June 1982

Proceedings of the Third Annual Conference
of the National Computer Graphics Association
Anaheim, California, June 13-17, 1982

DATAPLOT—AN INTERACTIVE HIGH-LEVEL LANGUAGE FOR GRAPHICS, NON-LINEAR FITTING, DATA ANALYSIS, AND MATHEMATICS

James J. Filliben
Center for Applied Mathematics
National Bureau of Standards
Washington, D. C. 20234

ABSTRACT

This paper describes the design philosophy and features of DATAPLOT—a high-level (free-format English-like syntax) language for:

1) graphics (continuous or discrete);
2) fitting (linear or non-linear);
3) general data analysis;
4) mathematics.

DATAPLOT was developed originally in 1977 in response to data analysis problems encountered at the National Bureau of Standards. It has subsequently been the most heavily-used interactive graphics and non-linear fitting language at NBS. It is a valuable tool not only for "raw" graphics, but also for manuscript preparation, modeling, data analysis, data summarization, and mathematical analysis. DATAPLOT may be run either in batch mode or interactively, although it was primarily designed for (and is most effectively used in) an interactive environment. DATAPLOT graphics may appear on many different types of output devices. Due to its modular design and underlying ANSI FORTRAN (PFORT) code, DATAPLOT is portable to a wide variety of computers.

The paper is divided into four general parts: part 1 deals with background motivation and design philosophy; part 2 deals with capabilities and features; part 3 deals with examples and applications; part 4 deals with implementation features.

1. INTRODUCTION

The laboratory scientist in a research environment has become increasingly dependent on the computer for carrying out the various facets of his/her investigations. Indeed, over the last 10 years, there has been an evolution (a revolution!) of computer capabilities with the net result that no component in the research process is left untouched by the computer—it is being used to design the experiment, to run the experiment, to log the data, to verify, process, analyze, and summarize the data, and to prepare the final research manuscript.

My personal vantage point in this laboratory "computerization" has been that of a statistical consultant at the National Bureau of Standards. In my interactions with laboratory scientists over the last 12 years, I have been able to observe first hand how the individual laboratory scientists are coping with the advent of computers in the laboratory process. Many such scientists approach the computer with mixed reactions—being spurred on by the knowledge that there exists the potential for enormous time saving, and yet being somewhat hesitant by the reality that the computer demands an up-front commitment of time and effort in terms of familiarization with hardware and software. By and large, however, the trend is strikingly positive—there being fewer cases of active resistance against the intrusion of the computer, and many more cases of active acceptance of the merits of the computer.

The majority of "bench scientists" are viewing the computer as a tool (even a "friendly" tool) which can be profitably applied at every step of their research effort. It is no longer being applied as a last resort; rather, it is being considered as the prime option.

In this light, we see laboratory scientists asking (even demanding) more and more in the way of capabilities from their local computer operation, which in turn is resulting in upgraded requirements for hardware and software. These demands have been anticipated by the hardware manufacturers and have been met by the development of faster, more powerful maxi/mini computers with larger mass storage capabilities.

More importantly, however, the feature of computing which has had greatest impact on the average laboratory scientist has been the widespread introduction of interactive (as opposed to batch) oriented systems. This has (via terminals) literally bought the power of the computer into the scientist's laboratory, and has scaled down the time frame for computer analyses from days and hours to minutes and seconds. This compression of time has had the important practical effect of allowing the analyst the luxury of time-continuity in a research analysis—a luxury that so frequently was lost in the overnight turnarounds accompaning the batch-oriented systems of a decade ago.

Along with interactive-oriented operations, a second innovation which is impacting heavily on the scientist is the widespread availability of moderately-priced, high-resolution display terminals. These terminals with their enhanced resolution have given the scientist the ability to generate graphics output with resolution orders of magnitude better than discrete terminals and batch-hardcopy output. High-resolution terminals have allowed the laboratory scientist to "think continuous" (as opposed to discrete), and has provided a day-in/day-out graphics tool for research which has never before been practically available. The scope of potential graphics activities is limitless—as the variety of applications discussed in the computer graphics literature attests to. The graphics tool allows the research scientist to visualize process mechanisms, to gain insight into underlying structure, to thoroughly test modeling assumptions, to substantiate research conclusions, and to communicate such conclusions to the larger scientific community.

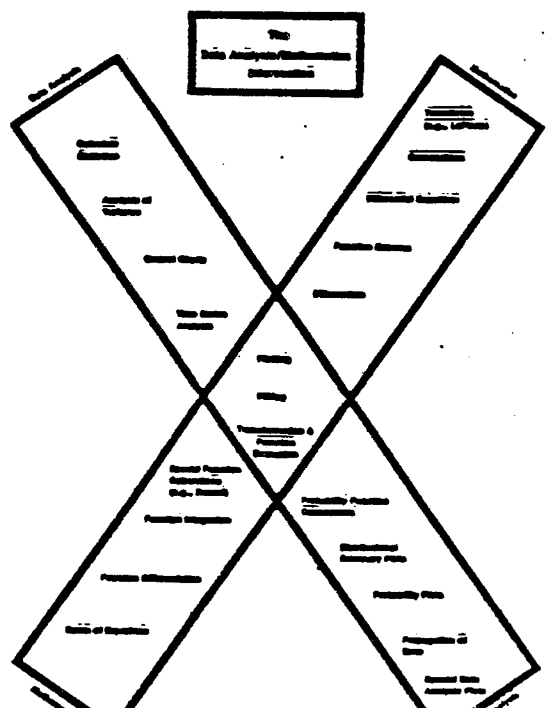
The remainder of this paper discusses a set of graphics activities which has high practical importance because of its universal presence in general scientific investigations. Further, the paper describes the accompaning development of software aimed at allowing the research scientist to take full advantage of the above-described advent of interactively-oriented operating systems and high-resolution display terminals. Such complementary software is directed at permitting the laboratory scientist to carry out necessary graphics activities in as easy and effective a fashion as possible—thus enhancing the graphics tool in the true sense of the word "tool". From the laboratory scientist's point of view, such ease-of-use is so critically important, because it not only allows for more insightful and thorough analyses, but also preserves the researcher's time-continuity which is so important in effectivly carrying out a scientific investigation.

Section 2 describes a set of activities (graphics and otherwise) which are common to most scientific investigations and which in turn define the target set of activities for which a software system (DATAPLOT) has been developed. Section 3 decribes DATAPLOT capability features. Section 4 gives code and output examples for a variety of specific graphics activities. Section 5 describes DATAPLOT implementation features.

2. DATAPLOT DESIGN PHILOSOPHY: THE DATA ANALYSIS/MATHEMATICS INTERSECTION

The design philosphy of DATAPLOT is based on the importance and relatedness of the two general areas:

1) data analysis; and
2) mathematics.



In a scientific environment—whether at a university, in private industry, or in a government laboratory—it is near-impossible to carry out a research project without some facet of the analysis being drawn from both areas. Data analysis and mathematics are complementary areas, and a research project which is restricted (by software) to only one without the benefits of the other is necessarily limited in terms of the thoroughness of the analysis and in terms of the efficient expenditure of the analyst's time.

The X-diagram enumerates several important sub-areas of data analysis and mathematics—specific activities that an analyst might typically engage in to carry out a research project. The upper-left to lower-right diagonal is the data analysis diagonal—it includes the familiar tools which the statistician routinely applies. The lower-left to upper-right diagonal includes the tools that a mathematician might commonly employ.

Several comments are appropriate with respect to the X-diagram. First of all (as mentioned above), it is rare that an analysis (whether statistical or mathematical) does not include components from the complementary diagonal. Secondly, the individual components of the diagonals—especially the data analysis diagonal—have a heavy representation of techniques which are graphical in nature (control charts, probability plots, etc.). Thirdly, of all the individual elements on the two diagonals, there is a select subset of three activities which may be properly classified as both statistical and mathematical:

1) plotting;
2) fitting; and
3) function evaluation & variable transformation.

These 3 kernel activities are common to both diagonals and are of prime importance to both. In terms of factors which affect the use of statistical software, we thus see that the ability/inability to plot general data sets/ functions, to fit general models, and to evaluate/ transform general functions are key factors in the power and ease-of-use of the system.

With the above underlying interlocking nature of data analysis and mathematics in mind, the DATAPLOT design philosophy thus becomes straightforward:

1) Firstly, concentrate on developing 3 powerful and flexible primary commands so as to allow the analyst to easily carry out the above 3 kernal capabilities; these 3 kernal commands are:

    PLOT (for plotting);
    FIT (for fitting); and
    LET (for function evaluation & variable transformation;

2) Secondly, develop the necessary set of secondary commands to handle the associated activities on the arms of the diagonals (and for other activities not illustrated).

3) Thirdly, develop the necessary set of tertiary commands to handle the various subsidiary activities (e.g., defining the line colors on a plot) that the above primary and secondary commands might require.

Thus in short, the unique design philosophy for DATAPLOT is seen to be a direct by-product of the underlying interrelationship between data analysis and mathematics with its common kernal of 3 activities; it is these 3 kernal activities which play a central role in the overall development of the language.

3. DATAPLOT CAPABILITIES

The purpose of this section is to enumerate various DATAPLOT capability features. As discussed in the previous section, DATAPLOT has capabilities in 4 areas:

1) graphics;
2) fitting;
3) data analysis;
4) mathematics.

For sake of discussion, the above 4 areas will have 2 additional partitions: the graphics area will be partitioned into 2 parts—one dealing with general graphics and the other dealing with diagrammatic graphics; the data analysis area will also be partitioned into 2 parts—one dealing with graphical data analysis and the other dealing with non-graphical data analysis. Thus the framework for the discussion of capabilities will be the following 6 areas/sub-areas:

1) general graphics;
2) diagrammatic graphics;
3) fitting;
4) graphical data analysis;
5) non-graphical data analysis;
6) mathematics.

3.1 General Graphics Capabilities

Graphics capabilities include continuous display terminal plots (e.g., Tektronix), discrete (narrow-width or wide-carriage) terminal plots (e.g., TI 700), high-speed printer plots, high-quality secondary output plots (e.g., Calcomp); on-line interactive definition and plotting of functions; data plots; multi-trace plots; linear or log scale plots; plots with or without labels, titles, frames, tic marks, grid lines, legends, legend boxes, arrows, etc.; automatic hardcopying of plots; 3-d plots of functions and/or data; multi-colored graphics; all of above for full data sets or subsets of data.

