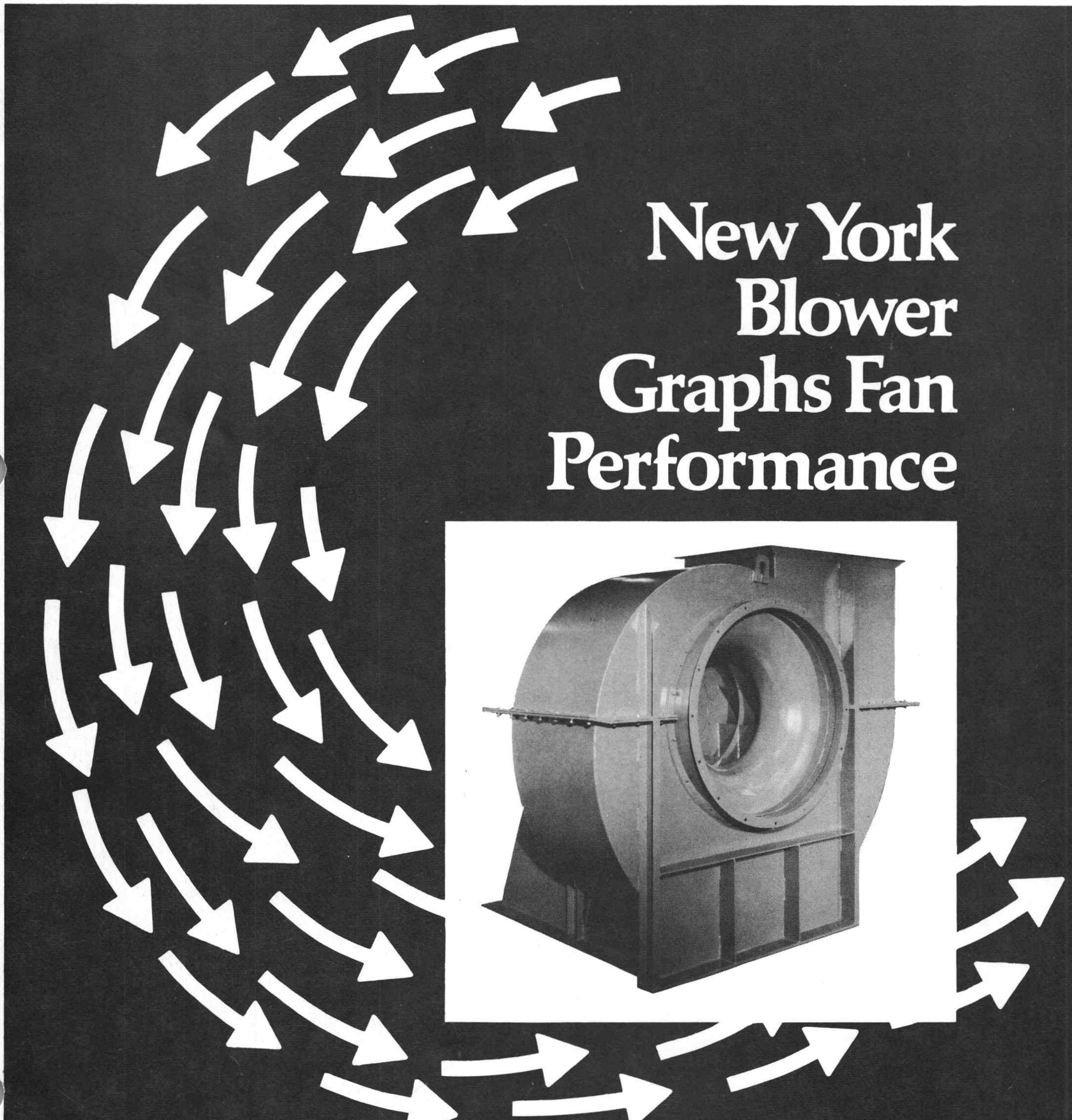
Tekniques

The 4050 Series Applications Library Newsletter August 1, 1980 Vol. 4 No. 5



Tekniques

In This Issue

TEKniques, the 4050 Series Applications Library Newsletter, is published by the Information Display Division of Tektronix, Inc., Group 451, P.O. Box 500, Beaverton, Oregon 97077. It is distributed to TEKTRONIX 4050 Series users and members of the 4050 Series Applications Library.

Publishing Manager	Ben Buisman
Managing Editor	Patricia Kelley
Editor	Terence Davis
Technical Editor	Dan Taylor
Graphic Design	John Ellis
Circulation	Rory Gugliotta

To submit articles to TEKniques or for information on reprinting articles, write to the above address. Changes of address should be sent to the 4050 Series Library serving your area (see Library addresses).

New York Blower Graphs Fan Performance

by Terry Davis
TEKniques staff

with Tim Barnhart
The New York Blower Co.
LaPorte, IND

Designing and building a system to move air is no simple task, whether the system is for air conditioning, fume evacuation, combustion air, or other uses. There are many factors to take into account in the design, several that relate directly to the fan to be used in the system: torque, horsepower required, and air flow volume and pressure are some of these. How, for instance, does the horsepower requirement relate to the delivered flow? Or how does air flow relate to static pressure? And how do these relationships change over the operating range of the fan? Fitting the correct fan into the system requires integrating these factors into the system design.

But that leads to another question: how does the fan maker communicate these parameters and their changing relationships to the air system designer? At The New York Blower Company, graphics were chosen as the answer some years ago. Fan Performance Curves and Speed-Torque Charts were developed to be presented to inquiring customers, depending upon their specific needs. At first these charts were calculated and plotted by hand. But starting four years ago, the 4050 Series Graphic Computing Systems began to take on that burden, with the help of the 4662 Plotter.

The New York Blower Company is located in LaPorte, Indiana, near the shores of Lake Michigan. From this quiet midwest setting, they manufacture and distribute industrial blowers throughout the United States and, through licensees, internationally as well. Blower sizes range from the smallest, an eight-inch wheel, to their huge 85″ wheel blower. Three basic centrifugal fan types are produced — forward curve (or squirrel cage), radial blade wheels, and curved radial blade wheels. Other special-purpose fans produced include vaneaxial fans, pressure blowers, tubular acoustafoil fans, and corrosion-resistant fiberglass fans for fume exhaust applications.

Basic Graphic Applications

There are two 4050 Series Graphic Computing Systems at New York Blower Company. The first, a 4051, has been there for about four years, and currently resides in the Research Laboratory. The recently-acquired 4052 is used most often; it's located in the Engineering Order Processing Section. Each machine is complemented by a 4662 Plotter. Their uses vary with the area that they're in.

The 4052 system in the Engineering Order Processing Section is used most often; that's where the charts are made. This department receives requests for fans to meet a certain set of specifications — for instance, a required air flow volume at a given pressure and temperature. (Other specifications, such as mounting and drive types, are found in their catalog.) A fan that meets that particular performance intersection can be selected directly from the catalog tables, but that doesn't tell much about the fan's performance over a range of operation conditions. Fan performance curves are required to tell the whole story, curves that *were* laboriously generated by hand. So a program was developed to let the 4050 System take over that task.

Performance Curves

The performance curves for a particular fan describe its operating characteristics over its entire operating range. While the full operating range is shown, emphasis is placed on the requested application.

The example in Fig. 2, for instance, shows the characteristics for a type 404 fan (with a wheel diameter of 40 inches), moving 15,000 CFM (Cubic Feet per Minute). At that flow volume, emphasized by the vertical line in the graph, the fan has a static pressure of 10 inches of water and requires 42.9 BHP (Brake Horsepower).

Other points in the static pressure curve represent the total safe operating range for the fan. For instance, at 10,000 CFM, the static pressure would be about 12 inches, while using 32 BHP. However, these operating areas must be balanced against a system requirement. Each set of conditions also has an operating efficiency. And mechanical efficiency translates to energy efficiency, a topic foremost in many designers minds.

Speed-Torque Curves

Figure 3 is a Speed-Torque Curve for a similar type of fan. In this graph, both BHP and Torque are displayed simultaneously, as a function of fan speed. This, along with wheel inertial (WR^2), is important to the designer of an air movement system, to efficiently select the proper motor type and size for an installation.

As these programs were put into use, the operating ease of the 4050 Series Graphic Computing Systems became an apparent asset as well. With the programs in place, the system can be operated by anyone in the group, even new staff members. That's a major factor when you send out as many as 100 Performance Curve plots each day.

Custom Shafts and Bearings

There is other customer-directed information that's generated by the 4052 in New York Blower's order processing section. When a custom-designed fan system is ordered, the 4052 is put to work on shaft and bearing calculations. Using the shaft size, overhangs, and type and size of bearings, the program calculates the first and second critical resonance for the fan. In addition, the system will calculate the required fan base. While these programs don't use the graphic capabilities, the higher speed of the 4052 processor shows forth in these calculations.

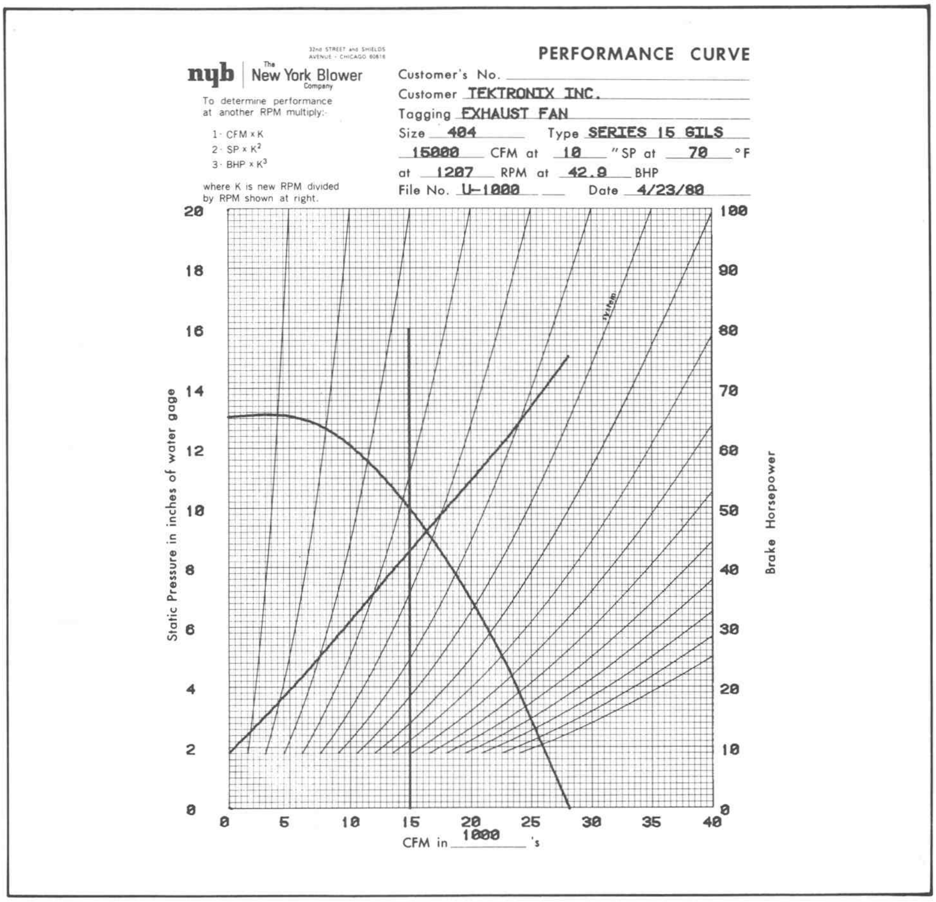


Fig. 2. A Performance Curve Plot for a size 404, Series 15 GILS fan, showing the relationships of brake horsepower, and fan speed.

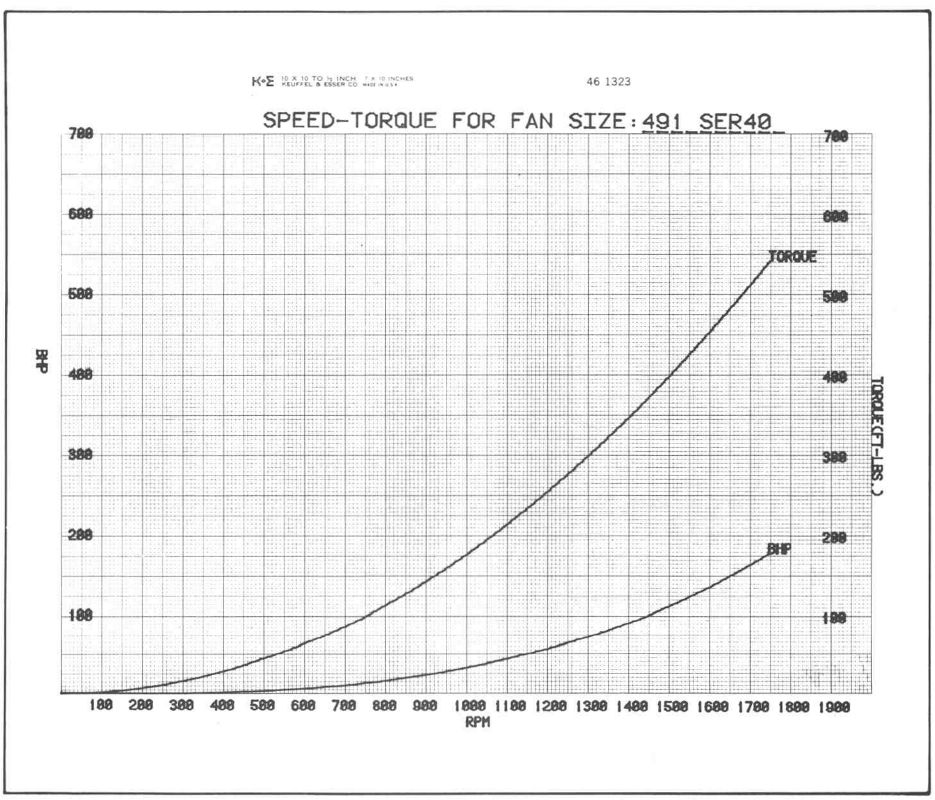


Fig. 3 A Speed-Torque curve for a size 491, Series 40 fan, showing the relationships of torque, brake horsepower, and fan speed.

