Tekniques

The 4050 Series Applications Library Newsletter   August 1, 1979   Vol. 3 No. 5

Powerful new Dynamic Graphics option brings refreshed graphic objects and refreshed alphanumerics to the Tektronix 4054.

Refreshed Graphic Objects On the 4054

by Miki Tokola

Imagine the graphics power if you could move objects around the screen of the 4054. Not just propel an object across the screen by overlaying many copies of it, but actually move an object from one side of the screen to the other without leaving a trace of its path. Imagine the fast user interaction if prompts and menus appeared and disappeared at appropriate times any place on the graphic screen. Both of these exciting capabilities and many more are now possible with the newly announced 4054 Option 30 Dynamic Graphics.

Dynamic Graphics adds powerful interactivity to the 4054. Unique BASIC keywords build, manipulate and store true graphic objects, and then display them dynamically. While the standard 4054 has some refresh capability—the cross-hair cursor, alphanumeric cursor, and "FULL PAGE" can all be displayed without being stored—Dynamic Graphics dramatically extends this capability.

Dynamic Objects

In designing, you often think in terms of standard graphic components such as a desk in facilities layouts, or a transistor symbol in a circuit diagram. Dynamic Graphics provides the graphic power to work directly with these graphic elements, in addition to points and lines.

In This Issue

BASIC language programs create the object. The object can be any meaningful unit: a simple line or word, or a complex 1,000 vector object or a program menu. A ROPEN statement begins the construction process; subsequent vectors and text strings define the object. The definition continues until a RCLOSE statement is encountered. A RAPPEND command will expand the definition. The specified object is reopened to add additional vectors and/or text. Object definitions can also be deleted or replaced.

The object definitions, along with some display information automatically provided by the 4054, are stored in the Dynamic Graphics memory. This separate memory permits efficient object storage which doesn't subtract from the 4054's read/write memory. It's easy to build a large set of objects.

An object appears on the 4054 screen through the VISIBILITY command. This command causes the object's definition to be repeatedly retraced (refreshed) producing a non-stored image on the screen. The displayed image of the object can just as easily be deleted; the image is no longer refreshed thus disappears leaving no copy on the screen. However, the definition remains in dynamic memory. This is useful for interactive prompts and responses without cluttering the screen with unnecessary text.

A BLINK command alternates an object between visible and invisible modes according to user-specified on and off times. It could be used to draw the operator's attention to a prompt or indicate an object needing some user action.

There is a great speed advantage with Dynamic Graphics. An object stored in the dynamic memory can be displayed on the screen up to 100 times faster than by drawing it directly from a program onto the screen.

In addition to objects appearing and disappearing on the screen without affecting the rest of the display, another major graphic enhancement is motion.

The refreshed image of the object can be moved around the screen either under program control or interactively with the thumbwheels, or by using an optional graphic input device such as a graphic tablet or joystick. The CURSOR command replaces the standard cross-hair cursor with a specified object and places it under direct control of the thumbwheels.

The FIX command copies the image onto the screen in storage mode. You can quickly place multiple images of an object by repeatedly positioning and fixing the object.

Let's look at a section of a BASIC program which uses some of the new Dynamic Graphics commands. The following listing defines a refresh object, moves the object around the screen with the thumbwheels, and stores copies of the object.

```
1000 MOVE 65,50
1010 REM - DEFINE A SQUARE AS OBJECT 1
1020 ROPEN 1
1030 RDRAW 10,0
1040 RDRAW 0,10
1050 RDRAW -10,0
1060 RDRAW 0,-10
1070 RCLOSE
1080 REM - USE THE SQUARE AS THE CURSOR
1090 REM - UNDER THE CONTROL OF THE THUMBWHEELS
1100 CURSOR 1
1110 POINTER X,Y,Z$
1120 REM - IF THE "F" KEY IS PRESSED
1130 REM - THEN STORE A COPY OF THE SQUARE
1140 REM - ON THE SCREEN
1150 IF Z$<>"F" THEN 1110
1160 FIX 1
1170 GO TO 1110
```

Language compatibility, a major feature of the 4050 Series, is maintained. The new commands are ignored by the 4052, and by 4054s without the option. Special provisions to interact with optional graphic input devices, such as the Joystick or Graphics Tablet, have been incorporated.

A Dedicated Microprocessor

By now you can appreciate not only the power of Dynamic Graphics but also the computing power and memory required to provide this capability. Dynamic Graphics is a single circuit board installed in the 4054. A high-speed microprocessor and 32K bytes of dynamic memory are dedicated to the creation and display of refresh objects, completely independent of the 4054's processor. When Dynamic Graphics commands are received, the microprocessor stores the objects created in its own memory. Timing circuits prompt the microprocessor to retrace its display memory, producing non-stored images on the screen.

The number of vectors displayable without flicker depends on the length of the vectors. Up to 1,000 vectors averaging 1/2 cm. in length may be displayed without flicker. The amount of displayable text is related to the number of strokes in each character. At least one full line of text, or several shorter lines, can be displayed in refresh mode.

Applications

The 4054 has the most powerful graphics features in a desktop computer. With the addition of Dynamic Graphics, the potential application areas are endless. One

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 — Ken Cramer
Managing Editor — Patricia Kelley
Editor — Terence Davis
Technical Editor — Dan Taylor
Graphic Design — John Ellis
Circulation — Rory Gugliotta

major category are those applications which build up a display from standard symbols or components. Examples include circuit schematics, facilities layout, architecture, landscape design, and subdivision planning.

Let's examine a facilities layout application more closely and see how the 4054 with Dynamic Graphics provides an efficient solution to the planning problem. To begin, a menu in refresh is displayed in the upper left corner of the screen, explaining the functions of the User Definable Keys. First, the planner draws the exterior walls of the office. By pressing one of the User Definable Keys, the planner brings up the cross-hair cursor. Positioning the cursor with the thumbwheels, the planner defines the corner points of the walls and doors. Once the planner has finished drawing the walls, pressing a button recalls the menu. The menu can be recalled and erased independently with Dynamic Graphics. It is no longer necessary to erase the entire screen to eliminate the menu.

The next stage places desks. Another User Definable Key brings up a menu of desks and other office furniture on the screen. After identifying the appropriate object, the planner moves the desk around the screen with the thumbwheels. The desk can be tried in different places in the office; upon deciding on the correct location, an image of the desk is stored on the office display with a keystroke. Numerous images of the desk can be stored around the office by positioning the object and storing copies (Figure 1).

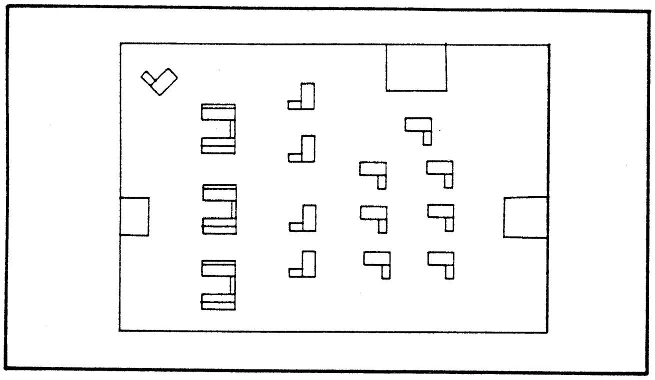


Fig. 1. Facilities layout is quick with the 4054 Dynamic Graphics option. User-defined objects, such as the desks and chairs in this plot, can be selected from a refreshed menu, then positioned with the thumbwheels.

Similarly, wall partitions, plants, file cabinets, and conference tables are quickly positioned and stored. As the office is being assembled, the BASIC program stores the coordinate location and type of each object as it is stored on the screen for editing and output. Once the plan is complete, the planner can have a 4662 or 4663 Digital Plotter plot the office plan (Figure 2). The procedure is quick, easy, and highly interactive.

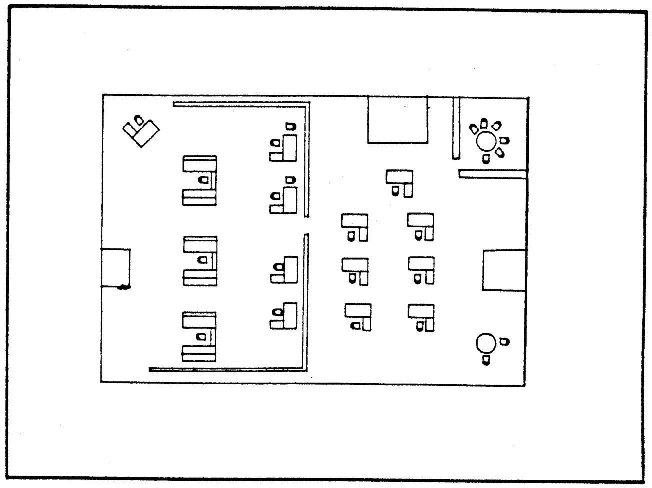


Fig. 2. Any stage of the layout may be sent to the 4662 Plotter from the 4054.

This refresh capability also provides real-time simulations. With the cursor command, it is possible to replace the cross-hair cursor with any refreshed object and place the object under the control of the thumbwheels. Imagine a 4054 with Dynamic Graphics as a tactical decision aid. A BASIC program creates an object resembling a submarine and specifies it as the cursor. The operator can then position the sub anywhere on the screen with the thumbwheels. A second object, a destroyer, displayed in refresh chases the submarine. The BASIC program moves the destroyer based on the location of the submarine. It is possible to simulate evasive maneuvers by allowing the operator to move the submarine around the screen, trying to avoid detection by the destroyer.

These two examples can only hint at the power of 4054 Option 30 Dynamic Graphics. We know that users will apply this increased graphic interactivity in many creative ways.

We've Enclosed Some Questions

When you opened this issue of TEKniques, you probably noticed an enclosed questionnaire. There are five questionnaire areas, all intended to provide information that will make us more responsive to you. We'd like to know your opinion of TEKniques, and of the Applications Library itself. For some further background, we'd like to know a little about you, your equipment, and your computing needs.

Please take a few minutes to fill out the questionnaire, and return it to us in the enclosed self-addressed stamped envelope. This information will be very helpful to us, as we plan for the future of the 4050 Series Applications Library, and TEKniques, your applications library newsletter.

We'd really like to hear from you!

Computer Aided Design Prepares Students for World of Work

by Patricia Kelley

At Ricks College, Rexburg, ID, the Design and Drafting Technology Department housed within this building is using a 4051 to teach their students computer aided design.

Architecture, engineering and other design segments of our economy are relying more and more on the computer to translate their ideas into tangible reality. This has not gone unnoticed by the industrious and farseeing educators at Ricks College, tucked away in the Upper Snake River Valley of Idaho at Rexburg. One of the goals of this two-year private school of 6,000 students is to "Provide experiences which promote early identification of satisfying career paths and efficient preparation for the world of work."

Several years ago, in line with these two objectives and the direction of the design industry, the Design and Drafting Technology department felt it was an opportune time to incorporate a computer into its curriculum. Department professors Blair Pincock and Mel Eckman sensed an upcoming market for graduates with computer experience from reviewing professional literature, and from an American Institute for Design and Drafting convention held about that time which Pincock attended. The convention dealt directly with computer graphics. Interactive computers aimed at design and drafting applications were coming out; industry was just starting to talk about it. Unfortunately, the equipment that was available was far too expensive for educational needs.

Pincock and Eckman researched what other schools were doing and found a dearth of information. Realizing Ricks College could get in on the ground floor of teaching computer graphics in design and drafting, they began exploring possibilities. They were adamant in their belief that the student required interaction with the computer; they deliberately avoided batch processing or host-terminal systems. They wanted a computer that could "teach" the student. The Tektronix 4051 seemed to fit the specifications. Working with a very supportive college administration, and John Hess, Tektronix Sales Engineer from Salt Lake City, Pincock and Eckman installed a 4051, 4956 Tablet, 4662 Plotter and 4051R05 Binary loader in their laboratory (Figures 1 and 2). The results have been resounding; the professors have no trouble getting the students to work with the 4051 to learn useful projects, and the students have no trouble finding jobs upon completion of their schooling.

Fig. 1. One of the initiators of computer aided instruction at Ricks College is Blair Pincock, shown above at the 4956 Tablet in the department's laboratory.

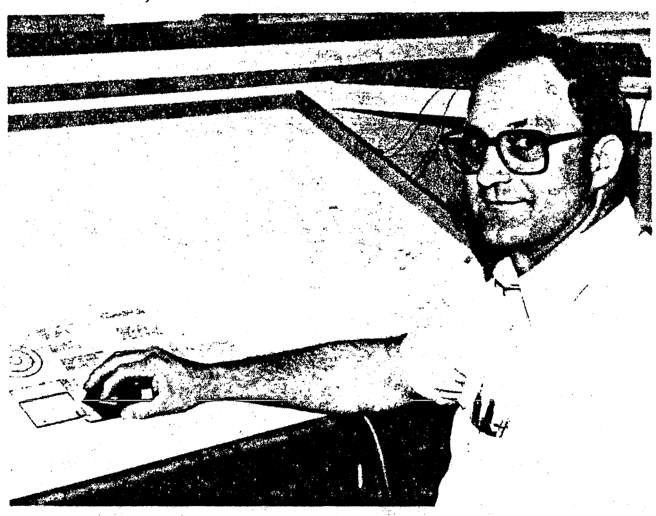


Fig. 2. Mel Eckman, Coordinator of Design and Drafting Technology at Ricks College, was also instrumental in putting together the program of computer aided design.

The Design and Drafting Technology Program

A student in Design and Drafting Technology has a choice of three Associate Degree programs. Design and Drafting Engineering Technology, Architectural Drafting Technology, and Pre-Architecture. The Design and Drafting Engineering Technology program is accredited by the nationally known Engineer's Council for Professional Development. The student completing course work in this area will either transfer to Brigham Young University or go right to a job.

The Architectural Drafting Technology student will enter the architectural drafting job market at the end of the two year program. This program and the previous one both graduate students with computer-aided graphics and design experience; someone who understands the computer sufficiently to operate it and someone with some engineering background. This graduate is capable of going to work in a hurry without a lot of on-the-job training, filling the gap between no experience and the four year graduate.

The student receiving an Associate Degree in Pre-Architecture will transfer to a university to become a licensed architect.

A student must complete a minimum of 64 semester credit hours to receive the Associate Degree. Of these a typical Design and Drafting major will have had five semester hours of computer aided graphics and design (120 actual class hours plus project time), including one BASIC programming class. However, Pincock pointed out the curriculum is gradually being revised to include the 4051 in other courses such as descriptive geometry, electrical drawing, residential drafting and planning, and so on. The result will be students who feel comfortable using the computer in any facet of design and drafting.

All courses are competency based which means a student must demonstrate proficiency in coursework before moving on to the next step. If he cannot meet the standards, he is required to repeat the step. The competency-based classes have established the credibility of Ricks College graduates with the business world. Pincock said the department works closely with local business and industry to maintain real-world standards for the coursework. An industrial committee comprised of four practicing engineers and architects plus the department staff sets and oversees the standards.