Examples of use of such graphics capabilities are as follows:

```
PLOT Y
PLOT Y X
PLOT Y X LAB
PLOT Y1 Y2 Y3 VERSUS X
PLOT Y1 Y2 Y3 VERSUS X1 Y4 Y5 VERSUS X2
PLOT Y PRED VERSUS X
```

```
PLOT EXP(-0.5*X**2)
PLOT EXP(-0.5*X**2) FOR X = -3 .1 3

3-D PLOT Y X1 X2
3-D PLOT Y X1 X2 LAB

3-D PLOT EXP(-0.5*(X**2+Y**2)) FOR X = -3 .1 3 FOR Y = -3 .1 3

PLOT Y X SUBSET LAB 4
PLOT Y X SUBSET X . 10 SUBSET Y , 100
PLOT Y X LAB SUBSET LAB 2 TO 8
PLOT Y1 Y2 Y3 VERSUS X SUBSET LAB 1 TO 4
PLOT Y1 Y2 Y3 VERSUS X SUBSET LAB 2 TO 5 SUBSET MONTH 3

3-D PLOT Y X1 X2 SUBSET LAB 3 5 6 10 15 SUBSET Y , 1000
3-D PLOT Y X1 X2 LAB SUBSET LAB 3 5 6 10 15 SUBSET Y , 1000
```

3.2 Diagrammatic Graphics Capabilities

Graphical diagrammatic capabilities relating to manuscript and slide preparation include sub-vocabularies for interactive diagramming such as chemical diagramming, electrical diagramming, IC and LSI device diagramming, printed circuit diagramming, flow charting, logo construction, and general text/equation writing (with choice of Hershey character fonts, character sizes, upper and lower case, mixture of English/Greek/math symbols, superscripting and subscripting, etc.). Capabilities exist for the saving, editing, combining, superimposing, and redisplaying of any diagram.

Examples of use of such diagrammatic graphics are as follows:

```
TEXT GRAPHICS
TEXT Y = INTE() SIN(ALPH() X) DX
MOVE 50 50
POINT 60 60
DRAW 40 40 60 60
ARROW 40 40 60 60
DRAW 20 20 40 20 40 40 20 40 20 20
BOX 40 40 60 60
```

```
CIRCLE 30 30 50 30
ELLIPSE 50 50 70 60 90 50
ARC 50 50 70 60 90 55
RESISTOR 30 30 50 30
AMPLIFIER 30 30 50 30
HEXAGON 60 60 70 60
NAND 50 50 60 50
NOR 50 50 60 50
```

3.3 Fitting Capabilities

Fitting capabilities include interactive on-line model specification; fitting of linear, polynomial, multi-linear, and non-linear models; fitting may be linear/non-linear, weighted/unweighted, constrained/unconstrained; non-linear fitting without need of derivatives; pre-fit analyses for determination of non-linear fit starting values; exact rational function fitting; spline fitting; least squares smoothing; robust smoothing; automatic storage of predicted values/residuals from all fitting and smoothing operations; superimposed raw and predicted value plots; residual plots; fitting and smoothing over full data sets or subset of data.

Examples of use of such fitting capabilities include:

```
CUBIC FIT Y X
QUARTIC FIT Y X
FIT Y = (A+B*EXP(-C**X))/(ALPHA+BETA*X)
FIT Y = A0+A1*EXP(A2*X1)+A3*EXP(A4*X2)+A5*EXP(A6*X3)
PRE-FIT Y = A+EXP(-B*X) FOR A = 10 10 100 FOR B = 1 .1 2
EXACT 2/1 RATIONAL FIT Y X
EXACT 3/3 RATIONAL FIT Y X Y2 X2
CUBIC SPLINE FIT Y X
QUARTIC SPLINE FIT Y X
MOVING AVERAGE SMOOTH Y
MEDIAN SMOOTH Y

CUBIC FIT Y X SUBSET LAB 2 TO 6
FIT Y = (A+B*EXP(-C**X))/(ALPHA+BETA*X) SUBSET X . 50
FIT Y = (A+B*EXP(-C**X))/(ALPHA+BETA*X) EXCEPT X = 50
FIT Y = (A+B*EXP(-C**X))/(ALPHA+BETA*X) EXCEPT Y . 100
FIT Y = (A+B*EXP(-C**X))/(ALPHA+BETA*X) SUBSET LAB 2 TO 5
FIT Y = (A+B*EXP(-C**X))/(ALPHA+BETA*X) SUBSET LAB 2 TO 5 SUBSET X . 50 EXCEPT X = 70
EXACT 3/3 RATIONAL FIT Y X Y2 X2 SUBSET X2 . 10
CUBIC SPLINE FIT Y X SUBSET LAB 4
MEDIAN SMOOTH Y FOR I = 1 1 50
MEDIAN SMOOTH Y SUBSET LAB 2 TO 7 EXCEPT Y . 100
```

3.4 Graphical Data Analysis Capabilities

Graphical data analysis capabilities include box plots; complex demodulation plots; control charts; correlation plots; distributional frequency plots; histograms; lag plots; percent point plots; auto and cross periodograms; probability plots (24 distributions); probability plot corr. coef. dist. analysis plots (3 families); auto- and cross- spectral plots; scatter plots; pie charts; Youden plots; graphical ANOVA/ANOCOV; runs plots; 3-d dist. frequency plots; 3-d histograms; 4-plot per page univariate analysis; all of above for full data sets or for subsets of data.

Examples of use of such data analysis graphics include the following:

```
BOX PLOT Y X                                   LAG 3 PLOT Y
COMPLEX DEMODULATION AMPLITUDE PLOT Y          LAG 5 PLOT Y1 Y2
COMPLEX DEMODULATION PHASE PLOT Y              BOX-COX NORMALITY PLOT Y
MEAN CONTROL CHART Y X                         PERCENT POINT PLOT Y
RANGE CONTROL CHART Y X                        PERIODOGRAM Y
AUTOCORRELATION PLOT Y                         PIE CHART Y
CROSS-CORRELATION PLOT Y1 Y2                   TUKEY LAMBDA PPCC PLOT Y
FREQUENCY PLOT Y                               NORMAL PROBABILITY PLOT Y
HISTOGRAM Y                                    WEIBULL PROBABILITY PLOT Y
I PLOT Y X                                     RUN SEQUENCE PLOT Y
BOX PLOT Y X SUBSET LAB 4 EXCEPT OPERATOR 3    SPECTRAL PLOT Y
COMPLEX DEMODULATION PLOT SUBSET LAB 3 TO 7    CROSS-SPECTRAL PLOT Y1 Y2
MEAN CONTROL CHART SUBSET X . 10 SUBSET X ,= 100
HISTOGRAM Y SUBSET OPERATOR 5 7 8 DAY 2 TO 5 EXCEPT Y . 100
SPECTRAL PLOT Y SUBSET STATE 2 10 13 16 SUBSET YEAR .= 1975
```

3.5 Non-graphical Data Analysis Capabilities

Non-graphical data analysis capabilities include elementary statistics (25 statistics); analysis of variance; median polish; tabulation of summary statistics; on-line definition and execution of functional transformations; cum. dist. functions (24 dist.); prob. density functions (24 dist.); percent point functions (24 dist.); random number generation (24 dist.); all operations may be over full data sets or subsets of data.

Examples of use of such non-graphical data analysis capabilities include:

```
ANALYSIS OF VARIANCE Y X1 X2
MEDIAN POLISH Y X1 X2
SUMMARY Y

LET M = MEAN Y                    ANALYSIS OF VARIANCE Y X1 X2 SUBSET LAB 3 SUBSET OPERATOR 7
LET M2 = MIDMEAN Y                MEDIAN POLISH SUBSET LAB 2 TO 10 EXCEPT LAB 3 EXCEPT Y . 1000
LET S = STANDARD DEVIATION Y      SUMMARY Y SUBSET LAB 2
LET UH = UPPER HINGE Y
LET C = CORRELATION Y1 Y2         LET M = MEAN Y SUBSET LAB 3
LET RC = RANK CORRELATION Y1 Y2   LET S = STANDARD DEVIATION Y SUBSET STATE 4 TO 8 SUBSET YEAR 1980
LET K = SQRT(2*3.14159)           LET C = CORRELATION Y1 Y2 SUBSET X .= 1970 SUBSET X ,= 1975
LET Y = (X**LAMBDA)/(LAMBDA-1)    LET Y = SQRT(X) FOR X , 1
LET C = NORPPF(.95)               LET Y = 1+LOG(X) FOR X .= 1
LET F = NORPDF(1.96)
LET Z = NORMAL RANDOM NUMBERS FOR I = 1 1 50
LET X = CHI-SQUARED RANDOM NUMBERS FOR N = 1 1 100
```

3.6 Mathematical Capabilities

Mathematical capabilities include interactive on-line definition & concatenation/ composition of functions; functional analyses; exact analytic symbolic differentiation; root extraction; definite integration; convolution.

Examples of use of such mathematical capabilities are as follows:

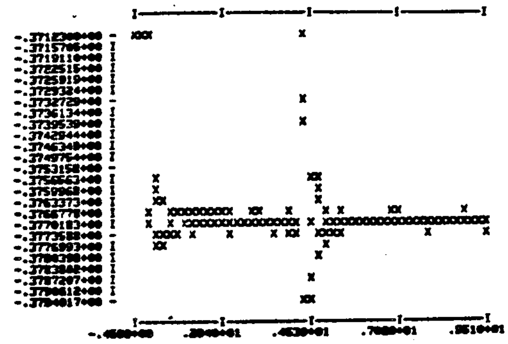
```
LET FUNCTION F = EXP(-ALPHA*X/Y)                 LET N = NUMBER Y
LET FUNCTION G = SIN(EXP(X))                     LET A = SUM Y
LET FUNCTION H = LOG(1+F)*G                      LET B = PRODUCT Y
LET FUNCTION F2 = DERIVATIVE F1 WRT X            LET C = INTEGRAL Y X
LET A = INTEGRAL F WRT X FOR X = 0 TO 4
LET B = ROOTS F WRT X FOR X = 0 TO 100           LET Y2 = CUMULATIVE SUM Y
                                                 LET Y2 = CUMULATIVE PRODUCT Y
LET N = NUMBER Y SUBSET LAB 2 YEAR 1980          LET Y2 = CUMULATIVE INTEGRAL Y
LET A = SUM Y FOR I = 1 1 10                     LET Y2 = SEQUENTIAL DIFFERENCE Y
LET A = SUM Y SUBSET X . 20                      LET Y2 = SORT Y
LET B = PRODUCT Y SUBSET Y . 0 SUBSET Y , 10     LET Y2 = RANK Y
LET C = INTEGRAL Y X SUBSET X . 10               LET Y2 = CODE Y
                                                 LET Y2 = CONVOLUTION Y X
LET Y2 = CUMULATIVE SUM Y SUBSET Y , 20
LET Y2 = CONVOLUTION Y X SUBSET Y . 2
```

4. EXAMPLES OF DATAPLOT CODE AND OUTPUT

The purpose of this section is to present a sampling of DATAPLOT code and the resulting output. The examples will focus on commonly-occurring graphical activities. Usually there will be several different ways within DATAPLOT to generate the same output; more than one of the different ways will at times be presented. Space considerations limit the detail associated with the discussion of each example.