In the Lab Too

A separate building at New York Blower houses the Research Laboratory, where their 4051 resides, also with a 4662 Plotter. In this lab, they test all ranges of fan sizes and types for air flow, static pressure, mechanical efficiency and sound level. The system plots the results of all lab tests, for easy reference to test results and quick communication of the results to others.

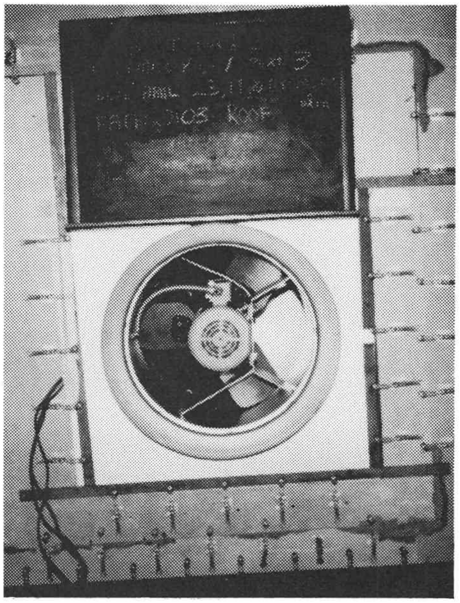


Fig. 4. Laboratory testing of a fan used as a small crop dryer.

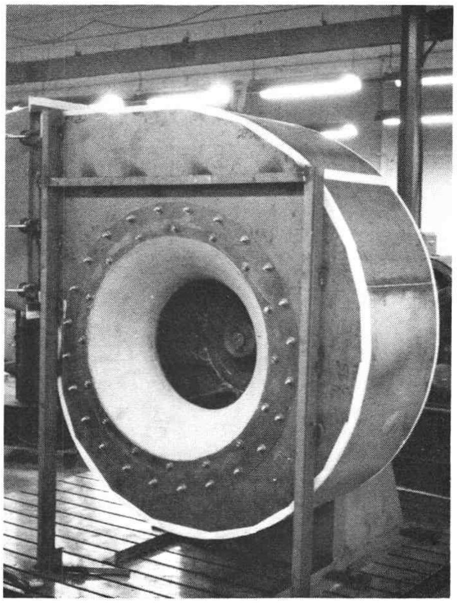


Fig. 5. A larger, typical centrifugal type fan undergoing analysis in the New York Blower Laboratory.

To test a fan in the lab, the fans are mounted in an appropriate housing to secure them for the test, and are then instrumented to gather the required data. Small fans might be mounted in the test chamber wall, as shown in Fig. 4, while the bigger blowers might be secured in a free-standing fashion (Fig. 5). Of course, the test being performed dictates much of the mounting requirements.

Plots of test data come in several forms; Fig. 6 is one example of a test lab plot. In that plot, three factors — Brake Horsepower, Static Pressure, and Mechanical Efficiency — are plotted as functions of air flow. Tests and test plots of this type help the engineering group at New York Blower to optimize these important factors in the design of new fans.

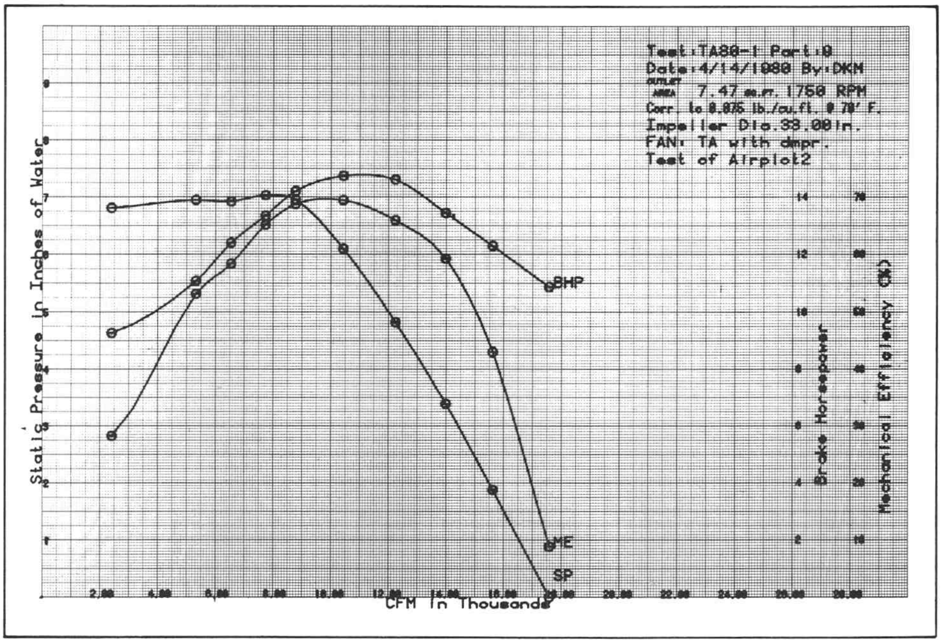


Fig. 6. An example of the test plots produced in the laboratory, showing static pressure, mechanical efficiency, and brake horsepower, as they relate to the air flow produced.

In addition to the local processing and plotting with the lab's 4051/4662 combination, some outside timeshare processing is done as well. Finite Element Modeling is used in the design process, for instance, using the processing and software of Structural Design Research Corporation in Cincinnati, Ohio. To make use of their services, data is transferred through the lab modem and into their network for analysis. The returned data can be plotted as well.

Measurable Results

Improvements are somewhat hard to measure in the lab, especially since they are using the system for things that they couldn't do before. But the changes are easier to see in the graphs and plots the order processing section prepares.

As noted earlier, graphs were formerly prepared through hand calculation and manual plotting. Using this process, each customer's Performance Curve took about 30 minutes to produce, while the 4052/4662 system completes them in about 2-3 minutes. The change in productivity was quickly apparent, allowing fewer people to complete more work. The first system paid for itself in a year.

In addition, there is greater acceptance of the validity of plots generated by the 4052/4662 combination. No matter how carefully a hand-prepared plot is calculated and drawn, it is still a hand-prepared plot, and subject to the human errors that attend such endeavors. But accuracy is assumed in the computer-drawn plots; the first hurdle is overcome. In the lab or in the engineering offices, the 4050 Series Graphic Systems continue to play an important role in increasing productivity at New York Blower Company. Easy use, processing power, and high resolution graphics are some of the reasons why.

Announcing 4050D02: Statistics Disk-based Tests and Distributions

Fig. 1. The 4050D02 PLOT 50 Statistics: Tests and Distributions disc-based software package.

by Terry Davis
TEKniques staff

If your application requires that you summarize lots of numeric data, or if you're a decision maker in need of statistical aid, you'll be happy to hear about the newest Plot 50 software package. 4050D02 PLOT 50 STATISTICS: TESTS AND DISTRIBUTIONS is a software tool that contains all of the functions of the tape-based 4050A01 PLOT 50: Statistics Vol. 1 version, and more. User interface features are enhanced. There are more statistical routines and more plotting options. And all of the programs are menu-driven and completely interactive.

The easy-to-use nature of Plot 50 Tests and Distributions has been made friendly in several ways. The flow through the various levels is a natural progression from data entry to exploratory analysis to testing and analysis, or finally to plotting of results. And extensive use of the User-Definable Keys (UDKs) makes operation easy even for the novice. The UDK overlay labels the 17 keys that are used for data entry, editing, and storage.

High-quality, useful plots are available from the analyses, including the new routines and options. Cumulative Histograms and Suspended Histograms are available, as well as Scatter Plots, Stem-and-Leaf Displays, and Box-and-Whisker plots.

In addition, an entirely new data management section has been added, as well as a new random number generation program. The Two-Sample F-test is also a new addition in this package, as well as the resistant line option for probability plots, distribution analysis, and regression analysis. We'll look at some of these in more detail, a little later in this article.

Standard File Format Featured

The PLOT 50 Standard File Format, described in TEKniques Vol 4. No. 1, is another feature; it allows data entered into, or generated by, any 4050DXX software package to be accessed and used by other "D-Series" software packages. Examples are Plot 50 Easy Graphing, and the disk version of PLOT 50 Business Planning and Analysis Vol. 2, both also described in previous TEKniques issues. The new PLOT 50 Statistics: Tests and Distributions can analyze data generated by another software package, or can generate data to be analyzed by another software package.

New Features

The Standard File Format eases the interface between 4050D02 and other compatible software packages. But the most noticeable change is in the interface between you and the system with Statistics Test and Distributions installed. 17 functions are defined and called through the User-Definable Keys, to let you select data entry, editing, and output options at the touch of a key, in addition to listing data and variables, and controlling program execution. The function overlay for the User-Definable keys is shown in Fig. 2.

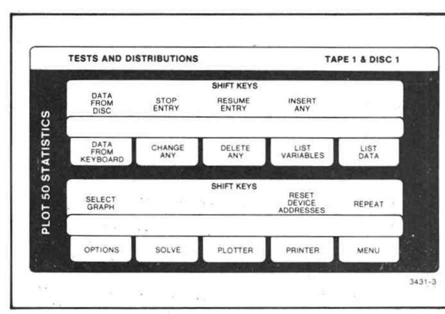


Fig. 2. The Statistics: Tests and Distributions User-Definable Key overlay.

The entry functions allow entering data from the keyboard (key 1), from the disk (key 11), and to stop entry (key 12). The Editing keys allow you to Change Any entry, Delete Any entry, Insert Any entry, or Resume Entry. Output control consists of directing output to a plotter or a printer, or resetting the addresses of the peripheral devices. Two other keys allow listing variables and listing data.

The program execution functions on User-Definable Keys give you easier and more-direct control over program execution. Key 6, for instance, is the Options function. You can press this key to list or perform program options. If the current program has no options, the following message will appear:

```
1:Options not enabled
2:
3:
```

If the current program has only one option, the option is executed immediately when the key is pressed. If more than one option is available, the available options will be displayed. For instance, pressing the Options key during the Paired t-Test will result in the following display, allowing you to choose the desired option:

```
4:
5:                               ** PAIRED t-TEST **
6:                                     OPTIONS
7:
8:1.  CONFIDENCE INTERVALS
9:2.  PUT DIFFERENCES ON DISC
10:
11:Please choose one of the above:  1
12:
```

The Solve key is used to rerun the program without recalling the menu, as when the variables have not changed; it's faster than recalling the program menu. If you haven't yet run a program when this key is pressed, the following message appears to direct you to the menu:

```
13:
14:Select menu item.  Press Function Key 10 (MENU)
15:
```

Pressing the Menu key recalls the Statistics Tests and Distributions menu, if you're in the Tests and Distributions facility:

```
16:
17:              PLOT 50 STATISTICS:  TESTS AND DISTRIBUTIONS
18:
19:                               ** MENU **
20:
21:
22:DATA ENTRY                      TWO SAMPLES
23:  DATA FROM KEYBOARD       1      T-TEST, EQUAL VARIANCE   10
24:  DATA FROM DISC           2      T-TEST, UNEQUAL VARIANCE 11
25:                                  T-TEST, PAIRED           12
26:DESCRIPTIVE STATISTICS     3      F-TEST                   13
27:
28:EXPLORATORY PLOTTING            SIMPLE REGRESSIONS         14
29:  HISTOGRAMS               4
30:  STEM-AND-LEAF DISPLAY    5    CONTINGENCY TABLES
31:  BOX-AND-WHISKER PLOT     6      2 X 2                    15
32:  SCATTER PLOT             7      R X C                    16
33:
34:PROBABILITY PLOTS AND           RANDOM NUMBER GENERATION   17
35: DISTRIBUTION ANALYSIS     8
36:                                PROBABILITY DISTRIBUTIONS
37:ONE SAMPLE t-TEST AND            (TABLES)                  18
38: CONFIDENCE INTERVALS      9
39:                                DATA MANAGEMENT UTILITIES  19
40:
41:Please choose one of the abov
42:
```

But if you're in the Utility facility, the utility menu is displayed:

```
43:
44:
45:PLOT 50 STATISTICS:  TESTS AND DISTRIBUTIONS UTILITIES
46:
47: 1.  GET FILE
48: 2.  PUT FILE
49: 3.  DIRECTORY
50: 4.  DUPLICATE DISC
51: 5.  KILL FILE
52:
53: 6.  APPEND FILE
54: 7.  LIST FILE
55: 8.  CHANGE TITLE, CHANGE LABELS
56: 9.  COMBINE FILES
57:10.  SORT FILE
58:11.  RENAME FILE
59:12.  TRANSFORM DATA
60:
61:13.  CHANGE CONFIGURATION PARAMETERS
62:14.  STATISTICS: TESTS AND DISTRIBUTIONS
63:
64:
65:Please choose one of the above:
66:
```

The other program control keys are the Repeat key, which repeats the last table of graph produced, and the Select Graph key, which is used to select the plotting mode. When the Select Graph key is pressed, one of two messages is displayed:

```
67:
68:Select graph mode is on.
```

OR

```
69:
70:Select graph mode is off.
71:
72:
```

If graph mode is on, you'll be asked to specify the plotting parameters; if it's off, the plotting parameters will be selected by the program.