The Facilities

The Design and Drafting Technology department is housed in the Mark Austin Industrial Science building on the 255 acre campus, overlooking Rexburg. The college has provided the department with over 5,000 square feet of space for their two drafting laboratories (figure 3), a large blueprinting area, resource library, and the computer aided drafting room.

The library contains a unique sound-on-slide display. Here the student may study his topic of interest at his own pace. Rather than having a complete subject recorded in one continuous lecture on tape, with the resultant searching through the tape for a particular segment of

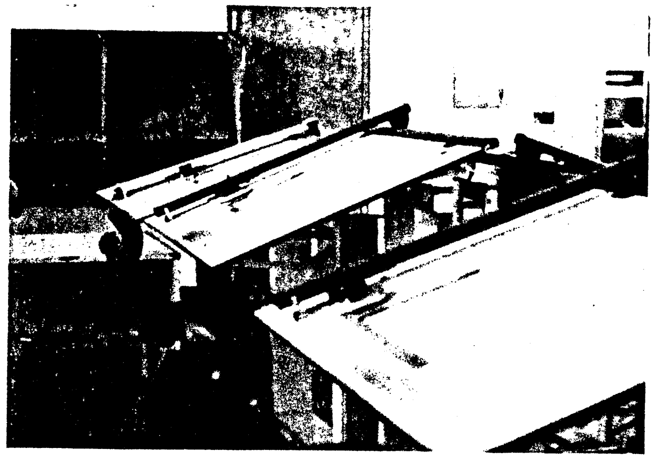


Fig. 3. Two drafting laboratories, a large blueprinting area, a resource library and a computer aided drafting room provide the learning environment for Design and Drafting Technology students at Ricks College.

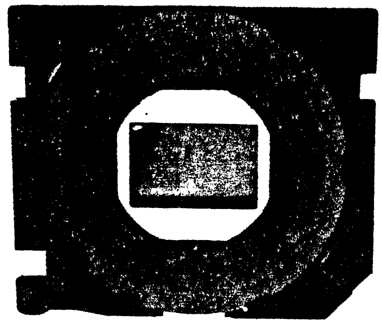


Fig. 4. As a slide is projected on the screen, the lecture pertaining to that slide is played from the sound track contained on the slide.

that subject, slides with a sound track on each are used (figure 4). From the library of over 4,000 slides, the student can quickly select the segment of interest, then view the graphics as he listens to the mini-lecture. Some of the slides have been original products of the department, many have come from Texas A & M University, and a few have been adapted from commercially-available tapes. According to Eckman, the concept has been well received by the students as demonstrated by their constant use of the slides.

A room across from the laboratories holds a Teletype terminal connected to the large college host computer that the students use while learning to program in BASIC. They also have easy access to the wood and metal shops located downstairs.

The Applications

The department abounds with programs generated by enthusiastic students and staff. Pincock noted that he and Eckman had to learn programming by themselves. He credited the ability to interact with the 4051, as well as its friendliness, for smoothing a somewhat arduous process. Although fairly proficient now, he says he continually discovers unsuspected power within the 4051 commands.

Digitizer Utility

A digitizer utility program was one of the first developed by a student, Ken Rock, to aid a user in building design programs. A choice of symbols-arrows, circles, lines, and so on—reside in a menu accessed from the tablet. Through extensive use of the User Definable Keys and the four-button cursor, the user indicates arrow direction, whether a line is dashed, center and radius of a circle, and many other refinements. A labeling routine prints a question mark on the 4051 screen, the user just moves the question mark to the desired location through the tablet cursor. The labeling desired at this location can then be typed in.

Points may be deleted, working backwards (last in, first out). Additionally, the tablet is divided into nine windows, with provision for zooming into a window. Arrays containing the coordinate points of a drawing may be accessed through the cursor; that is, the user simply picks the desired point by digitizing from the tablet and requests the 4051 to display that array on the screen. Therefore, if a loop needs to be completely closed or some other change made where the correct coordinate is known, that coordinate may be inserted directly into the proper array. Once the user is happy with the design on the screen, the program may be saved on magnetic tape or sent to the plotter. Figure 5 displays a plot generated using this program.

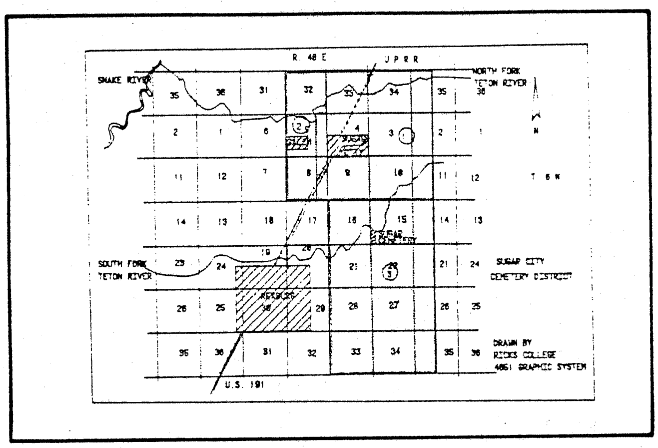


Fig. 5. A map of Sugar City Cemetary District demonstrates the digitizer utility program developed by Ricks College student, Ken Rock, and later refined by Mel Eckman.

Numerical Control

A formidable task for any draftsman is to draw the schematics for etched circuit boards and translate the holes for the wires into coordinates required by the milling machines. The coordinates must be precise so these holes will be accurately punched. Doing this manually takes a lot of time. However, the student now takes a component drawing twice the size of the finished product (figure 6), places it on the tablet, and digitizes the positions for the wire connections. The 4051 draws the drill diagram on the screen and lists the coordinates for each hole. It then reduces it to actual size and stores the coordinates on magnetic tape which could be used to punch out the NC tape. Note the photo negative (center) and the finished product next to the top in figure 6.

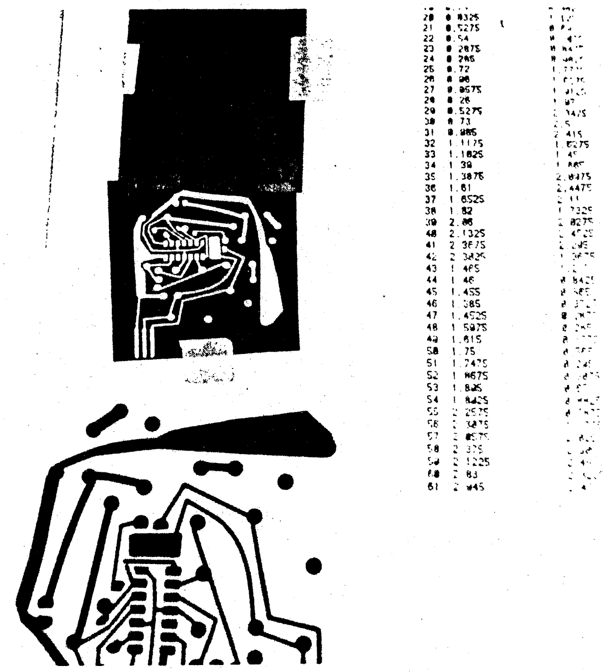


Fig. 6. The wire connections from an enlarged component drawing (bottom) are digitized on the 4956 Tablet. The 4051 draws the drill diagram (top) on the screen, lists the coordinates for each hole (right-hand side), and reduces the drawing to actual size. From the photo negative (center) the finished product (shown next to the top) is produced.

Sheet Metal Patterns

Another real time saver is the pattern layout program for sheet metal development. For instance, when an object is to be constructed from flat pieces of sheet metal, the draftsman has the responsibility for drawing out flat patterns so they can be folded and fit together. Tom Egbert, a student involved in such a project, became exasperated and told himself there must be a better way to do it. Consequently, with only one BASIC language programming course behind him, he sat down at the 4051 and began to develop his pattern program. The result was an easy-to-use tutorial for designing all different shapes and sizes of sheet metal patterns, with attendant time saving. The dust collection unit in Figure 7 is similar to the pattern this program will produce.

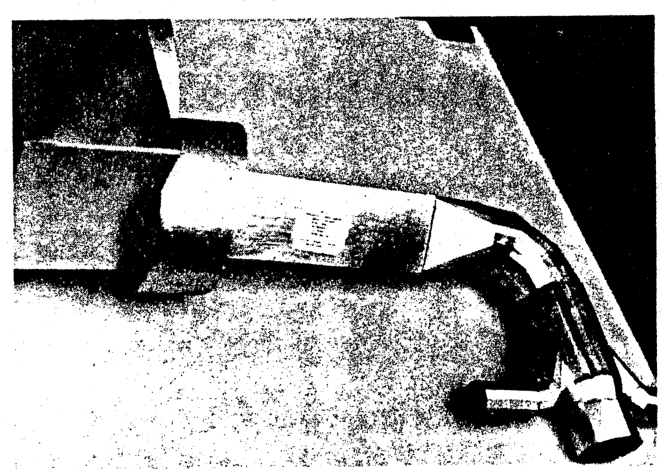


Fig. 7. The designing and dimensioning of flat pieces of sheet metal so they will fold and fit together to produce components similar to this dust collector unit are done on the 4051.

Geometric Dimensioning

Student-developed geometric dimensioning programs save a lot of time as well as space. Eckman described geometric dimensioning as the shorthand of dimensioning, a common system that allows one to give a very accurate and definitive description of a product requirement. For example, if a rod needed to be manufactured round and straight to within .005" from end to end, a lot of written instruction would need to be given to achieve this. The geometric dimensioning system allows a very simple symbol combination to give the same instruction without confusion for those familiar with the symbology. An example describing the requirement for the rod is:

⌭	.005

The programs draw the standard geometric boxes and symbols automatically at the proper size from menu selection (Figure 8).

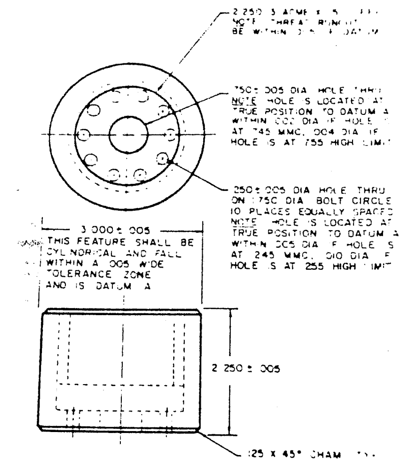


Fig. 8a. Text describes product requirements.

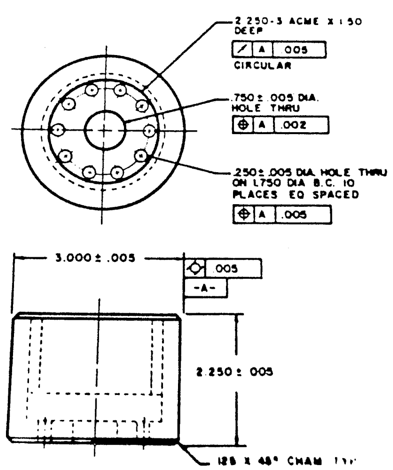


Fig. 8b. Symbols and standard geometric boxes are the shorthand of dimensioning replacing the text shown in Fig. 8a.

Architecture

A student program that gets a lot of use is the architectural planning aid, designed by Lance Murri and Jim Wiebe. The user can sit down at the 4051 and design a typical section for a home using this tutorial program. The user specifies the size of foundation (8" in figure 9), the type of siding, roofing and so on. A cross-section of the building may then be plotted as shown in figure 9. Pincock stressed the time savings for drawings that have to be repeated many times.

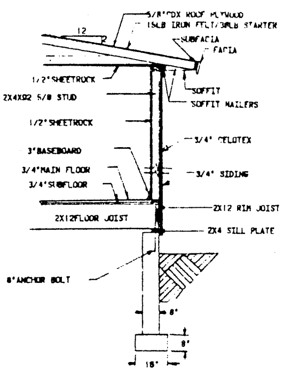


Fig. 9. Cross-section of a home designed on the 4051 using the architectural planning aid program.

A related routine compiles a bill of materials. The user simply responds to the prompts on the 4051 screen for the dimensions of the house. From this the 4051 quickly calculates the quantity of 2 x 4's, sheetrock, shingles, floor joist, and other raw materials required. Pincock compared several materials lists done manually with the 4051 output and noted they came within one-half of a 2 x 4 or one sheet of sheetrock. It's very accurate and fast. He hopes to have a similar program soon to calculate the finish package for the inside of the home. Pincock is also planning a demolition package which would determine the amount of material in an already-constructed building. Through such a program, a user could calculate how many truck loads of material would need to be hauled away and consequently the cost to demolish a building.

Another routine developed for a student project is a structural design program. Sizes of floor joists and beams are calculated from the span length keyed into the 4051.

The Model in Figure 10 was constructed from plans

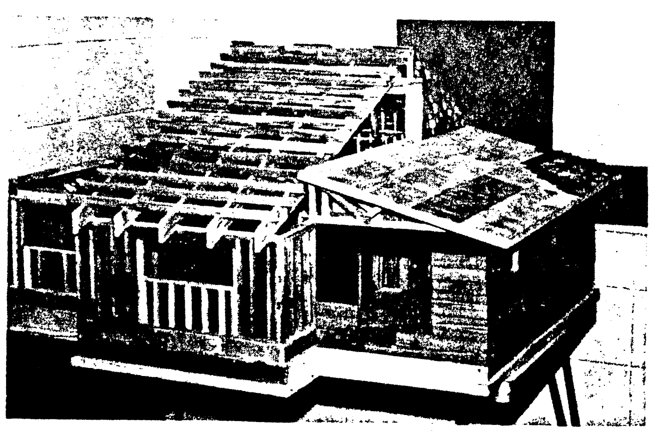


Fig. 10. Students constructed this model from plans similar to those formulated on the 4051 using the architectural planning aid and structural design programs.

similar to those formulated on the 4051 using these programs.

An ambitious undertaking by student Kip Carpenter designs and plots landscapes. A typical plot plan shown in figure 11 exhibits the standard landscape architect symbols. These symbols are called using the digitizer. Various sizes and shapes of patios and swimming pools are available and either of these may be rotated. Note the swimming pool comes complete with diving board and decking. Different sizes of the various types of trees and shrubs may be specified. Stepping stones may be located anywhere through the tablet cursor. When the design is complete, it may be sent to the plotter for drawing on translucent vellum which is later run through the blueprinting machine.

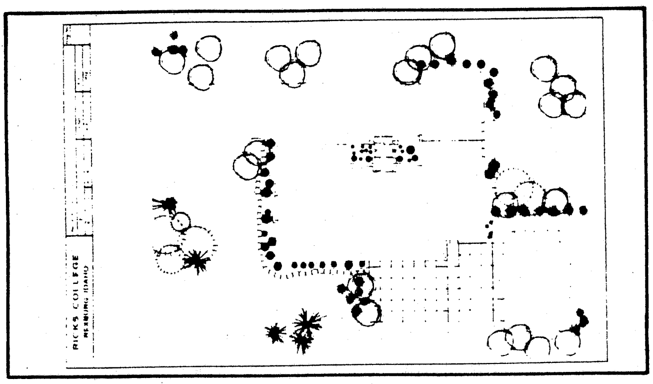


Fig. 11. A blueprint of a landscape plot designed on the 4051, then drawn onto translucent vellum using the 4662 Plotter.

