, the program will calculate the function using all of the data points which are currently stored. If this number is large, the calculation time may be quite long. For this reason, the options described below allow the user to specify some smaller number, or subset, of the entire list of data points to be used in the calculation. If it is desired to return to the use of all points on the list, the USE ALL POINTS key should be pressed.

USE RANGE [SF Key 6]:
This key allows the user to specify a starting and an ending point number on the list of data points. When the function is evaluated, only those point numbers from the starting point number to the ending point number inclusive, will be used in the calculation. This specification will remain in effect until a new specification for point usage is given.

USE SUBSET [SF Key 7]:
The user may also specify that only certain points from the list of stored data points be used in the evaluation of the function. When USE SUBSET is pressed, the program will ask for the number of points to be used, and then for the point numbers. Enter each point number to be used, pressing EXECUTE after each entry. These points will be used in all evaluations until a new set of points to be used is specified.

EVALUATE [SF Key 9]:
When the EVALUATE key is pressed, the characters "FNA" appear in the display. The value of x at which the function is to be evaluated is then entered. If x is negative, it should be enclosed in parentheses. When EXECUTE is pressed, the value of the function at the given x is evaluated and displayed.

Note: The evaluation of the function at a given x has been programmed as a function, FNA( ). Thus, once the data points have been entered and the list of points to be used has been specified, FNA may be used within an expression or as part of a program in mainline memory (see operation manual for details).

TABLE [SF Key 8]:
In addition to evaluating the function at a single point, a table of interpolated values may be printed. When the TABLE key is pressed, the program will ask for the X-MIN, X-MAX, and STEP SIZE. When these values are entered and EXECUTE is pressed, the program will print a table of values from X-MIN to X-MAX at the specified increment.

PRINT RESULTS ON [SF Key 17]:

PRINT RESULTS OFF [SF Key 18]:

These two keys are used to put the program into the on or off mode for printing results. If PRINT RESULTS is on, the values of x and of f(x) will be printed.

PRINT POINTS ON [SF Key 15]:

PRINT POINTS OFF [SF Key 16]:

These two keys are used to put the program into the on or off mode for printing the points used. If PRINT POINTS is on, the number of points used, the smallest point number and the largest point number are printed.

Note: When START is pressed, all print options are off. If a printing device is not plugged into the calculator, these print options should not be turned on, since the program would be "hung up" trying to execute the print command.

## EXAMPLE

```
GIVEN THE TABLE BELOW FOR SINH(X),
FIND THE VALUE OF SINH(1.35).

RUN, START
READY

ENTER DATA
NUMBER OF POINTS?5
X( 1     ) = ?1
Y( 1     ) = ?1.1752
X( 2     ) = ?1.25
Y( 2     ) = ?1.60192
X( 3     ) = ?1.5
Y( 3     ) = ?2.12923
X( 4     ) = ?1.75
Y( 4     ) = ?2.79041
X( 5     ) = ?2
Y( 5     ) = ?3.62686

PRINT POINTS
POINT NO.          X               Y
 1                1              1.1752
 2                1.25           1.60192
 3                1.5            2.12923
 4                1.75           2.79041
 5                2              3.62686

PRINT POINTS ON
PRINT RESULTS ON

FNA 1.35
POINTS =  5     RANGE =  1       TO    5

X= 1.35         F(X) =  1.799091408

ACTUAL:  SINH(1.35) =  1.799092635

TABLE
X-MIN,X-MAX,STEP SIZE =?1,1.5,.05
X               F(X)
 1               1.1752
 1.05            1.253821408
 1.1             1.335609328
 1.15            1.420752048
 1.2             1.509450848
 1.25            1.60192
 1.3             1.698386768
 1.35            1.799091408
 1.4             1.904287168
 1.45            2.014240288
 1.5             2.12923
```

START

```
10 DIM X[50],Y[50],U[50]
20 U=P=R=N=S7=0
25 A=B=M=U[1]=1
30 FOR I=1 TO 30
40 X[I]=Y[I]=0
50 NEXT I
60 DISP "READY"
70 END
```

ENTER DATA

```
10 DISP "NUMBER OF POINTS";
20 INPUT K
30 N=0
40 DISP "X(";N+1;") = ";
50 INPUT X
60 DISP "Y(";N+1;") = ";
70 INPUT Y
80 N=FNZ1
90 IF N<K THEN 40
100 END
```

ADD POINT

```
10 DISP "ADD: X,Y=";
20 INPUT X,Y
30 N=FNZ1
40 END
```

DELETE POINT

```
10 DISP "DELETE: X=";
20 INPUT X
30 N=FNZ2
40 END
```

DELETE POINT NUMBER

```
10 DISP "POINT NUMBER =";
20 INPUT K
30 N=FNZ3
40 END
```

PRINT POINTS

```
10 PRINT "POINT NO.";"    X";"    Y"
20 FOR I=1 TO N
30 PRINT I,X[I],Y[I]
```

```
40 NEXT I
50 PRINT
60 END
```

LOAD DATA SET

```
10 DISP "DATA FILE NO.";
20 INPUT F
30 LINK F,100,40
40 READ N
50 FOR I=1 TO N
60 READ X[I],Y[I]
70 NEXT I
80 END
```

PRINT POINTS ON

```
10 P=1
20 END
```

PRINT POINTS OFF

```
10 P=0
20 END
```

PRINT RESULT ON

```
10 R=1
20 END
```

PRINT RESULT OFF

```
10 R=0
20 END
```

USE ALL POINTS

```
10 U=0
20 END
```

USE RANGE OF POINTS

```
10 DISP "STARTING POINT NO. =";
20 INPUT A
30 DISP "ENDING POINT NO. =";
40 INPUT B
50 U=1
60 END
```

USE SUBSET OF POINTS

```
10 DISP "NO. POINTS TO USE =";
20 INPUT M
30 FOR I=1 TO M
40 DISP "POINT NO. =";
50 INPUT U[I]
60 NEXT I
70 U=2
80 END
```

TABLE

```
10 DISP "X-MIN,X-MAX,STEP SIZE =";
20 INPUT A7,B7,S7
30 PRINT "X","F(X)"
40 FOR X=A7 TO B7 STEP S7
50 PRINT X,FNAX
60 NEXT X
70 S7=0
80 DISP
90 END
```

EVALUATE

```
10 DEF FNA(Z)
20 S=0
25 Z1=Z
30 IF U=2 THEN 200
40 IF U=1 THEN 80
50 A1=1
60 B1=N
70 GOTO 100
80 A1=A
90 B1=B
100 FOR I=A1 TO B1
110 T=1
120 FOR J=A1 TO B1
130 GOSUB 400
140 NEXT J
150 S=S+T*Y[I]
160 NEXT I
170 GOTO 500
200 FOR I1=1 TO M
210 T=1
220 I=U[I1]
230 FOR J1=1 TO M
240 J=U[J1]
250 GOSUB 400
260 NEXT J1
270 S=S+T*Y[I]
280 NEXT I1
```

```
290 GOTO 500
400 IF X[I]=X[J] THEN 420
410 T=T*(Z-X[J])/(X[I]-X[J])
420 RETURN
500 IF P=0 THEN 520
510 T=FNP1
520 IF R=0 THEN 540
530 T=FNP2
540 RETURN S
```

(FNP)

```
10 DEF FNP(Z)
15 IF S7#0 THEN 100
20 IF Z=1 THEN 60
30 PRINT "X=";Z1,"F(X) = ";S
40 PRINT
50 RETURN 0
60 PRINT "POINTS = ";N*(U=0)+(B-A+1)*(U=1)+M*(U=2),
70 PRINT "RANGE = ";(U=0)+A*(U=1)+U[1]*(U=2);
80 PRINT "  TO  ";N*(U=0)+B*(U=1)+U[M]*(U=2)
90 PRINT
100 RETURN 0
```

(FNZ)

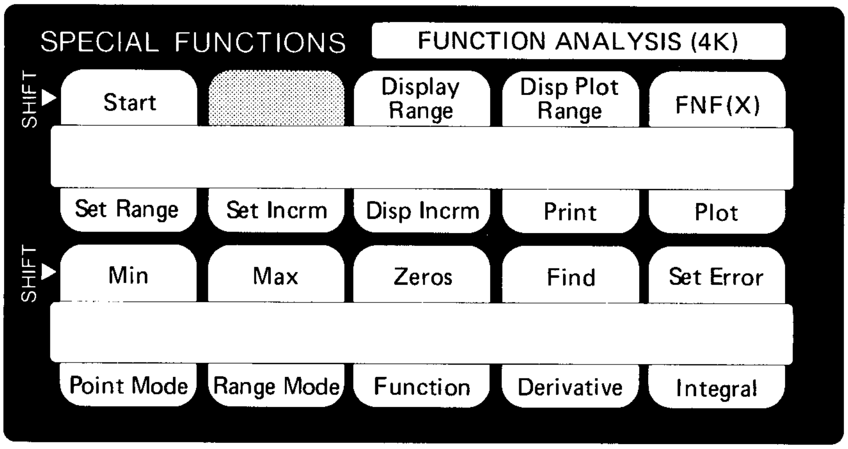
```
10 DEF FNZ(Z)
20 IF Z#1 THEN 200
30 FOR I=N TO 1 STEP -1
40 IF X<X[I] THEN 80
50 X[I+1]=X
60 Y[I+1]=Y
70 RETURN N+1
80 X[I+1]=X[I]
90 Y[I+1]=Y[I]
100 NEXT I
110 I=0
120 GOTO 50
200 IF Z#2 THEN 400
210 FOR I=1 TO N
220 IF X#X[I] THEN 250
230 GOSUB 300
240 RETURN N-1
250 NEXT I
260 DISP "POINT NOT FOUND";
270 RETURN N
300 FOR J=I TO N
310 X[J]=X[J+1]
320 Y[J]=Y[J+1]
330 NEXT J
```

```
340 RETURN
400 IF K>N THEN 260
410 I=INTABSK
420 GOTO 230

VARIABLES USED: A,B,F,I,J,K,M,N,P,R,S,T,U,X,Y
                A1,A7,B1,B7,I1,J1,S7,Z1.
```

MODEL 30

# FUNCTION ANALYSIS (4K)

| | |
|---|---|
| DESCRIPTION: | Given any function which can be defined in closed form, this program finds its maximums, minimums, and zeros over a given range of the argument. In addition, the function, its derivative, or its integral may be evaluated at a point, tabulated over a range, or plotted (with or without printed coordinate values). |
| METHODS: | The find routine for maximums, minimums, and zeros evaluates the function over the range given at the specified intervals. If the sign of the function for two successive points is different, a search is conducted for the zero in that interval. If the difference quotients for three successive points are of opposite sign, a search for a max/min is conducted in that interval.<br><br>The derivative of the function at a point is found by moving away from the given point until the change in the function is large enough to give significant results. Four equally spaced points are then used to calculate the leading differences, from which the derivative is obtained.<br><br>Integrals are obtained by Simpson's rule methods, with the sub-intervals successively halved until the error is within acceptable limits. The operator should be aware that if the error is set too small, the integral may not converge to a solution for functions that oscillate too rapidly in the range of integration. |
| SPECIAL CONSIDERATIONS: | When tabulating a function or derivative over the range, or when evaluating an integral, the range should be set so that the function does not pass through a singularity, as this may result in an error. The maximum number of points that may be tabulated for ploting is 256. |
| ACKNOWLEDGMENTS: | John H. Nairn, Hewlett-Packard |
| REFERENCES: | Kunz, Kaiser S., <u>Numerical Analysis</u> (New York: McGraw-Hill Book Company, 1957), Chapter 7. |
| SYSTEM SPECIFICATIONS: | 9830 (4K R/W)<br>9866 Printer or 9861 Typewriter |




**OPERATION:**

1. Type LOADKEY 3, and press EXECUTE.
2. Press FETCH, FNF [SF Key 14]. The display will read "KEY".
3. Define the function to be analyzed as FNF(X), either as a single- or a multi-line function.
4. Press RUN, START [SF Key 10]. The display should read "READY". If RUN was not pressed, the display will read "ERROR 41 IN LINE 2".
5. Any combination of the available operations described below may now be used.
6. To define a new function for analysis, press SCRATCH, FNF [SF Key 14] and go to step 2 above.

**SPECIAL FUNCTIONS:**

SET RANGE [SF Key 0]:
Sets X-MIN and X-MAX for other operations which require a range over X.

SET INCREMENT [SF Key 1]:
Sets the X-increment for other operations which require an increment. When SF Key 1 is pressed, the calculator will ask for the X-step size. If it is more convenient to enter the number of increments into which the range should be divided, indicate this by entering a zero for the X-step size.

DISPLAY RANGE [SF Key 12]:

DISPLAY INCREMENT [SF Key 2]:
These keys display the current range and increment that is set.

SET ERROR [SF Key 19]:
This key allows the operator to specify the allowable error in the calculating of max/min, zeros, integrals, and derivatives.
Note: When START is pressed, "default" values are given to the above variables. These are: X-MIN = 0, X-MAX = 1, NO. INCRMS. = 20, and ERROR = 1E-09.

FIND [SF Key 18]:
When FIND is pressed, the program searches for the maximums, minimums, and zeros of the function over the specified range, using the given increment (see METHODS section). The value of X and of FNF(X) are printed for X-MIN, X-MAX, and any max/min or zeros found in the range.

MINIMUM [SF Key 15]:

MAXIMUM [SF Key 16]:

ZEROS [SF Key 17]:
These are "mode" keys which may be turned on or off. When START is pressed, all three are on. By pressing the appropriate key, the MAX, MIN, or ZERO may be placed in the alternate mode. When FIND is pressed, only those options (MIN, MAX, ZEROS) which are on will be searched for over the range.