New Utilities Too

The statistical routines all operate under the Tests and Distributions facility, which is the default operating condition of the system. But a set of 13 utilities has been added as well, accessible by selecting program 19, Data Management Utilities. Some of the utility functions are as follows: You can GET a file from a PLOT 50 Standard Data Tape or another PLOT 50 Standard File compatible disk, or PUT a file on another Standard Data File or another Statistics: Tests and Distributions disk. You can use DIRECTORY to obtain a listing of all files stored on the data disk or a PLOT 50 Standard Data Tape, or use DUPLICATE to make a copy of the 4050D02 PLOT 50 STATISTICS: TEST AND DISTRIBUTIONS disk for archiving or as another data disk.

Other utilities let you delete data files, append data from one file to another, and list all or part of a file. You can change titles, change labels, combine variables from two files to create a new file, or sort an old file into ascending or descending order to create a new file. Data file names can be changed as well. The Transform utility lets you create a new file in several ways: select specific variables or specific observations from another file, or perform mathematical transformations on data from another file.

New Programs, New Plots

In addition to the easier interface to you, and the new utilities to ease data management, there are several new statistical analysis programs in Statistics: Tests and Distributions. For instance, F- test has been added, allowing you to choose one- or two-sample tests. These tests display the sample mean, variance, degrees of freedom, the value of the statistic and the test conclusion: whether or not to accept the null hypothesis. Let's look at some of the other new programs:

Probability Plots and Distribution Analysis is another addition to the statistics package. This program displays the probability plot of any variable. It will plot using normal, log normal, logistic Weibull, exponential, or uniform distribution. The fluctuations of the data points from the fitted line show how much the data deviates from the theoretical distribution. In addition, you have the option of plotting with least squares method or resistant line method. Figure 3 shows a sample plot of normal probability for water quality data.

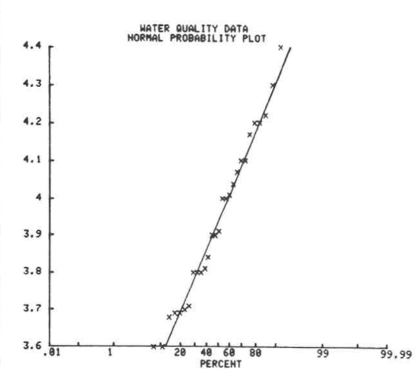


Fig. 3. A sample plot of normal probability for water quality data.

Regression analysis is another valuable program addition. Simple Regressions can be used to fit one of nine models to the data. The program will fit a staight line, using least squares or resistant line method, and plot it. Figure 4 is a sample of a regression analysis plot of water quality data, where the chosen model is Y = A + B * X.

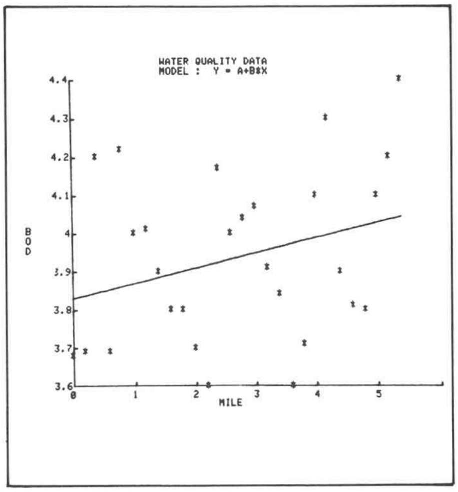


*Fig. 4. A regression analysis plot of sample water quality data. The chosen model is Y=A+B*X.*

In addition to the new statistical analysis programs, the menu of plot possibilities is greatly expanded. This adds the flexibility of being able to explore data in a number of ways, graphically. Plots can include regular histograms, cumulative histrograms, and suspended histograms, such as the one shown in Fig. 5. In a suspended histogram, the distance of the cells from the zero line shows how much the sample distribution deviates from the normal distribution.

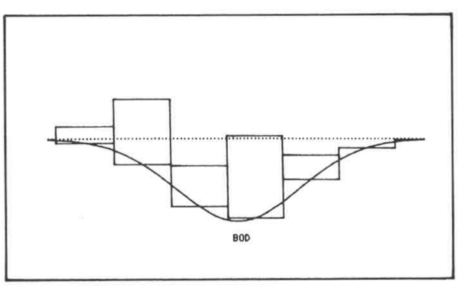
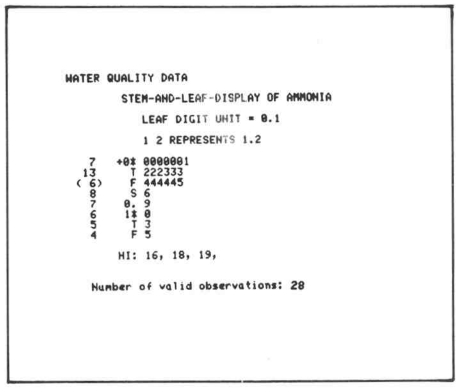


Fig. 5. A sample of a suspended histogram. The distance of the cells from the zero line shows how much the sample distribution deviates from the normal distribution.

Some of the other new plots include the Stem-and-Leaf Display, Box-and-Whisker Plots, and Scatter Plots. Stem-and-Leaf Displays are used to examine the distribution of a variable and determine if any values are extremely high or low. Similarly, Box-and-Whisker Plots give a quick look at the data, quickly show values that are extreme, and therefore likely to be strays, out-liers, or questionable data points. Scatter plots are a third type of "check" plot, which can show the relationship between two variables, in addition to an overall look at the data. Samples of these three plot types are shown in Figs. 6-8.



```
WATER QUALITY DATA
        STEM-AND-LEAF-DISPLAY OF AMMONIA
          LEAF DIGIT UNIT = 0.1
          1 2 REPRESENTS 1.2

   7    +0* 0000001
  13      T 222333
 ( 6)     F 444445
   8      S 6
   7     0. 9
   6     1* 0
   5      T 3
   4      F 5

        HI: 16, 18, 19,

     Number of valid observations: 28
```

Fig. 6. A Stem-and-Leaf display, which can be used to quickly examine the distribution of a variable.

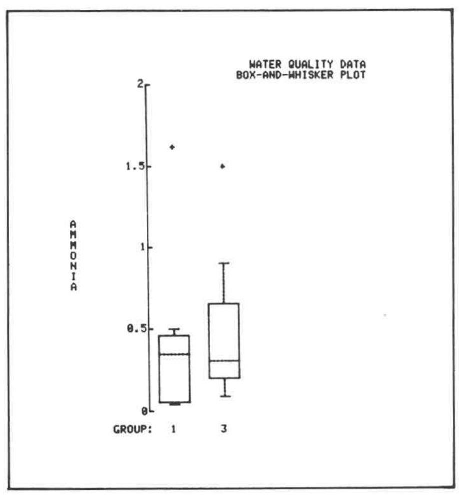


Fig. 7. A Box-and Whisker Plot, which can be used to find strays or "outliers" in the data.

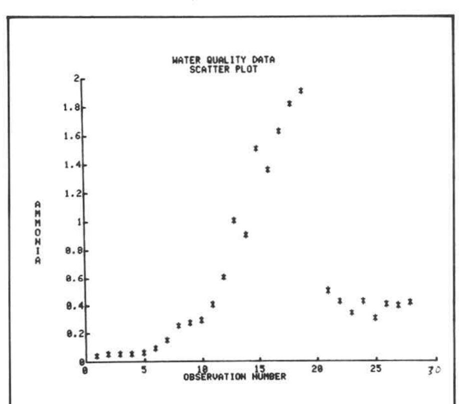


Fig. 8. A Scatter Plot sample. Scatter Plots show data concentration by plotting one variable against another.

More Information

This new software package will enhance your system's effectiveness, in any environment that requires statistical analysis of data. If you'd like more information about 4050D02 PLOT 50 Statistics: Tests and Distributions, contact your local Tektronix office.

On-Line Calorimetry Data Acquisition and Analysis

by D. L. Raschella
J. R. Uchida
J. R. Peterson
Department of Chemistry
University of Tennessee
Knoxville, TN

Background

Our research efforts are focused on determining the basic physico-chemical properties of radioactive transplutonium elements, man-made elements produced in nuclear reactors. In a previous article we described the vital role of the 4051 Graphic System in the acquisition and subsequent analysis of spectroscopic data from these materials. This article outlines the utilization of the 4051 in our solution calorimetry program.

A calorimeter is an instrument that measures the quantity of heat absorbed or evolved in some process. Solution calorimetry involves determination of the heat absorbed or liberated when a substance (usually a solid) dissolves in a liquid. Since transplutonium element samplès available for calorimetric study are typically submilligram in size, our calorimeter was designed to detect the small heats involved in such dissolution experiments. We call our system a solution microcalorimeter because of these design characteristics.

The fundamental principles of calorimetry are straightforward. The experimental heat is measured with a carefully calibrated thermometer. This calibration is usually done by inputting an accurately known quantity of electrical energy (calibration heat) into the calorimeter and noting the response of the thermometer. The thermometer's response to an experimetal heat is then compared with its response to the calibration heat, thereby determining the experimental heat. Research calorimeters are more complex, depending on the type of thermometer employed, the precision of measurement desired, and the extent to which thermal exchange with the calorimeter's surroundings is controlled.

In our microcalorimeter a thermistor is used as the thermometer, so changes in temperature are detected as changes in resistance. A Wheatstone bridge (shown schematically in Fig. 1) is used to measure this resistance change; the change of resistance is monitored by the bridge detector. A block diagram of the circuit used in the calibration of the thermistor's response is given in Fig. 2. The voltages E_h and E_{std} are needed to determine the electrical energy delivered to the calorimeter during a calibration heating.

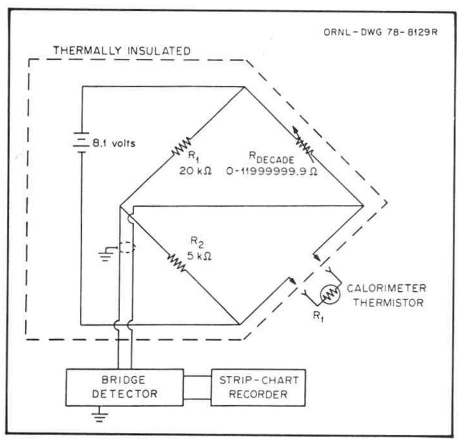


Fig. 1 Temperature measurement system.

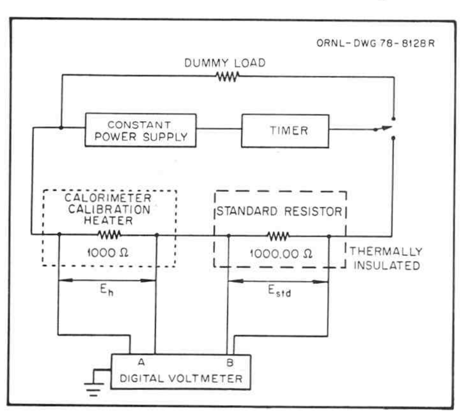


Fig. 2 Calibration heating system.

The microcalorimeter operates under isoperibolic conditions. That is, the temperature of the microcalorimeter itself is allowed to change, but the temperature of the calorimeter's environment is not. The calorimeter's thermal excursions with time can be monitored on a strip-chart recorder or by an automatic measurement system. Our calorimetry facility has the capability for computer-assisted data acquisition and analysis, made possible by the 4051 Graphic System.