Energy Conservation

Last summer Pincock took part in a three week energy conservation workshop held at Pacific University in Forest Grove, Oregon. As a result he and another attendee, Don Singer, developed a program on the 4051 to analyze energy loss and savings. Armed with a completed worksheet (Figure 12), the user keys in responses to the prompts printed on the 4051 screen. The program calculates the heat loss and the annual fuel cost. Changing a variable such as insulation or window type allows the user to compare the cost of optimum heat conservation with the payback from fuel savings. Every component of a home may thus be looked at in terms of cost/benefit before construction is started. This comparison routine used to take two or three days of full time work; it can now be done in minutes on the 4051.

These are just a sampling of the many programs designed by students and staff in the Design and Drafting Technology department at Ricks College. But to provide the students with continuing useful projects, Pincock and Eckman are hoping to extend their business and industry contacts. While the Rexburg location and atmosphere are conducive to learning, Pincock notes that it is out of the mainstream of industry. The department would like to reach those individuals with real life design problems which could be addressed by the students. Eckman and Pincock comment that the department would be very willing to spend the time developing solutions to such problems and return these to the originators free. It would be a real benefit for the students and should be something that would work well for industry.

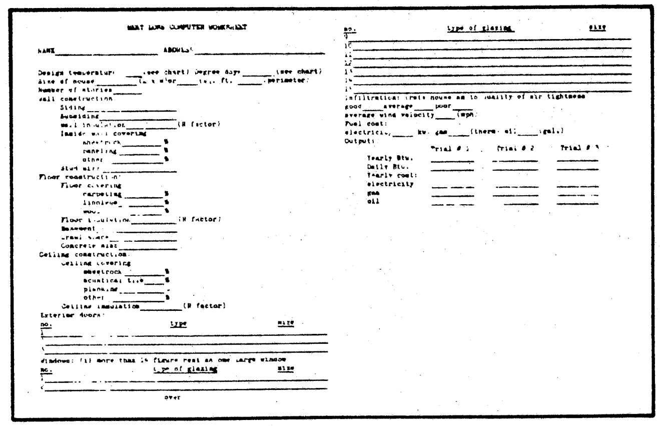


Fig. 12. Once the user has entered the data on the worksheet into the 4051, the 4051 calculates heat loss and annual fuel costs.

The venture into computer aided drafting is paying off for the department and the students. Enrollment in the program is increasing at over 19% each year. Students graduating from the program are finding successful careers. Two examples are Ken Rock and Debbie McKay (Figure 13), both working for Bell Laboratories in Denver doing computer aided design of printed circuits for telephone communication. Their work is completely done on the computer including design, drawings, and quality control. Another graduate finished his last two years at Brigham Young University and is now working for Boeing in Seattle. And many others are working for architectural and engineering firms who are applying computer aided design.

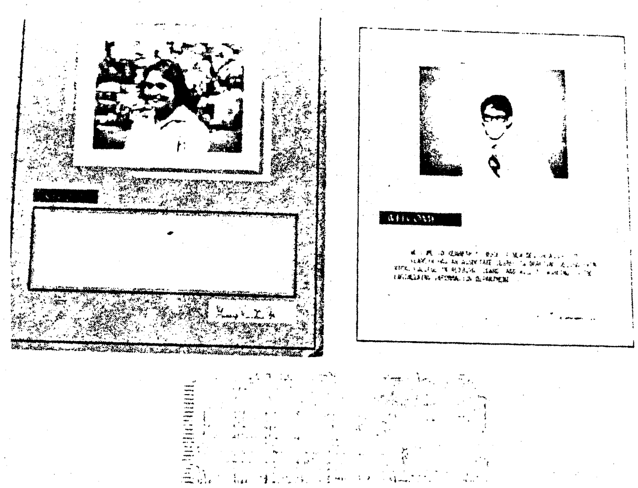


Fig. 13. Two recent graduates of Ricks College with Associate Degrees in Drafting Design are employed at Bell Laboratories in Denver. Deborah McKay is designing circuit boards such as the one shown at the bottom of their Bell Lab welcoming photos. Ken Rock works in the engineering information department; he was the designer of the digitizer utility program described earlier.

Both Pincock and Eckman have had a chance to use their computer aided skills in redesigning their homes. Three years ago on June 6 the Teton Dam located 15 miles above Rexburg on the Teton River washed out, flooding the entire valley. Both men and their families lived near the banks of this river. Pincock stood on a hill and watched the wall of water, 7 to 8 miles wide and more than 10 feet high, sweep his home and outbuildings downstream. Eckman standing next to him saw his home heavily damaged but intact. Both say "it was an interesting experience to have had." Eckman scraped the mud and debris away and repaired his home. Pincock rebuilt his in the same location.

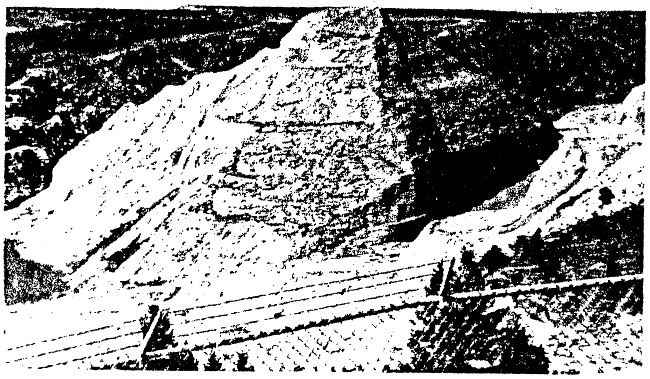


Fig. 14. The pyramid-shaped "mountain" is all that's left of the earthen dam above Rexburg on the Teton River. The dam washed out on its north side (left) sending a wall of water down into the valley.

Fig. 15. Faintly visible here, Sugar City lies 11 miles below the dam. On its rush downstream, the surge of water tore out trees and cut away the bank. Below, it spread out into the valley eight miles across, but was still 10 feet high as it inundated Rexburg 15 miles below.

Editors Note: Blair Pincock and Mel Eckman would be very willing to discuss their program with other colleges or industry. They may be reached at: Ricks College, Dept. of Design and Drafting Technology, Rexburg, Idaho 83440. John Hess, Tektronix Sales Engineer, who helped install the equipment and arranged for this interview may be reached at the Tektronix Salt Lake City field office.

PLOT 50 Business Planning and Analysis Volume 2: A Unique and Powerful Tool for the Decision Maker

by Dennis Heckman

A new software package broadens the business uses of the 4050 Series Graphic Computing Systems. PLOT 50 Business Planning and Analysis Volume 2 (BPA-2) allows you to do time-series analysis and forecasting quickly and easily. A wide range of forecasting methods are available in the package; you identify the data to be used, and specify parameters if desired. The system then takes over to provide the analysis and forecast. The package will handle any kind of data—weekly, monthly, quarterly, accounting period, and so on. High-quality graphic output is an added benefit. Results are also available in tabular form.

Time-series analysis is based on the theory that historical performance indicates some underlying process. Therefore, statistical analysis of the process can provide the basis for projecting future performance. BPA-2 is designed to provide better information for decision makers and planners. Beyond its applications in business, the package's versatility as a statistical analysis tool will allow the same easy use in statistical and scientific environments.

Program Structure

The program is organized into six phases. These phases correspond to the normal steps you would take in developing a forecast. The Data Management group is used to handle and store your data, so that analysis may be performed. Pre-Forecast Analysis determines the nature of the data so that the proper forecasting methods may be used. Deseasonalization identifies and isolates seasonal factors prior to running the forecast computations. These three groups are called upon prior to running the forecast computations.

Forecast Computations is the heart of BPA-2. The Forecast Computations group contains a variety of methods for time-series analysis and forecasting. Once the forecast is completed, two post-forecast program groups are available. Forecast Seasonalization is a unique feature that lets you introduce seasonal factors into the forecast. Post Forecast lets you evaluate the forecast you've just generated. The six groups and the functions they contain are summarized in Table 1.

Table 1

PROGRAM STRUCTURE

Program Group	Functions Performed
Data Management	Data Entry File Manager
Pre-Forecast Analysis	Autocovariance Simple Differences Compound Differences Comparative Analysis Moving Totals
Deseasonalization	Simple Seasonals Averaged Seasonals Smoothed Seasonals Transcendental Smoothing Winters' Method
Forecast Computations	Moving Averages Exponential Smoothing Direct Smoothing Adaptive Smoothing I Adaptive Smoothing II Regression Analysis Gompertz' Curve
Forecast Seasonalization	Seasonalized Forecasts Cumulative Forecasts
Post Forecast	Measures of Dispersion Confidence Intervals

Using the Program

Data Management

When you use Business Planning and Analysis Volume 2, you work through the forecast stages interactively with your Graphic System through a set of functions and through a master menu of programs. Once you choose a program, the BPA-2 functions are accessed through the System's User-Definable Keys. See figure 1.The first step is entering the data that will be used as the basis for the forecast. You can enter data by hand, from the keyboard, or you can call it in from files on tape or disc. The **Data Management** phase of the program allows easy data handling and storage, interactive definition of the graph parameters and rescaling of your data.

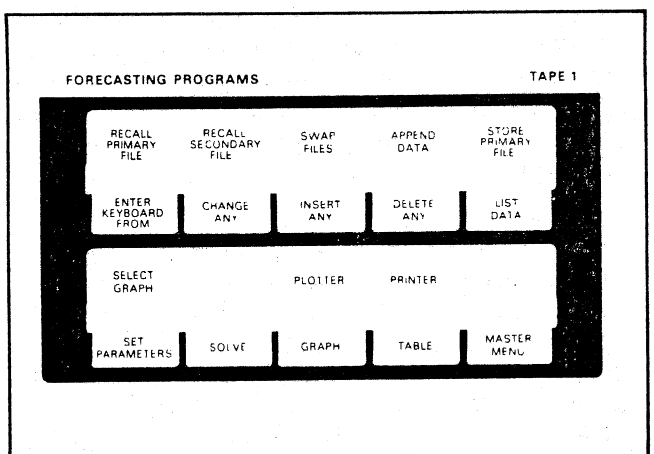


Fig. 1. BPA-2 overlay for the User-Definable Keys. There are 18 functions on the overlay.

Pre-Forecast Analysis

Next you'll want to run a Pre-Forecast Analysis to check for patterns in the data. Determining which patterns are present allows selection of the correct forecasting method. In addition, patterns such as seasonality can be identified and later can be removed by calling up the appropriate Deseasonalization program. The programs may be run repetitively in any order to identify and isolate the trend, seasonal and cyclic patterns that may be present in your data. Figure 2 is an example of the graphs produced by the Pre-Forecast **Autocovariance Analysis** program. This program measures how strongly data values within a time series are related to each other.

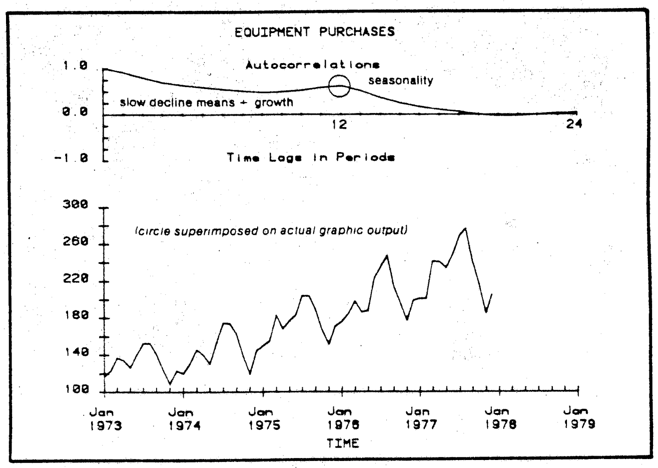


Fig. 2. Autocorrelation coefficient graph from BPA-2, Pre-Forecast Analysis phase. The upper graph gives the autocorrelation coefficients for consecutive time lags which measure the strength of the relationship. Significant coefficient values (say less than −0.8 or greater than 0.8) will appear as peaks. Here the peaks occur at annual (12-period) intervals inferring seasonality. The bottom graph is a graph of the time series.

Still in Phase II looking for patterns, we can look at the **Mean Differences** and **Mean Absolute Differences**. The mean difference for a particular time lag measures the tendency for the difference between new data pairs to be greater than, smaller than, or equal to the difference between older pairs of data. Figure 3 is an example of the graph produced by the **Simple Differences** program.

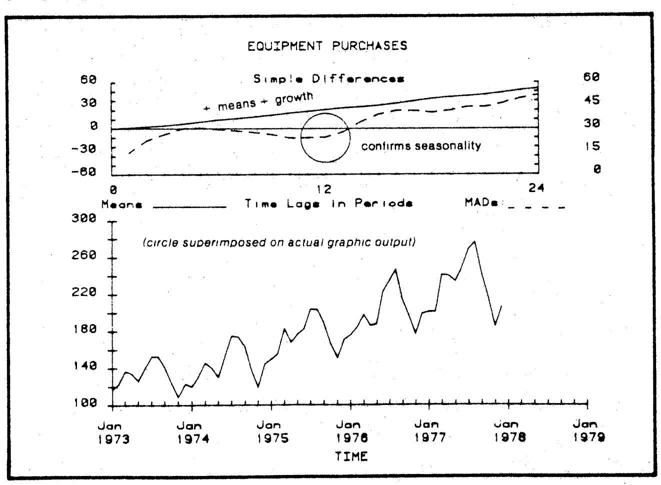


Fig. 3. Mean Differences and Mean Absolute Differences computed from BPA-2, Pre-Forecast Analysis phase. Note the two sets of axes in the upper graph. Graphed on the lefthand axis is the computed Mean Difference

shown as a solid line. Because the means get larger in a positive direction, the growth pattern is positive.

The mean absolute difference (MADS) measures the tendency for the difference between data pairs to be equal. The MADS are graphed on the righthand axis. Here the scale runs from 0 to 60 (always a positive scale). When data repeats itself, the average mean absolute difference is smaller. Note the valleys in the MADS graph correspond directly to the peaks in the autocorrelation coefficient graph in Figure 2.

The lower graph is a graph of the time series.

Deseasonalization

If you have detected a seasonal pattern in your data, you'll want to run a program from the Deseasonalization phase to remove these recurring patterns which may mask underlying trends in the data. There are two kinds of seasonals. Seasonals which depend on trend, that is, the value of the seasonal may vary as the trend varies, are called "multiplicative" seasonals. Seasonals which do not depend on trend are called "additive" seasonals. An example of the former would be if a significant portion of your annual sales are made to the federal government, with the major sales happening at the beginning and end of their fiscal year. However, the amount of their spending in either case is dependent upon the state of the economy, or the general trend. Conversely, no matter how well or poorly business is doing, Joe calls up every year in March and orders 100 units; this is additive seasonality.