4.1 Plot Data at a Discrete Terminal

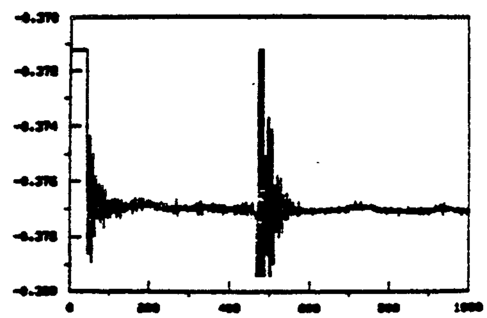
Data exists on a file as a series of (x,y) pairs. Two numbers—an x and a y—are on each line image. There is an indeterminate (and unimportant) number of line images in the file. Read in the (x,y) data points from the file in a format-free fashion. Plot the data on a narrow-width (72 character), discrete terminal (e.g., TI 700, Hazeltine, Omron). (Note that DATAPLOT has discrete (and batch) analogues to almost all of the continuous plots as shown in the remaining examples.) See figure 1 for the discrete terminal plot.



```
READ file name X Y
DISCRETE
PLOT Y X
```

4.2 Plot Data

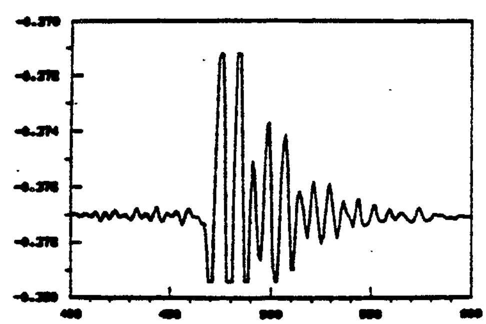
Data exists on a file as a series of (x,y) pairs. Two numbers—an x and a y—are on each line image. There is an indeterminate (and unimportant) number of line images in the file. Read in the (x,y) data points from the file in a format-free fashion. Plot them. See figure 2 for the output.



```
READ file name X Y
PLOT Y X
```

4.3 Plot a Subset of the Data

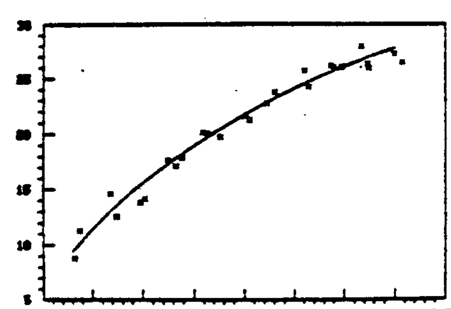
Read in (x,y) points from a file. Plot them but restrict the plot to only those points between x = 400 to x = 600. This demonstrates the subsetting feature and will here have the effect of "blowing up" the plot. In general, subsetting may be done on any variable (not just those involved in the plot), may be done for combinations of variables, and may be appended to all graphics ana analysis commands. See figure 3.



```
READ file name X Y
PLOT Y X SUBSET X 400 TO 600
```

4.4 Plot a Mixture of Data and Functions

Read in (x,y) pairs. Plot these points as discrete X's. Superimpose a continuous functional trace with a solid line. See figure 7.

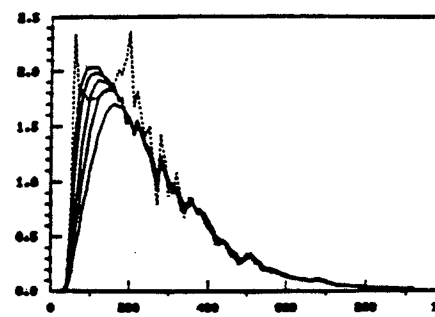


```
READ file name X Y
.
LET A0 = -181
LET A1 = 142
LET B1 = 2.8
LET FUNCTION F = (A0+A1*X)/(1+B1*X)
.
CHARACTERS X BLANK
LINES BLANK SOLID
PLOT Y X AND
PLOT F FOR X = 1.65 .1  3.3
```

4.5 Plot Multiple Data Traces

Data exists on a file for several variables (data vectors) with corresponding points on the same line image, and with some indeterminate (and unimportant) number of line images. Read in the data points for the several different variables. Plot them (6 traces will result for this example). Have all traces solid except for the first trace which is to have a dotted line. See figure 5.

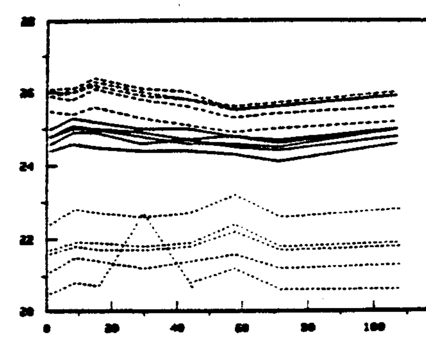
```
READ file name X Y1 Y2 Y3 Y4 Y5 Y6
LINES DOTTED SOLID SOLID SOLID SOLID SOLID
PLOT Y1 Y2 Y3 Y4 Y5 Y6 VERSUS X
```



4.6 Generate a Multi-Trace Plot to Emphasize Between-Group Effect

A response variable Y is dependent on 3 independent variables—time X, operator OP, and replication (within operator) REP. Data exists on a file for these 4 variables consisting of the 4 corresponding values on a given line image. Read in the data. The operator variable will have 3 distinct values (1, 2, and 3). The replication variable will have 5 distinct values (1 through 5). Generate a multi-trace plot of Y versus X consisting of 3 x 5 = 15 distinct traces. Emphasize between-operator effects by having the 5 traces for operator 1 solid, the 5 traces for operator 2 dashed, and the 5 traces for operator 3 dotted. See figure 6.

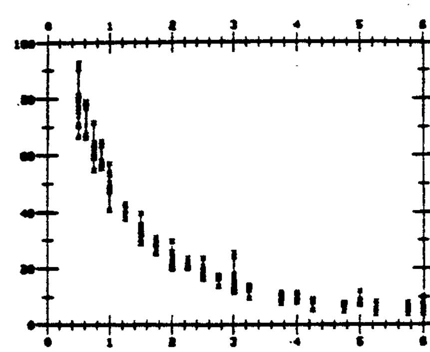
```
READ file name Y X OP REP
LET TAG = 3*(OP-1)+REP
LINES SO SO SO SO SO DA DA DA DA DA DO DO DO DO DO
PLOT Y X TAG
```



4.7 Generate a Plot to Emphasize Spread Due to Replication

A response variable Y is dependent on a single independent variable X. Data exists on a file consisting of corresponding values of Y and X on a given line image. Read in the data. Note that replication may exist—that is, a given value of X may have several values of Y. Plot Y versus X but emphasize the spread between replicates for each fixed X value by having these replicates for a given X connected by individual solid traces. Have the data values themselves displayed by using x's as a plot character. Have the height of the x's equal to 1.75% of the total vertical screen height. See figure 7.

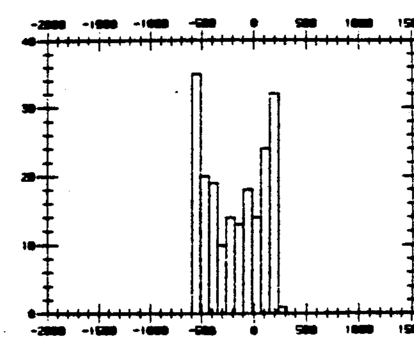
```
READ file name Y X
CHARACTERS X ALL
CHARACTER SIZE 1.75 ALL
PLOT Y X X
```



4.8 Generate a Histogram

Data exists on a file as a series of values (1 value per line image). Read in the data. Generate a histogram. Have the class limits automatically determined. The histogram is a graphical data analysis technique for displaying distributional information in a data set. See figure 8.

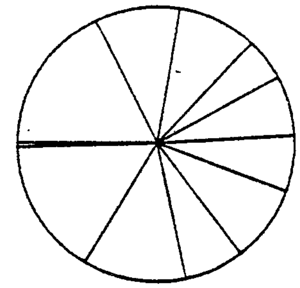
```
READ file name X
HISTOGRAM X
```



4.9 Generate a Pie Chart

Data exists on a file as a series of values (1 value per line image). Read in the data. Generate a pie chart. Have the class limits automatically determined. The pie chart is a graphical data analysis technique for displaying distributional information in a data set. See figure 9.

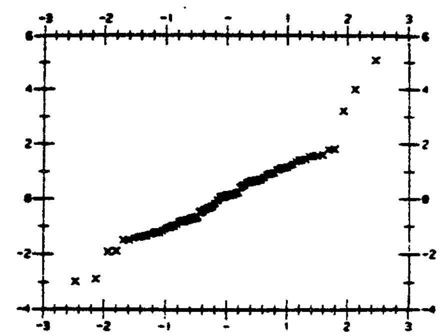
```
READ file name X
PIE CHART X
```



4.10 Generate a Normal Probability Plot

Data exists on a file as a series of values (1 value per line image). Read in the data. Generate a probability plot for the normal distribution. The probability plot for a given distribution is a graphical data analysis technique for determining if the given distribution provides a good distributional fit to the data. The vertical axis is the ordered raw data. The horizontal axis is the order statistic medians from the given distribution. The linearity of the resulting plot is of interest—the more linear the plot, the better the distributional fit. See figure 10.

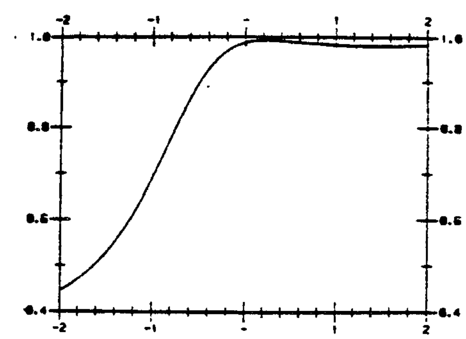
```
READ file name X
NORMAL PROBABILITY PLOT X
```



4.11 Generate a Probability Plot Correlation Coefficient Plot

Data exists on a file as a series of values (1 value per line image). Read in the data. Generate a PPCC (probability plot correlation coefficient) plot for the Tukey lambda distribution family. The PPCC plot for a given distributional family is a graphical data analysis technique for determining which member of the family provides the best distributional fit to the data. The maximum probability plot correlation coefficient criterion is the measure of goodness of fit. The vertical axis is the probability plot correlation coefficient—a measure of linearity in a probability plot. The horizontal axis is the parameter (in this case, lambda) which defines the various members of the distributional family. The lambda value where the PPCC attains a maximum is of interest. See figure 11.