Interfacing

Figure 3 is a block diagram of the configuration of the data acquisition/analysis system. The output of the bridge detector and E_h and E_{std} are input to a Hewlett-Packard Model 3495A scanner. At designated times the controller (the 4051) commands the scanner to connect the appropriate channel. The 4051 then directs a Dana Model 5900 digital voltmeter (DVM), through a Dana Model 55 interface, to read the voltage and send it to the 4051. (Dana Model 55 converts the voltmeter's BCD output to the IEEE-488 standard.) Data are collected on a real-time basis, with the aid of a TransEra Model 641 real-time clock, and are stored on magnetic tape using the 4924

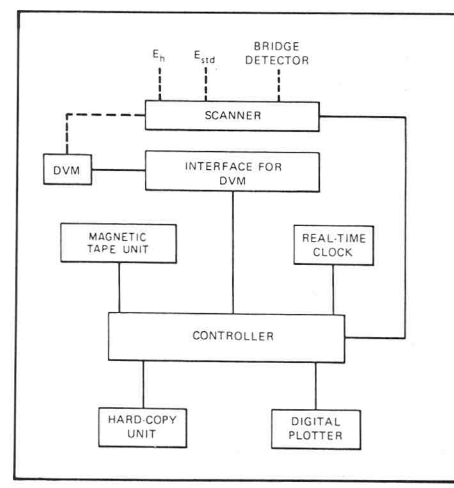


Fig. 3. Block diagram of the data acquisition/analysis system.

Fig. 4. Microcalorimetry facility with 4051 system.

Tape Unit. A 4631 Hard Copy Unit and a 4662 Interactive Digital Plotter provide permanent records of the output. All interfacing is achieved through the GPIB. Dr. Raschella is shown with the 4051 and microcalorimeter systems in Fig. 4.

Data Acquisition and Analysis Software

The program is divided into two parts: acquisition, and analysis/evaluation. The input section is stored in one program file; the analysis/evaluation section is located in several files, with overlaying achieved by depressing appropriate User-Definable Keys (UDKs).

Data Acquisition. The overlay card for the data acquisition routine is shown in Fig. 5. The program tape is inserted into the 4924 tape drive, while the data tape is placed into the 4051 tape drive. New data tapes are automatically MARKed by pressing UDK 15. After entering the sample identification number and date, the data acquisition process is begun by depressing UDK 1. The scanner then samples the bridge detector output at a regular rate. Reading rates may be altered from about three readings per second to one reading every three seconds by pressing UDKs 6 through 10.

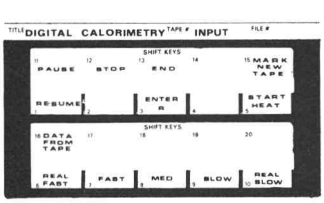


Fig. 5. User-definable keys' overlay card for data acquisition.

The collected experimental data consist of bridge calibrations (commonly referred to as chart sensitivities), calibration heatings, and the dissolution reaction heat of the sample. Typically, several chart sensitivities are determined, interspersed with calibration heatings, before the dissolution reaction heat of the sample is measured. Then several more chart sensitivity and calibration heating determinations are made following dissolution of the sample. (For further information regarding calorimetry and the details of the experimental procedures, refer to reference 2.) Chart sensitivities provide the relationship between changes in voltage (ΔV) and changes in resistance (ΔR) and are a measure of the combined sensitivites of the thermistor and bridge detector. They are performed by changing the decade resistance setting (see Fig. 1) and noting the resultant change in bridge detector output. Decade resistance values are input from the 4051 keyboard after depressing UDK 3. Calibration heatings effectively calibrate the thermistor in units of heat (joules) per decade resistance change caused by the heat input.

The 4051 is prepared for a calibration heating by depressing UDK 5, at which point E_h and E_{std}, as well as the bridge detector channel, are scanned sequentially at a rapid rate. The actual input of heat as electrical energy occurs in a fixed time period and is initiated by a switch on the calorimeter. After the calibration heating the values of E_h [EMF(1)] and E_{std}[EMF(2)] are displayed, and the new bridge resistance setting must be entered. All events must be identified: I for initial entry, C for chart sensitivity, H for calibration heating, and R for the dissolution reaction. Examples of the 4051's CRT displays during an experiment are given in Figures 6a, 6b, and 6c. User-definable key 11 temporarily stops data collection but not the clock, while UDK 12 stops both the data collection and clock and closes the data file. User-definable key 12 is useful if an interruption in data collection is prolonged, such as for the adjustment of some equipment. The end of the experiment is designated by UDK 13, which closes the current data file and stores the sample identification number and a summary of information about the chart sensitivites and heat inputs in file 1 of the data tape. Analysis of the data is begun by pressing UDK 16.

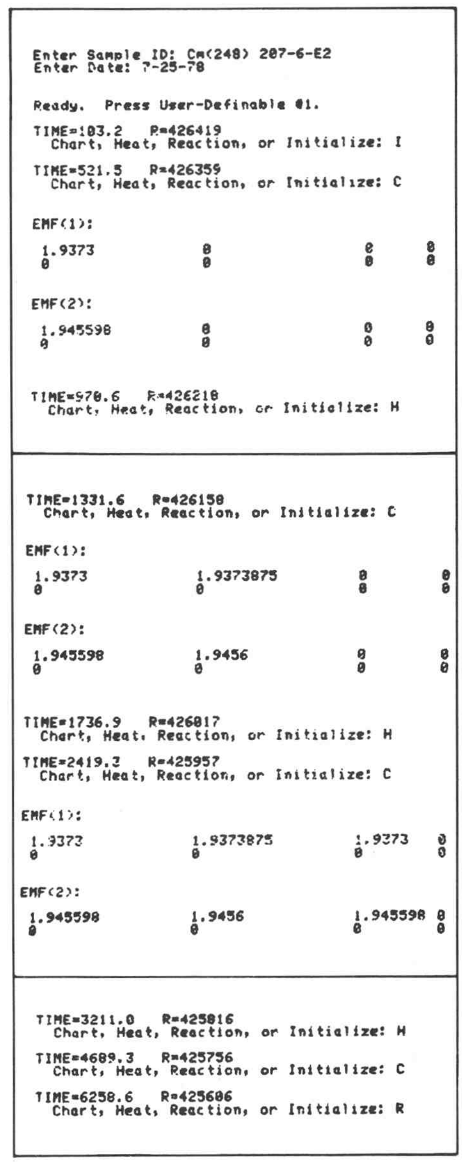


Fig. 6., a, b, c Sample CRT displays during data acquisition.

Data Analysis/Evaluation The overlay card for the data analysis/evaluation routine is shown in Fig. 7. User-definable key 5 is used to plot the voltage-time data. An example of such a plot is given in Fig. 8, where chart sensitivities occur around 450 and 1180 seconds and calibration heatings at about 750 and 1600 seconds. User-definable key 1 initiates the chart sensitivity analysis routine. This extrapolates the drift line segments prior to and following the resistance change towards an "imaginary" line that intercepts the voltage-time curve halfway through the event which caused the change in resistance. Then the distance (ΔV) between the points at which the extrapolated drift line segments intercept the "imaginary" line is determined. The drift line segments to be extrapolated are selected by the operator by using UDKs 6 through 10. This procedure is illustrated in Figs. 9a and 9b.

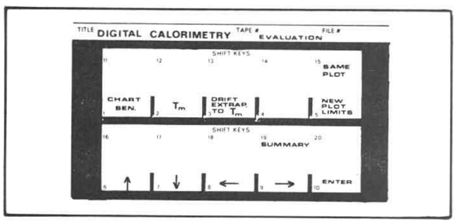


Fig. 7. User-definable keys' overlay card for data analysis/evaluation.

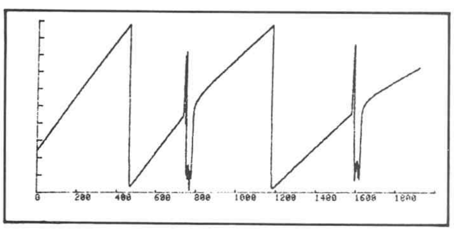


Fig. 8. Voltage-time plot of some typical calorimetric data.

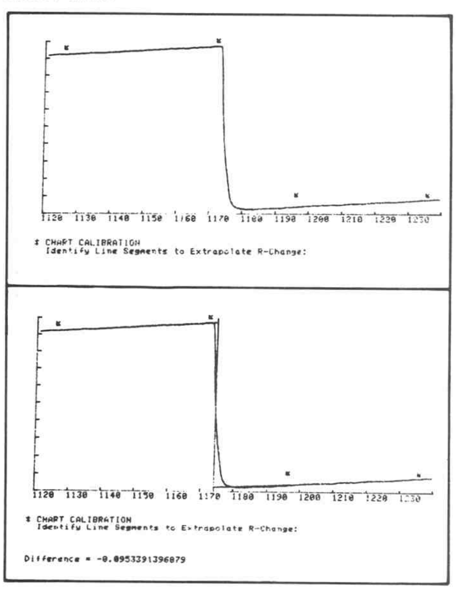


Fig. 9., a, b Analysis of chart sensitivities.

Calibration heatings are evaluated in a similar manner. Drift line segments before and after the heat input are extrapolated towards a point midway into the heating period. The "imaginary" line for calibration heatings and the dissolution reaction heat is known as "T_M." (An example of its determination is given later.) The distance between the extrapolated drift line segments at T_M is then measured. User-definable keys 3 and 6 through 10 are used in this case. The extrapolations are illustrated in Figs. 10a and 10b. The beginning (B), end (E), and T_M are automatically marked by the program.

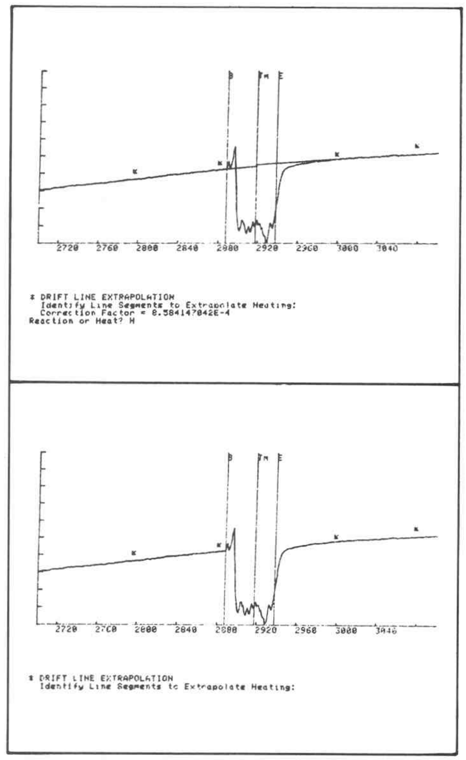


Fig. 10., a, b Analysis of calibration heating.

A stepwise illustration of a determination of T_M is given in Figs. 11a through 11d. Using UDKs 2 and 6 through 10, line segments are identified (Fig. 11a). These line segments are extrapolated inward, tangentially from the inner point chosen, in discrete steps, and the areas between the extrapolated lines and the experimental curve are calculated for each step, as shown in Figs. 11b, 11c, and 11d. These areas are determined by a modified Newton's integration technique. (Alternatively, a nonlinear extrapolation of the chosen line segment might be used.) T_M is the one point at which the areas between each extrapolated line, the experimental curve, and the T_M line are equal. Once T_M is found, the drift line segments are extrapolated to T_M as before, and the difference between the segments at T_M is determined (see Figs. 12a and 12b). Upon concluding the data analysis, depressing UDK 19 yields a summary of all pertinent information. This information is used in subsequent calculations to determine the heat of solution of the sample.

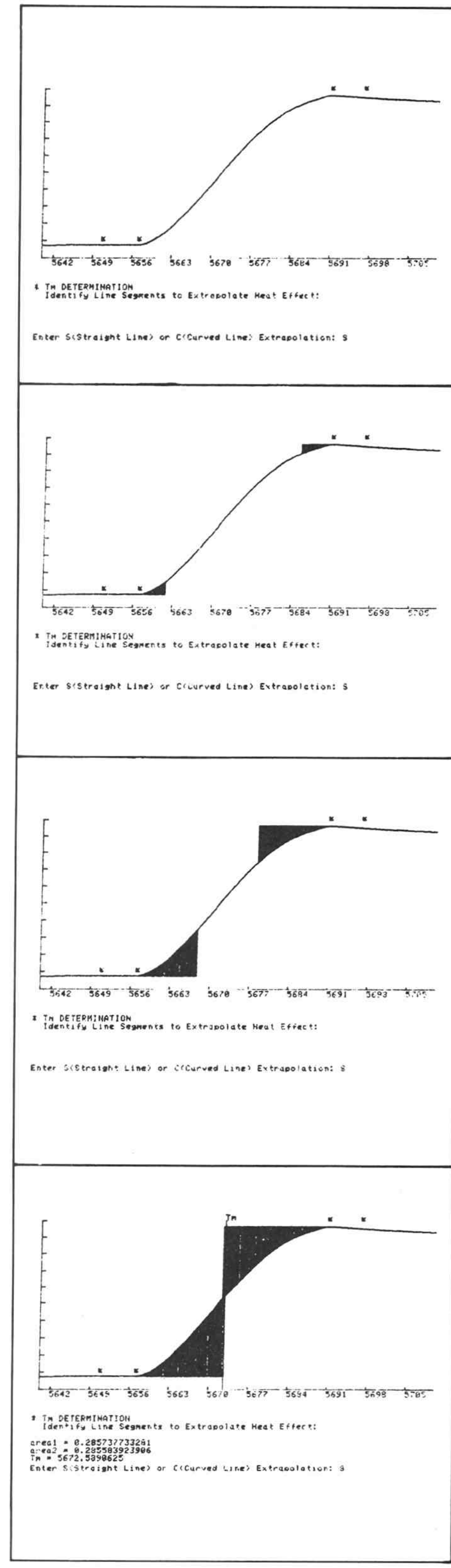


Fig. 11., a, b, c, d Determination of T M.