PRINT [SF Key 3]:
Print is a "mode" key. When START is pressed, the calculator is put into the print mode. Pressing the PRINT key will put the calculator in the alternate mode. The output of other program keys varies according to whether the print mode is on or off.

POINT MODE [SF Key 5]:
Pressing POINT puts the program in the point-mode (see FUNCTION below) and asks for the point (i.e., value of X) to be used in the calculation.

RANGE MODE [SF Key 6]:
Pressing RANGE puts the program in the range-mode (see FUNCTION below).

FUNCTION [SF Key 7]:
The result of pressing the FUNCTION key depends on the current mode of POINT/RANGE and PRINT. If the program is in the point-mode, the function is evaluated at the given point and the results are printed. If the program is in the range-mode, a table of values for the function is evaluated over the given range and at the specified intervals. This table is stored and may then be used to plot the function (see PLOT below). If the print-mode is on, this table of values will also be printed as each value is calculated.

DERIVATIVE [SF Key 8]:
This key operates in a manner identical to the FUNCTION key above, except the derivative of the function, rather than the function itself, is evaluated.

INTEGRAL [SF Key 9]:
This key operates in a manner identical to the FUNCTION key above, except the integral of the function, rather than the function itself, is evaluated. In addition, initial conditions may be specified. In the point-mode, the integral is evaluated from the lower limit of the range (i.e., X-MIN) to the given point. In the range-mode, the calculator will ask for the initial value of the integral at X-MIN. Each successive integral in the table will then be the accumulated integral from X-MIN to the current point (specified by the given interval size).

PLOT [SF Key 4]:
When the PLOT key is pressed, the values currently stored in the table will be plotted. The size of the plot is automatically adjusted so that the plot covers the maximum printer-paper width. If the program is in the print-mode, the value of X and Y will also be printed alongside of the plotted point.

DISPLAY PLOT RANGE [SF Key 13]:
When the function, its derivative, or its integral is tabulated and stored, the values of X are determined by the range and interval that is set. Once the values have been tabulated, the range (but not the interval) may be reset for plotting purposes. If the range is reset to be different from the range over which the values were tabulated, and PLOT is pressed, only the tabulated values of X which lie within the new range will be plotted, and the plot size will be adjusted accordingly. All values are retained in the stored table. If the range has been reset in this manner, the range of X-values for which the function, derivative, or integral is stored may be displayed with the DISPLAY PLOT RANGE key.

[SF KEY 11]:
This key is used by the program, and is not available to the user.

## EXAMPLE

```
10 DEF FNF(X)=EXP(-X/5)*SINX

RUN, START
READY

SET RANGE
X-MIN=?0
X-MAX=?10
SET INCREMENT
X-STEP SIZE=?1
FIND

                       X                    F(X)
X-MIN           0.00000000            0.00000000
MAX             1.37340160            0.74506001
ZERO            3.14159266           -0.00000000
MIN             4.51500800           -0.39748064
ZERO            6.28318531            0.00000000
MAX             7.65657600            0.21205119
ZERO            9.42477795            0.00000000
X-MAX          10.00000000           -0.07362525

SET POINT MODE
X=?1
POINT= 1
X= 1                    FUNCTION= 0.688938173

X= 1                    DERIVATIVE= 0.304574636

SET ERROR
ERROR=?1E-5
X= 1                    INTEGRAL= 0.403702170

SET ERROR
ERROR=?1E-9
SET RANGE MODE
DERIVATIVE
          X                   DERIVATIVE
         0.00000000            1.00000000
         1.00000000            0.30457464
         2.00000000           -0.40085537
         3.00000000           -0.55880909
         4.00000000           -0.22569021
         5.00000000            0.17490714
         6.00000000            0.30602919
         7.00000000            0.15350812
         8.00000000           -0.06932554
         9.00000000           -0.16423287
        10.00000000           -0.09883105
```

## EXAMPLE

```
SET INCREMENT
X-STEP SIZE=?0
NO. INCRMS=?40
PRINT OFF
FUNCTION

PLOT
```

```
PRINT ON
PLOT

                        *     0.00000,     0.00000
                                      *     0.25000,     0.23534
                             0.50000,     0.43380   *
                                      0.75000,     0.58669   *
                                            1.00000,     0.68894   *
                                                1.25000,     0.73907   *
                                               1.50000,     0.73896   *
                                             1.75000,     0.69340   *
                                        2.00000,     0.60952   *
                                 2.25000,     0.49612   *
                        2.50000,     0.36299  *
                                      *     2.75000,     0.22020
                             *     3.00000,     0.07745
                     *     3.25000,     -0.05648
             *     3.50000,     -0.17419
       *     3.75000,     -0.26999
   *     4.00000,     -0.34005
*     4.25000,     -0.38253
*     4.50000,     -0.39743
*     4.75000,     -0.38647
  *     5.00000,     -0.35277
     *     5.25000,     -0.30057
          *     5.50000,     -0.23485
              *     5.75000,     -0.16094
                   *     6.00000,     -0.08416
                       *     6.25000,     -0.00951
                            *     6.50000,     0.05863
                               *     6.75000,     0.11667
                                  *     7.00000,     0.16201
                                    *     7.25000,     0.19307
                                     *     7.50000,     0.20930
                                     *     7.75000,     0.21110
                                    *     8.00000,     0.19975
                                   *     8.25000,     0.17719
                                 *     8.50000,     0.14587
                               *     8.75000,     0.10856
                            *     9.00000,     0.06812
                          *     9.25000,     0.02734
                       *     9.50000,     -0.01124
                     *     9.75000,     -0.04546
                   *     10.00000,     -0.07363
```

```
START

1 DIM T5[256]
2 X=A9=I9=T[1]=0
3 N9=20
4 E9=1E-09
5 B9=T9=M8=N9=Z9=S9=1
6 DISP "READY"
7 END

SET RANGE

1 DISP "X-MIN=";
2 INPUT A9
3 DISP "X-MAX=";
4 INPUT B9
5 END

SET INCREMENT

1 DISP "X-STEP SIZE=";
2 INPUT N9
3 IF N9=0 THEN 6
4 I9=1
5 END
6 DISP "NO. INCRMS=";
7 INPUT N9
8 I9=0
9 END

DISPLAY RANGE

1 DISP A9":"B9
2 END

DISPLAY INCREMENT

1 IF I9=0 THEN 4
2 DISP "STEP SIZE=";N9
3 END
4 DISP "NO.INCRMS=";N9
5 END

DISPLAY PLOT RANGE

1 DISP A8":"A8+(N8-1)*I8
2 END

PRINT
```

```
1 T9=(T9=0)
2 IF T9=0 THEN 5
3 DISP "ON"
4 END
5 DISP "OFF"
6 END
```

PLOT

```
1 FORMAT F12.5
2 FORMAT E12.5
10 D8=-INT((A8-A9)/I8)
12 D9=INT((B9-A8)/I8)
14 Y8=Y9=T[D8+1]
16 FOR J9=D8+1 TO D9+1
18 IF T[J9]<Y9 THEN 22
20 Y9=T[J9]
22 IF T[J9]>Y8 THEN 26
24 Y8=T[J9]
26 NEXT J9
28 PRINT
30 IF T9=1 THEN 42
32 FOR J9=D8 TO D9
34 H9=70*(T[J9+1]-Y8)/(Y9-Y8)
36 PRINT TABH9"*"
38 NEXT J9
40 GOTO 68
42 FOR J9=D8 TO D9
44 P8=A8+J9*I8
46 P9=T[J9+1]
48 H9=79*(P9-Y8)/(Y9-Y8)
50 IF H9>40 THEN 58
52 PRINT TABH9"*";
54 GOSUB 72
56 GOTO 64
58 PRINT TAB(H9-27);
60 GOSUB 72
62 PRINT "   *";
64 PRINT
66 NEXT J9
68 PRINT
70 END
72 GOSUB 78
74 PRINT ",";
76 P8=P9
78 IF P8<1E+05 THEN 84
80 WRITE (15,2)P8;
82 RETURN
84 WRITE (15,1)P8;
86 RETURN
```

```
SET MIN

1 M8=(M8=0)
2 IF M8=0 THEN 5
3 DISP "MIN ON"
4 END
5 DISP "MIN OFF"
6 END

SET MAX

1 M9=(M9=0)
2 IF M9=0 THEN 5
3 DISP "MAX ON"
4 END
5 DISP "MAX OFF"
6 END

SET ZEROS

1 Z9=(Z9=0)
2 IF Z9=0 THEN 5
3 DISP "ZEROS ON"
4 END
5 DISP "ZEROS OFF"
6 END

FIND

10 H9=(B9-A9)/N9*(I9=0)+N9*(I9=1)
12 PRINT
14 PRINT TAB24"X"TAB42"F(X)"
16 P8=X8=A9
18 P9=Y8=FNFX8
20 PRINT "X-MIN"TAB11;
22 P8=FNZ0
24 X9=A9+H9
26 Y9=FNFX9
28 D9=Y9-Y8
30 GOSUB 114
32 FOR J9=X9+H9 TO B9-H9 STEP H9
34 GOSUB 54
36 NEXT J9
38 J9=B9
40 GOSUB 54
42 P8=X9
44 P9=Y9
46 PRINT "X-MAX"TAB11;
48 P8=FNZ0
50 PRINT
```

```
52 END
54 X7=X8
56 X8=X9
58 Y8=Y9
60 D8=D9
62 X9=J9
64 Y9=FNFJ9
66 D9=Y9-Y8
68 IF SGND8=SGND9 THEN 114
70 W8=SGN(SGND8-SGND9)
72 IF M8*(W8<0)+M9*(W8>0)=0 THEN 114
74 P8=V8=X7
76 V9=X9
78 H8=(V9-V8)/10
80 P9=W8*FNFV8
82 FOR J8=V8 TO V9 STEP H8
84 W9=W8*FNFJ8
86 IF W9<P9 THEN 92
88 P8=J8
90 P9=W9
92 NEXT J8
94 IF ABS(P9-W8*FNF(P8-H8))<E9 THEN 102
96 V9=P8+H8
98 V8=P8-H8
100 GOTO 78
102 P9=FNFP8
104 IF W8=1 THEN 110
106 PRINT "MIN"TAB11;
108 GOTO 112
110 PRINT "MAX"TAB11;
112 P8=FNZ0
114 IF Z9=0 THEN 126
116 IF ABSY9>E9 THEN 128
118 P8=X9
120 P9=Y9
122 PRINT "ZERO"TAB11;
124 P8=FNZ0
126 RETURN
128 IF (SGNY8#SGNY9) AND (ABSY8>E9) THEN 132
130 RETURN
132 V8=X8
134 W8=Y8
136 V9=X9
138 W9=Y9
140 H8=H9
142 H8=H8/2
144 P8=V8+H8
146 P9=FNFP8
148 IF ABSP9>E9 THEN 152
150 GOTO 122
```

```
152 IF SGNP9=SGNW9 THEN 160
154 V8=P8
156 W8=P9
158 GOTO 142
160 V9=P8
162 W9=P9
164 GOTO 142
```

SET ERROR

```
1 DISP "ERROR=";
2 INPUT E9
3 END
```

SET POINT MODE

```
1 DISP "X=";
2 INPUT X
3 S9=0
4 DISP "POINT=";X
5 END
```

SET RANGE MODE

```
1 S9=1
2 DISP "RANGE"
3 END
```

FUNCTION

```
1 X7=FNZ1
2 END
```

DERIVATIVE

```
1 X7=FNZ2
2 END
```

INTEGRAL

```
1 X7=FNZ3
2 END
```

(FNZ)

```
10 DEF FNZ(Z)
12 GOTO INT((Z-1)/3)+1 OF 44,18
14 FORMAT F20.8
16 FORMAT E20.8
18 GOSUB 22
```

```
20 RETURN 0
22 IF T9=0 AND Z#0 THEN 32
24 GOSUB 34
26 P8=P9
28 GOSUB 34
30 PRINT
32 RETURN
34 IF P8<1E+05 THEN 40
36 WRITE (15;16)P8;
38 RETURN
40 WRITE (15;14)P8;
42 RETURN
44 GOTO S9*(T9+1) OF 132,118
46 X9=X
48 X8=A9
50 PRINT "X="X;TAB25;
52 GOSUB Z OF 60,66,90
54 PRINT Y9
56 PRINT
58 RETURN 0
60 PRINT "FUNCTION=";
62 Y9=FNFX9
64 RETURN
66 PRINT "DERIVATIVE=";
68 H8=(X9+(X9=0))/1E+09
70 W9=FNFX9
72 W8=FNF(X9+H8)
74 IF ABS(W9-W8)/(ABSW9+1E-06)>1E-04 THEN 84
76 H8=2*H8
78 IF ABS(H8/(X9+(X9=0)))<1 THEN 72
80 Y9=0
82 RETURN
84 Y9=25*W9-48*W8+36*FNF(X9+2*H8)-16*FNF(X9+3*H8)+3*FNF(X9+4*H8)
86 Y9=-Y9/(12*H8)
88 RETURN
90 PRINT "INTEGRAL=";
92 H8=X9-X8
94 V9=H8*(FNFX8+FNFX9)
96 W9=0
98 V9=(V9+W9)/2
100 W9=0
102 FOR Y8=X8+H8/2 TO X9 STEP H8
104 W9=W9+FNFY8
106 NEXT Y8
108 W9=W9*H8
110 Y9=(V9+2*W9)/3
112 H8=H8/2
114 IF ABS(V9-W9)/ABSY9>SQRE9 THEN 98
116 RETURN
118 PRINT TAB11"X"TAB28;
```

```
120 GOTO Z OF 130,126
122 PRINT "INTEGRAL"
124 GOTO 132
126 PRINT "DERIVATIVE"
128 GOTO 132
130 PRINT "FUNCTION"
132 IF Z#3 THEN 144
134 DISP "INIT. VALUE=";
136 INPUT P9
138 T[1]=D9=P9
140 P8=A9
142 GOSUB 22
144 X8=A8=A9
146 N8=(Z=3)
148 I8=(B9-A9)/N9*(I9=0)+N9*(I9=1)
150 FOR X9=A9+N8*I8 TO B9 STEP I8
152 N8=N8+1
154 GOSUB Z OF 62,68,92
156 P8=X9
158 P9=Y9
160 T[N8]=Y9
162 IF Z#3 THEN 170
164 T[N8]=T[N8]+T[N8-1]
166 D9=D9+P9
168 P9=D9
170 GOSUB 22
172 X8=X9
174 NEXT X9
176 PRINT
178 RETURN 0

VARIABLES USED:   X,A8,A9,B9,D8,D9,E9,H8,H9,I8,I9,
                  J8,J9,M8,M9,N8,N9,P8,P9,S9,T9,V8,
                  V9,W8,W9,X7,X8,X9,Y9,Z9.
```

MODEL 30

## FUNCTION ANALYSIS (2K)

**DESCRIPTION:**

Given any function which can be defined in closed form, the function, its derivative, or its integral may be evaluated at a point, tabulated over a range, or plotted (with or without printed coordinate values).