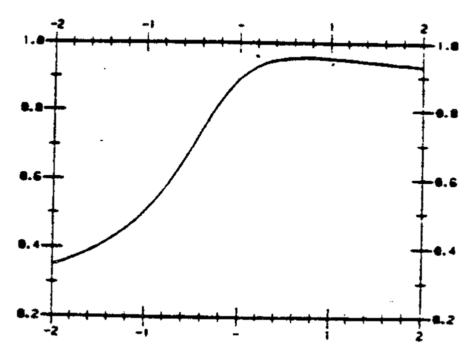
```
READ file name X
TUKEY LAMBDA PPCC PLOT X
```



4.12 Generate a Normality Plot

Data exists on a file as a series of values (1 value per line image). Read in the data. Generate a normality plot for the Box-Cox transformation family. The normality plot for a given transformation family is a graphical data analysis technique for determining which member of the family (that is, which transformation of the raw data) will yield a transformed variable which is most closely normally distributed. The maximum normal probability plot correlation coefficient criterion is the measure of goodness of fit. The vertical axis is the normal probability plot correlation coefficient—a measure of linearity in a normal probability plot. The horizontal axis is the parameter (in this case, lambda) which defines the various members of the transformation family. The lambda value where the normal PPCC attains a maximum is of interest. See figure 12.

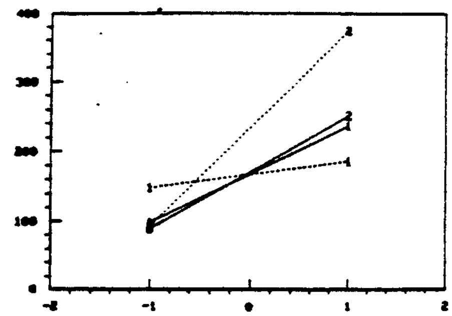
```
READ file name X
BOX-COX NORMALITY PLOT X
```



4.13 Carry Out a 3-Factor Graphical ANOVA (GANOVA) for a 2-Cubed Full Factorial Design

Data exists on a file for a response variable Y and independent variables X1, X2, and X3; the 4 corresponding values of the variables are on each line image. Variables X1, X2, and X3 each have 2 levels (1 and 2). There are 2 x 2 x 2 = 8 line images. Read in the data. Carry out a graphical ANOVA (GANOVA) by generating a multi-trace plot of Y versus X1 consisting of 2 x 2 = 4 traces. Emphasize the 2 levels of the X2 variable by having different line types—solid and dotted. Emphasize the 2 levels of the X3 variable by having different character types—1 and 2. See fig. 13.

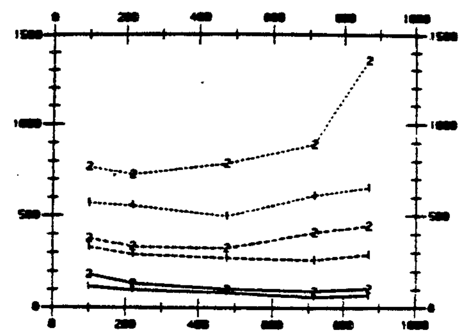
```
READ file name Y X1 X2 X3
LET ID = 2*(X2-1)+X3
LINES SOLID SOLID DOTTED DOTTED
CHARACTERS 1 2 1 2
PLOT Y X1 ID
```



4.14 Carry Out a 3-Factor Graphical ANOVA (GANOVA) for a General Full Factorial Design

Data exists on a file for a response variable Y and independent variables X1 (5 levels), X2 (3 levels), and X3 (2 levels). The 4 corresponding values of the variables are on each line image. There are 5 x 3 x 2 = 30 line images. Read in the data. Carry out a graphical ANOVA (GANOVA) by generating a multi-trace plot of Y versus X1 consisting of 3 x 2 = 6 traces. Emphasize the 3 levels of the X2 variable by having different line types. Emphasize the 2 levels of the X3 variable by having different character types. See fig. 14.

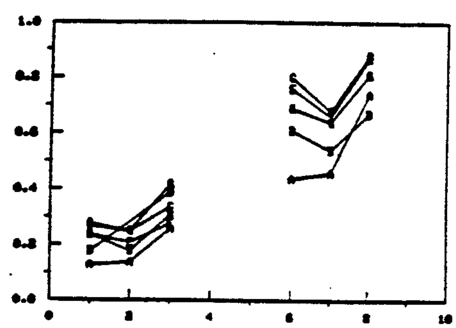
```
READ file name Y X1 X2 X3
LET ID = 2*(X2-1)+X3
LINES SOLID SOLID DASHED DASHED DOTTED DOTTED
CHARACTERS 1 2 1 2 1 2
PLOT Y X1 ID
```



4.15 Carry Out a Nested Graphical ANOVA (Nested GANOVA) of a 3-Factor Experiment

Data exists on a file for a response variable Y and independent variables X1 (2 levels), X2 (3 levels), and X3 (6 levels when X1 = 1 and 5 levels when X1 = 2). The 4 corresponding values of the variables are on each line image. There are (3 x 5) + (3 x 6) = 33 line images. Read in the data. Carry out a nested graphical ANOVA (nested GANOVA)—plot Y versus X1, but visually nest all levels of X2 and X3 within each level of X1; have all line types solid; have the 6 levels of X3 denoted by different character types. See fig. 15.

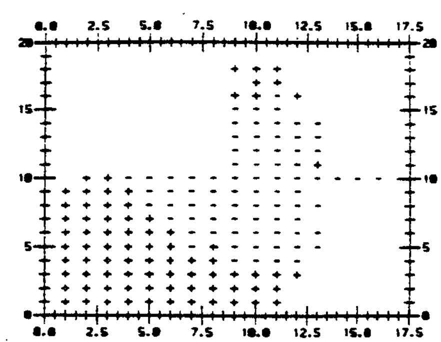
```
READ file name Y X1 X2 X3
LET X1PRIME = 5*(X1-1)+X2
LET TAG = 2*(X1-1)+X3
CHARACTERS A B G E C D A B G E C
PLOT Y X1PRIME TAG
```



4.16 Generate a Discrete Contour Plot to Assess 2-Dimensional Homogeneity

A physical specimen has a measured response at equi-spaced points on its surface. The response Z and the coordinate values X and Y are on a file. Read in the data. Carry out a quick check to see if the response is homogeneous over the surface by computing the median response, and then generating a discrete contour plot in which all values smaller than the median have plot character −, and all values greater have +. See fig. 16.

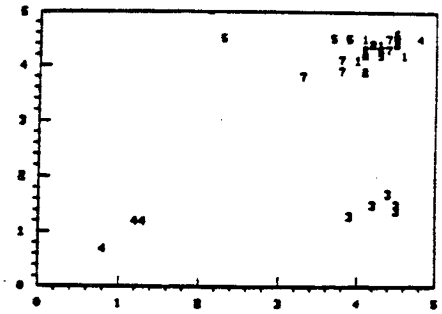
```
READ file name X Y Z
LET MED = MEDIAN Z
LINES BLANK ALL
CHARACTERS - +
PLOT Y X SUBSET Z , MED AND
PLOT Y X SUBSET Z .= MED
```



4.17 Generate a Youden Plot for Interlaboratory Analysis

Two specimens of a material are circulated to 7 laboratories. A given laboratory makes 5 measurements Y1 for specimen 1, and 5 measurements Y2 for specimen 2. The data exists on a file. A given line image has 4 values—response Y1 for specimen 1, response Y2 for specimen 2, laboratory identification LAB, and replication identification REP. There are 7 x 5 = 35 line images. Read in the data. Carry out an interlaboratory analysis via a Youden plot. The vertical axis is Y1; the horizontal axis is Y2. The plot character is laboratory. See figure 17.

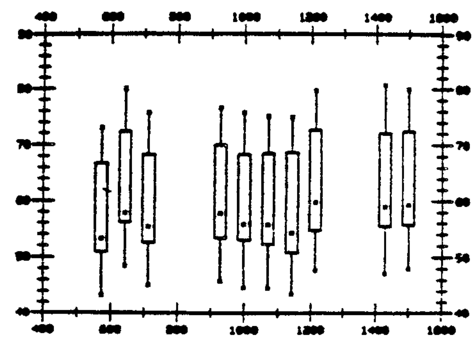
```
READ file name Y1 Y2 LAB REP
LINES BLANK ALL
CHARACTERS 1 2 3 4 5 6 7
PLOT Y1 Y2 LAB
```



4.18 Generate a Box Plot

Data exists on a file as a series of line images with 2 values per line image. The first value is a response variable value; the second value is the treatment identifier for a given response. Read in the data. A given treatment will typically have more than 1 response value associated with it. Generate a box plot. The box plot is a graphical data analysis technique for assessing whether significant differences exist between various treaments; it is akin to ANOVA. See figure 18.

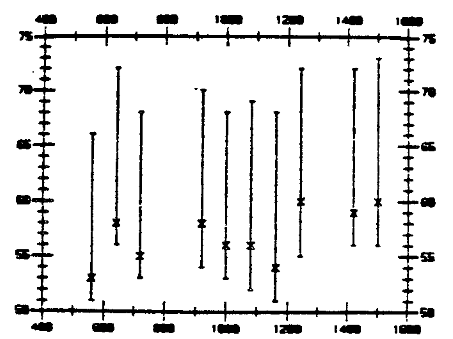
```
READ file name Y X
CHARACTERS BOX PLOT
LINES BOX PLOT
BOX PLOT Y X
```



4.19 Generate an I Plot

Data exists on a file as a series of line images with 2 values per line image. The first value is a response variable value; the second value is the treatment identifier for a given response. Read in the data. A given treatment will typically have more than 1 response value associated with it. Generate an I plot. The vertical bar on the I plot will extend from the minimum response to the maximum response within a treatment. The I plot is a graphical data analysis technique for assessing whether significant differences exist between various treatments. It is also a graphical data summary technique for displaying a typical value and analyst-selected uncertainty limits. See fig. 19.

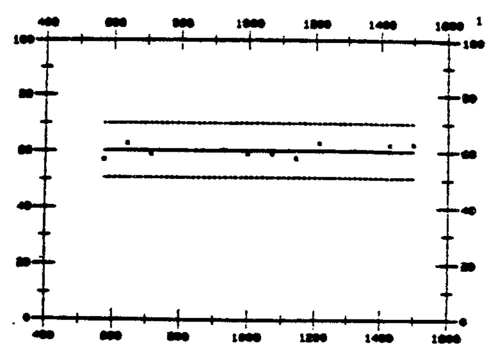
```
READ file name Y X
CHARACTERS I PLOT
LINES I PLOT
I PLOT Y X
```



4.20 Generate a Control Chart

Data exists on a file as a series of line images with 2 values per line image. The first value is a response variable value; the second value is the set (e.g., time) identifier for a given response. Read in the data. Generate a mean control chart. To use control charts, the various sets usually have more than 1 observation per set, and ideally have the same number of observations for all sets. The control chart (in general) is a graphical data analysis technique for assessing whether significant shifts in location or dispersion have occurred in a measurement process. See figure 20.