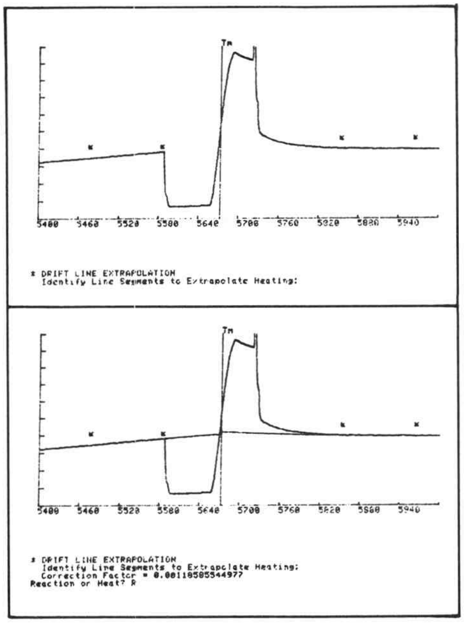


Fig. 12.,a, b   Analysis of dissolution reaction heat.

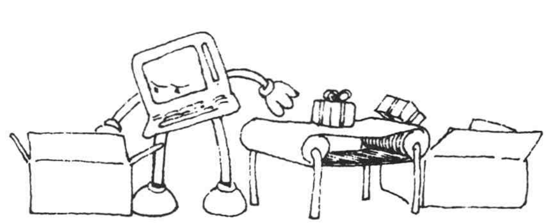
Conclusions

An appreciable number of extrapolations and integrations are required in evaluating calorimetric data; the 4051 is particularly helpful in these aspects of data analysis. Not only can the 4051 reduce the time required for evaluation of the experimental data, but it can also eliminate "eye-balling" errors inherent in "manual" extrapolations and integrations. The 4051 has also increased our flexibility, by providing the ability to perform the necessary extrapolations on a nonlinear basis, as well as on a linear basis. The 4051 offers unsurpassed accuracy and ease in this regard; its capability for regression analysis and integrations is invaluable.

References

1. D. L. Raschella, R. L. Fellows, and J. R. Peterson, "Isoperibol Microcalorimeter for Measuring Enthalpies of Solution of Radioactive Elements and Compounds; Enthalpy of Solution of Curium-248 Metal," Journal of Chemical Thermodynamics (in press).
2. D. L. Raschella, Ph.D. Dissertation, "Solution Microcalorimeter For Measuring Heats of Solution of Radioactive Elements and Compounds," University of Tennessee, Knoxville; U.S. Department of Energy Document No. ORO-4447-081 (December 1978).

INPUT / OUTPUT



Jerry W. Anderson, Phillips Chemical Company of Bartlesville, Okla, has several questions. **Howard Sanders, Technical Support Specialist at Tektronix** replies.

Plotter/Printer On or Off

Is it possible to determine, under 4052 program control, whether or not the plotter and/or printer is turned on? (The motive here is to output a plot to the plotter if it is turned on; otherwise output it to the screen.)

The 4052 by itself cannot determine if a plotter or printer is turned on before it attempts to output (or input). If the output is to the plotter and it isn't on, you'll receive an error message and your program will halt. If the output is to the printer which isn't on, the 4050 system will "go busy" waiting to output. However, a new 4050 ROM Pack, the 4052R09 Real Time Clock ROM Pack* provides the facility to interrupt or continue a program after a prescribed amount of time has elapsed. Taking advantage of this ability, the following routine recovers your program if the plotter isn't on.

```
1 INIT
2 SET KEY
3 GO TO 100
84 IF NOT(F) THEN 86
85 RETURN
86 D=32
87 GO TO 150
100 CALL "ONTIME",0.21
110 F=0
120 D=1
130 MOVE @D:0,0
140 F=1
150 PRINT "DEVICE ADDRESS IS ";D
160 REM CONTINUE WITH PLOTTING PROGRAM
```

*For use only with the 4052 and 4054 systems.

The "ONTIME" command tells the ROM Pack to branch to line 84 after a specified amount of time. If the program is running when the time is up, it branches and when it encounters a RETURN, returns to the statement following the one it executed prior to branching. If a program isn't running, the ROM Pack starts the program executing at statement 84 (a RETURN in this case would return it to idle mode).

Applying this, the above routine branches to line 84 after .21 seconds. The MOVE in statement 130 will generate an error if the plotter isn't on and the program will halt; the variable, F, remains 0. If the plotter is on, F is set to 1 and the program continues. Now, when the ROM Pack instructs the 4050 system to branch, the value in F at line 84 tells whether the program was executing. If it is 1, it simply returns to the program. If it's 0, the device address is set to 32 and the program directed to the plotting routine.

However, there is no way with or without the ROM Pack to recover from the busy state, and determine if the printer,whether interfaced through Option 10 or Option 1, is turned off.

Enabling BREAK on Option 1

Using the Data Communications Interface, is it possible to enable the "BREAK" key without also enabling the "PAGEFULL" break?

This is accomplished by disabling the PAGEFULL break. To do this set the first parameter in CALL "MARGIN" to 0.

From **Bill Fowee at Proctor & Gamble in Cincinnati** comes the next question; again **Howard Sanders** responds.

DIR @1:0,"@"

Thanks for your new column in TEKniques. I found last month's "TLI @1:" useful. How do you DIR @1:0,"@"?

The 4050 Series can't directly output the directory of the 4907 to the GPIB, however, it can output to the tape. Thus, using the tape as an intermediate step and a small program to input and output the directory tape, the directory can easily be sent to the plotter.

```
100 REM Routine to put disk directory on
110 REM tape and then to plotter
120 INIT
130 FIND 1
140 DIRECTORY @33:0,"@"
150 CLOSE
160 FIND 1
170 ON EOF (0) THEN 210
180 INPUT @33:A$
190 PRINT @1:A$
200 GO TO 180
210 END
```

```
USMAP/SEG67
USMAP/SEG76
USMAP/SEG85
USMAP/SEG94
USMAP/SEG5
USMAP/SEG14
USMAP/SEG23
USMAP/SEG32
USMAP/SEG39
USMAP/SEG41
USMAP/SEG48
```

The 4907 directory can be output directly to the Option 10 Interface (RS-232-C) and thus to the printer by simply typing:

```
DIR @51:0,"@"
```

The CALL "FILE" command may also be used. A full file status will be generated for each file in the library designated and then stored in a string variable. The variable may then be output over the GPIB to the plotter.

```
100 INIT
110 DELETE A$
120 DIM A$(300)
130 CALL "FILE",0,"@USMAP/SEG48",A$
140 PRINT @1:A$
150 END

RUN
```

```
              0 USED        29-NOV-79 17:18 USED
       0 OPEN 0 REC LEN     29-NOV-79 17:18 CREATED
USMAP/SEG48
```

The limitation, however, is plotter paper area. If you ask for a total file directory dump, e.g., CALL "FILE",0,"@",A$, one sheet of plotter paper may not be enough to hold it all and you can't change paper without losing some of the string.

Fitting Data

Dan Taylor, Design Engineer at **Tektronix** and **TEKniques' Technical Editor** writes: Twice recently a question has come to me about data fitting. The customers had a set of (X_i,Y_i) observations and a model Y = F(X,C) and wanted to find the "best" C to fit the data (where C is defined as equal to one or more "constants" which are chosen to fit the data).

There are several different programs in PLOT 50 Statistics Vol. 4 that will do this (find C). Marquardt's method was used by both of these customers. However, what they really wanted was Y = F(X,C) with the additional constraint that C is greater than or equal to 0.

This is quite possible and easily done; just code the model in BASIC as Y=F(X,ABS (C)). For example, the model Y=aE^{bx} would be coded as:

```
1100 Y=ABS(C(1))*EXP(ABS(C(2))*X(I))
```

instead of

```
1100 Y=C(1)*EXP(C(2)*X(I))
```

This will work for any sort of program which fits (X,Y) data through a user coded Y = F(X,C) function.

Howard Sanders has compiled three of the most frequently asked questions of the **Technical Support Specialists**. Their answers may save you a telephone call to your Systems Analyst.

ON SRQ and POLLing

When using a 4050 with an LSI (Digital Equipment), the second SRQ doesn't get trapped coming from the LSI although a POLL has been executed.

The ON SRQ THEN . . . statement traps the first SRQ, but to arm the 4050 System to respond to another SRQ, a RETURN must be incorporated somewhere after your POLL routine to complete the SRQ loop. When the RETURN is executed, the ON SRQ THEN . . . will respond to the second interrupt, and so on. For a thorough discussion on the handling of interrupts from peripheral devices on the GPIB, see "The POLL Statement" Section of your 4050 Series Graphic Systems Reference Manual.

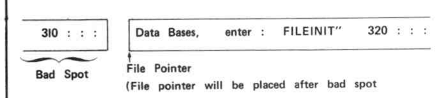
```
110 ON SRQ THEN 1970
120 REM Begin main program
!
1970 POLL M,W;5;8;3
1980 GO TO M OF 2000,3000,4000
!
2000 REM Service Routine for Device Number 5
!
2990 RETURN
!
3000 REM Service Routine for Device Number 8
!
3990 RETURN
!
4000 REM Service Routine for Device Number 3
!
4990 RETURN
!
```

Tape Error When OLDing

I saved a program on tape in ASCII format. When I attempted to OLD the program back, a tape error occurred in the middle of the file. How can the rest of the program be retrived?

OLD in your file. When the error message is received, key in INPUT A$. You will probably get the last part of a statement. So do another INPUT A$ to get the next line number that's retrievable. Now key in that statement. Then type in the next line number followed by REM; APPend the rest of the program to that line. This procedure can be repeated if more errors occur.

```
                             FIN1
                             OLD
Error Message                MAG TAPE READ ERROR - MESSAGE NUMBER 53
indicating bad spot

              Key in         INPUT @33:A$
       See what's in         A$
       A$                    Data Bases,  enter:  FILEINIT"

     Do it again to          INPUT @33:A$
     get line number         A$
                             320 PRINT "  for Mapping Demonstration,
                                          enter:  SEGUSMAP"

Key in statement             320 PRINT "  for Mapping Demonstration,
                                          enter:  SEGUSMAP"
Key in next line #           330 REM
Append rest of               APPEND 330
program
```




```
LIS
100 INIT
110 PAGE
115 C$="4-JUL-76"
116 D$="11:11:11"
120 PRINT "                        4907 AUTOLOAD  "
130 PRINT "                        ------------JJ"
140 PRINT "ENTER TODAY'S DATA (format: DD-Mon-YY, e.g.: 4-JUL-76): ";
150 INPUT A$
155 IF A$<>"" THEN 160
156 A$=C$
160 PRINT "ENTER CURRENT TIME (format: HH:MM:SS, e.g.: 18:58:46): ";
170 INPUT B$
175 IF B$<>"" THEN 180
176 B$=D$
180 A$=A$&" "
190 A$=A$&B$
200 CALL "SETTIM",A$
210 CALL "MOUNT",0,""
220 PAGE
320 PRINT "       for Mapping Demonstration,       enter:  SEGUSMAP"
330 PRINT "Ifor Data Sorting,                  enter:  SORTINGJ"
340 INPUT A$
350 CALL "USERLIB","DEMOS"
360 OLD A$
```

Salvaging Accidentally Re-MARked Files

How do I recover files which have become inaccessible because the tape was inadvertently re-MARked?

Two programming tips for recovering such files on a 4051 have been published in TEKniques Vol. 2, No. 3 and Vol. 3, No. 3 (pages 39 and 75 in the new Programming Tip Handbook) and one tip for recovering these files on a 4052/4054 in TEKniques Vol. 3, No. 8 (page 102 in the Programming Tip Handbook).

Editor's Note

New Catalog Arrives

With your last TEKNIQUES issue, you should have found a copy of the new 4050 Series Applications Library catalog. The catalog has added 67 new programs to the library since last year's catalog. Look through it; you'll probably find some you can put to work in your application.

Additional copies are available for the asking, from the Applications Library office serving you. Their addresses are located at the back of each issue. The Applications Library staff thanks all of you who contributed programs to the library, for making these programs available to others.

The Programming Tip Handbook is Here Too!

Programming Tips have proven to be one of our most-read features in TEKniques over the past few years. They've proven to be such a valuable reference for most of our readers, in fact, that we've compiled all of the Programming Tips and BASIC Bits from Volumes 1, 2, and 3, and put them together in a Programming Tip Handbook.

The Programming Tip Handbook is available through the Applications Library office serving you. It's listed in the Resource Materials section of the new catalog; its Abstract Number is 51/00-7004/0. And again, the Applications Library staff would like to thank those of you who contributed your valuable Programming Tips. They've proven very helpful to many users, and made our handbook possible.