Five programs deseasonalize the data, depending on which kind of seasonality you detect and if the value of each data point is as important as another, or if some points carry more weight than others. Both Transcendental Smoothing (for additive seasonality) and Winters' Method compute seasonalized forecasts which eliminates the need to run any programs from the Forecast Computation group or the Seasonalized Forecast program.

The **Averaged Seasonal** method uses a moving average technique. It computes final seasonals by forcing the latest set of moving averages to total 100 times the number of periods per year. This allows more importance to be placed on recent data and allows the seasonal factors to change. The results are displayed as two graphs (Figure 4).

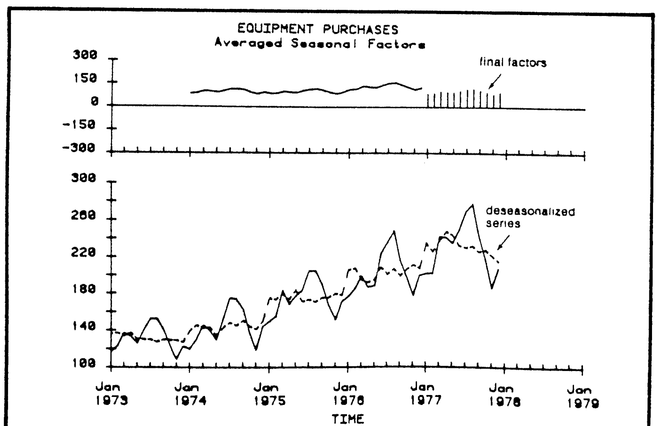


Fig. 4. Averaged Seasonals computed from BPA-2, Deseasonalization phase. The averaged seasonals are plotted as the solid line and the final seasonal factors as vertical bars in the upper graph. Note the averaged seasonals are centered, because they are moving averages, as they are plotted. Here three years of data were used for each averaged seasonal. In the lower graph the solid line depicts the original time series; the deseasonalized version is shown as a dashed line. Removing the seasonality results in a more stable time series.

If you suspect the seasonality is additive (independent of any trends), try the **Transcendental Smoothing** program. As mentioned earlier, this method deseasonalizes the time series, forecasts, and seasonalizes it again. The graph in Figure 5 is produced by this method.

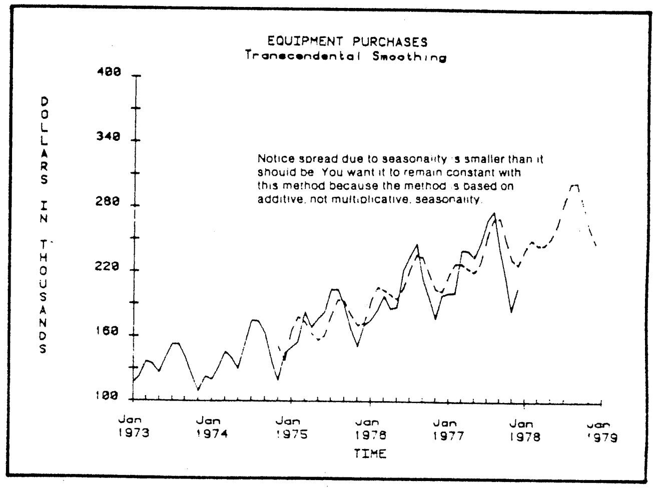


Fig. 5. Transcendental Smoothing graph from BPA-2, Deseasonalization phase. As you inspect the graph, mentally draw lines along the upper and lower limits of the time series data (solid line) and forecast (dashed line). In this example seasonality is growing in its variation as the trend grows so Transcendental Smoothing is not the program to use. If the seasonal contribution were independent, the bandwidth would remain approximately constant.

Let's step back into the track of multiplicative seasonals. The **Winters' Method** provides a total forecast. However, it requires **lots** of data (a minimum of two years) to cover the initial period and provide the comparison values for the initial forecasts. The Winters' Method will graphically depict seasonality changes which would never be picked out from a table of numbers. A graph of the results of the Winters' Method on our data is shown in Figure 6.

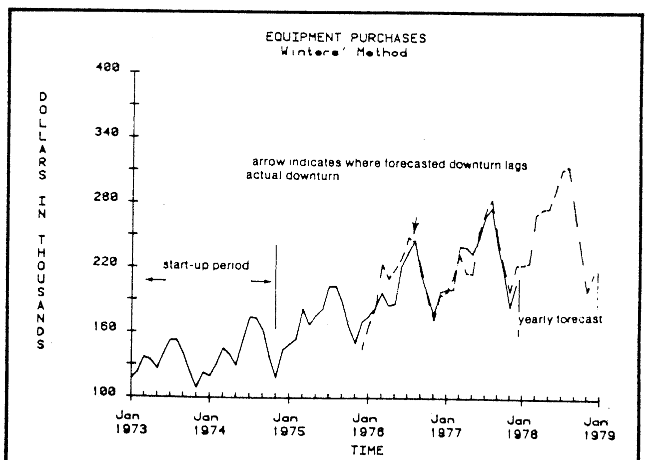


Fig. 6. Winters' Method graph computed from BPA-2, Deseasonalization phase. Of concern here is the amount of variation between the forecast (dotted line) and the actual data (solid line). Although every forecast won't match

the data exactly, the slope of the forecast should match that of the data and the turning points should match. The forecast slope here matches the historical data pretty well; however, the actual downturn in the summer of 1976 lagged the forecast. This graph would suggest further analysis using the Post-Forecast programs.

Forecast Computations

By this step in the forecasting process, you should have a good feel for your time series. You should know whether seasonality needs to be considered or not. You should be able to view your time series (in deseasonalized form, if applicable) and determine which trend pattern best fits your data (Figure 7).

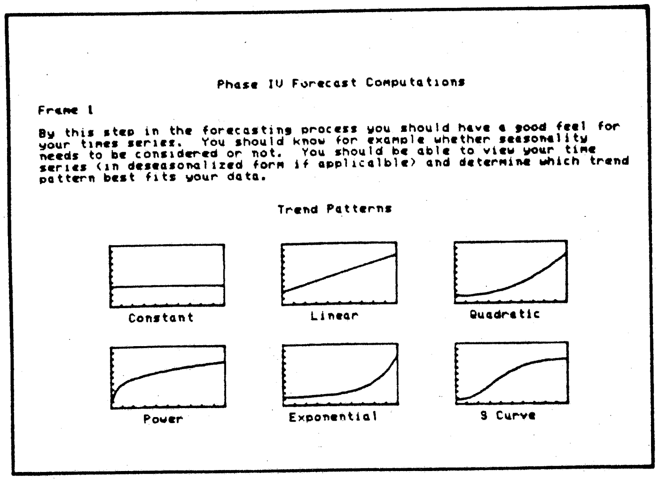


Fig. 7. The description of the pattern is an equation which is a function of time. Such an equation is called a model.

There are seven programs in the Forecast Computation section. Each program is designed for data that fits a certain model, or your choice of several models. Table 2 summarizes the programs and their models.

Table 2
PROGRAMS, METHODS, AND MODELS

Program	Computational Methods	Model
Program 13 MOVING AVERAGES	Single Moving Averages Single Weighted Moving Averages Double Moving Averages Double Weighted Moving Averages	Constant Constant Linear Linear
Program 14 EXPONENTIAL SMOOTHING	Single Exponential Smoothing Double Exponential Smoothing Triple Exponential Smoothing	Constant Linear Quadratic
Program 15 DIRECT SMOOTHING	Single Direct Smoothing Double Direct Smoothing Triple Direct Smoothing	Constant Linear Quadratic
Program 16 ADAPTIVE SMOOTHING I	Single Adaptive Smoothing Double Adaptive Smoothing Triple Adaptive Smoothing	Constant Linear Quadratic
Program 17 ADAPTIVE SMOOTHING II	Single Adaptive Smoothing Double Adaptive Smoothing Triple Adaptive Smoothing	Constant Linear Quadratic
Program 18 REGRESSION ANALYSIS	Linear Regression Quadratic Regression Exponential Regression Power Regression	Linear Quadratic Exponential Power
Program 19 GOMPERTZ CURVE	Gompertz Method	S Curve

Each forecasting method has its own technique for computing the estimates of the model parameters. The data in the previous examples seems to fit a linear model. Therefore, the **Direct Smoothing** program will be used. The graph in Figure 8 displays the forecast based on deseasonalized data.

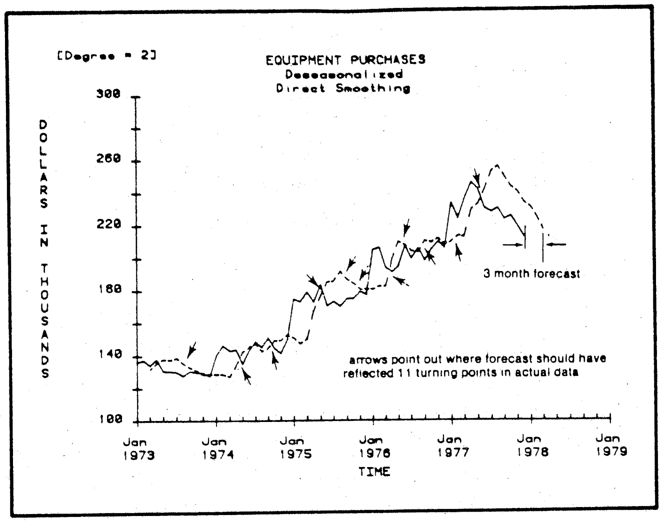


Fig. 8. Direct Smoothing forecast from BPA-2, Forecast Computation phase. Notice the degree is 2, or double smoothing, and the data has been deseasonalized. The forecast line (dashed) intersects the data (solid line) often. If it didn't a higher order model would be called for. However, the forecast does not reflect 11 critical turning points in the actual data. And although exponential smoothing will cause forecasted peaks and valley to lag actual data values, here they lag too far. Once the forecast computation is performed, the unique Forecast Seasonalization capability can be called upon. Introducing the seasonal factors that were identified earlier may explain some of the discrepancies.

Forecast Seasonalization

The Forecast Seasonalization phase introduces previously identified seasonal factors into the forecast. Note the **Seasonalized Forecast** shown in Figure 9.

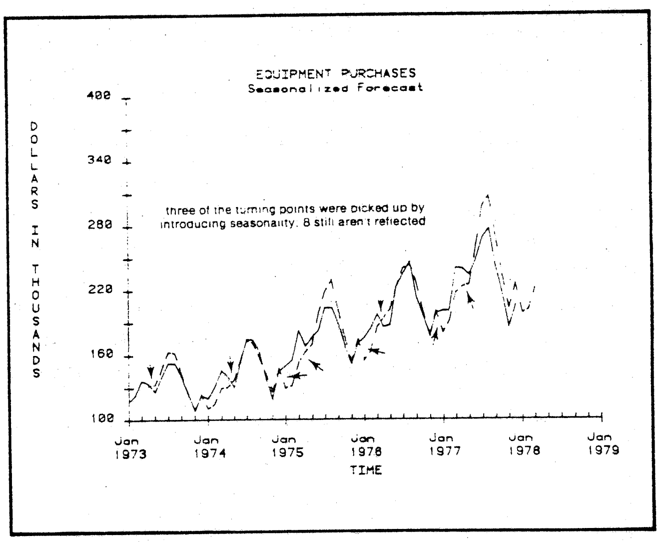


Fig. 9. Seasonalized Forecast graph from BPA-2, Seasonalization phase. Note the scaling has been changed from the graph in Figure 8. Introducing seasonality explained only three of the turning point discrepancies. The forecast is adding in a trough at the beginning of each year. We need to look at the forecasting method and the seasonal factors.

Post-Forecast Analysis

Post-Forecast Analysis programs analyze the errors of previously computed forecasts. **Measures of Dispersion** shows the dispersion of the errors and helps you judge the effectiveness of the forecasting method you chose. **Confidence Intervals** uses the errors to compute a confidence interval for each forecast. Both programs first seasonalize the forecasts if necessary. Results from these two programs may point out changes in the seasonal pattern or in the trend-cycle pattern of the time series. These changes may require that you adjust or change the current forecasting method or deseasonalization program. Figure 10 graphs the forecast errors for our previous example data. Acceptance or rejection of the forecast depends on the size of the errors and your accuracy criteria.

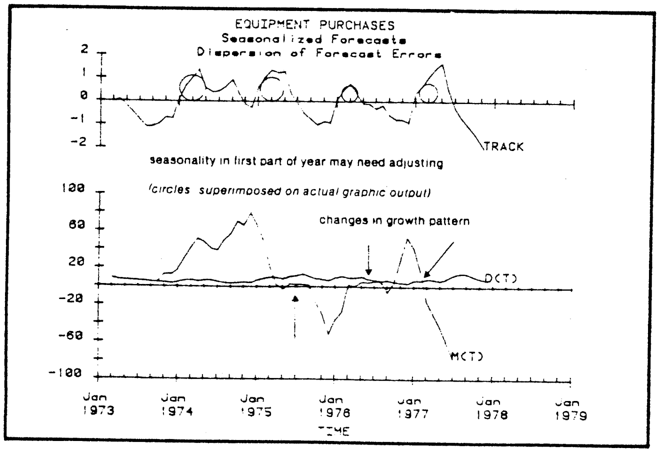


Fig. 10. Measures of Dispersion analysis from BPA-2, Post-Forecast Analysis phase. Note the pattern of the tracking signal in the upper graph at the beginning of each year. The uniform oscillation at yearly intervals indicates the deseasonalization method may be failing.

The smoothed mean absolute deviation (m.a.d.) of the errors, shown in the lower graph as a solid line labeled D(T), seems to be increasing. This indicates the forecast is diverging from the process. The cumulative errors are graphed (in centered form) as a solid line labeled M(T) in the lower graph. It is the yearly moving totals of the errors.

Values of the cumulative errors and tracking signal are negative when the forecast exceeds the time series, and positive when the forecast is less than the series. Comparing these results against Figure 9 will bear this out.

Returning to the Forecast Computation phase once more, we'll run the **Regression Analysis** program on the deseasonalized time series. (Since BPA-2 is organized in phases, it is possible to stop at any point and go back only as far as necessary to try another method.) This technique determines the equation of a line that best fits the data. In Figure 11 we are using a quadratic model—note the formula in the title.

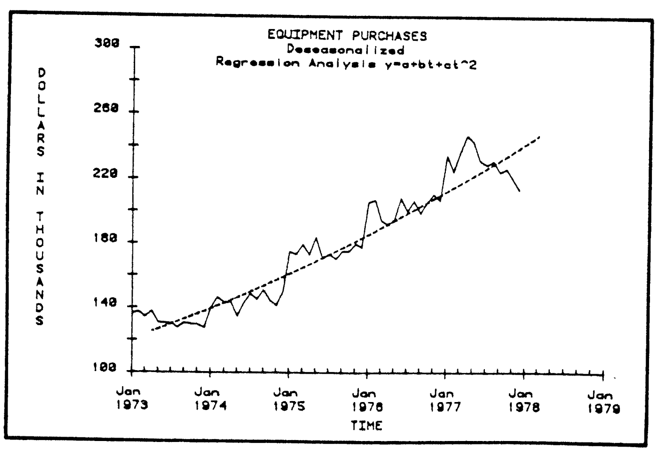


Fig. 11. Regression Analysis computed from BPA-2, Forecast Computation phase. The slope of the forecast line (dashed) doesn't look too bad. The three month forecast indicates it may be leveling out.