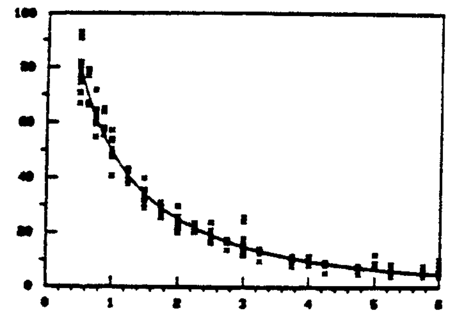
```
READ file name Y X
CHARACTERS CONTROL CHART
LINES CONTROL CHART
MEAN CONTROL CHART Y X
```



4.21 Generate a Superimposed Plot of Raw Data and Fitted Values from a Non-Linear Fit

Data exists on a file as a series of line images with 2 values per line image (response variable Y and independent variable X). Theory suggests an exponential/linear non-linear relationship. Read in the data. Carry out a least squares fit. Generate a plot of the raw data Y (as discrete x's) and the superimposed fitted curve (as a solid line) versus X. Such a superimposed plot is the first step in assessing goodness of fit. See fig. 21.

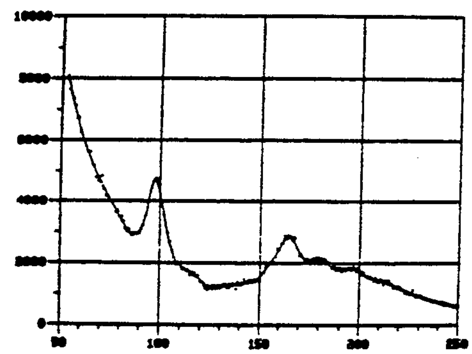
```
READ file name Y X
LET ALPHA = .1
LET A = .001
LET B = .01
FIT Y = EXP(-ALPHA*X)/(A+B*X)
CHARACTERS X BLANK
LINES BLANK SOLID
PLOT Y PRED VERSUS X
```



4.22 Generate a Superimposed Plot of Raw Data and Fitted Values from a Spline Fit

Data exists on a file as a series of line images with 2 values per line image (response variable and independent variable). The knots for such a fit are on a second file (1 knot point per line image). Read in the data; read in the knots. Carry out a least squares cubic spline fit. Generate a plot of the raw data Y (as discrete x's) and the superimposed fitted curve (as a solid line) versus X. See figure 22.

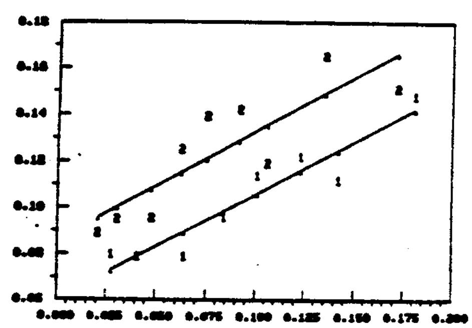
```
READ file name Y X
READ file 2 name KNOTS
CUBIC SPLINE FIT Y X KNOTS
CHARACTERS X BLANK
LINES BLANK SOLID
PLOT Y PRED VERSUS X
```



4.23 Generate a Superimposed Plot of Raw Data and Fitted Values from 2 Fits

Data exists on a file as a series of line images with 3 values per line image (response variable Y, and independent variables X and LAB). Read in the data. Fit a line to the data from lab 1; fit a line to the data from lab 2. Generate a plot showing raw data and fitted curves from both labs. Lab 1 data has character 1; lab 2 has 2. The 2 fitted lines are solid and dotted. See figure 23.

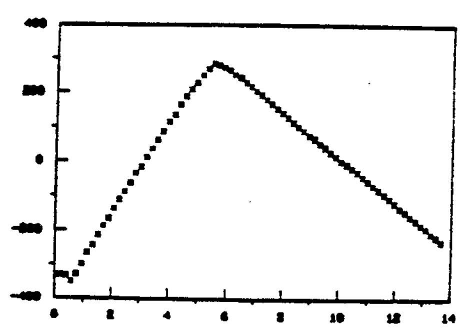
```
READ file name Y X LAB
FIT Y = A1+B1*X SUBSET LAB 1
LET PRED1 = PRED
FIT Y = A2+B2*X SUBSET LAB 2
CHARACTERS 1 BLANK 2 BLANK
LINES BLANK SOLID BLANK DOTTED
PLOT Y PRED1 VS X SUBSET LAB 1 AND
PLOT Y PRED  VS X SUBSET LAB 2
```



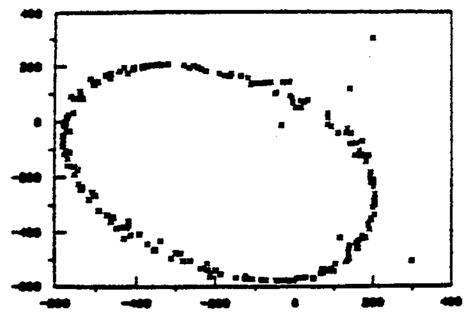
4.24 Generate a Residual Plot

Data exists on a file as a series of line images with 2 values per line image (response variable and independent variable). Theory suggests a linear relationship. Read in the data and carry out a least squares fit. Generate a plot of the residuals (as discrete X's) from the fit (vertically) versus values of the independent variable (horizontally). Such a plot is drawn from a battery of graphical procedures known as residual analysis—they are essential for assessing goodness of fit of a general model. See fig. 24.

```
READ file name Y X
FIT Y = A+B*X
CHARACTERS X
LINES BLANK
PLOT RES X
```



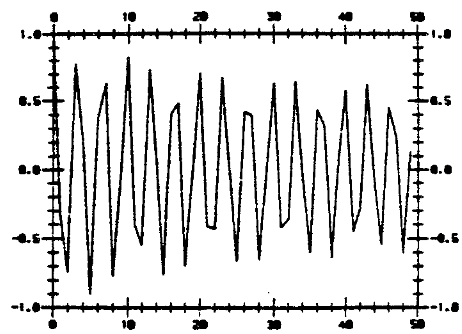
4.25 Generate a Lag Plot



Data exists on a file as a series of values (1 value of the variable X per line image). Read in the data. Generate a lag plot with lag = 1. Use the symbol x as the plot character. The lag plot (with lag = 1) is a plot of X(i) versus X(i-1). It is a graphical data analysis technique for assessing autocorrelation structure in time series. See figure 25.

```
READ file name X
CHARACTERS X
LINES BLANK
LAG 1 PLOT X
```

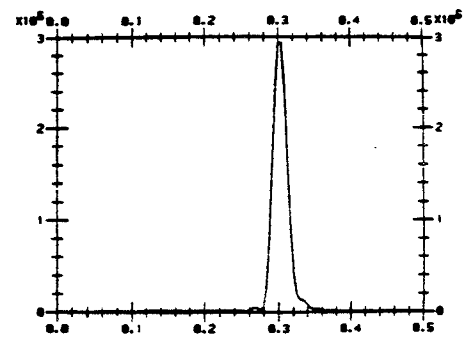
4.26 Generate an Autocorrelation Plot



Data exists on a file as a series of values (1 value of the variable X per line image). Read in the data. Generate an autocorrelation plot. The autocorrelation plot is a plot of the correlation between X(i) and X(i+h) versus lag h. It is a graphical data analysis technique for assessing autocorrelation structure in time series. See figure 26.

```
READ file name X
AUTOCORRELATION PLOT X
```

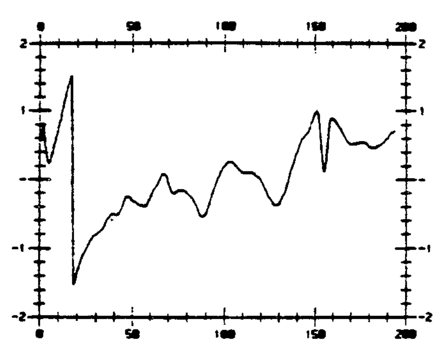
4.27 Generate a Spectral Plot



Data exists on a file as a series of values (1 value of the variable X per line image). Read in the data. Generate a spectral plot. The spectral plot is a frequency domain plot of power versus frequency (0 to .5). The spectral power function is the smoothed Fourier transform of the autocorrelation function. The spectral plot is a graphical data analysis technique for assessing autocorrelation and cyclic structure in time series. See figure 27.

```
READ file name X
SPECTRAL PLOT X
```

4.28 Generate a Complex Demodulation Plot



Data exists on a file as a series of values (1 value of the variable X per line image). Read in the data. Generate a complex demodulation phase plot at the demodulation frequency of 0.3. The complex demodulation phase plot is a plot of locally-estimated phase versus time. The complex demodulation amplitude and phase plots are graphical data analysis techniques for assessing (among other things) whether amplitude and phase are constant ove the entire time domain in single-cycle time series. See figure 28.

```
READ file name X
DEMODULATION FREQUENCY .3
COMPLEX DEMODULATION PLOT X
```

4.29 Plot a Function

Plot the Normal N(0,1) density function. See figure 29.

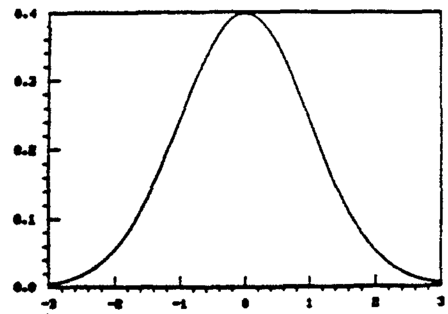
```
LET PI = 3.1415926
PLOT (1/SQRT(2.0*PI))*EXP(-0.5*X**2) FOR X = -3 .1 3
```

or

```
LET PI = 3.1415926
LET FUNCTION F1 = 1/SQRT(2*PI)
LET FUNCTION F2 = EXP(-0.5*X**2)
LET FUNCTION G = F1*F2
PLOT G FOR X = -3 .1 3
```

or

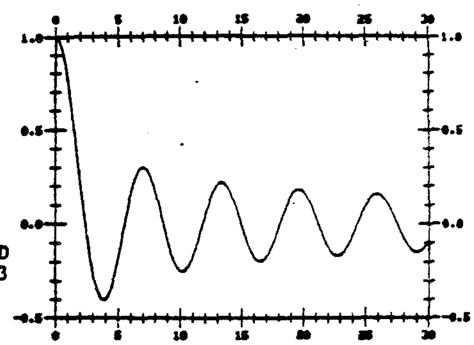
```
PLOT NORPDF(X) FOR X = -3 .1 3
```



4.30 Plot a Bessel Function

Plot J0(x)—a member of the family of Bessel functions of the first kind. See figure 30.

```
PLOT BESS0(X) FOR X = 0 .1 30
```



4.31 Plot Multiple Functions

Plot 2 Normal N(.,1) density functions—one centered at 0 and the other cetnered at 3. Have the first trace solid and the second trace dotted. See figure 31.