Back Issues are Gone

TEKniques has made a point of keeping some back issues available for those of you who may have lost one, or want to pass a copy on to someone else. However, we will no long provide back issues of copies from Volumes 1 through 3 (1977-1979). But the information is still available. Many back issue requests were from those who wanted all of the Programming Tips and Basic Bits; these are all contained in the Programming Tip Handbook. Others want a copy of an article relating to a particular application; these are now available in article reprint collections from the first three volumes of TEKniques. (See the following note.) And, of course, all of the Applications Abstracts are found in the new Applications Library Catalog.

Application Article Reprints

To continue to meet the needs of users who may need to review an application article from one of the previous issues of TEKniques, we've made some reprints available. We collected the application articles from our first three volumes, and put them together by application area. We currently have five sets of article reprints available, in the following areas:

Engineering and Design AX-4449
Mapping AX-4460
Data Acquisition and Analysis .. AX-4450
Business Graphing
and Reporting AX-4451
Peripherals and ROM Packs AX-4452

If you have need of an article from an issue in Vol. 1, 2, or 3, contact your local Tektronix office, or the Applications Library office serving you. They can provide the reprint volume that contains the article you want.

New Contest Coming

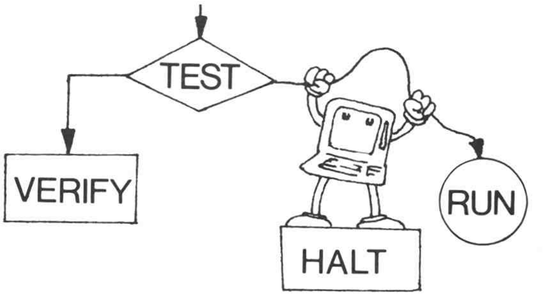
Response to the last Applications Library contest was tremendous. The entries provided several new and useful programs for the library. So we're going to have another contest very soon. Watch the next issues of TEKniques for details.

Wanted: Your Programs

We're always looking for more programs to add to the 4050 Series Applications Library. This helps us keep the library growing to better serve more of you, and provides a medium for you to share your programs with others who may be able to use them too. And you get three free programs for each program you send in to the library, so everybody wins.

Programming Tips Too

The Applications Library staff would also like to remind you that we're looking for those valuable Programming Tips and BASIC Bits. You'll receive any one of 12 programs from the library in exchange for your tip. Details are listed at the end of the BASIC Bits column in each issue.



PROGRAMMING TIPS

Checking Numeric Strings for Valid Numbers

by Jim Gish
Tektronix, Inc.
Irvine, CA

There are times when you may want to input numbers from the keyboard into a character string. One instance might be an indefinite loop to input numbers from the keyboard until the operator responds with "END" instead of number. However, if the program attempts to convert "END" into a number, it will abort the program. To avoid this, use a numeric parser. A parser, in this sense, determines if an input string contains a valid number without actually converting it to a numeric quantity.

If the parser determines the string has a valid number in it, then it's safe for the program to convert the string into a number. If not, the program can ask the operator to try again or check to see if the operator wants to exit.

This particular parser also edits the character string, removing any imbedded blanks or commas. It will accept integer, real or scientific notation.

The string to be parsed is O$. O$ will have a length equal to the value stored in variable I3. *The length of O$ must never be greater than I3.* When the subroutine is finished a numeric constant will be right justified in 0$ with blank fill from X$. If the parser doesn't find any numeric characters K0 will be set to one as a warning to the calling program.

```
1000 REM SUBROUTINE TO VERIFY NUMERIC STRING
1010 REM STRING PASSED IN O$
1020 RESTORE 1030
1030 DATA 0,0,0,0,0,0
1040 READ K0,H4,K1,K2,K3,K4
1050 DELETE J$,V$
1060 H1=LEN(O$)
1070 IF H1=0 THEN 1490
1080 DIM J$(H1),V$(1)
1090 J$=""
1100 V$=""
1110 FOR H2=1 TO H1
1120 Y$=SEG(O$,H2,1)
1130 K4=POS("0123456789.+-E, ",Y$,1)
1140 IF K4<=0 THEN 1490
1150 REM CASE 1 - Y$ = A COMMA OR BLANK  "IGNORE IT"
1160 GO TO K4-10 OF 1440,1400,1400,1330,1230,1230
1170 REM Y$ IS A NUMERIC CHARACTER
1180 K3=1
1190 REM ADD CHARACTER TO OUTPUT STRING (J$) AND INC COUNTER (H4)
1200 H4=H4+1
1210 J$=REP(Y$,LEN(J$)+1,0)
1220 V$=Y$
1230 NEXT H2
1240 REM CHECK J$ FOR ZERO LENGTH OR NO DIGITS
1250 IF K3=0 OR LEN(J$)<=0 THEN 1490
1260 REM PACK NUMERIC STRING RIGHT JUSTIFIED
1270 REM      X$ CONTAINS A STRING OF BLANKS AT LEAST I3 LONG
1280 O$=J$
1290 W$=SEG(X$,1,I3-LEN(O$))
1300 O$=REP(W$,1,0)
1310 RETURN
1320 REM CASE 2 - Y$= "E"
1330 IF LEN(V$)=0 THEN 1490

1340 IF ASC(V$)=43 OR ASC(V$)=45 THEN 1490
1350 K1=1+K1
1360 IF K1>1 THEN 1490
1370 K3=0
1380 GO TO 1190
1390 REM CASE 3 - Y$ = A PLUS OR MINUS SIGN
1400 IF V$="" THEN 1190
1410 IF ASC(V$)<>69 THEN 1490
1420 GO TO 1190
1430 REM CASE 4 - Y$ = A DECIMAL POINT
1440 K2=K2+1
1450 IF K2>1 THEN 1490
1460 IF K1=>1 THEN 1490
1470 GO TO 1190
1480 REM *** ILLEGAL ENTRY OR END OF INPUT
1490 K0=1
1500 RETURN
```

Ed. Note: This routine uses many scratch variables. An alternative method is to concatenate the input string with an absurd number and then test the value of O$, e.g.,

```
100 REM O$&",3.649287141697E304"
110 REM K0=(VAL(O$)<>3.649287141697E+304)
```

Where K0=1 means okay and K0=0 means error. This is similar to the programming tip in Vol 4. No. 1 'Use VAL Function to Avoid Errors.'

APPEND to Recover from Alternate Delimiters

by Dan Taylor
Tektronix, Inc.
Wilsonville, OR

TEKniques Vol. 2 No. 7 and Vol. 3 No. 1 (pages 56 and 63 in the Programming Tip Handbook) contain programming tips to recover ASCII programs SAVED with alternate delimiters, namely CR/LF.

Although you can't OLD in a program containing CR/LF (PRI @37,26:1) as its delimiters, you can APPEND it if your environmental parameters are set to CR/LF. Input the first line to get the line number, type a REM statement for that line number, then APPEND in the program.

```
PRI@37,26:1
FIND 1
INP @33:A$
A$
100 INIT

100 REM
FIND 1
APPEND 100

PRI @37,26:0
LIS
100 INIT
110 PRINT "TEST PROGRAM"
120 END
```

Input Without Echo

by S. Schicktanz
Physics Department
University of Munich
Germany

The following subroutines allow a user to key in a response without echoing the response to the screen. The first routine is used for multiple key entries for numbers or strings. Two examples for calling the subroutine are shown.

```
   100 REM Example call for multiple characters
*  110 GOSUB 1000
   120 REM Input is in T$

   100 REM Example call for numeric variable
   110 GOSUB 1000
*  120 IF T$>"9" OR T$<"0" THEN 110
   130 X = VAL(T$)
   140 REM Input is in X, transferred by line 130
   150 REM Line 120 checks for valid input


1000 REM Subroutine to input multiple cha or numbers
1010 GIN X9,Y9
1020 T$=""
1030 POINTER Z,Z,Z$
1040 IF Z$="" THEN 1070
1050 T$=T$&Z$
1060 GO TO 1030
1070 MOVE X9,Y9
1080 RETURN
```

* T$ cannot hold more than 72 characters unless redimensioned, and it must contain at least one character upon return.

The second routine is used for single key response. The examples which call it execute a GOTO or GOSUB branch on the response.

```
100 REM GO TO BRANCH ON RESPONSE
110 DIM Z$(1)
120 GOSUB 1000
130 GO TO POS("DAB?-+",Z$,1) OF 400,500,600, . . .
140 REM Program continues here if no match found


100 REM GOSUB BRANCH ON RESPONSE
110 DIM Z$(1)
120 GOSUB 1000
130 GOSUB POS("TESTSTRING",Z$,1) OF 400,500,600, .
140 REM Program continues here.


1000 GIN X9,Y9
1010 PRINT "G";
1020 POINTER Z,Z,Z$
1030 MOVE X9,Y9
1040 RETURN
```

Note: Pressing the RETURN key initiates the branch to the first line number following OF--in this case, line 400.

Editor's Note: Mr. Schicktanz has touched on areas of several previous Programming Tips--no echo, branching, checking for valid input and GOSUB/GOTO. For a review, see TEKniques Vol. 1, Nos. 7, 8, 10; Vol.2, Nos. 3, 6; (pages 16, 18, 24, 43, 92 in the Programming Tip Handbook); and Vol. 4, Nos. 1, 3; this issue also contains a tip which checks for valid numeric input.

Determination of a Graph's Upper Bound

by Leonard Weitman
Tektronix, Inc.
Beaverton, Oregon

While programming a 4050 series Graphics System to display data via line and bar graphs, a consistent problem arises: What will be the largest label on the vertical axis (the top of the graph)? Often, data from one graph to the next will vary by several orders of magnitude. For this reason, a general utility routine for determining the maximum graph value must be able to handle a large variety of magnitudes of data. Not only should the maximum value on the vertical axis suit the data's order of magnitude, but it should also be divisible by some "nice" number so that the graph will have a pleasing appearance. The following subroutine calculates the maximum graph value and the number of major horizontal tic marks that will appear on the vertical axis. The routine requires that:

1) The largest data value has been determined ("L").
2) The lower bound of the graph is zero.

```
1000 REMARK ** FIND UPPER BOUND & NUMBER OF TIC MARKS FOR GRAPH **
1010 REM
1020 REM ------  L=Largest data value to be graphed.
1030 REM ------ Q4=Logarithm (base 10) of the current maximum value on
1040 REM            the vertical scale.
1050 REM ------ Q5=Current maximum value on the vertical scale.
1060 REM ------  T=Smallest vertical scale value, greater than L, whose
1070 REM            logarithm base 10 is a multiple of .3010299 .
1080 REM ------  D=The number of Vertical Scale Tic Marks.
1090 REM ------  F=A counter.
1100 REM
1110 REM
1120 Q4=INT(LGT(L)+1)
1130 FOR F=1 TO 3
1140 IF L>INT(10↑(Q4-0.3010299)*10)/10 THEN 1170
1150 Q4=Q4-0.3010299
1160 NEXT F
1170 Q5=INT(10↑Q4*100)/100
1180 T=Q5
1190 FOR F=5 TO 3 STEP -1
1200 IF L>Q5-T/5 THEN 1230
1210 Q5=Q5-T/5
1220 NEXT F
1230 D=F
1240 RETURN
```

The number of major horizontal tic marks on the vertical axis will always be 3, 4 or 5. The increment of each of these tic marks will be evenly divisible by some pleasing quantity, such as multiples of 5 or 10. Also, the maximum data point will appear in at least the upper third of the Graphics System display window. Although this routine assumes a lower bound of zero, similar logic can be used to determine both a variable upper AND lower bound.

Multiple Dimensioned Arrays

by Nathan Oxhandler
Tektronix, Inc.
Santa Clara, CA

and Dan Taylor
Tektronix, Inc.
Wilsonville, OR

There are times you need to work with data arrays of three or more dimensions. While the 4050 Desktop Computers provide for two-dimensional arrays, it's very easy to develop routines that allow you to simulate arrays of almost unlimited dimensions.

Memory

One of the first things to look at is the amount of memory numeric data takes. Each single data item takes up eight bytes of memory. A single-dimensioned array takes eight times the number of elements in the array plus 18. Example: DIM A(20) requires 8 X 20 + 18 bytes. A two-dimensional array takes eight times the number of elements in the first dimension, times the number of elements in the second dimension, plus 18 bytes. Example: DIM A(20,30) requires 8 X 20 X 30 + 18 = 4,818 bytes. As you can see, no matter what type of array you're working with, you'll need to be conscious of the amount of memory it requires.