**METHODS:**

The derivative of the function at a point is found by moving away from the given point until the change in the function is large enough to give significant results. Four equally spaced points are then used to calculate the leading differences, from which the derivative is obtained.

Integrals are obtained by Simpson's rule methods, with the sub-intervals successively halved until the error is within acceptable limits. The operator should be aware that if the error is set too small, the integral may not converge to a solution for functions that oscillate too rapidly in the range of integration.

**SPECIAL CONSIDERATIONS:**

When tabulating a function or derivative over the range, or when evaluating an integral, the range should be set so that the function does not pass through a singularity, as this may result in an error.

The maximum number of points that may be tabulated for plotting is 101. If the function which is defined for analysis requires more than about 50 words of memory, ERROR 2 (memory overflow) may occur during running time. If this should happen, the DIM TS(101) statement in line 1 of the START key may be edited from 101 to some smaller number of elements. This will give the user more memory space for the defined function, but will also reduce the number of plot-points that may be stored.

**ACKNOWLEDGMENTS:**

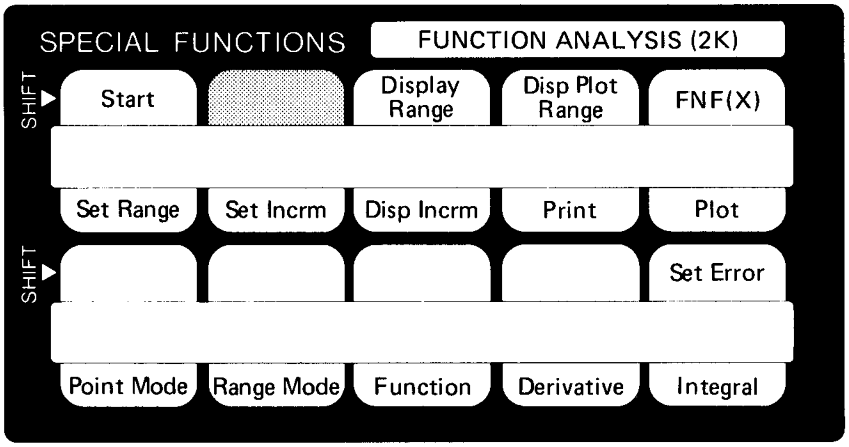
John H. Nairn, Hewlett-Packard

**REFERENCES:**

Kunz, Kaiser S., Numerical Analysis (New York: McGraw-Hill Book Company, 1957), Chapter 7.

**SYSTEM SPECIFICATIONS:**

9830 (2K R/W)
9866 Printer or 9861 Typewriter




OPERATION:

1. Type LOADKEY 4, and press EXECUTE.
2. Press FETCH, FNF [SF Key 14]. The display will read "KEY".
3. Define the function to be analyzed as FNF(X), either as a single- or a multi-line function.
4. Press RUN, START [SF Key 10]. The display should read "READY". If RUN was not pressed, the display will read "ERROR 41 IN LINE 2".
5. Any combination of the available operations described below may now be used.
6. To define a new function for analysis, press SCRATCH, FNF [SF Key 14] and go to step 2 above.

SPECIAL FUNCTIONS:

All of the special function keys for this program operate in a manner identical to that described in the SPECIAL FUNCTIONS Section of the program FUNCTION ANALYSIS (4K) with the following exception.

Due to memory limitations, the options FIND, MINIMUM, MAXIMUM, and ZEROS are not in the 2K version of this program, but are found in a separate program, MIN/MAX/ZEROS. If any of these keys [SF Keys 15, 16, 17, 18] are pressed in the 2K version of this program, ERROR 10 (Key undefined) will result.

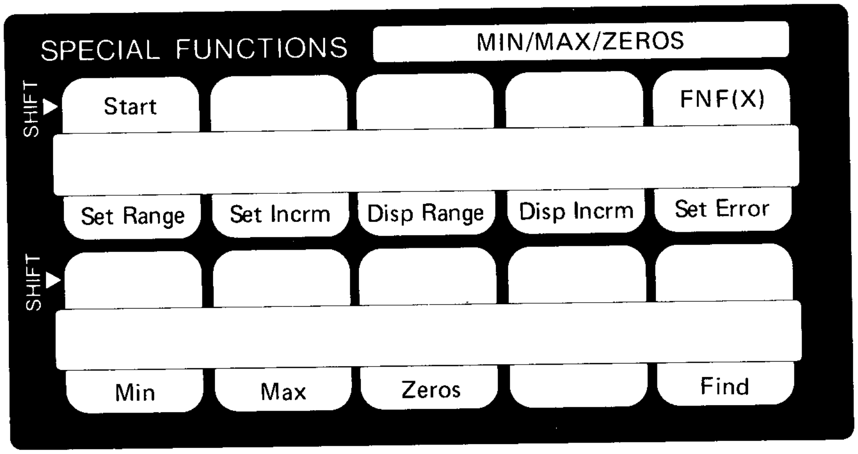
Note: The EXAMPLE and LISTING for this program are identical to the 4K version, with the above mentioned options deleted.

MODEL 30

# MIN/MAX/ZEROS

| | |
|---|---|
| DESCRIPTION: | Given any function which can be defined in closed form, this program finds its minimums, maximums, and zeros over a given range of the argument. |
| METHODS: | This program evaluates the function over the range given at the specified intervals. If the sign of the function for two successive points is different, a search is conducted for a zero in that interval. If the difference quotients for three successive points are of opposite sign, a search for a max/min is conducted in that interval. |
| SPECIAL CONSIDERATIONS: | The range should be set so that the function does not pass through a singularity, as this may result in an error or cause the program to go into an endless loop. |
| ACKNOWLEDGMENTS: | John H. Nairn, Hewlett-Packard |
| REFERENCES: | None |
| SYSTEM SPECIFICATIONS: | 9830 (2K R/W)<br>9866 Printer or 9861 Typewriter |

HEWLETT · PACKARD HEWLETT · PACKARD HEWLETT · PACKARD HEWLETT · PACKARD




OPERATION:

1. Type LOADKEY 5, and press EXECUTE.
2. Press FETCH, FNF [SF Key 14]. The display will read "KEY".
3. Define the function to be analyzed as FNF(X), either as a single- or a multi-line function.
4. Press RUN, START [SF Key 10]. The display should read "READY".
5. Any combination of the available operations described below may now be used.
6. To define a new function for analysis, press SCRATCH, FNF [SF Key 14] and go to step 2 above.

SPECIAL FUNCTIONS:

<u>SET RANGE</u> [SF Key 0]:
Sets X-MIN and X-MAX which define the range of X.

<u>SET INCREMENT</u> [SF Key 1]:
Sets the X-increment. When SF Key 1 is pressed, the calculator will ask for the X-step size. If it is more convenient to enter the number of increments into which the range should be divided, indicate this by entering a zero for the X-step size.

<u>DISPLAY RANGE</u> [SF Key 2]:
<u>DISPLAY INCREMENT</u> [SF Key 3]:
These keys display the current range or increment that is set.

<u>SET ERROR</u> [SF Key 4]:
This key allows the operator to specify the allowable error in the calculating of max/min, zeros.
Note: When START is pressed, "default" values are given to the above variables. These are: X-MIN = 0, X-Max = 1, NO. INCRMS. = 20, and ERROR = 1E-09.

<u>FIND</u> [SF Key 9]:
When FIND is pressed, the program searches for the maximums, minimums, and zeros of the function over the specified range, using the given increment (see METHODS section). The value of X and of FNF(X) are printed for X-MIN, X-MAX, and any max/min or zeros found within the range.

MINIMUM [SF Key 5]:
MAXIMUM [SF Key 6]:
ZEROS [SF Key 7]:
These are "mode" keys which may be turned on or off. When START is pressed, all three are on. By pressing the appropriate key, the MAX, MIN, or ZERO may be placed in the alternate mode. When FIND is pressed, only those options (MIN, MAX, ZEROS) which are on will be searched for over the range.

## EXAMPLE

```
10 DEF FNF(X)=X↑3-6*X↑2+11*X-6

START
READY
SET RANGE
X-MIN=?0
X-MAX=?5
SET INCREMENT
X-STEP SIZE=?0.5
FIND

                    X              F(X)
X-MIN         0.00000000      -6.00000000
ZERO          1.00000000       0.00000000
MAX           1.42265600       0.38490018
ZERO          2.00000000       0.00000000
MIN           2.57735040      -0.38490018
ZERO          3.00000000       0.00000000
X-MAX         5.00000000      24.00000000

********************

10 DEF FNF(X)=X*SINX+X/2

START
READY
SET RANGE
X-MIN=?-5
X-MAX=?5
SET INCREMENT
X-STEP SIZE=?0
NO. INCRMS=?20
FIND

                    X              F(X)
X-MIN        -5.00000000      -7.29462137
ZERO         -2.61799388      -0.00000000
MAX          -1.82892160       0.85386906
ZERO         -0.52359878      -0.00000000
MIN          -0.25552000      -0.06317769
ZERO          0.00000000       0.00000000
MAX           2.20467200       2.87872426
ZERO          3.66519143       0.00000000
X-MAX         5.00000000      -2.29462137
```

```
START

1 A9=I9=0
2 N9=20
3 E9=1E-09
4 B9=M8=M9=Z9=1
5 DISP "READY"
6 END

SET RANGE

1 DISP "X-MIN=";
2 INPUT A9
3 DISP "X-MAX=";
4 INPUT B9
5 END

SET INCREMENT

1 DISP "X-STEP SIZE=";
2 INPUT N9
3 IF N9=0 THEN 6
4 I9=1
5 END
6 DISP "NO. INCRMS=";
7 INPUT N9
8 I9=0
9 END

DISPLAY RANGE

1 DISP A9":"B9
2 END

DISPLAY INCREMENT

1 IF I9=0 THEN 4
2 DISP "STEP SIZE=";N9
3 END
4 DISP "NO. INCRMS=";N9
5 END

SET ERROR

1 DISP "ERROR=";
2 INPUT E9
3 END

SET MIN
```

```
1 M8=(M8=0)
2 IF M8=0 THEN 5
3 DISP "MIN ON"
4 END
5 DISP "MIN OFF"
6 END
```

SET MAX

```
1 M9=(M9=0)
2 IF M9=0 THEN 5
3 DISP "MAX ON"
4 END
5 DISP "MAX OFF"
6 END
```

SET ZEROS

```
1 Z9=(Z9=0)
2 IF Z9=0 THEN 5
3 DISP "ZEROS ON"
4 END
5 DISP "ZEROS OFF"
6 END
```

FIND

```
10 H9=(B9-A9)/N9*(I9=0)+N9*(I9=1)
12 PRINT
14 PRINT TAB24"X"TAB42"F(X)"
16 P8=X8=A9
18 P9=Y8=FNFX8
20 PRINT "X-MIN"TAB11;
22 P8=FNZ0
24 X9=A9+H9
26 Y9=FNFX9
28 D9=Y9-Y8
30 GOSUB 114
32 FOR J9=X9+H9 TO B9-H9 STEP H9
34 GOSUB 54
36 NEXT J9
38 J9=B9
40 GOSUB 54
42 P8=X9
44 P9=Y9
46 PRINT "X-MAX"TAB11;
48 P8=FNZ0
50 PRINT
52 END
```

```
54 X7=X8
56 X8=X9
58 Y8=Y9
60 D8=D9
62 X9=J9
64 Y9=FNFJ9
66 D9=Y9-Y8
68 IF SGND8=SGND9 THEN 114
70 W8=SGN(SGND8-SGND9)
72 IF M8*(W8<0)+M9*(W8>0)=0 THEN 114
74 P8=V8=X7
76 V9=X9
78 H8=(V9-V8)/10
80 P9=W8*FNFV8
82 FOR J8=V8 TO V9 STEP H8
84 W9=W8*FNFJ8
86 IF W9<P9 THEN 92
88 P8=J8
90 P9=W9
92 NEXT J8
94 IF ABS(P9-W8*FNF(P8-H8))<E9 THEN 102
96 V9=P8+H8
98 V8=P8-H8
100 GOTO 78
102 P9=FNFP8
104 IF W8=1 THEN 110
106 PRINT "MIN"TAB11;
108 GOTO 112
110 PRINT "MAX"TAB11;
112 P8=FNZ0
114 IF Z9=0 THEN 126
116 IF ABSY9>E9 THEN 128
118 P8=X9
120 P9=Y9
122 PRINT "ZERO"TAB11;
124 P8=FNZ0
126 RETURN
128 IF (SGNY8#SGNY9) AND (ABSY8>E9) THEN 132
130 RETURN
132 V8=X8
134 W8=Y8
136 V9=X9
138 W9=Y9
140 H8=H9
142 H8=H8/2
144 P8=V8+H8
146 P9=FNFP8
148 IF ABSP9>E9 THEN 152
150 GOTO 122
152 IF SGNP9=SGNW9 THEN 160
```