Inserting the seasonal patterns into the Regression Analysis forecast computation tells us we're still having a problem with the beginning of the year (Figure 12). The seasonal figures may need tweaking. Other factors may need to be considered such as union contracts, floods, material shortages, and so on.

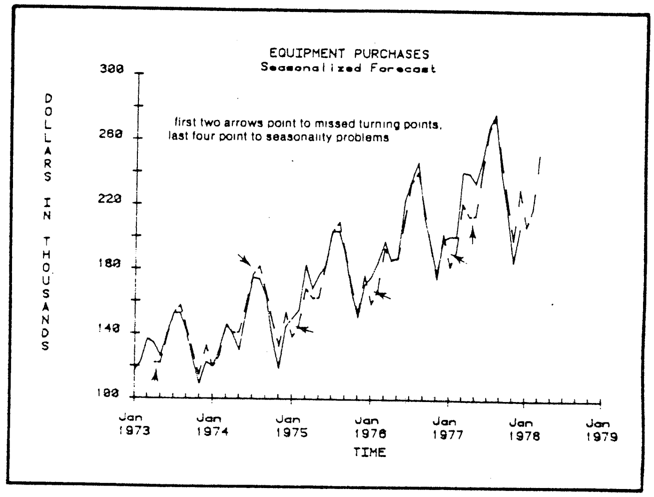


Fig. 12. Seasonalized Forecast (based on Regression Analysis) from BPA-2, Seasonalization phase. The troughs at the beginning of each year are still causing problems. It may be necessary to change some seasonal factors—this is easily done by pressing the SET PARAMETERS function key.

The analysis of the forecast errors from this latest run portrays the same pattern (Figure 13).

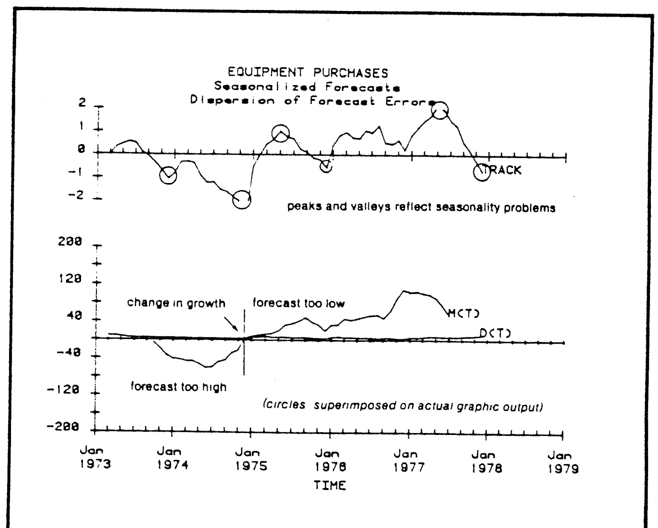


Fig. 13. Dispersion of Forecast Errors calculated by BPA-2, Post-Forecast Analysis phase. In this graph the pattern is showing up as valleys compared to peaks in Figure 10. The direction of the forecast isn't matching that of the data. We are getting more and more signals that the process is changing toward the end and that the seasonals need adjusting.

If the data is changing perhaps it will fit a **Gompertz Curve**. This pattern is typified by an elongated "S"-shaped curve. The curve describes three phases in the life of the underlying process. In the first phase there is limited but positive growth. During the second phase of rapid transition, the growth curve is exponential. Finally, in the third phase, the growth slackens again. The results in Figure 14 suggest the process is possibly starting the second stage.

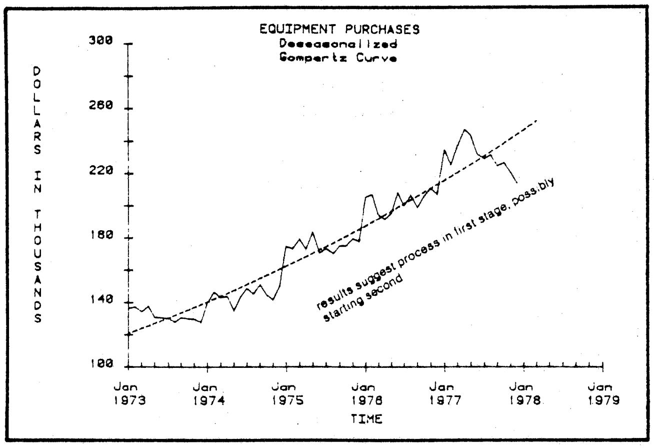


Fig. 14. Gompertz Curve fitted to deseasonalized data from BPA-2, Forecast Computation phase.

Figure 15 graphs another time series demonstrating a process in the third "mature" stage.

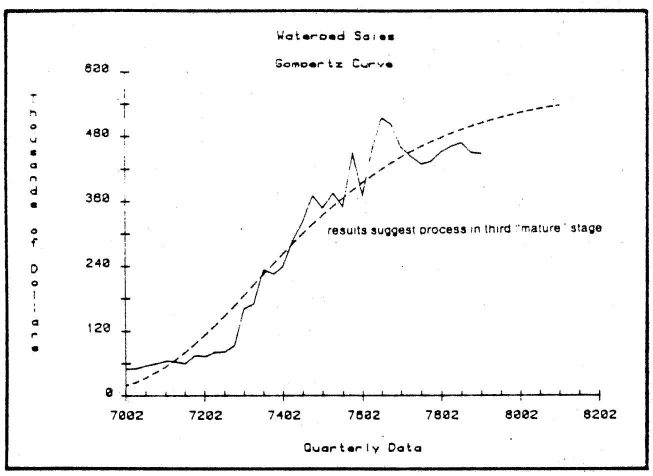


Fig. 15. Gompertz Curve fitted to another set of time series data. Although in the third stage, the curve still has a positive slope.

A final program in the Post-Forecast Analysis phase, **Confidence Intervals**, may be used to equate the size of the forecast errors and the resulting range of forecast values with probabilities. You can change the percent confidence value through the SET PARAMETERS function key.

The Manual

The Business Planning and Analysis Volume 2 User's Manual is designed for both novice and expert. For those not familiar with the 4050 Series, or even computers, an introductory section helps them through the basics. Another section thoroughly acquaints the user with the functions of the BPA-2 programs and how they're accessed through the User Definable Keys. A refresher course in basic forecasting techniques is provided, followed by a sample session which introduces the user to BPA-2 methods. The remaining sections carry the user through each phase, step-by-step with plenty of illustrations and graphics. The appendix contains a glossary of forecasting and computing terms for a basic understanding. It also contains the mathematical equations used in each of the programs.

In Summary

Objectively evaluating historical data and then using that evaluation to accurately predict future values is the basis for many of today's important commercial and scientific decisions. PLOT 50 Business Planning and Analysis Volume 2 is a powerful analysis tool that provides better decision data in the form of *graphic* results, for decision makers and planners in many areas. It frees the user from performing the many required complex computations manually, or from having to depend on a time-share system.

4050A11 is the tape version of BPA-2 which uses the 4924. A disc version, 4050A12, will soon be available. Contact your local Tektronix Sales Engineer for more information on Business Planning and Analysis Volume 2.

CAMSEQ/M: A 4051-Based Conformational Analysis System

by Herschel J. R. Weintraub

School of Pharmacy and Pharmacal Sciences
Dept. of Medicinal Chemistry and Pharmacognosy
Purdue University
West Lafayette, IN

CAMSEQ is a program package that has been developed for use in the analysis of organic chemical molecules[1]. The programs in the package are used in conformational analysis, a new and exciting area of medicinal chemistry. In conformational analysis, chemists use the principle that a molecule may have many different shapes in solution, some of which may be similar to that in the crystal state. The CAMSEQ programs use both empirical and quantum mechanical techniques to find the lowest free-energy state in free space and in solution; the latter is the most likely state that will occur in the body.

The CAMSEQ package has been running in batch mode at Purdue and many industrial and other academic sites for some time. An interactive version is also running on a timeshare basis on the National Institutes of Health-Environmental Protection Agency-Chemical Information System (NIH-EPA-CIS)[2]. The advent of the powerful "stand-alone" capabilities of the 4051, with its capacity for interactive graphics, prompted the development of CAMSEQ/M. This package also performs conformational analysis, but is designed to run locally on any 4050 Series Graphic Computing System.

Conformational Analysis

Conformational Analysis is a very new field, but one that is beginning to have important impacts in the field of chemical and molecular design. It is particularly useful in the field of medicinal chemistry, where knowledge about the state and shape of chemical molecules can be used to help predict their activity in the body. Three-dimensional shapes, electronic parameters, bond types, excluded volume—all of these factors and more contribute to how a molecule will interact in a living system. The technique of conformational analysis can be used to find the most likely active state of chemical compounds, and also to predict inactive states.

The technique uses computing power to rotate all of the bonds in a molecule. The effect is to theoretically change the molecule's three-dimensional shape, or conformation, with the goal of finding the shape with the lowest free-energy in solution. Statistically, the state with the lowest free energy is the one that's most likely to occur in the body.

By determining the characteristics of known compounds, and using this information for comparison with new, similar structures, useful modifications may be predicted. Besides predicting the activity level of various molecules, the assumption that similar compounds react in a similar fashion can be helpful in other ways. One of the newest applications uses the same assumptions and techniques to determine if compounds might be carcinogens. Conformational analysis is used to compare new compounds with the molecular shapes of known carcinogens. Again, the three-dimensional molecular structure is used as a predictor of biological activity.

Why Local Processing?

The batch versions of CAMSEQ have a fairly high core demand (30-50K words). The NIH-EPA-CIS interactive version also has a rather large memory requirement. CAMSEQ/M is designed to supplement both the NIH-EPA-CIS system and the batch processors by providing local processing power. This relieves the former system for data base manipulations rather than the time-consuming conformational analysis calculations.

While computer graphics *might* be viewed as a fancy, non-essential, luxury item, it becomes a necessary tool for the understanding of the chemistry and energetics of complex molecular systems. In many cases, the solutions to difficult theoretical problems are too intricate to be easily understood when presented in the form of tables. The sheer bulk of the data often is so great that it defies comprehension. That's where the high resolution graphic displays of the 4050 Series come into the picture.

CAMSEQ/M, as noted before, was developed around the 4051, but will run on any 4050 Series Graphic Computing System. Required hardware includes the 4051 with at least 16K memory (32K is preferred), a joystick, and a Matrix Function ROM Pack. A more versatile disc-based system has been implemented by including the 4907 File Manager into the system. With either system, communicating with the NIH-EPA-CIS computer (or other host system) necessitates adding the Option 1 Communications Interface for RS-232-C communication.

Using CAMSEQ/M

When the system is turned on and CAMSEQ/M is initiated, a menu is displayed on the 4050 Series display. The particular task to be performed is selected from this menu, as shown in Figure 1. In this example, the "crystallographic coordinate" input feature has been selected. As the example shows, the program prompts the user for various required parameters. And, at any point, a question mark can be typed by the user, prompting the system to highlight the appropriate information on the screen to help the user determine what to do.

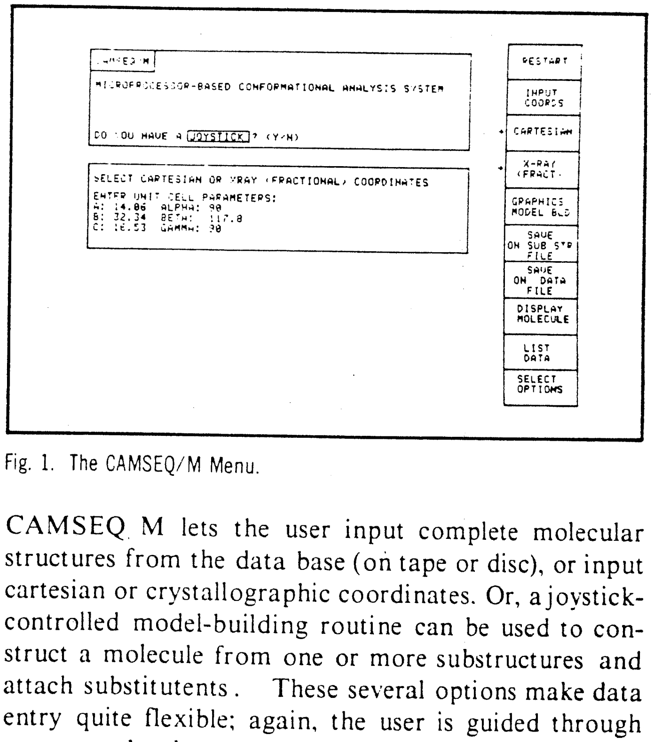


Fig. 1. The CAMSEQ/M Menu.

CAMSEQ/M lets the user input complete molecular structures from the data base (on tape or disc), or input cartesian or crystallographic coordinates. Or, a joystick-controlled model-building routine can be used to construct a molecule from one or more substructures and attach substitutents. These several options make data entry quite flexible; again, the user is guided through every step by the program's prompts.

Continuing with the example shown in Figure 1, the user is prompted to supply the appropriate coordinates, and the atom type (C, O, N, H, etc.). The program determines the exact atom type (sp3, etc.) by analyzing the molecular connectivity data found in a subsequent step. This connectivity data is calculated, following coordinate input, by locating bonds that are less than 1.6 Angstroms apart between atoms. A sketch of the molecule can then be produced to verify the data input.

Bonds can be added or deleted while viewing this display. The graphic view is more convenient for locating incorrect data (coordinates, bonds) than a table of similar information; but tables of all topological and topographical information are also available. To rotate the molecule, or to produce a stereoscopic view, the user simply requests a "stereo" view rather than a "quick" view. Various molecular parameters, including bond lengths, bond angles, and torsion angles, may be determined while viewing this display. Publication-quality stick models or "disk and stick" models may be produced, as shown in Figure 2.[3]

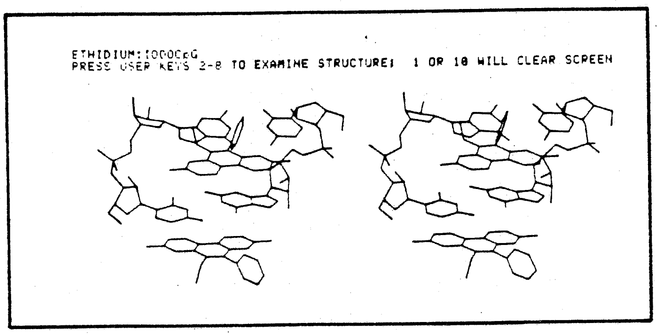


Fig. 2. A stereo view of a calculated molecular structure.