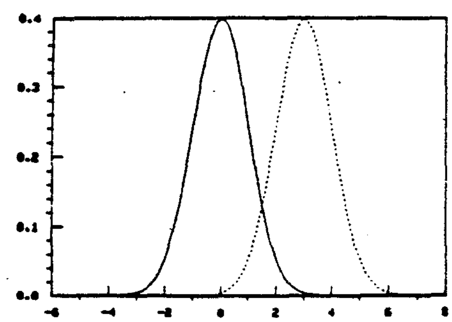
```
LINES SOLID DOTTED
LET PI = 3.1415926
PLOT (1/SQRT(2.0*PI))*EXP(-0.5*X**2) FOR X = -3 .1 3 AND
PLOT (1/SQRT(2.0*PI))*EXP(-0.5*(X-3)**2) FOR X = -3 .1 3
```

or

```
LINES SOLID DOTTED
LET PI = 3.1415926
LET FUNCTION F0 = 1/SQRT(2*PI)
LET FUNCTION F1 = EXP(-0.5*X**2)
LET FUNCTION F2 = EXP(-0.5*(X-3)**2)
LET FUNCTION G1 = F0*F1
LET FUNCTION G2 = F0*F2
PLOT G1 FOR X = -3 .1 3 AND
PLOT G2 FOR X = 0 .1 6
```

or

```
LINES SOLID DOTTED
PLOT NORPDF(X) FOR X = -3 .1 3 AND
PLOT NORPDF(X-3) FOR X = 0 .1 6
```



4.32 Plot Multiple Polar Coordinate Functions with Cross-Hatching

Generate the polar coordinate function (a spiral)

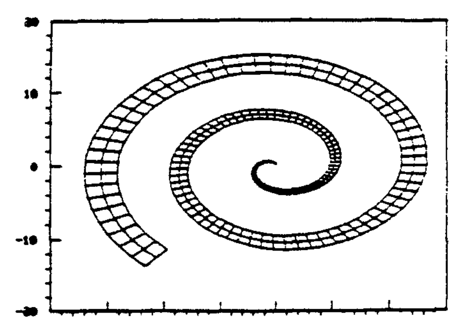
radius = theta

for theta = 0 (30) 1000 degrees. Do similarly for the functions

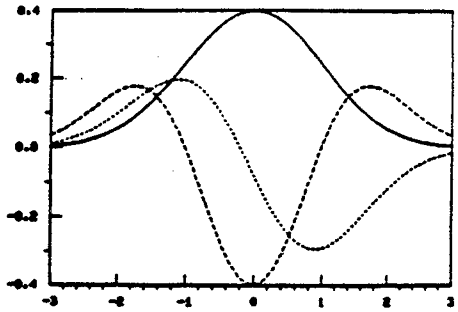
radius = 1.1 x theta
radius = 1.2 x theta

Draw the 3 spirals, and also cross-hatch the spirals at fixed theta values (theta = 0, 30, 60, etc.). See fig. 32.

```
LET THETA = SEQUENCE 0 30 1000 FOR I =  1 1  34
LET THETA = SEQUENCE 0 30 1000 FOR I = 35 1  68
LET THETA = SEQUENCE 0 30 1000 FOR I = 69 1 102
.
LET R = 1   FOR I =  1 1  34
LET R = 1.1 FOR I = 35 1  68
LET R = 1.2 FOR I = 69 1 102
.
DEGREES
LET Y = R*SIN(THETA)
LET X = R*COS(THETA)
.
PRE-SORT OFF
PLOT Y X R AND
PLOT Y X THETA
```



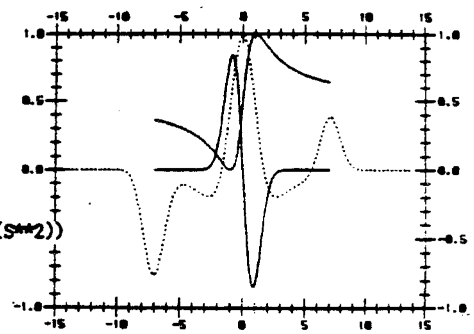
4.33 Plot a Function and Its Derivatives



Plot the Normal N(0,1) density function, along with its first and second derivatives. Have the line types solid, dotted and dashed, respectively. See figure 33.

```
LET FUNCTION F = (1/SQRT(2*PI))*EXP(-0.5*X**2)
LET FUNCTION D1 = DERIVATIVE F WRT X
LET FUNCTION D2 = DERIVATIVE D1 WRT X
.
LINES SOLID DOT DASH
PLOT F FOR X = -3 .1 3 AND
PLOT D1 FOR X = -3 .1 3 AND
PLOT D2 FOR X = -3 .1 3
```

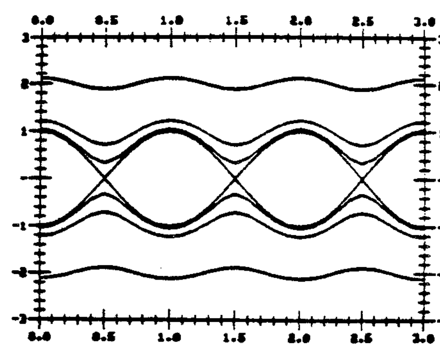
4.34 Plot 2 Functions and Their Convolution



Define equi-spaced values from a function of interest. Define equi-spaced values from a second function of interest. Compute the convolution. Plot the 2 functions and the resulting convolution. Have the line types solid, solid, and dotted, respectively. See figure 34.

```
LET TMIN = -7
LET TDEL = .1
LET TMAX = 7
LET T = SEQUENCE TMIN TDEL TMAX
.
LET S = .85
LET Y1 = ((T+1)**2)/((T**2)+1)/2
LET Y2 = 2**((-(T+1.3/2)**2)/(S**2))-2**((-(T-1.3/2)**2)/(S**2))
LET Y3 = CONVOLUTION Y1 Y2 TDEL
.
LET T3MIN = TMIN+TMIN
LET T3MAX = TMAX+TMAX
LET T3 = SEQUENCE T3MIN TDEL T3MAX
LINES SOLID SOLID DOTTED
PLOT Y1 Y2 VS T AND
PLOT Y3 VS T3
```

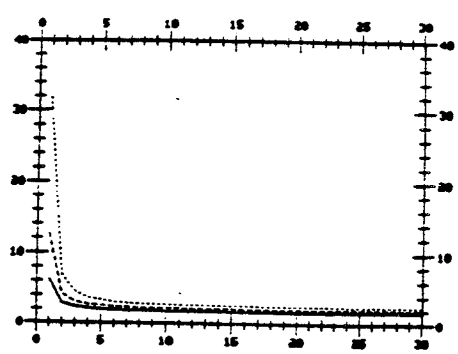
4.35 Generate a Differential Equation Phase Diagram



Generate a phase diagram for the differential equation associated with the simple pendulum. The phase diagram (= phase portrait) is a graphical technique for displaying the nature of the solution for a differential equation. The phase diagram of a differential equation y'' = f(y',y) or y' = f(y) is a plot of y' (vertically) versus y horizontally. The individual phase paths (= trajectories = integral curves) represent solution contours for the differential equation. See fig. 34.

```
LET FUNCTION F = 0.5*COS(2*3.14159*T)
PLOT  SQRT(F+0.5) FOR T = 0 .1 3 AND
PLOT -SQRT(F+0.5) FOR T = 0 .1 3 AND
PLOT  SQRT(F+0.6) FOR T = 0 .1 3 AND
PLOT -SQRT(F+0.6) FOR T = 0 .1 3 AND
PLOT  SQRT(F+1.0) FOR T = 0 .1 3 AND
PLOT -SQRT(F+1.0) FOR T = 0 .1 3 AND
PLOT  SQRT(F+4.0) FOR T = 0 .1 3 AND
PLOT -SQRT(F+4.0) FOR T = 0 .1 3
```

4.36 Generate a Multi-Trace Plot of Percent Point Functions

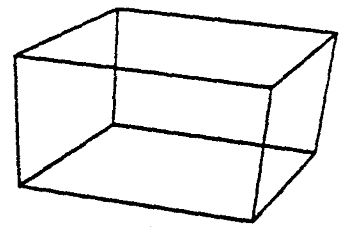


Compute and generate the 95 percent points for the members of the t distribution family for degrees of freedom parameter nu = 1 (1) 30. Do similarly for the 97.5 and 99 percent points. Plot the 3 functions versus nu. Have the 3 traces solid, dashed, and dotted. See figure 5.36.

```
LET NU = SEQUENCE 1 1 30
LET Y1 = TPPF(.95,NU)
LET Y2 = TPPF(.975,NU)
LET Y3 = TPPF(.99,NU)
LINES SOLID DASH DOT
PLOT Y1 Y2 Y3 VS NU
```

4.37 Generate a 3-Dimensional Data Plot

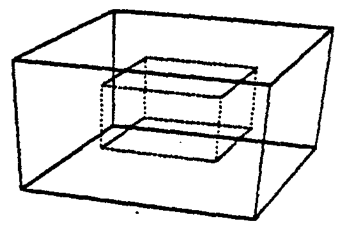
(x,y,z) coordinates exist on a data file. These data points may define any general 3-dimensional entity—a line in space, a plane, a surface, a figure, etc. Read in the data. Generate a 3-dimensional plot. The plot will consist of a box. Position the eye at (x = 2, y = 5, z = 3). Omit the reference axes. See figure 37.



```
READ file name X Y Z
FRAME OFF
EYE COORDINATES 2 5 3
3D-PLOT Z X Y
```

4.38 Generate 3-Dimensional Multiple Data Traces

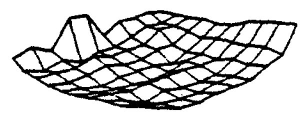
Data exists on a file as (x,y,z) triples along with a fourth variable—a tag variable—which identifies the trace that this (x,y,z) triple belongs to. A trace may define any general 3-dimensional entity. Read in the (x,y,z,tag) data points. Plot them all as individual traces, have the first trace solid and the second trace dotted. The plot will consist of 2 boxes—one within the other. The outer box will be solid, the inner dotted. Omit the reference axes. Position the eye at (x = 2, y = 5, z = 3). See figure 38.



```
READ file name X Y Z ID
LINES SOLID DOTTED
FRAME OFF
EYE COORDINATES 2 5 3
3-D PLOT Z X Y ID
```

4.39 Generate a 3-Dimensional Data Surface

A response Z has been measured at equi-spaced values of 2 independent variables X and Y. Z, X, and Y values have been collected on a file. Read in the data. Examine the nature of the response surface by generating a cross-hatch plot of the surface. Omit the reference axes. Positon the eye at (x = 20, y = 30 z = 53). See figure 39.