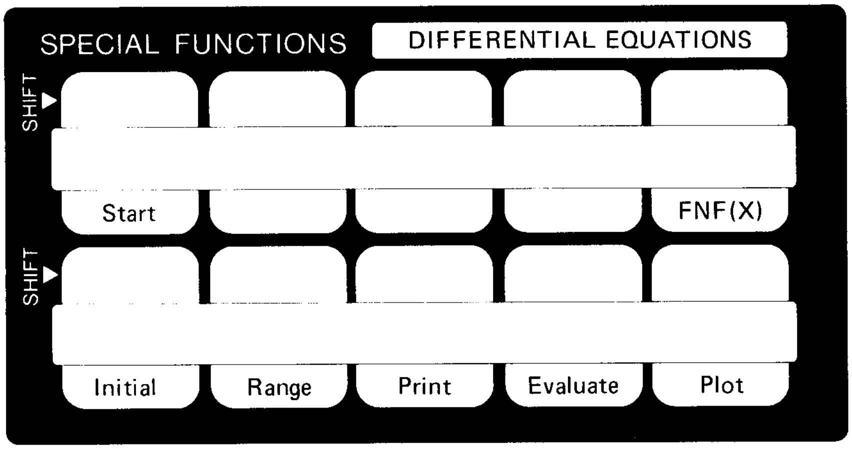
```
154 V8=P8
156 W8=P9
158 GOTO 142
160 V9=P8
162 W9=P9
164 GOTO 142
166 FORMAT F20.8
168 FORMAT E20.8
170 DEF FNZ(Z)
172 GOSUB 182
174 P8=P9
176 GOSUB 182
178 PRINT
180 RETURN 0
182 IF P8<1E+05 THEN 188
184 WRITE (15,168)P8;
186 RETURN
188 WRITE (15,166)P8;
190 RETURN

VARIABLES USED: A9,B9,D8,D9,E9,H8,H9,I9,J8,
                J9,M8,M9,N9,P8,P9,V8,V9,W8,
                W9,X7,X8,X9,Y8,Y9,Z9.
```

MODEL 30

HEWLETT · PACKARD HEWLETT · PACKARD HEWLETT · PACKARD HEWLETT · PACKARD

# DIFFERENTIAL EQUATIONS

| | |
|---|---|
| DESCRIPTION: | This program solves the initial value problem for a first or second order differential equation. |
| METHODS: | The solution of the differential equation is found using the second order Runge-Kutta method. |
| SPECIAL CONSIDERATIONS: | The range should be set so that the differential equation does not pass through a singularity, as this may result in an error.<br><br>The Program can store up to 100 points for plotting. |
| ACKNOWLEDGMENTS: | John H. Nairn, Hewlett-Packard |
| REFERENCES: | Abramowitz, M., and Stegun, I. A., Handbook of Mathematical Functions (Washington D. C., U. S. Government Printing Office, NBS Applied Mathematics Series #55, 1964), Section 25.5.6<br><br>Time-Shared BASIC Program Library Handbook (Cupertino, California: Hewlett-Packard, Software Center, 1971), Programs DE-1OR and DE-2OR. |
| SYSTEM SPECIFICATIONS: | 9830 (2K R/W)<br>9866 Printer or 9861 Typewriter |




OPERATION:

1. Type LOADKEY 6 and press EXECUTE.
2. Press FETCH, FNF [SF Key 4]. The display will read "KEY".
3. Define the differential equation as FNF(X), either as a single- or a multi-line function.
   For a first order differential equation, the function should be entered as FNF(X) = f(X, Y) where f(X, Y) is any function of X and Y. For example, to solve the differential equation $y' = -y^2/x$, it would be entered as

   ```
   10 DEF FNF(X) = - Y*Y/X .
   ```

   For a second order differential equation, the function should be entered as FNF(X) = f(X, Y, Z) where f(X, Y, Z) is any general function of X, Y, and Z, and Z is used to denote y'. For example, to solve the differential equation $y'' = y^2/x^2 - 2y\,y'/x$, it would be entered as

   ```
   10 DEF FNF(X) = (Y*Y)/(X*X) - 2*Y*Z/X .
   ```
4. Press RUN, START [SF Key 0]. Wait until the display shows "READY".
5. Press the key INITIAL [SF Key 5] to set up the initial conditions. The program will ask for the order of the differential equation. Enter a 1 or a 2 and press EXECUTE. The program will then ask for the initial X, initial Y, and if the differential equation is second order, the initial Y'.
6. Press the key RANGE [SF Key 6] to set up the conditions for integration of the differential euqation. The program will ask for the X-MAX, the integration step size, and the Print step size. When EVALUATE (see below) is pressed, the differential equation will be integrated from the initial X to the X-MAX and the accumulated results will be printed in increments specified by the print step size. The integration step size determines the intervals into which the range is divided for integration of the differential equation. The user should be aware of the fact that by reducting the integration step size, both the accuracy of the results and the running time are increased.
7. When all conditions have been entered, EVALUATE [SF Key 8] should be pressed. The y-values of the solution to the differential equation will be calculated and printed. (If the equation is second order, the values of y' will also be calculated and printed.)

8. After the y-values have been calculated, the user may call for the function to be plotted by pressing PLOT [SF Key 9].

Note: The PRINT key [SF Key 7] is used to put the program in either the print-on or print-off mode. (When START is pressed, the program is in the print-on mode.) If EVALUATE is pressed when the program is in the print-off mode, the y-values will be calculated and stored for plotting, but not printed. The x and y coordinates of the points being plotted will be either included or deleted from the printout when PLOT is pressed, depending on the current status of the print-on/off mode of the program.

9. At this point, the user may respecify the range and/or the initial conditions, and re-evaluate the function. Or a new differential equation may be entered for analysis by pressing SCRATCH, FNF [SF Key 4], and continuing at step 2 of the operating instructions.

## EXAMPLE

USING NUMERICAL METHODS, VERIFY THAT THE TWO DIFFERENTIAL EQUATIONS Y''= (2XY-3)/X↑3 AND Y''= -(Y'X↑2-XY+2)/X↑3 HAVE THE SAME SOLUTION, WITH THE INITIAL VALUES X=1, Y=1, Y'=0; AND SHOW THAT THIS SOLUTION IS Y = (LOGX+1)/X .

```
10 DEF FNF(X)=(2*X*Y-3)/X↑3
RUN, START
READY
INITIAL
ORDER=?2
INITIAL X=?1
INITIAL Y=?1
INITIAL Y'=?0
RANGE
X-MAX=?2
INTEGRATION STEP SIZE=?.01
PRINT STEP SIZE=?.2
EVALUATE
VALUE OF X              VALUE OF Y      VALUE OF Y'

 1                        1.000000        0.000000
 1.2                      0.985255       -0.126635
 1.4                      0.954600       -0.171704
 1.6                      0.918720       -0.183638
 1.8                      0.882061       -0.181466
 2                        0.846520       -0.173343

FETCH FNF(X)
10 DEF FNF(X)=-(Z*X↑2-X*Y+2)/X↑3
EVALUATE
VALUE OF X              VALUE OF Y      VALUE OF Y'

 1                        1.000000        0.000000
 1.2                      0.985256       -0.126621
 1.4                      0.954605       -0.171682
 1.6                      0.918730       -0.183609
 1.8                      0.882078       -0.181431
 2                        0.846544       -0.173302
```

THIS SHOWS THAT THE TWO EQUATIONS HAVE THE SAME SOLUTION.

RUNNING THE SHORT PROGRAM ...

```
10 FOR X=1 TO 2 STEP 0.2
20 PRINT X,(LOGX+1)/X
30 NEXT X
40 END
```

```
RUN
 1             1
 1.2           0.985267964
 1.4           0.954623026
 1.6           0.918752268
 1.8           0.882103703
 2             0.846573590

... SHOWS THAT THE SOLUTION IS (LOGX+1)/X .
```

```
START

1 DIM TS[100]
2 P=T[1]=Z0=1
3 DISP "READY"
4 END

INITIAL

1 DISP "ORDER=";
2 INPUT C
3 IF  NOT (C=1 OR C=2) THEN 1
4 DISP "INITIAL X=";
5 INPUT X0
6 DISP "INITIAL Y=";
7 INPUT Y0
8 IF C=1 THEN 11
9 DISP "INITIAL Y'=";
10 INPUT Z0
11 END

RANGE

1 DISP "X-MAX=";
2 INPUT B
3 DISP "INTEGRATION STEP SIZE=";
4 INPUT H
5 DISP "PRINT STEP SIZE=";
6 INPUT L
7 END

PRINT

1 P=(P=0)
2 IF P=0 THEN 5
3 DISP "ON"
4 END
5 DISP "OFF"
6 END

EVALUATE

10 FORMAT F15.6
20 FORMAT E15.6
30 X=X0
35 Z=Z0
40 T[1]=Y=Y0
50 I=1
60 E=1/L
70 S=SGN(B-X0)
```

```
80 H=S*ABSH
90 L=S*ABSL
100 IF P=0 THEN 290
110 PRINT "VALUE OF X"TAB20"VALUE OF Y"TAB35;
120 IF C=1 THEN 140
130 PRINT "VALUE OF Y'";
140 PRINT
150 PRINT
160 PRINT X,
170 Y1=Y
180 GOSUB 800
190 IF C=1 THEN 220
200 Y1=Z
210 GOSUB 800
220 PRINT
230 GOTO 290
240 P1=Y+H*FNFX
250 X=X+H
260 Y2=Y+P1
270 Y=P1
280 Y=(Y2+H*FNFX)/2
290 V=((B-X-H)*S>1E-07)
300 R=V*(X+H)+(1-V)*B
310 A=INT(E*R)/E+L*(S-1)/2
320 A=A+L*(S*(A+L-R)<1E-07)
330 IF (X-A)*S+1E-07 >= 0 THEN 360
340 Q=A
350 GOSUB C OF 540,590
360 IF R=B THEN 380
370 GOTO C OF 240,430
380 IF ABS(R-A)<5E-07 THEN 410
390 Q=R
400 GOSUB C OF 540,590
410 PRINT
420 END
430 P1=Y+H*Z
440 Q1=Z+H*FNFX
450 Y=(Y+P1+H*Q1)/2
460 Y1=Y
470 Z1=Z
480 Y=P1
490 Z=Q1
500 X=X+H
510 Z=(Z1+Q1+H*FNFX)/2
520 Y=Y1
530 GOTO 290
540 Y2=Y
550 Y=Y+(Q-X)*FNFX
560 Y1=(Y2+Y+(Q-X)*FNFQ)/2
570 Y=Y2
```

```
580 GOTO 700
590 P2=Y+(Q-X)*Z
600 Q2=Z+(Q-X)*FNFX
610 Y2=(Y+P2+(Q-X)*Q2)/2
620 Y1=Y
630 Z1=Z
640 Y=P2
650 Z=Q2
660 Z2=(Z1+Q2+(Q-X)*FNFQ)/2
670 Y=Y1
680 Z=Z1
690 Y1=Y2
700 I=I+1
710 T[I]=Y1
720 IF P=0 THEN 790
730 PRINT Q;
740 GOSUB 800
750 IF C=1 THEN 780
760 Y1=Z2
770 GOSUB 800
780 PRINT
790 RETURN
800 IF Y1<1E+05 THEN 830
810 WRITE (15,20)Y1;
820 RETURN
830 WRITE (15,10)Y1;
840 RETURN
```

PLOT

```
1 FORMAT F12.5
2 FORMAT E12.5
10 Y1=Y2=T[1]
20 FOR J=1 TO I
30 IF T[J]<Y2 THEN 50
40 Y2=T[J]
50 IF T[J]>Y1 THEN 70
60 Y1=T[J]
70 NEXT J
80 PRINT
100 FOR J=1 TO I
110 GOSUB P+1 OF 160,190
120 NEXT J
130 PRINT
140 END
160 Z1=70*(T[J]-Y1)/(Y2-Y1)
170 PRINT TABZ1"*"
180 RETURN
190 P1=(X0+(J-1)*L)*(J#I)+B*(J=I)
200 Q1=79*(T[J]-Y1)/(Y2-Y1)
```

```
210 IF Q1>40 THEN 250
220 PRINT TABQ1"*";
230 GOSUB 300
240 GOTO 280
250 PRINT TAB(Q1-27);
260 GOSUB 300
270 PRINT "   *";
280 PRINT
290 RETURN
300 GOSUB 330
310 PRINT ",";
320 P1=T[J]
330 IF P1<1E+05 THEN 360
340 WRITE (15,2)P1;
350 RETURN
360 WRITE (15,1)P1;
370 RETURN


VARIABLES USED: A,B,C,E,H,I,J,L,P,Q,R,S,V,X,Y,Z,
                P1,P2,Q1,Q2,X0,Y0,Y1,Y2,Z0,Z1,Z2.
```

MODEL 30

# POLYNOMIAL REGRESSION

HEWLETT · PACKARD

**DESCRIPTION:**

This program takes a set of data points (x, y) and calculates the coefficients of a polynomial (up to 6th degree) using a least-squares fit. Basic statistics on the data, $r^2$-measure of fit, and tables of calculated values may also be obtained.