Some More Examples

Figure 3 shows the menu obtained when the graphics model builder option is requested. A list of up to nine frequently-used substructures is displayed to speed the selection of appropriate subunits. Also shown in Figure 3 is an example of joystick-assisted construction of a simple compound, amphetamine. The sequence required to construct this molecule is as follows:

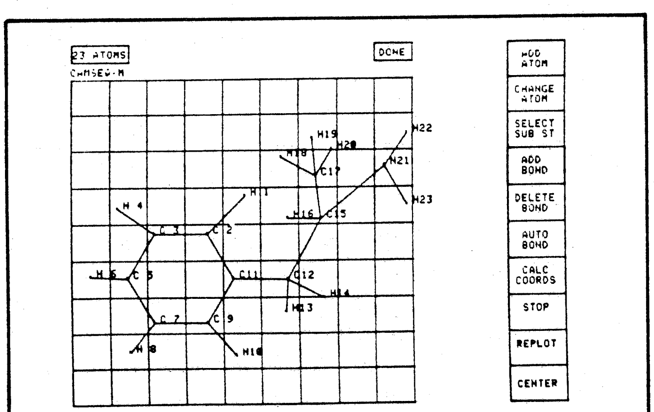


Fig. 3. The Graphics Model Builder menu, with a joystick-constructed model.

Step 1. The benzene substructure is selected from the list (by "pointing" to it with the joystick-controlled cursor), and placed in the desired location on the screen. The grid is provided for convenience in estimating distances. This is only a "ghost" structure; it becomes part of the molecule only after step 3, below.

Step 2. The "add atom" feature is selected and the number of atoms is defined.

Step 3. The cursor is moved about the screen using the joystick, and the desired atom letter (C, O, N, H, etc.) is typed. If the cursor is placed near a substructure atom and a "space" is typed, the substructure atom is added to the molecule. In this way, only the desired portion of the substructure need be incorporated into the molecule.

Step 4. After all atoms have been approximately located, bonds are added, either explicitly or implicitly, by specifying the "add bonds" or "auto bond" options, respectively. The implicit bonds are determined by calculating the interatomic distances as drawn on the screen, and bonding all atoms within 1.6 Angstrom of one another. A "delete bond" feature allows one to remove unwanted bonds. The "add bond" feature allows the bonds to be assigned explicitly by indicating the atom pairs to be bonded and depressing a key on the keyboard. The bonding operations automatically determine the connectivity data.

Step 5. Exact, "standard" bond lengths and bond angles are then used to transform the approximate structure displayed on the screen into an accurate molecular structure. The bond angles and distances to be used in the calculation are displayed, and may be modified if desired. This procedure is accomplished by specifying "calculate coordinates" from the menu.

Figure 4 illustrates the "standard" model generated by CAMSEQ/M. This structure may be saved for later use, or may be used as an input structure for a conformational study. The structure may later be used as a substructure for studying other congeners by modifying its substitutents. For example, phenethylamine may easily be constructed by removing the alpha-methyl group on the

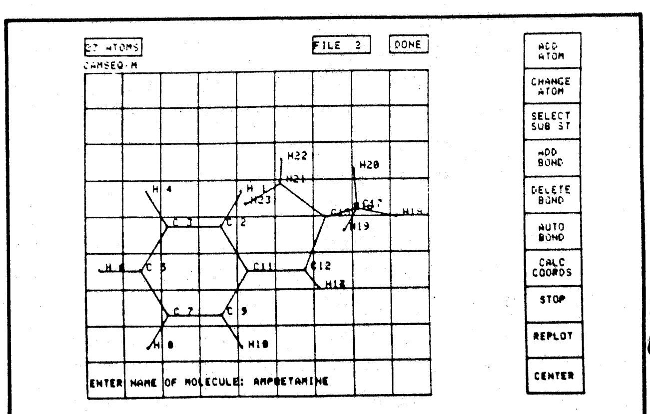


Fig. 4. The "standard" CAMSEQ/M model.

ethylamine sidechain. The stereochemistry of the molecule is determined by the input order of the substitutents.

Several substructures may be connected together to form more complex molecules. As an example, a polycyclic aromatic hydrocarbon consisting of "standard" bond lengths and angles may be constructed by connecting several benzene rings together. The sequence of operations is as follows.

Step 1. Select the benzene substructure from the list and place it in position on the screen (as previously described for the general graphics model builder example).

Step 2. Enter the total number of atoms in the complete structure by using the "add atom" feature.

Step 3. Attach appropriate substitutents to the first ring, and assign bonds using either the "add bond" or "auto bond" menu items. Next, select another substructure (by "pointing" to "select substructure" on the menu). Using the joystick, "point" to the "common atoms" of the two structures and CAMSEQ/M will attach the structures to one another.

Step 4. Repeat this sequence of operations (step 3) until the entire structure is generated. If desired, all substructures may be connected initially, and the substitutents added last. The resultant structure may be saved as either a substructure or as a molecule for later use.

In this example it would be more convenient to maintain the substructure in the substructure file, and to generate smaller bi- or multi-cyclic ring systems by using portions of the larger molecule.

Figure 5 is another example of complex model generation; it illustrates the generation of the DDT structure from two benzene rings and other atoms. The first step involves the selection of the first benzene ring, the assignment of 28 total atoms to the molecule, and the location of the first 18 atoms (Figure 5a). The calculation

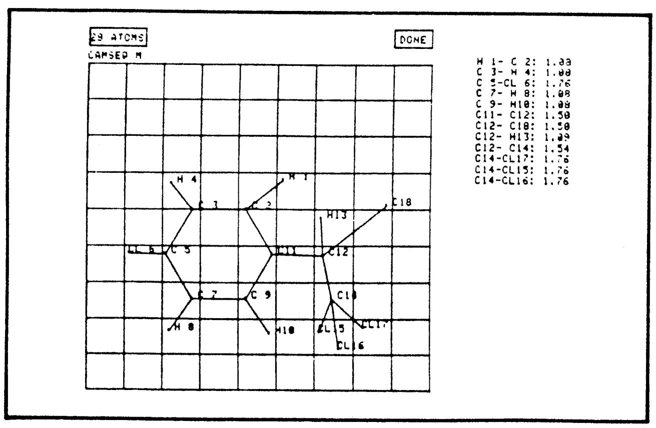


Fig. 5a. Beginning a complex model generation.

of the "standard" geometry has begun, as indicated by the list of bond lengths. Figure 5b illustrates the "uniting" of two molecular fragments: the 18-atom structure depicted in the first figure and the second benzene ring substructure (shown in the upper right corner). The two vectors or line segments connecting the two substructures define the common bond linkage (i.e., C12-C18 and C-X become a single bond). The transformation of the two fragments

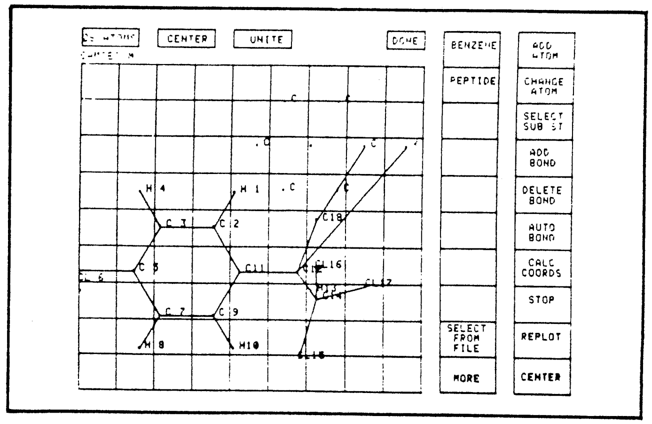


Fig. 5b. Continuing the generation of a complex molecule, two molecular fragments are united.

into one molecular structure is shown in Figure 5c. The remaining 10 atoms have been approximately located in this illustration. The "standard" geometry for the entire

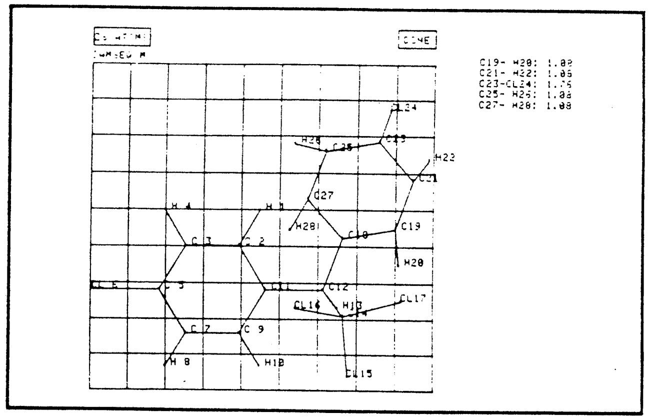


Fig. 5c. Completing the transformation of two fragments into one molecular structure.

structure is illustrated in Figure 5d, and a stereoscopic view is provided in Figure 5e.

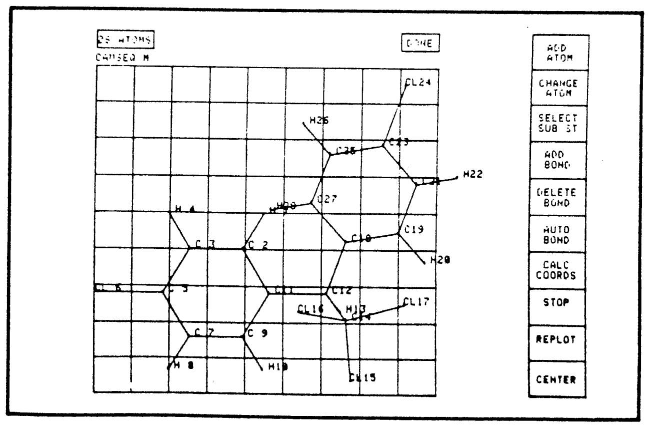


Fig. 5d. The "standard" geometry for the newly-created molecular structure.

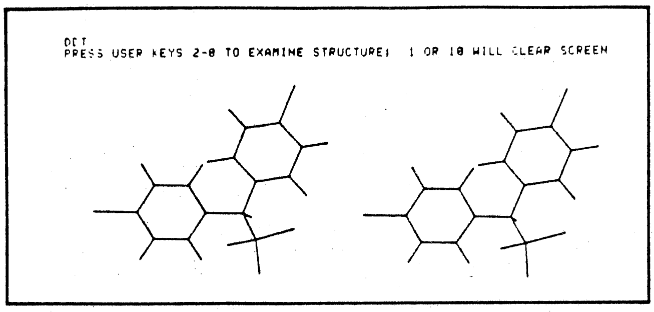


Fig. 5e. A stereoscopic view of the newly-created structure.

At this stage, the molecular coordinates have been determined, and the structure has been verified for accuracy by inspection of the drawings and tables provided by CAMSEQ/M. The next step sets up the desired conformational study to be performed. A well-documented procedure has been developed for CAMSEQ/M to assist the user in this often difficult task. The difficulty lies in the large number of available options; some of these options are alphabetically listed in Table 1.

Table 1

ANGLE:	Lists all bond angles in the molecule
CLOCK:	Performs "clockwise" rather than "counterclockwise" bond rotations
CMAPS:	Produces iso-energy contour maps
DISTANCE:	Lists all interatomic distances in the molecule
ELECTRO:	Selects appropriate electrostatic potential energy function
HBOND:	Determines hydrogen-bonding energies
LINPLOT:	Generates an energy vs. rotation angle plot
LISTCOORD:	Lists all atomic coordinates
LJPAR:	Allows modification of Lennard-Jones, nonbonded potential function parameters
LOCALMIN:	Lists all low energy conformers
MINIMIZE:	Performs multi-dimensional minimization
RANDOM:	Performs random type scan of conformational hyperspace
REFERENCE:	Allows user to specify a "standard" reference conformation
SCAN:	Performs a uniform sequential scan of conformational hyperspace
THERMO:	Lists thermodynamic probabilities
TORSIONAL:	Allows user to specify torsional correction functions

Since CAMSEQ/M has been designed to be totally compatible with the batch versions of CAMSEQ, there are additional options that are appropriate for the batch version. Since the two versions are totally compatible, CAMSEQ/M may be used either as a stand-alone conformational analysis device, or it may be used as an interactive, conversational off-line input device for batch CAMSEQ.

The option selection process is simplified by providing tables of options arranged according to function. A complete description of each option is also available by pressing a single key and specifying the option name. When an option name is selected, CAMSEQ/M prompts for each input item with a short description.

Figure 6 illustrates such an input sequence for two representative options, "CMAPS" and "TORSIONAL". As the example shows, the user need not be familiar with the internal workings of CAMSEQ/M in order to select the various options needed to describe the job to be done.

```
THE AVAILABLE OPTIONS, IN ALPHABETICAL ORDER, ARE:

ANGLE       ATOMSIZE    CAVITY      CHARGE      CHKBL
CLOCK       CMAPS       CONTACT     COORD       DEBUG
DIFSOLU     DISTANCE    ELECTRO     HBOND       LINPLOT
LISTCOORD   LJPAR       LOCALMIN    MAPAXES     MAPPARAM
MESSAGE     MINIMIZE    MODEL       MODPIC      NEWDEL
NEWSOLU     NEWTOR      NOCONF      NOHEADING   NOMODEL
NOPIC       ORTEP       QUANTUM     RANDOM      REFERENCE
REFPIC      RENUMBER    SCAN        SCREEN      SELECT
SKIPDATH    SOLUDATA    THERMO      TIMELEFT    TORSIONAL
WAITING     UTLPIC      VACONLY     XRAY        ZMATRIX

ENTER OPTION NAME (OR COMMAND BY PRESSING USER KEY "1"): torsional

ENTER TYPE OF FUNCTION (STANDARD, 2-D, DISCRETE, 2-D AND STANDARD):
: standard

FUNCTION:  Etor= Barrier1*(sp1+sp2*cos(Mult1*Bond+phase1)↑Power) +
                 Barrier2*(1-cos(Mult2*Bond+phase2))

BOND # > 12
USE INTERNAL FUNCTION ? no
BARRIER 1 > 2.5
SHAPING PARAMETER (sp1) > 1
SHAPING PARAMETER (sp2) >-1
POWER > 2
MULTIPLIER > 1.5
PHASE ANGLE 1 > 0
CIS/TRANS BARRIER (BARRIER 2) ? 0
CIS/TRANS MULTIPLIER (MULT 2) > 0
CIS/TRANS PHASE ANGLE (PHASE 2) > 0

ENTER OPTION NAME (OR COMMAND BY PRESSING USER KEY "1"): cmaps

NORMAL MAPS (N) OR SOLVENT EFFECT ONLY MAPS (S) ? n

BOND 1 (PLOTTED ON HORIZONTAL AXIS) : 11
BOND 2 (PLOTTED ON VERTICAL AXIS) : 14

NORMAL RANGE (0 TO 360 DEGREES) ? yes

LEVEL LINES DESIRED: 1 2 5 10
LEVEL LINE LABELING: yes


ENTER OPTION NAME (OR COMMAND BY PRESSING USER KEY "1"):
```

Fig. 6. "Torsional" and "CMAPS" options have been chosen to demonstrate CAMSEQ/M's interactive prompts.