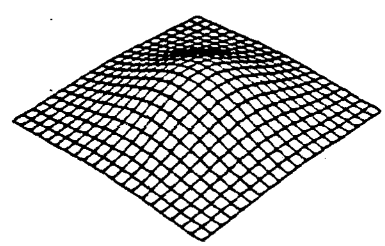


```
READ file name Z X Y
FRAME OFF
EYE COORDINATES 20 30 53
3D-PLOT Z X Y X AND
3D-PLOT Z X Y Y
```

4.40 Generate a 3-dimensional Function Plot

Generate the bivariate normal N(0,0,1,1,0) density function. Position the eye at (x = 10, y = 10, z = 11). Omit the reference axes. See figure 40.

```
LET FUNCTION E = -0.5*((X**2)+(Y**2))
LET FUNCTION F = (1/(2*PI))*EXP(E)
FRAME OFF
EYE COORDINATES 10 10 11
3D-PLOT F FOR X = -3 .1 3 FOR Y = -3 .1 3
```



4.41 Generate a Boxed Title

Go to the middle (= (50,50)) of the screen and generate the centered string: graphics in upper case triplex italic. Surround it with a box with lower left corner at (30,45) and upper right corner at (70,59). See figure 41.

```
FONT TRIPLEX ITALIC
CASE UPPER
JUSTIFICATION CENTER
HEIGHT 5
WIDTH 4
.
ERASE
MOVE 50 50
TEXT GRAPHICS
BOX 30 45 70 59
```

4.42 Generate a Mathematical Equation

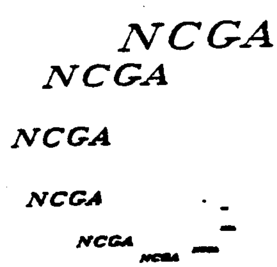
Go to a point 20 percent of the way across the screen and 50 percent of the way up the screen and write the following equation in lower case triplex:
y = integral sin(alpha x)dx + exp(bx)
Note the DATAPLOT convention of () as an appendage to certain in-line keywords (e.g., INTE). See figure 42.

```
FONT TRIPLEX
CASE LOWER
JUSTIFICATION LEFT
HEIGHT 5
WIDTH 4
.
ERASE
MOVE 20 50
TEXT Y = INTE()SIN(ALPH()X)DX + ESUP()(BX)
```

4.43 Generate a Logo

Generate an NCGA logo on a spiral—have the string NCGA appear on the spiral in increasingly larger letters. Have triplex italic font. Note the use of DATAPLOT subprogram and looping capabilitites. See fig. 43.

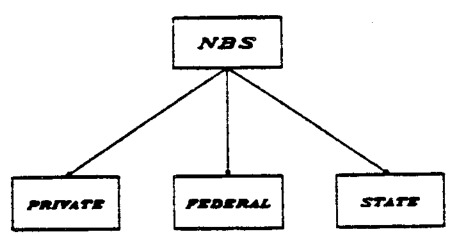
```
FONT TRIPLEX ITALIC
JUSTIFICATION CENTER
DEGREES
FEEDBACK OFF
PRE-ERASE OFF
ERASE
CALL LOGOSUB. FOR THETA = 390 -30 90
_______________
. THE FOLLOWING IS THE DATAPLOT SUBPROGRAM LOGOSUB.
LET H = (390-THETA)/30
LET H = H*H/10
LET W = 0.67*H
HEIGHT H
WIDTH W
LET R = 50*(1-THETA/360)
LET X = R*COS(THETA) + 50
LET Y = R*SIN(THETA) + 50
MOVE X Y
TEXT NCGA
```



4.44 Generate a Word Chart

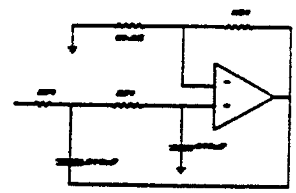
Generate a word chart. Note in passing the use of the DATAPLOT terminator character (here = ;) which allows packing of DATAPLOT commands on a line. See fig. 44.

```
TERMINATOR CHARACTOR ;
FEEDBACK OFF; FONT TRIPLEX ITALIC; JUSTIFICATION CENTER;
ERASE; HEIGHT 4; WIDTH 3; MOVE 50 80; TEXT NBS
HEIGHT 3; WIDTH 2; MOVE 20 40; TEXT PRIVATE;
MOVE 50 40; TEXT FEDERAL; MOVE 80 40; TEXT STATE
BOX 40 75 60 88; BOX 10 35 30 48; BOX 40 35 60 48; BOX 70 35 90 48
HEIGHT .5; WIDTH 1; ARROW 50 75 20 48; ARROW 50 75 50 48
ARROW 50 75 50 48; COPY
```



4.45 Generate an Electrical Diagram

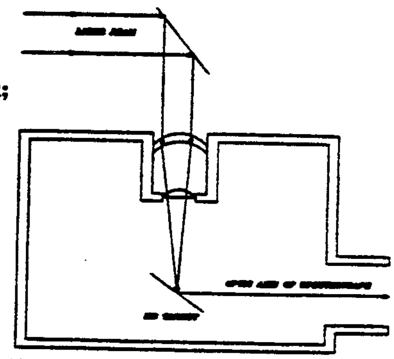
Generate a simple electrical diagram. Note that parameters (rather than numbers) could have been used throughout (e.g., DRAW A B rather than DRAW 25 65). Such parameters would be defined via the LET command or interactively via the CROSSHAIR command. Note also the packing of commands on a line via the use of the terminator character (here the terminator charactor = ;). See fig. 45.



```
TERMINATOR CHARACTOR ;
FEEDBACK OFF; ERASE; HEIGHT .6; WIDTH .6
MOVE 20 65; GROUND 20 63; DRAW 20 65 20 70 27 70;
RESISTOR 27 70 33 70; DRAW 33 70 47 70; RESISTOR 47 70 53 70;
DRAW 53 70 60 70 60 30 20 30 20 35; CAPACITOR 20 35 20 36;
DRAW 20 36 20 50 17 50; RESISTOR 17 50 13 50;
DRAW 13 50 10 50; CIRCLE 10 50 9.7 50
MOVE 20 50; DRAW 20 50 27 50; RESISTOR 27 50 33 50; DRAW 33 50 46 50
MOVE 40 70; DRAW 40 70 40 55 46 55; MOVE 46 52.5;
AMP 46 52.5 57 52.5; DRAW 57 52.5 60 52.5;
MOVE 40 50; DRAW 40 50 40 40; CAPACITOR 40 39;
DRAW 40 39 40 35; GROUND 40 35 40 33
HEIGHT 1.5; WIDTH 1; FONT TRIPLEX ITALIC; JUSTIFICATION CENTER;
MOVE 30.2 66.5; TEXT 39.2K
```

4.46 Generate a Physical Diagram

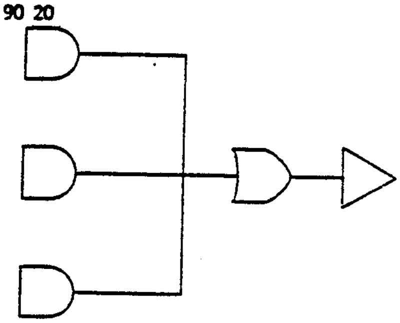
Generate a physical diagram. Note the packing of multiple commands on a line via the use of the terminator character (here = ;). See figure 46.



```
TERMINATOR CHARACTOR ;
FEEDBACK OFF; ERASE;
DRAW 90 10 80 10 80 5 20 5 20 60 45 60 45 45
DRAW 45 45 47 45 47 43 43 43 43 58 22 58 22 7 78 7 78 12 90 12 90 10
DRAW 90 28 78 28 78 58 57 58 57 43 53 43
DRAW 53 43 53 45 55 45 5 60 80 60 80 30 90 30 90 28
ARC 45 57 50 60 55 57; ARC 45 55 50 58 55 55; ARC 47 44 50 46 53 44;
DRAW 47 44 53 44; DRAW 55 75 45 92; DRAW 45 25 55 15;
HEIGHT .5; WIDTH 1; ARROW 20 90 30 90 46.5 90 46.7 57 50 20 90 20
ARROW 20 80 30 80 52 80 51.95 57.4 50 20
HEIGHT 2; WIDTH 1; MOVE 30 85; TEXT LASER BEAM;
MOVE 44 13; TEXT MO TARGET
MOVE 59 22; TEXT OPTIC AXIS OF SPECTROGRAPHY
```

4.47 Generate a Logic Diagram

Generate a simplified logic diagram. See figure 47.



```
FEEDBACK OFF
ERASE
AND 20 20 30 20
AND 20 50 30 50
AND 20 80 30 80
DRAW 30 20 50 20 50 50 30 50 50 50 50 80 30 80
DRAW 50 50 60 50
OR 60 50 70 50
DRAW 70 50 80 50
AMP 80 50 90 50
```

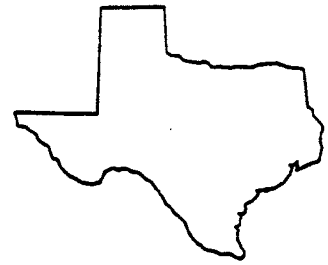
4.48 Generate a Chemical Equation

Generate a chemical equation. Note HW = alternate combined HEIGHT and WIDTH commands. See fig. 48.

```
FEEDBACK OFF
FONT TRIPLEX ITALIC
HW 4 3
MOVE 20 50
TEXT SUB()7UNSB()NSUP()13UNSP()
HW .5 1
ARROW 32 51 40 51
HW 4 3
TEXT SUB()6UNSB()CSUP()13UNSP() + LC()BETA()SUP()+UNSP() + NU()
```

4.49 Generate a Map

Map information exists on a file as (x,y) coordinate pairs. The (x,y) pairs are ordered in the sense that they are to exist in the file in the same sequence that they are to be drawn. Read in the data. Plot out a map. See fig. 49.