**METHODS:**

A triangular factorization of the X'X matrix is used to calculate the coefficients and the statistics (see Reference).

**SPECIAL CONSIDERATIONS:**

Although polynomials of up to 6th degree may be fit to the data, the running time of the program increases with the maximum degree of fit specified by the user. Thus, this maximum degree should not be specified to be larger than actually required.

When the degree of fit is large, and the range of the data is wide, considerable round-off error may occur. If n is the degree of fit, round-off errors enter when the largest and the smallest x-value, each raised to the 2n power, differ by about ten or more orders of magnitude. An R-SQUARE value of more than 1 indicates that this has occured and that the coefficients calculated may not be reliable.

If the program is to be run on a 4K machine, the maximum degree may be increased to 20 by changing the following lines on the START key [SF Key 0] to:

```
10 DIM C(253), B(22)
20 FOR I = 1 TO 22
50 FOR I = 23 TO 253
```

**ACKNOWLEDGMENTS:**

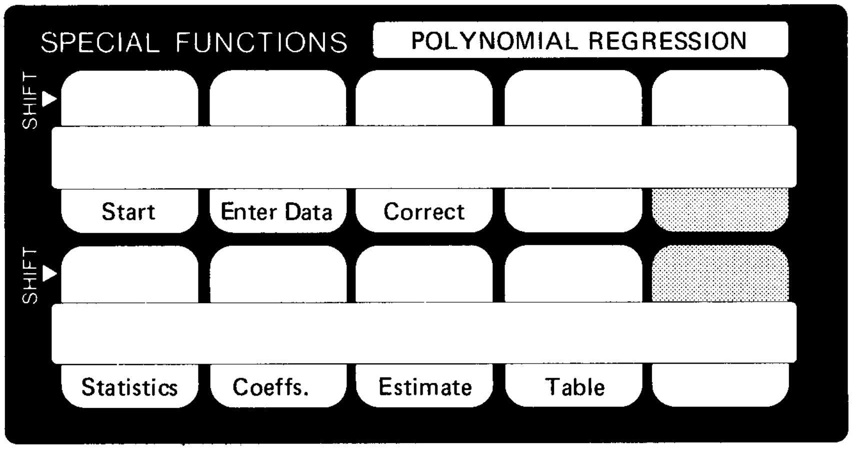
Robert W. Kopitzke, Hewlett-Packard

**REFERENCES:**

Graybill, F. A., An Introduction to Linear Statistical Models (New York: McGraw-Hill Book Co., 1961), Vol. 1

**SYSTEM SPECIFICATIONS:**

9830 (2K or 4K R/W)
9866 Printer or 9861 Typewriter




OPERATION:

1. Type LOADKEY 7 and press EXECUTE.
2. Press RUN, START [SF Key 0]. The program will ask for the maximum degree of polynomial to be fit to the data (see Special Considerations section). Enter this number (less than or equal to 6) and press EXECUTE.
3. Press ENTER DATA [SF Key 1]. Each time the display shows " X, Y = ? ", enter the x and y values of the point and press EXECUTE.
4. If an error is made in entering a data point, press CORRECT [SF Key 2]. When the display reads " WRONG X, Y = ? " enter the x and y values of the incorrectly entered point and press EXECUTE. If more data points are to be entered, the ENTER DATA key [SF Key 1] may be pressed and data entry continued.
5. When all data is correctly entered, any combination of the available options described below may be used, with the restriction that COEFFICIENTS [SF Key 6] must preceed ESTIMATE [SF Key 7] or TABLE [SF Key 8].
6. To enter a new set of data for analysis, go to step 2 above.

SPECIAL FUNCTIONS:

<u>STATISTICS</u> [SF Key 5]:
When the STATISTICS key is pressed, the mean and standard deviation of the x's and the y's, as well as the simple correlation coefficient, are printed. At any time during data entry, this key may be pressed to see the statistics on all data entered to that point. The ENTER DATA key [SF Key 1] will then cause data entry to be resumed.

<u>COEFFICIENTS</u> [SF Key 6]:
When all data has been entered, press COEFFICIENTS. The program will ask for the degree of the polynomial to be fit. Enter this number (less than or equal to the maximum degree specified) and press EXECUTE. The coefficients and the value of R-SQUARE will be printed. To fit a different degree polynomial, simply press the COEFFICIENTS key again and enter the new degree.

Note: If the maximum degree to be fit is n, n + 1 data points are required. If the number of points entered is not at least n + 1 when the COEFFICIENTS key is pressed, a message to this effect will be displayed. Pressing ENTER DATA [SF Key 1] will allow further points to be entered. Once a sufficient number of data points have been entered and COEFFICIENTS is pressed, data points may no longer be entered or corrected.

ESTIMATE [SF Key 7]:
When the ESTIMATE key is pressed, the program will ask for a value of x. When this value is entered, the corresponding value of y will be calculated, based on the last set of coefficients printed. This key may be used repeatedly for any number of x-values.

TABLE [SF Key 8]:
If a table of calculated y-values (YHAT) is desired, press the TABLE key. The program will ask for the minimum x, the maximum x, and the step size for x (i.e., the x increment for the table). Enter these values and press EXECUTE, and the table will be printed, based on the last set of coefficients calculated.

```
RUN, START
MAX DEGREE = 4

ENTER DATA
  NO.         X           Y
   1       1.0000      31.0000
   2       2.0000      59.0000
   3       3.0000      90.0000
   4       4.0000     120.0000
   5       5.0000     151.0000
   6       6.0000     181.0000
   7       7.0000     212.0000
   8       8.0000     243.0000
   9       9.0000     273.0000
  10      10.0000     304.0000
  11      11.0000     334.0000
  12      12.0000     365.0000

STATISTICS

NO. POINTS = 12

X:   MEAN= 6.5                 ST.DEV.= 3.605551275
Y:   MEAN= 196.9166667         ST.DEV.= 109.9210116

CORR.COEFF.= 0.999980668

DEG.REG.=?3

COEFFICIENTS

B(  0)=       1.3333
B(  1)=      28.8526
B(  2)=       0.2539
B(  3)=      -0.0113


R SQUARE =  0.999984155

TABLE
XMIN,XMAX,STEP=?1,12,1

X= 1                     YHAT= 30.4285714
X= 2                     YHAT= 59.96403596
X= 3                     YHAT= 89.87212789
X= 4                     YHAT= 120.0852481
X= 5                     YHAT= 150.5357976
X= 6                     YHAT= 181.1561772
X= 7                     YHAT= 211.8787879
```

## EXAMPLE

```
X= 8                      YHAT= 242.6360306
X= 9                      YHAT= 273.3603063
X= 10                     YHAT= 303.9840160
X= 11                     YHAT= 334.4395604
X= 12                     YHAT= 364.6593407
```

```
START

10 DIM C[36],B[8]
20 FOR I=1 TO 8
30 C[I]=B[I]=0
40 NEXT I
50 FOR I=9 TO 36
60 C[I]=0
70 NEXT I
80 B[1]=1
90 W=N=S1=S2=S3=S4=S5=0
100 DISP "MAX.DEGREE=";
110 INPUT D2
120 PRINT "MAX DEGREE ="; D2
130 PRINT
140 END

ENTER DATA

1 FORMAT F4.0,2F12.4
2 FORMAT 2F12.4
10 IF W=0 THEN 40
20 DISP "NOT ALLOWED"
30 END
40 IF N#0 THEN 60
50 PRINT "  NO."TAB12"X"TAB24"Y"
60 DISP "X,Y=";
70 INPUT B[2],Y
80 WRITE (15,1)N+1,B[2],Y
90 Y=FNX1
100 GOTO 60

CORRECT DATA

10 IF W=0 THEN 40
20 DISP "NOT ALLOWED"
30 END
40 DISP "WRONG X,Y=";
50 INPUT B[2],Y
60 PRINT "DELETE: X="B[2],"Y="Y
70 Y=FNX(-1)
80 DISP
90 END

STATISTICS

10 S8=SQR((S2-S1↑2/N)/(N-1))
20 S9=SQR((S4-S3↑2/N)/(N-1))
30 R9=(S5-S1*S3/N)/(N-1)/S8/S9
```

```
40 PRINT
50 PRINT "NO. POINTS ="N
60 PRINT
70 PRINT "X:   MEAN="S1/N;TAB25"ST.DEV.="S8
80 PRINT "Y:   MEAN="S3/N;TAB25"ST.DEV.="S9
90 PRINT
100 PRINT "CORR.COEFF.="R9
110 PRINT
120 END
```

COEFFICIENTS

```
5 IF N <= D2-W THEN 350
10 DISP "DEG.REG.=";
20 INPUT D1
30 IF D1 <= D2-W THEN 60
40 DISP "MAX DEG=";D2-W
50 END
60 IF W=0 THEN 340
70 T=0
80 FOR I=1 TO D1+1
90 B[I]=0
100 FOR J=1 TO D1-I+2
110 R=(I+J-1)*(D2+2-0.5*(I+J))
120 B[I]=B[I]+C[T+J]*C[R]
130 NEXT J
140 T=I*(D2+(3-I)/2)
150 NEXT I
160 R1=0
170 FOR I=2 TO D1+1
180 R1=R1+C[I*(D2+(3-I)/2)]↑2
190 NEXT I
200 T0=C[(D2+1)*(D2+2)/2]
210 T0=T0-C[D2+1]↑2
220 PRINT
230 PRINT "COEFFICIENTS"
240 PRINT
250 FORMAT F3.0,F12.4
260 FOR I=1 TO D1+1
270 WRITE (15,250)"B("I-1")="B[I]
280 NEXT I
290 PRINT
300 PRINT
310 PRINT "R SQUARE = "R1/T0
320 PRINT
330 END
340 IF N>D2 THEN 370
350 DISP "NOT ENOUGH POINTS"
360 END
370 P=W=1
```

```
380 D2=D2+1
390 FOR J=1 TO D2
400 C[P]=SQR(C[P])
410 FOR I=1 TO D2-J+1
420 C[P+I]=C[P+I]/C[P]
430 NEXT I
440 R=P+I
450 S=R
460 FOR L=1 TO D2-J
470 P=P+1
480 FOR M=1 TO D2+2-J-L
490 C[R+M-1]=C[R+M-1]-C[P]*C[P+M-1]
500 NEXT M
510 R=R+M-1
520 NEXT L
530 P=S
540 NEXT J
550 T=(D2+1)*(D2+2)/2
560 FOR I=1 TO D2-1
570 T=T-1-I
580 C[T]=1/C[T]
590 FOR J=1 TO D2-I
600 P=D2+1-I-J
610 P=P*(D2+1-(P-1)/2)-I
620 R=P-J
630 S=0
640 U=I+J+1
650 V=P
660 FOR K=1 TO J
670 V=V+U-K
680 S=S-C[R+K]*C[V]
690 NEXT K
700 C[P]=S/C[R]
710 NEXT J
720 NEXT I
730 C[1]=1/C[1]
740 GOTO 70
```

ESTIMATE

```
10 DISP "X=";
20 INPUT A
30 B=A
40 C=1
50 Y=FNZ0
60 END
```

TABLE

```
10 DISP "XMIN,XMAX,STEP=";
```

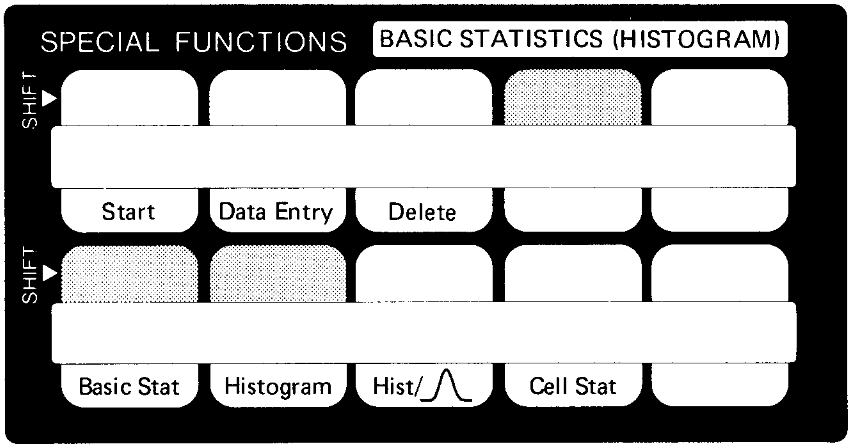
```
20 INPUT A,B,C
30 Y=FNZ0
40 END

(FNX)

10 DEF FNX(Z)
20 FOR I=2 TO D2
30 B[I+1]=B[I]*B[2]
40 NEXT I
50 B[D2+2]=Y
60 R=0
70 FOR I=1 TO D2+2
80 FOR J=I TO D2+2
90 R=R+1
100 C[R]=C[R]+B[I]*B[J]*Z
110 NEXT J
120 NEXT I
130 S1=S1+B[2]*Z
140 S2=S2+B[2]↑2*Z
150 S3=S3+Y*Z
160 S4=S4+Y*Y*Z
170 S5=S5+B[2]*Y*Z
180 N=N+Z
190 RETURN 0

(FNZ)

10 DEF FNZ(Z)
20 PRINT
30 FOR I=A TO B STEP C
40 Y=B[D1+1]
50 FOR J=D1 TO 1 STEP -1
60 Y=Y*I+B[J]
70 NEXT J
80 PRINT "X="I;TAB20"YHAT="Y
90 NEXT I
95 DISP
100 RETURN 0

VARIABLES USED: A,B,C,I,J,K,N,P,R,S,T,U,V,W,Y,
                D1,D2,R1,R9,S1,S2,S3,S4,S5,S8,S9,T0.
```