In the Figure 6 example, all necessary information is provided to enable the option to be used effectively. Thus, the form of the torsional barrier function is shown, indicating the various parameters involved. The "shaping parameters" are shown to be terms used to describe such functions as "cos" as "0+1 * cos", or "sin" as "1-1 * cos ^{2}", etc. Note that only those options that are required to specify parameters or control the conformational search must be specified in advance. Other options, such as those that affect the ultimate data analysis, may be specified after the conformational search is completed. Of course, if CAMSEQ/M is used as a data preparation device for batch CAMSEQ, these options must all be specified. However, even data sets produced by the batch CAMSEQ versions may be analyzed using CAMSEQ/M.

Specifying the options in this manner allows the rapid description of the conformational analysis task. If desired, the user is prompted for additional options at the appropriate point in the program. For example, prior to setting up the bond rotation data, the user is asked if he wishes to describe a reference conformation. If "REFERENCE" was specified previously, the user will be prompted for the appropriate information at this time without the query. In this manner, even if one forgets to select an option in advance, CAMSEQ/M provides a reminder that the option is available. If using CAMSEQ/M as a stand-alone system, it is not even necessary to specify the "select options" feature depicted on the menu in Figure 1. One need merely press the key labeled "perform conformational search", and the system will request the appropriate options.

The 4051 has been used for conformational analysis at

Purdue's Dept. of Medicinal Chemistry and Pharmacognosy in the School of Pharmacy and Pharmacal Sciences since the introduction of the TEKTRONIX Graphic Computing Systems. Its easy use and high-resolution graphics capabilities are a valuable asset, especially in an area where the graphic views of a molecule are a great aid to the chemist user.

Current efforts in the CAMSEQ/M system development include interfacing a low-cost microcomputer to the 4051; the microcomputer would then handle the actual conformational search and energy calculations. This would free the 4051 for other computing tasks, including set-up of additional molecules for conformational study. In concept, banks of such low-cost microcomputers could be linked to a single 4050 Series Graphic Computing System, providing an even more efficient conformational analysis system.

References
1. H. J. R. Weintraub, A. J. Hopfinger, *International Journal of Quantum Chemistry*, **QBS2**, 203 (1975). H. J. R. Weintraub, Ph.D. Dissertation, Case Western Reserve University, Cleveland, Ohio, 1975.
2. S. R. Heller, G. W. A. Milne, R. J. Feldmann, *Science*, **195 (4275)**, 253 (1977). R. Potenzone, Jr., E. Cavicchi, H. J. R. Weintraub, A. J. Hopfinger, *Computers and Chemistry*, **1**, 187 (1977).
3. Crystal structure coordinates for this display: S. C. Jain, C. C.–Tsai, H. M. Sobell, *Journal of Molecular Biology*, **114**, 317 (1977).

*Editor's Note

Programming Tip Exchange Continues

Have you found a small piece of programming information in the course of your work that you think others might find useful as well? We'd like to hear about it, and publish it for others as a Programming Tip or a Basic Bit. Send it in to the Applications Library. Any one of the following Applications Library Programs will be yours when it's published. Simply jot down a brief description of its function, the code, and your choice of program. Mail the information to the Applications Library address serving you; the addresses can be found at the back of each issue.

51/00-0101/0	51/00-5503/0
51/00-0702/0	51/00-7002/0
51/00-0715/0	51/00-8006/0
51/00-1401/0	51/00-9505/0
51/00-1402/0	51/00-9511/0
51/00-5401/0	51/00-9521/0

*Documentation and listing only.

New Catalogs are Available

By now you should have received your new 4050 Series Applications Library Catalog, which includes 40 new program additions. If you haven't received yours, or if you'd like additional copies, just drop a line to the Applications Library office serving you. They're free for the asking.

New ROM Pack Enhances Data Analysis Capabilities of 4050 Series

by Dave Barnard and Cliff Morgan

Have you ever had a set of evenly spaced data values and wanted to:

- Determine the frequency and amplitude of the cyclic components; or
- Highlight the cyclic content of the data and at the same time suppress most of the random ups and downs; or
- Measure the degree of association between that set and another data set even though one may lead or follow the other, and also determine the number of points of lead or lag that *maximizes* the amount of association; or
- Apply discrete time or digital filtering to the set?

If you have answered "yes" to any of the above, you may find the newest 4050 Series ROM Pack will solve your problem more simply and more quickly than you imagined.

The newest product, the Signal Processing ROM Pack #2 (FFT), is actually a pair. One, 4051R08, is compatible with the 4051. The other, 4052R08, is compatible with the 4052 and 4054.

The new ROM Pack lets you:

- Perform the Fourier Transform of an array of data containing from 16 to 1024 points.
- Calculate the autocorrelation of a data array of 8 to 512 points.
- Perform cross-correlation of two sets of data each containing from 8 to 512 points each (for every position of one relative to the other).
- Calculate the convolution of two sets of data points. One set represents the digital filter or any other discrete time system, the other set is a set of data to be processed.

Signal Processing ROM Pack #2 (FFT) begins where Signal Processing ROM Pack #1 left off. Its eight new commands include FFT, IFT, Convolution, Correlation plus four others for converting data to alternate formats or for pre-processing the data for easier display and analysis.

If you already use any of the functions mentioned you will find them easier than ever to use in their ROM Pack form. Each of them may be executed directly from the Graphic System keyboard or from a 4050 BASIC Program.

As with other ROM Packs, adding the new functions by plugging the new ROM Pack into one of the slots provided, consumes no read write memory. All of the new functions are performed with just one statement per function. The FFT command, for example, can reduce a typical FFT program in 4050 BASIC (50 lines or more long) to a single statement: a memory saving.

With more space freed for data and analysis programs you would probably like to set your sight higher toward newer and tougher problems. If you were just thinking it would help if the ROM Packs were faster than their 4050 BASIC equivalents, you're right. We really did it this time!

FFTs in a Flash

Their speed and speed range is astounding. Installed in a 4051, the 4051R08 increases the speed of Fast Fourier Transform (FFT) operations by a factor of 7 over an equivalent 4050 BASIC program. On the 4051, FFT and related operations are slightly faster than a 4052 running equivalent BASIC programs. The already-fast 4052 and 4054 experience even greater enhancements of FFT calculation speeds due to their bit-slice architecture and micro-coded floating point operations. FFT or IFT transformation in a 4052 or 4054 equipped with a 4052R08 jumps to 22 times the speed of equivalent 4050 BASIC programs running on either system. And, when comparing the 4052/4052R08 system to a 4051 using the fastest FFT algorithm we could find in BASIC, the factor takes a quantum leap: over 100 to 1!

Table 1 illustrates speed increases when processing random arrays of data employing the new ROM Pack #2 FFT and IFT functions.

Table 1. Fourier Transform Computation Time

Function		Computation Time In Seconds			
		4051 In BASIC	4051 + 4051R08	4052 In BASIC	4052 + 4052R08
512 points	FFT	205	26	40	2
1024 points	FFT	446	65	88	4
512 points	IFT	207	27	42	2
1024 points	IFT	446	64	90	4

The 4050 BASIC FFT algorithm is extremely fast: it contains no sine or cosine calculations, but rather uses a stored table of sine values only. Most of the FFT, or IFT, calculation times were devoted to indexing through various arrays and subroutines of the very fast algorithm. Because of this, the factor of only approximately 5 for the 4052 versus the 4051 resulted.

Repertoire of Waveform Analysis Functions

The advanced signal processing and data analysis commands are:

"FFT" (Fast Fourier Transform)—transforms a one-dimensional array into frequency domain data consisting of an interleaved array of its cosine (real) and sine (imaginary) spectral components.

"IFT" (Inverse Fourier Transform)—converts an array of interleaved frequency components (spectral) into its equivalent time series.

"UNLEAV" (un-interleave)—separates the interleaved complex results of the "FFT" command into two separate arrays to simplify display or manipulation of frequency data.

"INLEAV" (Interleave)—prepares real (cosine) and imaginary (sine) spectra contained in separate arrays into a format compatible with the "IFT" function.

"POLAR" (Rectangular to Polar)—converts and separates the intermixed complex output of the "FFT" command into separate arrays containing magnitude (distance) and phase (bearing) data.

"TAPER" (Programmable Cosine Window)—multiplies any chosen one-dimensional array by a cosine window whose tapering width is programmable. When the width is selected to be 0.5 (50%), the well known Hanning Window results.

"CORR" (Correlation)—performs the cross-correlation of two data arrays. If an array is cross-correlated with itself, the autocorrelation is obtained.

"CONVL" (Convolution)—forms the convolution series from two arrays of data. This is the discrete time (sampled data) analogue of the convolution integral.

Some Working Environments

An engineer might easily put the new ROM pack's capabilities to work analyzing signals or waveforms for laser fusion studies, component testing, power supply design, or audio-through-radio frequency design and analysis. Source signals might be captured by a Digital Processing Oscilloscope, a 7912AD or other digitizers including GPIB-compatible multimeters.

Potential uses include a chemist's analysis of strip chart data in the areas of biomedical and chemical data analysis. Or general time series analysis, for autocorrelation or cross-correlation of financial, business, or other data. (It will be particularly useful in this area, since the autocorrelation is more revealing than the single coefficient of correlation statistic provided by most statistics packages.) And, in the area of general linear system modeling, the ROM Pack can be used for discrete time (sampled) systems using convolution of the model (time series) by the input in economics, operations research, physics, and engineering.

An FFT Application

Since the FFT transforms a signal from the time domain to the frequency domain, use this routine to study the frequency spectrum of a signal. For example, in harmonic distortion analysis, you compute the FFT of the signal, then derive the magnitude spectrum via the POLAR routine. Next, the amplitude of each harmonic is determined and the total harmonic distortion (THD) is computed. Measurement of total harmonic distortion is a common measurement in amplifiers and sound systems.

An IFT Application

The IFT command does the inverse operation of the FFT command. That is, it transforms a signal from the frequency domain to the time domain. When used in conjunction with the FFT command, IFT can be used to perform very selective digital filtering on a waveform sample. As an example, suppose that the magnitude spectrum of a signal has been obtained and that you wish to attenuate all harmonics of a signal to a certain specified level. This could be done by selectively multiplying all harmonics by a certain value. Following this procedure, you could then convert the signal back to the time domain by computing the IFT of the resulting spectrum.

A CORR Application

In a practical sense, correlation can be thought of as successively shifting (by some horizontal increment), multiplying, and integrating the two signals to be correlated. From a mathematical standpoint, correlation is achieved by computing the FFT of each signal to be correlated, then performing a complex conjugate multiplication on these FFTs, and finally taking the IFT of the resulting product. When the two signals being correlated are the same, the process is referred to as autocorrelation; when they are different, it is called cross-correlation.

A common application for autocorrelation is detecting the presence of signals buried in noise. When a noisy signal is autocorrelated, the result appears as a periodic waveform modulated with a triangular envelope. This technique of signal detection is used in biomedical studies, astronomy, tone control systems, and numerous other applications.

Cross-correlation is commonly applied in the detection and ranging of radar, sonar, and other pulsed waveforms, whereby the transmitted and received signals are cross-correlated to determine the delay (and hence distance) to a target. Cross-correlation is also useful in certain business applications for determining the association between sets of data (such as sales and pricing).

A CONVL Application

Like correlation, convolution can be thought of as successively shifting, multiplying, and integrating the two arrays (waveforms) to be convolved. However, in the case of convolution, one of the waveforms is reversed in time before performing the shifting-multiplication-integration process. Mathematically, convolution is performed by computing the FFT of each signal to be convolved, multiplying these two FFTs, and then computing the IFT of the resulting product.

A common engineering application for convolution is determining the output of a linear, time invariant system (such as a passive filter or network). Given the input signal, x(t), and the impulse response, h(t), the output y(t) can be predicted simply by convolving x(t) with h(t).

Getting Started

The SPS ROM Pack #2 command formats are shown in Figure 1.

```
CALL"FFT",a
CALL"IFT",a
CALL"POLAR",d,X,Y[,v]
CALL"CONVL",b,c,X
CALL"CORR",b,c,X
CALL"TAPER",a[,p]
CALL"INLEAV",d,e,X
CALL"UNLEAV",d,X,Y
```

Fig. 1. Signal Processing ROM Pack #2 (FFT) Command Formats.

The variable legend is shown below:

```
a   Vector array input/output
b,c Vector array inputs (Overwritten)
d,e Vector array inputs
X,Y Vector array outputs
v   Phase delay (Optional)
p   Taper fraction (.5 is 50% width
        tapering for the Hanning window)
```

The program contained in Figures 2 and 3 computes and displays the FFT of an array of data.

```
LIS100,195
100 N2=N/2+1
110 DIM X(N),A(N2),B(N2)
120 GOSUB 200
130 INPUT Z$
140 CALL "FFT",X
150 CALL "UNLEAV",X,A,B
160 GOSUB 300
170 INPUT Z$
180 CALL "POLAR",X,A,B
190 GOSUB 300
195 END
```

Fig. 2. FFT Main Program.

```
LIS200,600
200 REM DRAW THE N POINTS OF DATA IN FUNCTION FORM
210 CALL "MAX",X,Y2,I
220 CALL "MIN",X,Y1,I
230 WINDOW 1,N,Y1,Y2
240 PAGE
250 VIEWPORT 10,110,10,100
260 CALL "DISP",X
279 RETURN
300 REM THE UPPER GRAPH IS OF A(I), THE LOWER IS B(I)
305 PAGE
310 CALL "MAX",A,A2,I
320 CALL "MIN",A,A1,I
330 WINDOW 1,N2,A1 MIN 0,A2 MAX 0
340 VIEWPORT 10,110,50,90
350 FOR I=1 TO N2
360 MOVE I,0
370 DRAW I,A(I)
380 NEXT I
390 VIEWPORT 10,110,10,45
400 FOR I=1 TO N2
410 MOVE I,0
420 DRAW I,B(I)
430 NEXT I
440 RETURN
```

Fig. 3. Graphic Subroutines.

Lines 100 through 120 perform dimensioning and housekeeping.* The subroutine at line 200 graphs the waveform (Fig. 4.)

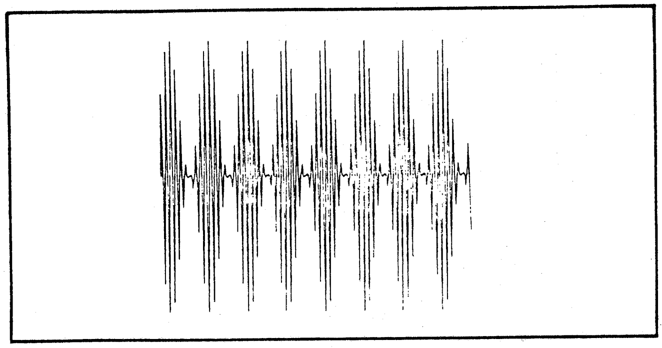


Fig. 4. A waveform to be transformed—256 points. This waveform was retrieved from a tape file and stored in array X.

Notice the calls to Signal Processing ROM Pack #1 in this graphic subroutine which extensively reduces the amount of coding and execution time.

Line 150 invokes the FFT. Line 160 separates the complex results (in array X) into separate real (A) imaginary (B) arrays. The subroutine at line 400 graphs the results in spectral form, shown in Figure 5.