Three-Dimensional Data Array in Two Dimensions

The following routine simulates a three-dimensional data array using a two-dimensional array. The application is simple, but demonstrates the subscript handling to achieve the simulation.

```
100 INIT
110 REM 3D ARRAY IN 2D
120 N1=4
130 N2=3
140 N3=2
150 DIM A(N1,N2*N3)
160 A=0
170 R=RND(-0.5)
180 FOR I=1 TO N1
190 FOR J=1 TO N2
200 FOR K=1 TO N3
210 A(I,(J-1)*N3+K)=INT((RND(1)+0.005)*100)
220 NEXT K
230 NEXT J
240 NEXT I
250 PRINT A
260 REM TO ACCESS ALL N3 ELEMENTS IN THE N2 DIMENSION IN N1(3)
270 FOR I=3 TO 3
280 FOR J=1 TO N2
290 FOR K=1 TO N3
300 PRINT A(I,(J-1)*N3+K)
310 NEXT K
320 NEXT J
330 NEXT I
340 END
```

Notice that to simulate DIM A (N1,N2,N3) and access it in the manner A(I,J,K) . . ., you need merely

```
DIM A(N1,N2*N3)
```

(statement 150)

and access it as

```
A(I,(J-1)*N3+K) = . . .
```

(statements 180-240)

Multi-Dimensioned Array in Two Dimensions

Based on the foregoing, let's extend the number of dimensions.

```
100 INIT
110 REM MULTI-DIM ARRAY IN 2D
120 REM X=NUMBER OF DIMENSIONS
130 X=4
140 REM N CONTAINS THE LENGTH OF THE DIMENSIONS
150 DIM N(X)
160 DATA 4,3,2,5
170 READ N
180 REM D IS LENGTH OF 2ND DIM
190 D=1
200 FOR I=2 TO X
210 D=N(I)*D
220 NEXT I
230 DIM A(N(1),D)
240 A=0
250 R=RND(-0.5)
260 FOR I=1 TO N(1)
270 FOR J=1 TO N(2)
280 FOR K=1 TO N(3)
290 FOR L=1 TO N(4)
300 A(I,((J-1)*N(3)+(K-1))*N(4)+L)=INT((RND(1)+0.005)*100)
310 NEXT L
320 NEXT K
330 NEXT J
340 NEXT I
350 PRINT A
360 REM TO ACCESS ALL N3 ELEMENTS IN THE N2 DIMENSION IN N1(3)
370 FOR I=3 TO 3
380 FOR J=1 TO N(2)
390 FOR K=1 TO N(3)
400 FOR L=1 TO N(4)
410 PRINT A(I,((J-1)*N(3)+(K-1))*N(4)+L)
420 NEXT L
430 NEXT K
440 NEXT J
450 NEXT I
460 END
```

To DIM A(N1,N2,N3,N4) and access it A(I,J,K,L) = . . ., you simply

```
DIM A(N1,N2*N3*N4)
```

(statements 180 — 230)

and access it by extending the subscript

```
A(I,((J-1)*N3+(K-1))*N4+L) = . . .
```

(statements 260 — 340)

For five dimensions:

```
DIM A(N1,N2*N3*N4*N5)
```

and access it:

```
A(I,(((J-1)*N3+(K-1))*N4+(L-1))*N5+M)
```

and so on.

Multi-Dimensional Array in One Dimension

A three-dimensional array simulated in one dimension is illustrated in the following routine:

```
100 INIT
110 REM 3D ARRAY IN 1D
120 N1=4
130 N2=3
140 N3=2
150 DIM A(N1*N2*N3)
160 A=0
170 R=RND(-0.5)
180 FOR I=1 TO N1
190 FOR J=1 TO N2
200 FOR K=1 TO N3
210 A(((I-1)*N2+(J-1))*N3+K)=INT((RND(1)+0.005)*100)
220 NEXT K
230 NEXT J
240 NEXT I
250 PRINT A
260 REM TO ACCESS ALL N3 ELEMENTS IN THE N2 DIM IN N1(3)
270 FOR I=3 TO 3
280 FOR J=1 TO N2
290 FOR K=1 TO N3
300 PRINT A(((I-1)*N2+(J-1))*N3+K)
310 NEXT K
320 NEXT J
330 NEXT I
340 END
```

Notice the subscript in statements 210 and 300 which allows you to access your arrays in the normal I, J, K, loop indexing.

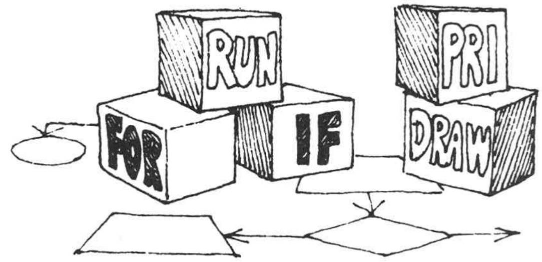
The subscript could easily be incorporated as a record number in a disk file.

As the dimensions increase, it's a simple matter to expand the array subscript to reflect them:

```
100 INIT
110 REM MULTI-DIM ARRAY IN ONE DIMENSION
120 REM X=NUMBER OF DIMENSIONS
130 X=4
140 REM N CONTAINS THE LENGTH OF THE DIMENSIONS
150 DIM N(X)
160 DATA 4,3,2,5
170 READ N
180 REM D IS LENGTH OF ARRAY
190 D=1
200 FOR I=1 TO X
210 D=N(I)*D
220 NEXT I
230 DIM A(D)
240 A=0
250 R=RND(-0.5)
260 FOR I=1 TO N(1)
270 FOR J=1 TO N(2)
280 FOR K=1 TO N(3)
290 FOR L=1 TO N(4)
300 A((((I-1)*N(2)+(J-1))*N(3)+K-1)*N(4)+L)=INT((RND(1)+0.005)*100)
310 NEXT L
320 NEXT K
330 NEXT J
340 NEXT I
350 PRINT A
360 REM FIND ALL ELEMENTS OF N1(3)
370 FOR I=3 TO 3
380 FOR J=1 TO N(2)
390 FOR K=1 TO N(3)
400 FOR L=1 TO N(4)
410 PRINT A((((I-1)*N(2)+(J-1))*N(3)+K-1)*N(4)+L)
420 NEXT L
430 NEXT K
440 NEXT J
450 NEXT I
460 END
```

Of course, the amount of data you may have in your 4050 computer is limited by the size of your memory. However, it is possible to simulate very large arrays of many dimensions using the 4050 Series and the 4907 File Manager. This will be covered in a subsequent issue of TEKniques.

BASIC BITS



String Concatenation

by Stan Jensen
State of California
Air Resources Board
Sacramento, CA

The following program and sample run demonstrate an alogrithm for string concatenation which is a major improvement over the Programming Tip on the same subject in TEKniques, Vol. 1 No. 1.

Editor's Note: This algorithm is intended for building a string of the same character. The length of C$ must be equal to 1.

```
100 REM
110 INIT
120 DIM B$(18),C$(1),E$(18)
130 C$="*"
140 L=2↑14+1
150 PRINT "↑JJJJJIILength - ";L
160 PRINT "JJJJJIIFAST METHOD :"
170 CALL "TIME",B$
180 DIM S$(L)
190 S$=""
200 N=INT(LOG(L)/LOG(2))
210 S$=C$
220 IF L=1 THEN 310
230 FOR I=1 TO N
240 S$=REP(S$,LEN(S$)+1,0)
250 NEXT I
260 IF LEN(S$)=L THEN 310
270 DIM T$(L-LEN(S$))
280 T$=SEG(S$,1,L-LEN(S$))
290 S$=REP(T$,LEN(S$)+1,0)
300 DELETE T$
310 CALL "TIME",E$
320 PRINT "IISTART @ ";B$
330 PRINT "IISTOP  @ ";E$
340 PRINT "JJJJJIISLOW METHOD :"
350 CALL "TIME",B$
360 DIM S$(L)
370 S$=""
380 FOR I=1 TO L
390 S$=REP(C$,LEN(S$)+1,0)
400 NEXT I

RUN
```

```
Length - 16385

FAST METHOD :
START @ 20-NOV-79 11:14:23
STOP  @ 20-NOV-79 11:14:25

SLOW METHOD :
START @ 20-NOV-79 11:14:25
STOP @ 20-NOV-79 11:18:34
```

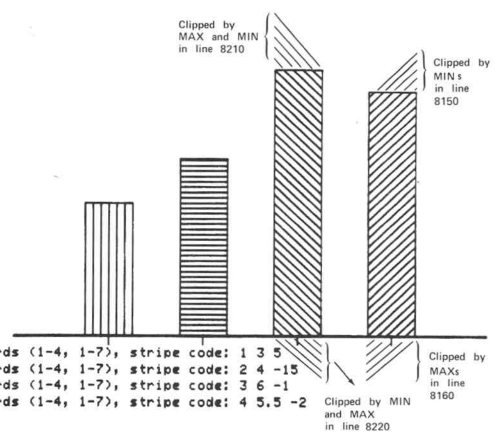
Bar Shading

by John J. Brophy
Pentagon
Washington, D.C.

The following subroutine quickly draws and shades bars. The driver program (statements 100 — 280) would have to be optimized for a user's requirements. In this case up to four bars are provided for varying in height. L, R, B, T are coordinates of the LEFT side, RIGHT side, BOTTOM and TOP, respectively. The stripe code, P, is defined as follows:

P > 0 P vertical lines/bar (not counting sides)
P = 0 No shading
P = —1 Lines slanted to upper left
P = —2 Lines slanted to upper right
P > 0 P horizontal lines/inch

```
8000 REM DRAW PERIMETER OF BAR
8010 PRINT @32,21:L,B
8020 PRINT @32,20:L,T,R,T,R,B,L,B
8030 GO TO P+3 OF 8140,8200,8330
8040 IF P>0 THEN 8260
8050 REM HORIZONTAL SHADING; D1 IS DISTANCE BETWEEN STROKES
8060 D1=18/-P
8070 IF B+D1>T THEN 8330
8080 FOR K=B+D1 TO T STEP D1
8090 PRINT @32,21:L,K
8100 PRINT @32,20:R,K
8110 NEXT K
8120 GO TO 8330
8130 REM SHADING SLANTED TO UPPER RIGHT
8140 FOR K=B TO T+R-L STEP 2
8150 PRINT @32,21:R MIN R-K+T,K MIN T
8160 PRINT @32,20:L MAX R-K+B,K-R+L MAX B
8170 NEXT K
8180 GO TO 8330
8190 REM Shading slanted to upper left
8200 FOR K=B TO T+R-L STEP 2
8210 PRINT @32,21:L MAX L+K-T,K MIN T
8220 PRINT @32,20:R MIN K-B+L,K-R+L MAX B
8230 NEXT K
8240 GO TO 8330
8250 REM Vertical shading
8260 W2=(R-L)/2
8270 REM
8280 D1=2*W2/(P+1)
8290 FOR K=L+D1 TO R-D1 STEP D1
8300 PRINT @32,21:K,B
8310 PRINT @32,20:K,T
8320 NEXT K
8330 RETURN
```




```
Input coords (1-4, 1-7), stripe code: 1 3 5
Input coords (1-4, 1-7), stripe code: 2 4 -15
Input coords (1-4, 1-7), stripe code: 3 6 -1
Input coords (1-4, 1-7), stripe code: 4 5.5 -2
```

```
100 INIT
110 Q=15
120 PAGE
130 PRINT @32,21:20,20
140 PRINT @32,20:120,20,20,19.9,120,19.8,20,19.7,120,19.7,100,20,100,18
150 PRINT @32,20:100,20,80,20,80,18,80,20,60,20,60,18,60,20,40,20,40,18
160 FOR I=1 TO 4
170 PRINT @32,21:0,Q
180 PRINT "Input coords (1-4, 1-7), stripe code: ";
190 INPUT X,Y,P
200 L=20*X+15
210 R=L+10
220 B=20
230 T=B+10*Y
240 GOSUB 8000
250 Q=Q-3.8
260 NEXT I
270 MOVE 0,100
280 END
```

Programming Tip Exchange

Send in your programming tip. Any one of the following 4050 Series Applications Library programs will be yours when it's published. Simply jot down a brief description of the function, the code, and your choice of program. Mail it to the 4050 Series Applications Library serving you; Library addresses are listed at the back of each TEKniques issue.*

51/00-0501/0	51/00-6002/0
51/00-0901/0	51/00-8004/0
51/00-1403/0	51/00-8017/0
51/00-1603/0	51/00-8022/0
51/00-4002/0	51/00-9507/0
51/00-5204/0	51/00-9533/0

***Documentation and listing only.**

4050 Series Applications Library Program Abstracts

Order

Documentation and program listings of each program are available for a nominal charge. Programs will be put on tape or disc for a small recording fee per program plus the charge for the tape cartridge or flexible disc. One tape/disc will hold several programs. Programs will be recorded on like media only, i.e., programs on tape cannot be sent on disc and vice versa unless so noted in the abstract.

(The program material contained herein is supplied without warranty or representation of any kind. Tektronix, Inc. assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from the use of this program material or any part thereof.)

Domestic U.S. Prices:

Documentation and listings	*$20 per program*
Recording Fee	*5 per program*
Tape Cartridge	*30 per tape*
Flexible Disc	*15 per disc*

Contribute

Contribute one program to the Library and receive three in exchange. Send in the membership card from your 4050 Series Graphic System Reference Manual to get the details. Or call us (503) 682-3411, ext. 3618.

Forms

Please use the Applications Library Order Form. Order forms are included in the Membership Packet and are available from your local Tektronix Sales Engineer.

Outside U.S.

Program contributions or orders outside the U.S. must be processed through the local Tektronix sales office or sent to one of the Libraries serving your area. See Library Addresses section.