MODEL 30

HEWLETT·PACKARD HEWLETT·PACKARD HEWLETT·PACKARD HEWLETT·PACKARD

# BASIC STATISTICS, HISTOGRAM

| | |
|---|---|
| DESCRIPTION: | This program takes raw data and calculates the mean, standard deviation, skewness and kurtosis for the input data. A histogram and/or histogram with normal curve overlay may be plotted and cell statistics may be printed. |
| METHODS: | See reference. |
| SPECIAL CONSIDERATIONS: | The maximum number of cells allowed is 50. As data is input, the appropriate number of cells will be calculated. If data is deleted, then the maximum, minimum and range may be incorrect. |
| ACKNOWLEDGMENTS: | Robert W. Kopitzke, Hewlett-Packard |
| REFERENCES: | Croxton, F. E., Elementary Statistics, with Applications in Medicine and the Biological Sciences (Dover Publications, Inc., 1953). |
| SYSTEM SPECIFICATIONS: | 9830 (2K or 4K R/W)<br>9866 Printer or 9861 Typewriter |




OPERATION:

1. Type SCRATCH A, press EXECUTE.
2. Type LOADKEY 8, press EXECUTE.
3. Press RUN, START [SF Key 0].
4. The display reads, "ENT 1 FOR HISTOGRAM". If a histogram is desired, enter 1, press EXECUTE, go to step 5. Otherwise, enter 0, press EXECUTE, go to step 7.
5. The display reads, "OFFSET = ?". Enter offset, press EXECUTE.
6. The display reads, CELL WIDTH?". Enter cell width, press EXECUTE.
7. The display reads, "ENT 1 TO PRINT DATA?". If it is desired to print raw data, enter 1, press EXECUTE, go to step 8. Otherwise, enter 0, press EXECUTE, go to step 8.
8. The display reads, "PRESS DATA ENTRY KEY".
9. Use DATA ENTRY [SF Key 1] to enter data.
10. Any combination of the special function keys described below may be utilized with the limitation that BASIC STAT [SF Key 5] must precede the histogram key [SF Key 6] or [SF Key 7].

SPECIAL FUNCTIONS:

<u>DATA ENTRY</u> [SF Key 1]:
This key allows input of raw data. To use:
1. Press DATA ENTRY [SF Key 1].
2. The display reads, "X (I) = ?". Enter data point, $x_i$, press EXECUTE. Repeat step 2 for each data value.

<u>DELETE</u> [SF Key 2]:
This key allows incorrect data to be deleted. To use:
1. Press DELETE [SF Key 2].
2. The display reads, "DELETE X = ?". Enter the value to be deleted, press EXECUTE. Repeat steps 1 and 2 as often as desired.

Note: If this key is used, the minimum, maximum and range as printed may be incorrect.

<u>BASIC STAT</u> [SF Key 5]:
This key prints the mean, standard deviation, skewness, kurtosis, minimum, maximum and range.

HISTOGRAM [SF Key 6]:
This key prints the histogram.

HIST [SF Key 7]:
This key prints the histogram with normal curve overlay.

CELL STATS [SF Key 8]:
This key prints the cell number, lower cell limit, cell count, and % relative frequency.

```
RUN, START
HISTOGRAM
OFFSET=  0.5
CELL WIDTH =  1
MAX NO. OF CELLS =?10
DATA ENTRY
      3.3000
      1.4000
      4.6000
      1.2000
      5.6000
      9.4000
      2.8000
      6.3000
      5.2000
      3.3000
      5.9000
      8.7000
      9.9000
      7.8000
      9.5000
      3.3000
      2.5000
      3.6000
      8.2000
      4.9000
      6.5000
      2.1000
      6.4000
      4.1000
      3.3000

BASIC STATISTICS

N= 25
MEAN=       5.1920
STD.DEV=       2.6059
SKEWNESS=       0.3146
KURTOSIS=       1.8496
XMIN=       1.2000
XMAX=       9.9000
RANGE=       8.7000



HISTOGRAM WITH PLOT OVERLAY
```

```
EACH X = 0.50  PERCENT


      0.0000 .    *
             .     *
      1.0000 .        *
             .XXXXXXXXXXX*XXXX
      2.0000 .              *
             .XXXXXXXXXXXXXXXXXX*XXXXX
      3.0000 .                     *
             .XXXXXXXXXXXXXXXXXXXXXXXXX*XXXXXXXXXXXXXX
      4.0000 .                            *
             .XXXXXXXXXXXXXXXXXXXXXXXX      *
      5.0000 .                               *
             .XXXXXXXXXXXXXXXXXXXXXXXX      *
      6.0000 .                             *
             .XXXXXXXXXXXXXXXXXXXXXXXX   *
      7.0000 .                        *
             .XXXXXXXX             *
      8.0000 .                 *
             .XXXXXXXXXXXXXX*X
      9.0000 .           *
             .XXXXXXXX*XXXXXXXXXXXXXXX
     10.0000 .      *
```

CELL STATISTICS

| CELL# | LOWER LIMIT | NO. OF OBS | %RELATIVE FREQ |
|---|---|---|---|
| 1 | 0.0000 | 0 | 0.00000 |
| 2 | 1.0000 | 2 | 8.00000 |
| 3 | 2.0000 | 3 | 12.00000 |
| 4 | 3.0000 | 5 | 20.00000 |
| 5 | 4.0000 | 3 | 12.00000 |
| 6 | 5.0000 | 3 | 12.00000 |
| 7 | 6.0000 | 3 | 12.00000 |
| 8 | 7.0000 | 1 | 4.00000 |
| 9 | 8.0000 | 2 | 8.00000 |
| 10 | 9.0000 | 3 | 12.00000 |

```
START

10 DIM A[50]
20 FOR I=1 TO 50
30 A[I]=0
40 NEXT I
45 PRINT "HISTOGRAM"
50 S1=S2=S3=S4=N=R3=R4=R5=T1=0
60 R1=1E+99
70 R2=-R1
80 DISP "ENT 1 FOR HISTOGRAM";
90 INPUT S5
100 IF S5#1 THEN 165
110 DISP "OFFSET =";
120 INPUT O
121 PRINT "OFFSET= "O
130 DISP "CELL WIDTH";
140 INPUT C
145 PRINT "CELL WIDTH = "C
160 B=50
165 DISP "ENT 1 TO PRINT DATA";
170 INPUT D
180 IF D=0 THEN 200
190 PRINT "DATA"
200 DISP "PRESS DATA ENTRY KEY";
210 END

DATA ENTRY

10 DISP "X("N+1")=";
20 INPUT X
50 Z=FNX(1)
60 GOTO 10
70 END

DELETE

10 DISP "DELETE X =";
20 INPUT X
30 PRINT "DELETE:";
41 T1=1
50 Y=FNX(-1)
60 END

BASIC STATISTICS

10 M=S1/N
20 S=SQR((S2-M*M*N)/(N-1))
30 M3=S3/N-3*M*S2/N+2*M↑3
40 M4=S4/N-4*M*S3/N+6*(M↑2)*S2/N
```

```
50 M4=M4-3*M↑4
60 FORMAT F12.4,/,F12.4,/,F12.4,/,F12.4
65 PRINT
70 PRINT "N="N
80 WRITE (15,60)"MEAN="M,"STD.DEV="S,"SKEWNESS="M3/S↑3,"KURTOSIS="M4/S↑4
111 IF T1=0 THEN 120
112 PRINT "MIN,MAX,RANGE MAY BE INCORRECT"
120 WRITE (15,60)"XMIN="R1,"XMAX="R2,"RANGE="R2-R1
150 IF R3=0 THEN 170
160 PRINT "NO. TOO SMALL="R3
170 IF R4=0 THEN 190
180 PRINT "NO. TOO LARGE="R4
190 FOR J=1 TO 10
200 PRINT
210 NEXT J
220 END

HISTOGRAM

10 P=0
20 T=FNY(0)
30 END

HISTOGRAM/PLOT

10 P=1
20 W=FNY(1)
30 END

CELL STATISTICS

10 PRINT
11 FORMAT F5.0,F12.4,F11.0,F17.5
12 PRINT "CELL#";"   LOWER LIMIT";
13 PRINT TAB18;"   NO. OF OBS";"  %RELATIVE FREQ"
30 Y=FNC(1)
40 FOR I=1 TO B
45 WRITE (15,11)I,O+(I-1.5)*C,A[I],100*A[I]/N
50 NEXT I
60 FOR J=1 TO 10
70 PRINT
80 NEXT J
90 END

(FNX)

180 DEF FNX(Z)
182 IF D#1 THEN 190
183 WRITE (15,184)X
184 FORMAT F12.4
```

```
190 IF X>R1 THEN 210
200 R1=X
210 IF X<R2 THEN 230
220 R2=X
230 N=N+Z
240 S1=S1+X*Z
250 Y=X*X
260 S2=S2+Y*Z
270 Y=Y*X
280 S3=S3+Y*Z
290 Y=Y*X
300 S4=S4+Y*Z
310 IF S5#1 THEN 400
320 Y=INT((X-O)/C+1.5)
330 IF Y<1 THEN 370
340 IF Y>B THEN 390
350 A[Y]=A[Y]+Z
360 GOTO 400
370 R3=R3+Z
380 GOTO 400
390 R4=R4+Z
400 RETURN 0

(FNY)

10 DEF FNY(X)
11 FOR I=1 TO 50
12 IF R5>A[I] THEN 14
13 R5=A[I]
14 NEXT I
20 W=2.5*R5/N
30 U=(N*C/(2.5066*S))*(40/R5)
40 FORMAT 2F5.2
45 PRINT
46 PRINT
50 WRITE (15,40)"EACH X ="W;"  PERCENT"
60 PRINT
70 PRINT
75 Y=FNC(1)
80 FOR I=1 TO B+1
90 Y=O+(I-1.5)*C
100 FORMAT 2F12.4
110 WRITE (15,100)Y" .";
120 IF P=0 THEN 140
130 Z=FNZ(Y)
140 PRINT
145 IF I=B+1 THEN 190
150 Y=Y+0.5*C
160 PRINT TAB13;".";
170 Z=FNP(Y)
```

```
180 NEXT I
190 FOR J=1 TO 10
191 PRINT
192 NEXT J
193 END
200 DEF FNP(Y)
220 T=INT(U*EXP(-((Y-M)/S)↑2/2)+0.5)
230 R=INT((100*A[I]/N)/W)
240 IF T <= R THEN 320
245 IF R=0 THEN 280
250 FOR J=1 TO R
260 PRINT "X";
270 NEXT J
280 IF P=0 THEN 300
285 IF T=0 THEN 296
290 FOR J=R TO T-2+(R#0)
292 PRINT " ";
294 NEXT J
296 PRINT "*";
300 PRINT
310 RETURN 0
320 IF T=0 THEN 345
321 FOR J=1 TO T-(T=R)*P+(P=0)
330 PRINT "X";
340 NEXT J
345 IF P=0 THEN 360
350 PRINT "*";
360 IF T=R THEN 400
370 FOR J=1 TO R-T-1
380 PRINT "X";
390 NEXT J
400 PRINT
410 RETURN 0
420 DEF FNZ(Y)
440 T=INT(U*EXP(-((Y-M)/S)↑2/2)+0.5)
460 PRINT TABT"*";
480 RETURN 0

(FNC)

10 DEF FNC(Y)
20 J=50
30 IF A[J]#0 THEN 70
40 J=J-1
50 GOTO 30
70 B=J*(J<B)+B*(J >= B)
80 RETURN 0

VARIABLES USED: B,C,D,I,J,M,N,O,P,S,T,U,W,X,Y,Z,
                M3,M4,R1,R2,R3,R4,R5,S1,S2,S3,S4,S5,T1
```

# MATRIX INVERSION



**DESCRIPTION:**

This program takes an N x N square matrix and computes its inverse and determinant, with printed input and results. The matrix inversion routine is also available for use within a user program.

**METHODS:**

The program employs a modified Gauss-Jordan reduction technique using the maximum pivot strategy. This method is superior to the standard Gauss-Jordan elimination or the diagonal pivot strategy since it will successfully invert all but singular or very near-singular matrices. Also, by using maximum elements as the pivots, the accuracy of the results is maximized.

**SPECIAL CONSIDERATIONS:**

On the basic (2K) machine, this program will invert a 14 x 14 matrix. If the program is to be run on a 4K machine and a larger matrix is required, up to a 26 x 26 matrix may be inverted, by changing the dimension statement in line 10 of the START key [SF Key 0] to:

```
10 DIM A(26,26), RI(26), CI(26), Y(26)
```

If the user wishes to incorporate the defined function, FND(N), into his own program, in order to invert the matrix A, the maximum size matrix which can be inverted will depend on the size of the user program. In general, however, this function uses 4N + 430 words, where N is the size of the matrix, A.