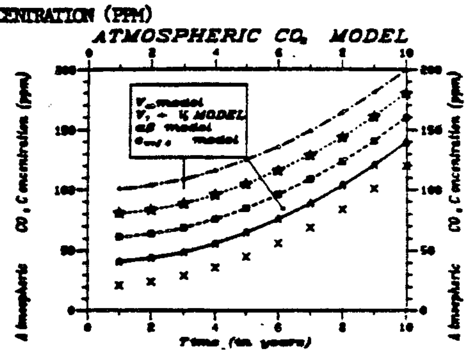


```
READ file name X Y
PRE-SORT OFF
CORNER COORDINATES 20 20 80 90
PLOT Y X
```

4.50 Generate a Multi-Trace, Hershey-Font Plot

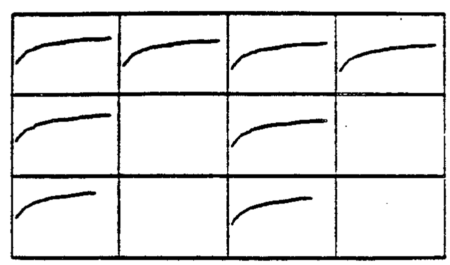
Generate a multi-trace plot with a variety of plot character types and line types. Have the titles and labels in Hershey triplex italic. See figure 50.

```
FONT TRIPLEX ITALIC
TIC MARKS OUTSIDE
TITLE UC()ATMOSPHERIC COSUB()2INSB() MODEL
YLABEL UC()ALC()TMOSPHERIC UC()COSUB()2INSB() CLC()ONCENTRATION (PPM)
XLABEL UC()TLC()IME (IN YEARS)
CHARACTERS X A BOX START TRIANGLE
LINES BLANK SOLID DASH DOT DA2
PLOT X**2 + 20  FOR X = 1 1 10 AND
PLOT X**2 + 40  FOR X = 1 1 10 AND
PLOT X**2 + 60  FOR X = 1 1 10 AND
PLOT X**2 + 80  FOR X = 1 1 10 AND
PLOT X**2 + 100 FOR X = 1 1 10
```



4.51 Generate a Multi-Frame Plot

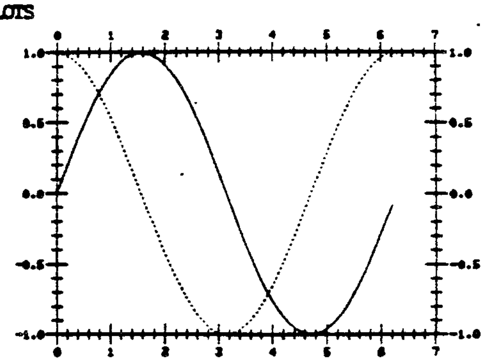
Data exists on a file for a response variable Y and independent variables X1 (many levels), X2 (3 levels) and X3 (4 levels). Read in the data. Carry out an analysis to determine the effect of the 3 variables. Do so by generating a multi-frame plot consisting of distinct subplots where each is a plot of Y versus X1 for some fixed value of the levels of X2 and X3. This technique is also applicable to residual analyses. My apprecia- tion to Wes Nicholson for pointing out this technique to the author. Note that some of the 3 x 4 = 12 sub- plots may be empty for unbalanced designs. See fig. 50.



```
READ Y X1 X2 X3
FEEDBACK OFF
TIC MARKS OFF
TIC MARK LABELS OFF
LET X0 = 10
LET Y0 = 10
LET XDEL = 20
LET YDEL = 20
LET NUMROW = 3
LET NUMCOL = 4
PRE-ERASE OFF
ERASE
CALL PLOTSUB. FOR ROW = 1 1 NUMROW FOR COL = 1 1 NUMCOL
———
. THE FOLLOWING IS A SUBPROGRAM USED FOR MULTI-FRAME PLOTS
LET X1 = X0 + (COL-1)*XDEL
LET X2 = X1 + XDEL
LET Y1 = Y0 + (ROW-1)*YDEL
LET Y2 = Y1 + YDEL
FRAME CORNER COORDINATES X1 Y1 X2 Y2
PLOT Y X1 SUBSET X2 ROW SUBSET X3 COL
```

4.52 Generate a Color Plot

Generate a multi-trace plot where one trace is green and the other is yellow. (Photographic restrictions on the NCGA Proceedings preclude color plots). See fig. 52.




```
COLOR ON
LINES SOLID DOTTED
LINE COLORS GREEN YELLOW
PLOT SIN(X) FOR X = 0 .1 6.3 AND
PLOT COS(X) FOR X = 0 .1 6.3
```

5. DATAPLOT IMPLEMENTATION INFORMATION

The DATAPLOT system was initially (1977) written for implementation at the National Bureau of Standards. The underlying code consists of about 600 subroutines and 200,000 lines of code (including abundant internal documentation in the form of comment statements). On NBS's Univac 1108, the entire system (with overlaying and segmentation) takes 74K words. Without segmentation it would take about 600K words. On a Vax, the system occupies about 2.3 Mbytes.

For continuous graphics, the DATAPLOT system as implemented at NBS makes use primarily of Tektronix 40XX series display terminals. It has run on all of the Tektronix terminals from the 4006 on up to the 4054. The 4027 is used for color graphics. It is also compatible with the 4112 and 4114 Tektronix terminals.

When only discrete graphics is needed, DATAPLOT can be used with a wide variety of terminals (Texas Instrument, Omron, Hazeltine, etc.). It can, of course, be run in batch mode with output directed to a high-speed printer. The system may also be run with off-line continuous graphics devices such as the Tektronix penplotter, Calcomp plotter, Versatec plotter, and Zeta plotter. For construction of printed circuit diagrams, the Tektronix penplotter on paper or acetate medium is of use. Output may be directed to several devices simultaneously.

The system as implemented at NBS does make use of Tektronix PLOT-10 software, but the linkage to such software is through a single subroutine and so substitution of graphics software subsystems other than PLOT-10 is feasible. The 74K words size as quoted above is inclusive of this low-level PLOT-10 software.

The above information pertaining to DATAPLOT at NBS is for illustrative purposes only; the system and the underlying code is in no sense local or peculiar to the NBS computer; DATAPLOT has been designed and coded for generality; consider the following enhancements:

1) Transportability—machine constants for a wide variety of computer types (IBM, CDC, DEC, Honeywell, Univac, Cray, Burroughs, Vax, Interdata, Data General, HP, etc.) are automatically included within the DATAPLOT code;

2) Computer/Compiler Generality—a safe, conservative discipline of restrict-ing underlying code; code; source code to subsets of FORTRAN Standard 66 and 77 has been imposed.

3) Portability—the system is coded (& verified) in PFORT (subset of ANSI FORTRAN); both FORTRAN 66 and FORTRAN 77 versions are available;

4) Core-size Generality—the dimensions of the main internal data storage array are hard-coded only within 1 subroutine and symbolically referred to in all of the remaining DATAPLOT code;

5) Modularity—the system has been partitioned into functionally-independent segments;

6) Implementability—any system-dependent local operations (e.g., opening a mass storage file) are isolated to a few subroutines, and referred to in a system-independent symbolic fashion in all calling subroutines;

7) Debuggability—a dynamic, interactive debugging/tracing capability is included as a standard command in the DATAPLOT language;

8) Maintainability—a modularized, structured, uniform style for the underlying code has been rigorously adhered to;

9) Expandibility—the system has been partitioned into expandible blocks of logically similar subroutines;

10) Graphics Output Generality—primary output may be on continuous display terminals (e.g., Tektronix 4014), on discrete narrow-width display terminals (e.g., TI 700), on discrete wide-carriage terminals (e.g., AJ 832), or on the discrete wide-carriage high-speed batch printer; secondary output may, of course, be directed to any locally-available secondary output device (e.g., Calcomp, Versatec, Pr-80, etc.);

11) Graphics-Hardware Independence—all device-dependent local operations (e.g., generating a local hardcopy of the screen contents) are isolated to 1 subroutine, and referred to in a system-independent symbolic fashion in all calling subroutines;

12) Graphics-Software Independence—all calls to low-level graphics software (e.g., accessing the Tektronix PLOT-10 software) are isolated to 1 subroutine, and referred to in a system-independent symbolic fashion in all calling subroutines;

Extensive external documentation describes not only the language, but more importantly the application of the language to a wide variety of "real world" problems. The documentation for the important FIT command alone consists of over 250 pages, starting with a few pages of FIT specifications, followed by many pages of data analysis considerations for model fitting, and finishing with a detailed presentation of 20 commonly-encountered fitting examples.

6. AVAILABILITY

TM YOUR RUN IS CALLING FOR A UNLBL BLANK AGAIN.
As of March 1, 1982, DATAPLOT has been available for general distribution. It is being distributed through the National Technical Information Service. The distribution medium is magnetic tape. Included also is a second tape containing a systematic procedure for bringing up the system—with an extensive set of test problems. The acquisition fee from NTIS is $900. For further information, contact the author at the following mailing address:

James J. Filliben
Statistical Engineering Division
Center for Applied Mathematics
Administration Building, A-337
National Bureau of Standards
Washington, D.C. 20234
301-921-3651

or contact NTIS directly at:

National Technical Information Service
United States Department of Commerce
Springfield, Virginia 22161
301-487-4807

7. REFERENCES

Caporal, P. M. and Hahn, G. J. (1981). An Overview of Tools for Automated Statistical Graphics. General Electric Corporate Research and Development Technical Information Series Report No. 81CRD024, Schenectady, New York.

Cowell, W. (1977). Portability of Numerical Software. Lecture Notes in Computer Science No. 57, Springer-Verlag, New York.

Filliben, James J. (1978). DATAPLOT—An Interactive System for Graphics, Fortran Function Evaluation, and Linear/Non-linear Fitting. Proceedings of the Statistical Computing Section of the American Statistical Association: 1977.

Filliben, James J. (1979). Factors Affecting the Use of Statistical Graphical Software. Proceedings of the Twelfth Interface Symposium on Computer Science and Statistics. Toronto.

Filliben, James J. (1979). New Features in DATAPLOT—A Language for Graphics, Non-linear Fitting, Data Analysis, and Mathematics. Proceedings of the Statistical Computing Section of the American Statistical Association: 1977.

Filliben, James J. (1980). A Review of DATAPLOT—An Interactive High-Level Language for Graphics, Non-linear Fitting, Data Analysis, and Mathematics. Proceedings of the Statistical Computing Section of the American Statistical Association: 1980.

Filliben, James J. (1981). DATAPLOT—An Interactive High-Level Language for Graphics, Non-Linear Fitting, Data Analysis, and Mathematics. Computer Graphics, Volume 15, Number 3, August 1981, pages 199-213.

Francis, I. (1981). Comparative Review of Statistical Software (Edition 2). International Association for Statistical Computing. 358 Ives Hall, Cornell University, Ithaca, New York.

8. KEYWORDS

Computer Graphics, Graphics Languages, Fitting, Modeling, Statistics, Data Analysis, Mathematics, Mathematical Modeling, Schematics, Diagrams, Software, Portability, High-Level Languages, Interactive Computing.

9. COMPUTER REVIEWS CATEGORIES

8.2, 3.1, 3.2, 3.3, 4.0, 4.1, 4.13, 4.2, 4.20, 4.22, 4.6, 5.1, 5.12, 5.15, 5.16, 5.5