ABSTRACT #: 51/00-1407/0

Title: **Thevenin's Theorem**
Author: John G. Lingle
Tektronix, Inc.
Beaverton, OR
Memory Requirement: 16K
Statements: 378
Files: 1 ASCII Program

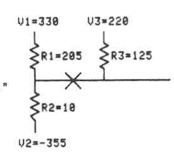
The program provides computer aided instruction on how to THEVENIZE. The user inputs the number of voltage sources; the program randomly selects values for the voltages and resistors. The following may be selected to solve the problem:

1. The program will progress through the solution, detailing each step.

2. The user may solve the problem. The user has the option of using the 4050 as a calculator to assist in computing the answer. Two attempts are allowed to input the correct answer. If the correct answer is not input the user may select to view the solution using the above process.

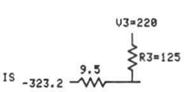
```
WOULD YOU LIKE TO SEE A THEVENIN PROBLEM SOLVED (Y OR N)?Y

ENTER NUMBER OF SUPPLIES  3

THEVENIN'S is a method of solving circuits with
TWO or more power supplies.

Using THEVENIN'S requires that only TWO supplies
be considered at a time.

Each complete step reduces the number of supplies
and resistive branches by one.

Eventually the circuit will be reduced to one
equivalent supply and resistor.
PUSH RETURN:
```

```
ASSUME THE CIRCUIT IS OPEN WHERE
THE "X" IS AND ONLY CONSIDER THE
PORTION TO THE LEFT OF THE "X".

TREAT THE CIRCUIT AS A SERIES BRANCH
AND CALCULATE THE VOLTAGE AT THE
POINT WHERE THE CIRCUIT WAS OPENED.

PUSH RETURN:
```




```
THE VOLTAGE IS -323.2 AND IS CALLED "Voc"
(open circuit VOLTAGE).

NOW COMPUTE THE PARALLEL RESISTANCE
OF THE TWO RESISTOR BRANCHES.

PUSH RETURN:

THE EQUIVALENT RESISTOR IS 9.5 AND IS CALLED Rth (Resistance Thevenin).

TO COMPLETE THIS STEP REPLACE THE TWO RESISTOR BRANCHES
WITH THE PARALLEL EQUIVALENT AND USE THE CALCULATED Voc
FOR THE VOLTAGE SUPPLY.

PUSH RETURN:
```

```
THERE IS JUST ONE RESISTOR BRANCH
AND POWER SUPPLY LEFT THAT HAS NOT
BEEN INCLUDED YET.

CALCULATE THE VOLTAGE BETWEEN THE
LAST TWO RESISTORS.

PUSH RETURN:
```




```
THE VOLTAGE -284.7 AT THIS JUNCTION IS
THE ACTUAL VOLTAGE.

WHAT IS THE PARALLEL EQUIVALENT OF
THESE TWO RESISTORS.

PUSH RETURN:

THIS IS THE FINAL Rth AND IS 8.8

NOW THAT YOU KNOW HOW TO SOLVE THIS PROBLEM,
HOW WOULD YOU LIKE TO TRY ANOTHER ONE?
```

ABSTRACT #: 51/00-6122/0

Title: **Data Communications Utility**
Author: Chris Rotvik
Tektronix, Inc.
Irvine, CA
Memory Requirement: 8K
Peripherals: Opt 1 (RS-232)
Statements: 135
Files: 1 ASCII Program

The program provides a means by which to transfer data files (or programs) between 4050 Series Desktop Computers utilizing telephone lines and the optional Data Communications Interface (Opt 1).

Features Include:

1. Input and output files may be ASCII or binary.
2. Interactive program allows operator selection of:
   A. Baud rate
   B. Transmit or receive
   C. Input or output file number
   D. Output file format (ASCII or binary) if file format is not currently defined
3. Software communication protocol.
4. Transmission error detection.

ABSTRACT #: 51/00-6123/0

Title: **CYBER 175/UNIVAC 1108 Terminal Control**
Author: Chris Rotvik
Tektronix, Inc.
Irvine, CA
Memory Requirement: 8K
Peripherals: Opt 1 (RS-232)
R06 Editor ROM Pack
Statements: 63 each
Files: 1 ASCII Program (CYBER 175)
1 ASCII Program (UNIVAC 1108)

The program consists of two separate programs. The first is CYBER 175 Terminal Control, the second is UNIVAC 1108 Terminal Control. Each sets the proper optional Data Communications Interface (Opt 1) parameters and modes to interface a 4050 Series Desktop Computer to the specified host computer.

Features are:

1. Function key driven
2. Provides the following modes of operation:
   A. Terminal mode
   B. Data receive mode (from the system editor)
   C. Data send mode (to the system editor)
   D. Editor mode (utilizing the R06 Editor ROM Pack)
3. Designed around the need to perform editing offline from the host computer.

ABSTRACT #: 51/07-8042/0

Title: **Coin Collection Inventory**
Author: Peter R. Hulick
Lynchburg, VA
Memory Requirement: 32K
Peripherals: 4907 File Manager
Statements: 354
Files: 1 ASCII Program

The program is used to create and maintain an inventory for a coin collection. A file is created for each series of coins, one record contains each coin. The first record of each series contains the name of the series, denomination and number of items. Files may be reviewed, updated, corrected and tabulated.

The following options are available:

1. Review of a series of coins

   Calls up a file of coins, displays the contents, tabulates and prints out the coins' values.

2. Add a new series of coins

   35 series of coins and suggested file identifiers are assigned in the program. These may be modified, added to or deleted.

3. Change an existing series of coins

   Allows the correction of a coin in a series.

4. Update the prices of an existing series

   Allows periodic updates of values of coins in a series.

5. Sum the values of all series currently on disk

   Lists all current series, each series total value, and the sum total of all series' values.

6. Add to a current series

   Allows the user to lengthen a series of coins that has been previously created.

About 3000 coins may be stored on a disk.

```
PROGRAM TO INVENTORY COIN COLLECTION

SELECT ONE OF THE FOLLOWING AND HIT RETURN KEY

    1  REVIEW OF A SERIES OF COINS
    2  ADD A NEW SERIES OF COINS
    3  CHANGE AN EXISTING SERIES OF COINS
    4  UPDATE THE PRICES OF AN EXISTING SERIES
    5  SUM THE VALUES OF ALL SERIES CURRENTLY ON DISK
    6  ADD TO A CURRENT SERIES

  YOUR CHOICE: _
```

```
### SERIES NAME                       FILE IDENTIFIER

  1 TEST SERIES OF DIMES              TEST10
  2 MISC HALF CENTS                   MISHAFCENT
  3 MISC LARGE CENTS                  MISLRGCENT
  4 MISC 3 CENT PIECES                MISC3CENT
  5 MISC 5 CENT PIECES                MISC5CENT
  6 MISC HALF DIMES                   MISHAFDIME
  7 MISC DIMES                        MISDIMES
  8 MISC 20 CENT PIECES               MIS20CENT
  9 MISC 25 CENT PIECES               MIS25CENT
 10 MISC 50 CENT PIECES               MIS50CENT
 11 MISC SILVER DOLLARS               MISSILVDOL
 12 US $1 GOLD                        USGOLD1
 13 US $2-1/2 GOLD                    USGOLD2HAF
 14 US $3 GOLD                        USGOLD3
 15 US $5 GOLD                        USGOLD5
 16 US $10 GOLD                       USGOLD10
 17 US $20 GOLD                       USGOLD20
 18 MEXICO 50 PESO GOLD               MEXGOLD50P
 19 MEXICO 2 PESO GOLD                MEXGOLD2P
 20 CANADA SOVEREIGN GOLD             CANGOLDSOV
 21 MISC SMALL CENTS                  MISCENTS
 22 BUFFALO NICKELS                   BUFF5
 23 JEFFERSON NICKELS                 JEFF5
 24 BARBER DIMES                      BARBER10
 25 MERCURY DIMES                     MERCURY10
 26 ROOSEVELT DIMES                   ROOS10
 27 BARBER QUARTERS                   BARBER25
 28 STANDING LIB. QUARTERS            STGLIB25
 29 WASHINGTON QUARTERS               WASH25
 30 BARBER HALVES                     BARBER50
 31 WALKING LIBERTY HALVES            WALKING50
 32 FRANKLIN HALVES                   FRANKLIN50
 33 KENNEDY HALVES                    KENNEDY50
 34 MORGAN SILVER DOLLARS             MORGAN100
 35 PEACE DOLLARS                     PEACE100


ENTER FILE IDENTIFIER DESIRED
```

ABSTRACT #: 51/00-9534/0

Title: **Drafting Digitizer**
Author: Tom Sutherlin
Cameron University
Lawton, OK
Memory Requirement: 16K
Peripherals: 4662 Plotter
Optional — 4956 Tablet
Statements: 406
Files: 2 ASCII Program

The program consists of two parts; the menu and the digitizer.

The menu is designed to be plotted on the 4662 Plotter. This is then placed on the 4956 Tablet to be used by the digitizer. The menu allows a blank area for sketching and digitizing with the basic symbols shown.

The digitizer allows the user to transform sketches into a finished drawing using the basic symbols selected from the menu. Input is from the 4956 Tablet, but may be easily modified to allow input from the 4662 Plotter. The User-Definable Keys are used to select the screen or the plotter for output.

200 X and Y coordinates may be input. For machines with more than 16K memory the program may be modified to accept additional X,Y coordinates.

Menu selections include:

Line — input points A and B, a line will be drawn between the two points

Hidden Line — same as Line, but draws a dashed line

Center Line — input points A and B, a dash .15 inches long will be drawn at the center distance of the line

Circle — input point A (center) and B (right of center), used as radius to compute and draw circle

Hidden Circle — same as Circle, but draws dashed circle

Partial Arc — input points A, B, and C, calculates distance between A and B for radius, length is angle calculated between AB and BC

Partial Hidden Arc — same as Partial Arc, but draws a dashed line arc

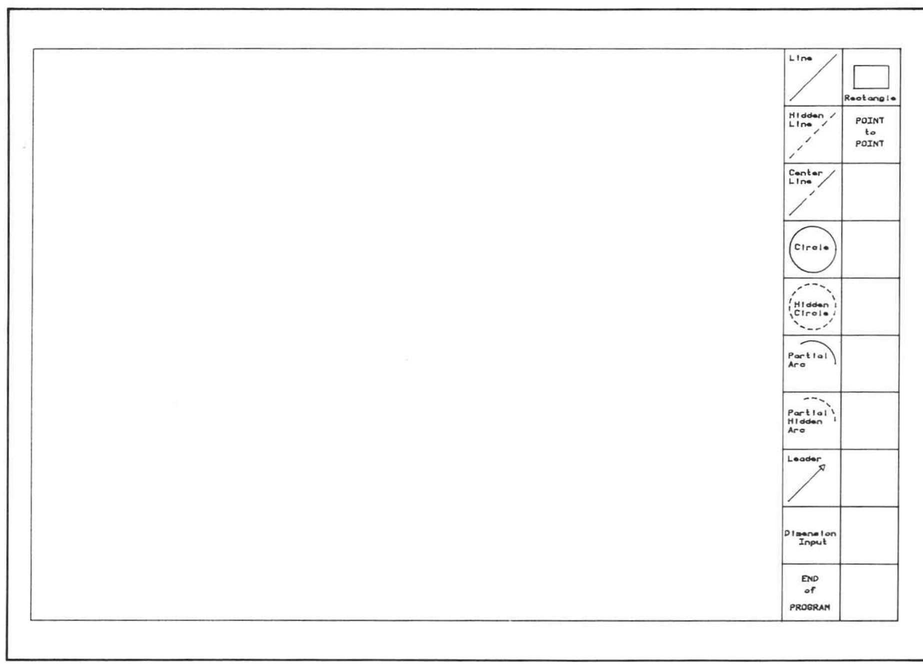
Leader — input points A and B, draws arrowhead at point A .07 inches wide and .1 inches long with line to B

Dimension Input — input points A and B (to the right), enter dimension from the keyboard (up to 16 alphanumeric characters) data is output in a horizontal position

End of Program — flashed "PROGRAM TERMINATED" on the screen and ends digitizing input

Rectangle — input points A (lower left corner) and B (upper right corner), calculates the length of the horizontal and vertical sides

Point-To-Point — input points A, B, C, D . . . , will draw straight line segments to connect the points, points may be at any position



4050 Series Applications Libraries

Africa, Europe, Middle East

Contact local sales office

Australia

4050 Series Applications Library
Tektronix Australia Pty. Limited
Sydney
80 Waterloo Road
North Ryde, N.S.W. 2113

Canada

4050 Series Applications Library
Tektronix Canada Ltd.
P.O. Box 6500
Barrie, Ontario
Canada L4M 4V3

Caribbean, Latin America and Far East (excl. Japan)

IDD Group
Export Marketing
Tektronix, Inc.
P.O. Box 500
Beaverton, Oregon 97077
U.S.A.

Japan

4050 Series Applications Library
Sony/Tektronix Corporation
9-31 Kitashinagawa-5
Tokyo 141 Japan

United States

4050 Series Applications Library
Tektronix, Inc.
Group 451
P.O. Box 500
Beaverton, Oregon 97077

Tektronix®
COMMITTED TO EXCELLENCE

TEKTRONIX, INC.
Information Display Division
Applications Library
Group 451
P.O. Box 500
Beaverton, Oregon 97077

ADDRESS CORRECTION REQUESTED