If the matrix is very near-singular, the inverse matrix may not be accurate. This situation is indicated by a very small value for the determinant. If this is the case, the matrix should be re-inverted and a comparison of the result with the original matrix will give an indication of the accuracy.

**ACKNOWLEDGMENTS:**

John H. Nairn, Hewlett-Packard

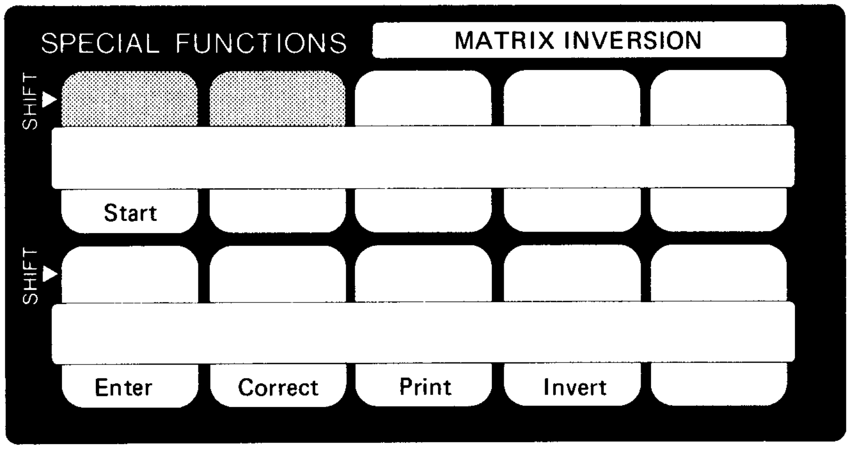
**REFERENCES:**

Carnahan, B., Luther, H.A., and Wilkes, J.O., Applied Numerical Methods (New York: John Wiley & Sons, Inc., 1969), pp. 282-284.

Kuo, Shan S., Numerical Methods and Computers (Reading, Mass.: Addison-Wesley Publishing Co., 1965), pp. 166-177.

**SYSTEM SPECIFICATIONS:**

9830 (2K or 4K R/W)
9866 Printer or 9861 Typewriter




**OPERATIONS:**

1. Type LOADKEY 9 and press EXECUTE.
2. Press RUN, START [SF Key 0]. If RUN is not pressed, the display will show "ERROR 41 IN LINE 20".
3. Press ENTER [SF Key 5]. The program will ask for the order of the matrix to be inverted. Enter this number and press EXECUTE. The display will show the value of i and j (the row and column respectively) of the element to be entered. Enter the elements of the matrix pressing EXECUTE after each entry. If an error is made, continue entering the elements in order. Corrections, if necessary, can be made after entry is complete.
4. When the display shows "MATRIX ENTERED", the PRINT key [SF Key 7] may be pressed to print the matrix.
5. If any elements of the matrix are incorrect, press the CORRECT key [SF Key 6]. The program will ask for the i and j values of the wrong element. Enter these numbers and press EXECUTE. The program will then ask for the correct value of that element. Enter this number and press EXECUTE. Repeat this procedure until all wrong elements have been corrected. The PRINT key [SF Key 7] may be used at any time.
6. When the matrix has been correctly entered, press INVERT [SF Key 8]. The value of the determinant and the inverse matrix will be calculated and printed.
7. The matrix may be re-inverted by pressing INVERT [SF Key 8] to check the results, or a new matrix may be entered by going to step 3.

**SPECIAL INSTRUCTIONS:**

There is a MATRIX ROM available for the 9830 which gives the user full matrix capability. For the user who does some matrix manipulation, but not enough to justify the inclusion of the MATRIX ROM into his system, the use of FOR-NEXT loops makes the writing of routines for addition, subtraction, and multiplication of matrices a simple matter. Obtaining the inverse of a matrix, however, is a much more difficult task. For this reason, we have placed the matrix inversion routine on file #10 of the tape cassette as a separate, defined function, FND(N) which the user may incorporate into his own program.

The listing for this function is identical to the one for the function FND(N) in the Matrix Inversion program, except that it contains the additional line

12 DIM RI(20), CI(20), Y(20) .

The user may include this function in his program by typing MERGE 10, L where L is the line number in his program where the function FND(N) is to begin. (Be sure that lines L through L + 126 are not used by the program.) In general, it is best to place the function at the end of the program.

To use the routine, the users program must dimension and define the elements of a matrix, A. When the function, FND(N) is executed within the users program the parameter N should be the size of the matrix, A. The value returned by the function is the determinant of the original matrix, and the matrix A is replaced by its inverse.

```
RUN, START
ENTER
ORDER=?3

ENTER THE VALUES: -3, 8, 5, 2, -7, 4, 1, 9, -6.

PRINT

     -3.000000          8.000000          5.000000
      2.000000         -7.000000          4.000000
      1.000000          9.000000         -6.000000


INVERT
DETERM =  235.0000000

INVERSE MATRIX:

      0.025532          0.395745          0.285106
      0.068085          0.055319          0.093617
      0.106383          0.148936          0.021277


TO CHECK THE RESULTS, WE WILL INVERT AGAIN.
INVERT
DETERM =  4.25532E-03

INVERSE MATRIX:

     -3.000000          8.000000          5.000000
      2.000000         -7.000000          4.000000
      1.000000          9.000000         -6.000000
```

```
START

10 DIM A[14,14],R[14],C[14],Y[14]
20 N=Y[1]=0
30 END

ENTER MATRIX

10 DISP "ORDER=";
20 INPUT N
30 FOR I=1 TO N
40 FOR J=1 TO N
50 DISP I;J;
60 INPUT A[I,J]
70 NEXT J
80 NEXT I
90 DISP "MATRIX ENTERED"
100 END

CORRECT

10 DISP "I,J TO CORRECT=";
20 INPUT I,J
30 DISP "CORRECT VALUE=";
40 INPUT A[I,J]
50 GOTO 10

PRINT

10 I=FNP0
20 END

INVERT

10 D9=FNDN
20 PRINT "DETERM = ";D9
30 PRINT
40 PRINT "INVERSE MATRIX:"
50 I=FNP0
60 END

(FND)

10 DEF FND(N)
14 D9=1
16 FOR K=1 TO N
18 P9=0
20 FOR I=1 TO N
22 FOR L=1 TO K-1
24 IF I=R[L] THEN 46
```

```
26 NEXT L
28 FOR J=1 TO N
30 FOR L=1 TO K-1
32 IF J=C[L] THEN 44
34 NEXT L
36 IF ABSA[I,J] <= ABSP9 THEN 44
38 P9=A[I,J]
40 R[K]=I
42 C[K]=J
44 NEXT J
46 NEXT I
48 IF P9#0 THEN 52
50 RETURN 0
52 D9=D9*P9
54 FOR J=1 TO N
56 A[R[K],J]=A[R[K],J]/P9
58 NEXT J
60 A[R[K],C[K]]=1/P9
62 FOR I=1 TO N
64 IF I=R[K] THEN 76
66 FOR J=1 TO N
68 IF J=C[K] THEN 72
70 A[I,J]=A[I,J]-A[I,C[K]]*A[R[K],J]
72 NEXT J
74 A[I,C[K]]=-A[I,C[K]]/P9
76 NEXT I
78 NEXT K
80 FOR J=1 TO N
82 FOR I=1 TO N
84 Y[C[I]]=A[R[I],J]
86 NEXT I
88 FOR I=1 TO N
90 A[I,J]=Y[I]
92 NEXT I
94 NEXT J
96 FOR I=1 TO N
98 FOR J=1 TO N
100 Y[R[J]]=A[I,C[J]]
102 NEXT J
104 FOR J=1 TO N
106 A[I,J]=Y[J]
108 NEXT J
110 NEXT I
112 FOR K=1 TO N
114 Y[R[K]]=C[K]
116 NEXT K
118 FOR I=1 TO N
120 FOR J=1 TO N-1
122 IF Y[J] <= Y[J+1] THEN 132
```

```
124 P9=Y[J]
126 Y[J]=Y[J+1]
128 Y[J+1]=P9
130 D9=-D9
132 NEXT J
134 NEXT I
136 RETURN D9
```

(FNP)

```
10 DEF FNP(X)
20 PRINT
30 FOR I=1 TO N
40 FOR J=1 TO N
50 K=A[I,J]
60 FORMAT 2F15.6
70 FORMAT 2E15.6
80 IF (ABSK<1E+06 AND ABSK>1E-06) OR K=0 THEN 110
90 WRITE (15,70)K,
100 GOTO 120
110 WRITE (15,60)K,
120 IF J/5#INT(J/5) THEN 140
130 PRINT
140 NEXT J
150 PRINT
160 NEXT I
170 PRINT
180 PRINT
190 RETURN 0
```

VARIABLES USED: I,J,K,L,N,D9,P9.

MODEL 30

HEWLETT · PACKARD

## ADDITIONAL FUNCTIONS

| | |
|---|---|
| DESCRIPTION: | This section contains a series of defined functions which may be used alone, or as part of the user's program. They are, for the most part, the more common mathematical functions that are not available in the standard BASIC compiler of the 9830. |
| METHODS: | Most of the mathematical functions are obtained by the straight-forward solution of the defining equations (see References), with some provisions for special cases. More detail concerning methods used is given under the description of each individual function when necessary. |
| SPECIAL CONSIDERATIONS: | None |
| ACKNOWLEDGMENTS: | John H. Nairn, Hewlett-Packard |
| REFERENCES: | Abramowitz, M., and Stegun, I. A., Handbook of Mathematical Functions (Washington D. C., U. S. Government Printing Office, NBS Applied Mathematics Series #55, 1964) |
| SYSTEM SPECIFICATIONS: | 9830 (2K R/W) |

## ADDITIONAL FUNCTIONS LIST

| FUNCTION | NAME | FILE | PARAMETERS | VARIABLES USED |
|---|---|---|---|---|
| ASIN(X) | FNA | 11 | none | none |
| ACOS(X) | FNB | 12 | none | none |
| ASEC(X) | FNC | 13 | none | none |
| ACSC(X) | FND | 14 | none | none |
| ACOT(X) | FNE | 15 | none | none |
| SINH(X) | FNF | 16 | none | none |
| COSH(X) | FNG | 17 | none | none |
| TANH(X) | FNH | 18 | none | none |
| SECH(X) | FNI | 19 | none | none |
| CSCH(X) | FNJ | 20 | none | none |
| COTH(X) | FNK | 21 | none | none |
| ASINH(X) | FNL | 22 | none | none |
| ACOSH(X) | FNM | 23 | none | none |
| ATANH(X) | FNN | 24 | none | none |
| ASECH(X) | FNP | 25 | none | none |
| ACSCH(X) | FNQ | 26 | none | none |
| ACOTH(X) | FNR | 27 | none | none |
| DEG(X) | FNS | 28 | none | none |
| RAD(X) | FNT | 29 | none | none |
| DDG(X) | FNU | 30 | D,M,S | D,M,S |
| DMS(X) | FNV | 31 | none | D,M,S |
| RECPOL(Z) | FNW | 32 | X,Y or R,A | X,Y,R,A |
| CINT(X) | FNY | 33 | none | none |
| ROUND(M) | FNZ | 34 | N | N |
| BIN(M) | FNA | 35 | N | N, Q0, Q1, Q2, Q3 |
| FACT(X) | FNB | 36 | none | Q4, Q5 |
| GAMMA(X) | FNC | 37 | none | Q6, Q7, Q8, Q9, K9 |
| BESSEL(X) | FND | 38 | N | N, E, Q0, Q1, Q2, Q3, Q4, Q5 |

OPERATION:

The most convenient method for using the additional functions is to place one or more of the available functions on the special function keys (one function per key).

1. Press FETCH, and the particular Special Function Key (SFK) on which that function is to be located. If that key is not being used, the display will read "KEY".
2. Type LOAD n where n is the file number on which the function to be called is located on the tape (see Additional Function List for file numbers), and press EXECUTE. When the function has been loaded onto the SFK, the display will again read "KEY".
3. If more functions are to be loaded onto the SFK's, repeat steps 1 and 2 for each function.
4. When all of the desired functions have been placed on the keys, press CLEAR, type END, and press EXECUTE.
5. If this particular combination of functions will be used frequently, the user may desire to place the entire set of SFK's that have just been defined onto a single tape file. This may be done by placing a previously marked tape cassette in the machine, and typing STOREKEY n , where n is some available file on which the set of SFK's is to be stored.
6. Any of the defined functions may now be used either from the keyboard or within a user program. (For a detailed description of the use of defined functions, see the 9830 Operating Manual.)

The available functions have been given arbitrary designations FNA, FNB, FNC, etc. The user may redefine the function names to suit his needs. In the case where two functions with the same name (say FNC) are to be used in the same program, it will be necessary to rename one of them.

Most of the available functions are functions of one variable (eg., ASIN(X) ) and require no special consideration. Others are functions of more than one variable (eg., BIN(M) which is the binomial coefficient of Nth order and Mth degree) and require that the other variables (called parameters) be defined either from the keyboard or within the program before the function is executed. If additional parameters must be defined, these are given in the description of that function and also in the "Parameters" column of the Additional Functions List.

Some of the functions require the use of additional variables in order to evaluate the function. If the user wishes to use these functions within his own program, he should avoid the use of these variables. A list of the variables used by each function is given in the Additional Functions List.

ADDITIONAL FUNCTIONS:

| | | |
|---|---|---|
| ASIN(X): | Inverse sine of x | $-1 \le x \le 1$ |
| ACOS(X): | Inverse cosine of x | $-1 \le x \le 1$ |
| ASEC(X): | Inverse secant of x | ABS(X) $\ge 1$ |
| ACSC(X): | Inverse cosecant of x | ABS(X) $\ge 1$ |
| ACOT(X): | Inverse cotangent of x | |

These five inverse circular functions (Inverse tangent of x is part of the BASIC language, ATN(X) ) are multivalued functions of x. The calculated value of the function is the principal value. (See Reference, sections 4.3.45 and 4.4.13)

| | | |
|---|---|---|
| SINH(X): | Hyperbolic sine of x | |
| COSH(X): | Hyperbolic cosine of x | |
| TANH(X): | Hyperbolic tangent of x | |
| SECH(X): | Hyperbolic secant of x | |
| CSCH(X): | Hyperbolic cosecant of x | $X \neq 0$ |
| COTH(X): | Hyperbolic cotangent of x | $X \neq 0$ |

(See Reference, sections 4.5.1 to 4.5.6)

| | | |
|---|---|---|
| ASINH(X): | Inverse hyperbolic sine of x | |
| ACOSH(X): | Inverse hyperbolic cosine of x | $X \ge 1$ |
| ATANH(X): | Inverse hyperbolic tangent of x | ABS(X) $< 1$ |
| ASECH(X): | Inverse hyperbolic secant of x | $O < X \le 1$ |
| ACSCH(X): | Inverse hyperbolic cosecant of x | $X \neq O$ |
| ACOTH(X): | Inverse hyperbolic cotangent of x | ABS(X) $> 1$ |

(See Reference, sections 4.6.20 to 4.6.25)

DEG(X): Radian to degrees conversion
Takes the given angle in radians and converts it to degrees.

RAD(X): Degrees to radians conversion
Takes the given angle in degrees and converts it to radians.

Note: The angle returned by the functions DEG(X) and RAD(X) is always greater than zero and less than 360 degrees or $2\pi$ radians. Thus, the combination function DEGRAD(X) will take any number of degrees and convert it to a number in the range zero to 360 degrees. The same is true of RADDEG(X) for radians.

DDG(X): Converts degrees, minutes, seconds to decimal degrees

The argument of the function is a "dummy" argument used to execute the function and may be set to any value. When the function is executed, D, M, and S must be defined. The program will take the values for D, M, and S and return the equivalent angle in decimal degrees.

DMS(X): Converts decimal degrees to degrees, minutes, seconds

This program takes an angle in decimal degrees and calculates the equivalent degrees, minutes, and seconds. These results are assigned to the variables D, M, and S respectively, and may be used within a program. If the function is used from the keyboard, the result which is returned is displayed as a single number whose integer part is the number of degrees, the first two digits after the decimal represent the number of minutes, and the remaining digits represent the number of seconds. For example, if we EXECUTE from keyboard the expression FNV(123.456), the result is 123.27216 which is interpreted as 123 degrees, 27 minutes, 21.6 seconds.

RECPOL(Z): Rectangular-Polar Conversions

This program converts rectangular coordinates to polar coordinates, and vice-versa. The argument of the function, Z, determines the direction of the conversion. If X and Y have been defined and the function is executed with Z=O, the polar coordinates are calculated and assigned to the variables R and A (the radius and angle respectively). If R and A are defined and the function is executed with Z=1, the rectangular coordinates are calculated and assigned to the variables X and Y.

CINT(X): Central integer value of x

The standard BASIC function INT(X) returns the largest integer less than or equal to the given value, x. For example, INT(-3.14) = -4 . It is sometimes useful to have a function which returns only the integral part of a number, whether that number is positive or negative. This is the function of CINT(X). For example, CINT(-3.14) = -3 .

ROUND(M): Rounds the number M to the power of N indicated

The parameter N is used to indicate where the number is to be rounded and must be defined before the function is executed. The argument of the function is rounded to the power of 10 indicated by N. For example, if N = -3, then ROUND (3.14159) would give 3.142.

BIN(M): Binomial Coefficient C(N,M)

This program calculates the binomial coefficient of Nth order and Mth degree. The parameter N must be defined (either from the keyboard or within the program) before the binomial function is executed. N and M must be integers, but are otherwise unrestricted (positive, negative, or zero).

FACT(X): Factorial function of x

This program calculates x! where x is zero or a positive integer, less than 70. If x is greated than or equal to 70, an overflow error will result.

GAMMA(X): Gamma function of x

This program calculates the gamma function of x for $x \le 70.75$ and x not a negative integer. If x is a negative integer, the gamma function has a singularity and an overflow error will result. If x is greater than 70.75, GAMMA(x) is greater than $10^{99}$, and the program will return the common (base 10) logarithm of GAMMA(x).

BESSEL(X): Bessell function, Jn(x)

This program calculates the nth order Bessel function, Jn(x), for n a positive integer or zero, and $X \ge 0$. The parameter N is the order of the Bessel function and must be defined before the function is executed. In addition to returning the value of the Bessel function, the variable E is assigned a value which is an estimate of the accuracy of the result. It roughly corresponds to the number of decimal places to which the result is reliable. E may be used within a program, or viewed from the keyboard by typing an E and pressing EXECUTE. If E is negative, the result is totally unreliable. In general, the function returns a good estimate of Jn(x) when x is less than about 20. The accuracy generally increases with the order of the Bessel function.

```
ASIN(X)

10 DEF FNA(X)=ATN(X/SQR(1-X*X+1E-99))

ACOS(X)

10 DEF FNB(X)=ATN(SQR(1-X*X)/(X+1E-99))+2*ATN1E+99*(X<0)

ASEC(X)

10 DEF FNC(X)=ATN(SGNX*SQR(X*X-1))+2*ATN1E+99*(X<0)

ACSC(X)

10 DEF FND(X)=ATN(SGNX/SQR(X*X-1+1E-99))

ACOT(X)

10 DEF FNE(X)=ATN(1/(X+1E-99*(X=0)))+2*ATN1E+99*(X<0)

SINH(X)

10 DEF FNF(X)=(EXPX-1/EXPX)/2

COSH(X)

10 DEF FNG(X)=(EXPX+1/EXPX)/2

TANH(X)

10 DEF FNH(X)=SGNX*(ABSX>99)+1-2/(EXP(2*X*(ABSX <= 99))+1)

SECH(X)

10 DEF FNI(X)=2*(ABSX<99)/(EXP(X*(ABSX<99))+1/EXP(X*(ABSX<99)))

CSCH(X)

10 DEF FNJ(X)
20 IF ABSX<99 THEN 40
30 RETURN 0
40 RETURN 2/(EXPX-1/EXPX)

COTH(X)

10 DEF FNK(X)
20 IF ABSX<99 THEN 40
30 RETURN SGNX
40 RETURN 1+2/(EXP(2*X)-1)
```

ASINH(X)

```
10 DEF FNL(X)=LOG(X+SQR(X*X+1))
```

ACOSH(X)

```
10 DEF FNM(X)=LOG(X+SQR(X*X-1))
```

ATANH(X)

```
10 DEF FNN(X)=LOG((1+X)/(1-X))/2
```

ASECH(X)

```
10 DEF FNP(X)=LOG(1/ABSX+SQR(1/(X*X)-1))
```

ACSCH(X)

```
10 DEF FNQ(X)=SGNX*LOG(1/ABSX+SQR(1/(X*X)+1))
```

ACOTH(X)

```
10 DEF FNR(X)=LOG((X+1)/(X-1))/2
```

DEG(X)

```
10 DEF FNS(X)=180*(X/PI-2*INT(X/2/PI))
```

RAD(X)

```
10 DEF FNT(X)=2*PI*(X/360-INT(X/360))
```

DDG(X)

```
10 DEF FNU(X)=D+(M+S/60)/60
```

DMS(X)

```
10 DEF FNV(X)
20 S=ABSX
30 D=SGNX*INTS
40 S=(S-ABSD)*60
50 M=SGNX*INTS
60 S=SGNX*(S-ABSM)*60
70 RETURN SGNX*(ABSD+ABSM/100+ABSS/10000)
```

RECPOL(Z)

```
10 DEF FNW(Z)
```

```
20 IF Z=0 THEN 70
30 X=R*COSA
40 Y=R*SINA
60 RETURN 0
70 R=SQR(X*X+Y*Y)
80 A=ATN(Y/(X+(1E-90)*(X=0)))+2*SGNY*ATN1E+99*(X<0)
100 RETURN 0
```

CINT(X)

```
10 DEF FNY(X)=SGNX*INTABSX
```

ROUND(M)

```
10 DEF FNZ(M)=SGNM*INT(ABSM*10↑(-N)+0.5)*10↑N
```

BIN(M)

```
10 DEF FNA(M)
20 IF M>0 THEN 40
30 RETURN M=0
40 Q0=ABSN+(M-1)*(N<0)
50 Q2=(N >= M)*(N >= 0)+((-1)↑M)*(N<0)
60 Q1=M*(Q0 >= 2*M)+(Q0-M)*(Q0<2*M)
70 FOR Q3=1 TO Q1
80 Q2=Q2*(Q0-Q1+Q3)/Q3
90 NEXT Q3
100 RETURN Q2
```

FACT(X)

```
10 DEF FNB(X)
20 Q4=1
30 FOR Q5=2 TO ABSX
40 Q4=Q4*Q5
50 NEXT Q5
60 RETURN Q4
```

GAMMA(X)

```
10 DEF FNC(X)
20 IF X<10 THEN 80
30 Q9=X
40 GOSUB 160
50 IF X>70.75 THEN 70
60 RETURN EXPQ8
70 RETURN Q8/LOG10
80 K9=10-INTX
90 Q6=1
100 FOR Q7=0 TO (K9-1)
```

```
110 Q6=Q6*(X+Q7)
120 NEXT Q7
130 Q9=X+K9
140 GOSUB 160
150 RETURN EXPQ8/Q6
160 Q8=(Q9-0.5)*LOGQ9-Q9+LOG(2*PI)/2
170 Q8=Q8+1/(12*Q9)-1/(360*Q9↑3)+1/(1260*Q9↑5)-1/(1680*Q9↑7)
180 RETURN
```

BESSEL(X)

```
10 DEF FND(X)
20 Q2=X/2
30 Q1=0
40 Q3=Q4=1
50 FOR Q5=2 TO ABSN
60 Q3=Q3*Q5
70 NEXT Q5
80 Q0=Q3=1/Q3
90 Q3=-Q3*Q2/Q4*Q2/(N+Q4)
100 IF ABSQ3/Q0<1E-10 THEN 150
110 Q0=Q0+Q3*(Q3>0)
120 Q1=Q1+Q3*(Q3<0)
130 Q4=Q4+1
140 GOTO 90
150 E=10+LGTABS(1+Q1/Q0)
160 RETURN (Q0+Q1)*(Q2+(Q2=0)*(N=0))↑N
```

# PROGRAM SUBMITTAL INSTRUCTIONS AND FORMS

# Program Submittal Instructions

Hewlett-Packard maintains a library of calculator programs available to HP users. New contributed programs are solicited to increase the number of programs available to you and other users.

Each program you submit will be evaluated for possible publication in the HP Keyboard or a program library, or inclusion in the HP Calculator Program Catalog.

Entries in the Calculator Program Catalog are of two types — supported programs from Hewlett-Packard and programs contributed by users. HP compiles but does not test or maintain contributed programs. Maintenance of these programs is the responsibility of the person submitting the program, since he is the most knowledgeable about his entry. Each program will bear either an original date or a revised date indicating a revision or correction.

Contributed programs should be typewritten and completely documented so that other users are able to easily understand and operate them. Most user questions can be obviated by having a person unfamiliar with your program read it and try the numerical example before submitting it. Completion of the Program Submittal Form and inclusion of the listed material and recorded mag cards or cassette will help insure adequately documented, publishable programs. The magnetic cards or cassette can be returned on request. Submittal and user instruction forms are found on the next few pages. Contact the HP sales office in your area if you need additional submittal forms.

Programs may be withdrawn from the Calculator Program Catalog by writing to Hewlett-Packard, and stating the reason for withdrawal, such as when the program is replaced with a redesigned version. Address all correspondence to:

Hewlett-Packard Company
Calculator Products Division
P. O. Box 301
Loveland, Colorado 80537

Attention: Applications Services

# Program Submittal Form

Hewlett-Packard Company
Calculator Products Division

1. Initial Submission ☐ Revision ☐
2. Equipment required: ______
3. Program Title: ______
4. Program Description and Application: ______
5. Contributor's Name: ______

   Organization: ______ Title: ______

   Address: ______

   Telephone: ______
6. Do you want your name to appear in the catalog?............... yes ☐ no ☐
7. Do you want your organization's name to appear in the catalog?.. yes ☐ no ☐
8. May an HP customer contact you directly?......................yes ☐ no ☐
9. Check (√) to be sure each item below is included.

   ☐ Program introduction, including equations solved and text reference.
   ☐ User instructions.
   ☐ Numeric input/output example, including sample plot or printer tape if applicable.
   ☐ Program listing (including program steps and step codes if applicable).
   ☐ Recorded magnetic cards or cassette.

   Shall we return original recorded cards/cassette ☐ or blank cards/cassette ☐ ?
10. Acknowledgement and Agreement

    To the best of my knowledge, this contributed program is free of any proprietary information belonging to any person or organization. I am making this program available to Hewlett-Packard, and I agree that HP may reproduce, publish, and use it, and authorize others to do so without obligation or liability of any kind.

______ (Signature) ______ (Date)