A few more lines provide and graph the results in magnitude and phase (Polar) form as shown in Figure 6. Since the original waveform was a sine wave amplitude modulated by a sine function, the results are simple to interpret, and agree with the expected result. As anticipated, the frequencies consist of the center frequency (carrier) and sidebands resulting from the modulation process.

*Whenever FFT, IFT, convolution or correlation are used, array sizes must be a power of two, in keeping with the Sande-Tukey (decimation in frequency) algorithm the ROM Pack uses.

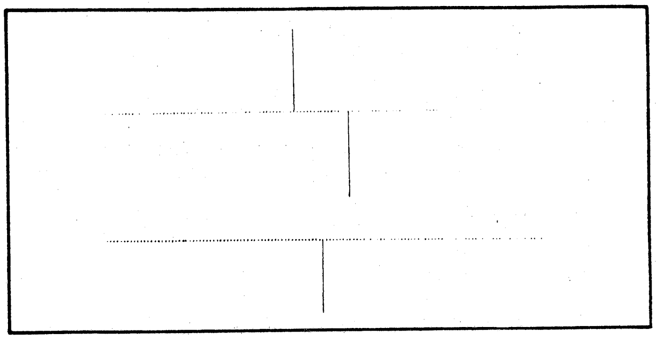


Fig. 5. Real (top) and Imaginary Transform Data for Figure 4.

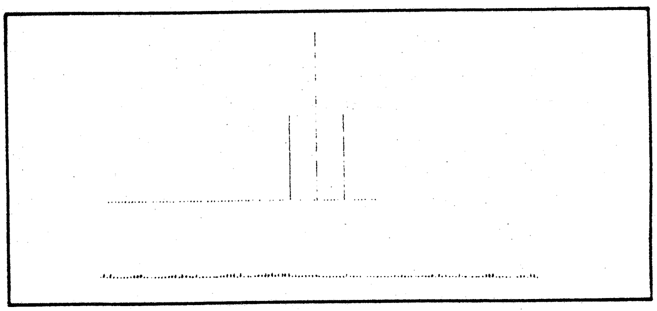


Fig. 6. Magnitude (top) and Phase Data—Figure 4 Transformed.

Although this example is a very simple one, it shows the simplicity of using the commands of the ROM Pack. Subsequent articles will illustrate the uses of other commands.

The transformation time for the waveform, including separation of results into separate arrays, took 12 seconds with a 4051 and 4051R08.

If you would like to get started using a 4051R08 (for the 4051) or 4052R08 (for the 4052 or 4054) Signal Processing ROM Pack #2 (FFT), contact your nearest Tektronix representative.

Manuals and Reference Cards Updated

4907 File Manager Operator's

The 4907 File Manager Operator's manual has been updated and arranged into a new easy-to-read format. It has been expanded to cover the operation of the File Manager with all of the 4050 Series.* A clear, concise General Operation section takes the user from hardware hookup through formatting a disc, through creating a file on that disc, to putting something into the file. It does it without overwhelming the user with details. "How To Write A File Identifier" is another section that has been simplified. The section on Command Descriptions has been revised into alphabetical order by command name rather than command function. More complete examples have been included. The part number for this February, 1979, version is 070-2380-01 and it's available through your local Tektronix Sales Engineer.

4907 File Manager Operator's Pocket Reference

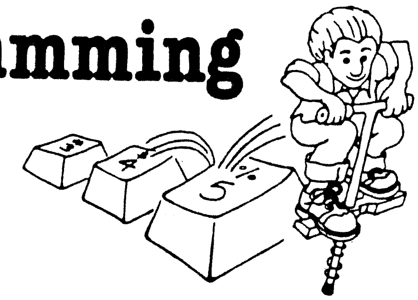
A handy, slim 20-page pocket-sized guide provides basic File Manager information and commands at your fingertips. It's also available through your Tektronix Sales Engineer under part number 070-2381-01.

4050 Series Basic Reference Guide

Another pocket-size guide gives you instant access to BASIC keywords and commands for all of the 4050 Series. Included within this little booklet also are GPIB addresses, the ASCII code chart, PRINT format operators and other operational information. Your Sales Engineer can order it for you under part number 070-2142-01.

*The main difference in operating the 4907 File Manager with the 4052/4054 units is the ROM Pack. Although it looks like the 4051 File Manager ROM Pack, the 4052/4054 ROM Pack is not interchangeable with the 4051. When a File Manager is ordered or upgraded for use with the 4052/4054 units, Option 40 must be specified, which will provide the proper ROM Pack for your 4907.

Programming Tips



Early Detection of Tape Wear with 4924 Tape Drive

by Jack Callahan
Honeywell Electro-Optics Center
Lexington, MA

The 4924 Tape Drive READ ERRORS command has helped us reduce the loss of data caused by tape wear. Appendix B of the 4924 Service Manual (070-2131-00) describes this command.

Each time the 4924 reads a physical record from the tape, it computes a "checksum" from the first 256 bytes of the record and compares this checksum with the final "checksum byte". If they do not agree, a "read error" has been detected; the 4924 reverses the tape and tries again to read the record. The 4924 keeps track of these reads following checksum failures, but the user is only aware of them when a program aborts following 10 unsuccessful reads.

When the READ ERRORS command

INPUT @D,24:E (D=device#)

is sent from the 4051 to the 4924, the 4924 replies with a count of reads following checksum failures that have occurred since power-on or since the last interrogation. Thus, by introducing the following statements at the end of file read routines from the 4924, checksum failures that indicate possible tape wear can be determined before they reach the fatal 10.

```
1000 INPUT @D,24:E
1010 IF E=0 THEN 1040
1020 PRINT E;" Re-reads were necessary to recover this file"
1030 PRINT "Tape replacement may be necessary"
1040 REM Continuation of program
```

String Sort

by J. J. Brown
Boart Hardmetals (Europe) Ltd.
Shannon Airport, Co. Clare, Ireland

This sorting routine alphabetizes words or character strings that are input in any sequence. The sorted list may be displayed on the screen, or it may be sent to the printer through User Definable Keys 6 and 7. User Definable Keys 8 and 9 repeat the sorted list. The number of memory bytes required is equal to the length of the longest word (or string) multiplied by the number of words (or strings), plus 1700 bytes for the routine.

```
1 PAGE
24 T=32
25 GO TO 100
28 T=41
29 GO TO 100
32 T=32
33 GO TO 550
36 T=41
37 GO TO 480
100 PRINT "                    WORD OR CHARACTER STRING SORTING"
110 PRINT "                    ---------------------------------"
120 PRINT "JENTER THE NUMBER OF CHARACTERS IN THE LONGEST WORD/STRING"
130 INPUT L
140 PRINT "ENTER THE NUMBER OF WORDS/STRINGS TO BE SORTED"
150 INPUT N
160 M=L*N
170 DELETE A$
180 REM "A$ DIMENSIONED TO SUIT NO OF WORDS AND MAX WORD LENGTH"
190 DIM A$(M)
200 A$=""
210 PRINT "ENTER WORDS/STRINGS"
220 FOR I=1 TO N
230 GO TO 260
240 PRINT "WORD/STRING IS LONGER THAN ORIGINALLY STATED"
250 PRINT "RE-ENTER OR RERUN"
260 PRINT I;"   ";
270 INPUT B$
280 IF LEN(B$)>L THEN 240
290 IF LEN(B$)=L THEN 320
300 B$=B$&" "
310 GO TO 290
320 A$=A$&B$
330 NEXT I
340 S=0
350 M1=M-L
360 FOR I=1 TO M1 STEP L
370 I2=I+L
380 C$=SEG(A$,I,L)
390 D$=SEG(A$,I2,L)
400 IF C$<=D$ THEN 450
410 Z$=SEG(A$,I,L)
420 A$=REP(D$,I,L)
430 A$=REP(Z$,I2,L)
440 S=1
450 NEXT I
460 IF S=1 THEN 340
470 IF T=32 THEN 550
430 PRINT @T:"J"
490 FOR I=1 TO M STEP L
500 E$=SEG(A$,I,L)
510 PRINT @T:E$,"J"
520 NEXT I
530 GO TO 600
540 REM ----- PRINT SORTED LIST ON SCREEN -----
550 PAGE
560 FOR I=1 TO M STEP L
570 E$=SEG(A$,I,L)
580 PRINT E$
590 NEXT I
600 END
```

Listing a Program Over the RS-232-C

by Wayne Miller
Missouri Center for Health Statistics
Jefferson City, MO

When you list a program in 4051 memory to a printer using the Option 10 Printer Interface, i.e., LIS @41:, the interface automatically intercepts control characters within strings or REM statements and converts them to their alphabetic equivalents, backspaces the printer, and prints the underlines. When the printer buffer fills, the Option 10 halts transmission until more room is available.

However, if you list a program to the printer using the RS-232-C (Option 1), i.e., LIS@40:, the printer executes any control characters, regardless of their placement. Also, the RS-232-C Interface is unaware of when the printer buffer is filled and may overwrite it. The following routine provides a quick way to overcome these limitations when transmitting directly to a printer over the RS-232-C Interface.

The program to be listed must be in ASCII format (here it's on the 4907 disc) which this routine brings into 4051 memory one line at a time. Lines 4150 through 4180 first check for a literal string or REM statement since control characters to be printed will only be found in one of these. Next, rather than testing each string or REM segment against a list of control characters, it simply looks for an ASCII value greater than 31. If the segment is a control character (ASCII value 31 or less), statement 4230 converts it to an "up arrow". If not, it passes to the next character. A timed delay, included at statements 4260—4270, ensures room in the printer buffer before the next statement is transmitted.

```
4000 PRINT "Enter the File Name to be Printed"
4010 INPUT F$
4020 PRINT "Enter the device: 40=PRINTER, 32=CRT"
4030 INPUT D
4040 CLOSE
4050 CALL "MOUNT",0,A$
4060 UNIT 0
4070 X=0
4080 OPEN F$;1,"R",M$
4090 ON EOF (1) THEN 4270
4100 U$=CHR(31)
4110 INPUT #1:X$
4120 X=POS(X$,"""",6)
4130 IF X>0 THEN 4160
4140 X=POS(X$,"REM",3)
4150 IF X=0 THEN 4220
4160 X1=LEN(X$)
4170 FOR I=X TO X1
4180 A$=SEG(X$,I,1)
4190 IF A$>U$ THEN 4210
4200 X$=REP("↑",I,1)
4210 NEXT I
4220 PRINT @D:X$
4230 FOR I=1 TO LEN(X$)*0.7
4240 NEXT I
4250 STOP
4260 GO TO 4110
4270 CLOSE
4280 END
```

Editor's Note: The PLOT 50 4050A08 Utility Programs package contains a program which emulates the Option 10 Interface, as well as providing a formatted listing.

Don't Abort SAVE To Disc

by John Carter
Tektronix, Inc.
Santa Clara Annex

Did you ever want to abort a SAVE to the 4907 disc, especially when it's a long one? Well, don't! The only safe way to terminate a SAVE is to let it happen naturally. Any attempt to abort a SAVE by pressing the 4051 BREAK key or by opening the 4907 disc drive door may interrupt a write to the directory and leave the directory chain broken. And you can't MOUNT a disc if the directory is damaged.

4051 Data Entry Routine

by Raymond DeMers
Tektronix, Inc.
Rochester, NY

This sample program demonstrates how a Data Entry routine may be programmed on the 4051. This approach uses the POINTER command to permit the asterisk key to emulate the ENTER key found to the right of many 10-key pads. It overcomes the need to use the 4051 RETURN key which is awkward for trained key operators.

The POINTER command in statement 200 returns the X and Y location of the arrow on the screen, as well as the character entered in Z$. Here, only Z$ is used. Statement 210 tests the ASCII value of Z$ to restrict entries to numbers and necessary control characters. An array is developed as the data is entered; after the last entry, the routine displays the data.

```
100 INIT
110 DELETE A,B
120 DIM A(100)
130 N$=""
140 S=1
150 E=0
160 PRINT "LJI  **** 4051 DATA ENTRY PROGRAM ****"
170 PRINT "JJJ 1. KEY IN SIGN (+ DEFAULT) THEN VALUE AND * TO ENTER."
180 PRINT "JJ 2. AFTER LAST *, USE / TO END AND DISPLAY ARRAY."
190 REM  -----  ASK FOR DATA ENTRY FROM SELECTED KEYS  -----
200 POINTER X,Y,Z$
210 IF ASC(Z$)<42 OR ASC(Z$)>57 THEN 200
220 GO TO ASC(Z$)-41 OF 510,200,200,310,410,570
230 GO TO 400
300 REM  -----  SET SIGN FOR NEGATIVE VALUE  ( - ADE 45)  -----
310 S=-1
320 GO TO 200
400 REM -----  INPUT IS NUMERIC  -----
410 N$=N$&Z$
420 GO TO 200
500 REM  -----  DATA ITEM COMPLETE  -----
510 E=E+1
520 A(E)=VAL(N$)*S
530 S=1
540 N$=""
550 PRINT "G"
560 GO TO 200
570 DIM B(E)
580 FOR K=1 TO E
590 B(K)=A(K)
600 NEXT K
610 PRINT "G                                              G"
620 PRINT "L ";E;" ENTRIES ARE:"
630 PRINT B
640 END
```

With the ENTER key problem solved, the excellent string and array handling capabilities of the 4051 permit you to develop a Data Entry System which could include such features as:

- Range Checking
- Check-digit Calculations
- Zero-fill
- Justification
- Verification
- Must-fill Field Checking
- Testing for Alpha or Numeric

Coordinate Transformation

by Stan Jensen and Ed Yotter

California Air Resources Board
Sacramento, CA

The Air Resources Board (ARB) in cooperation with the Department of Transportation (CALTRANS) develops and operates computer models of air pollutant emissions from motor vehicles. Part of the input data for these models is the geographic description of traffic analysis zones (TAZs), which are analogous to census tracts.

The task of the Air Resources Board's Planning Division staff is to digitize the boundaries of the zones, which have been drafted on USGS 7½′ maps, and record the Universal Transverse Mercator (UTM) coordinates. The UTM coordinate system was selected because it is a rectangular Cartesian system which uses metric units, and because it is also used in other ARB projects. The 7½′ map series was selected for its statewide coverage, convenient scale (1:24000) and the UTM reference ("tick") marks on the map edges.

The design of the coordinate transformation procedure had three major requirements. Since several hundred maps were to be digitized, the transformation control points had to be standard features on each map sheet; they could not require manual drafting or visual interpolation. Secondly, since the Tektronix digitizing system used a 4956 Graphics Tablet (20″ X 20″ surface), and, since the maps are about 19″ X 23″, the control points must lie on the tablet regardless of the map's orientation or positioning. Finally, the transformation equation should include terms for rotation, scaling and translation, should be accurate, and should still manage to transform coordinates quickly.

The program listing and sample run show how all of these requirements have been met. The map is positioned on the graphic tablet so the left edge and right edge tick marks for any Y value (3767000 in the example) are simultaneously on the tablet. The map is then shifted up or down until the top or bottom edge is also on the tablet. The program is RUN, the operator digitizes the left tick mark (point #1), the right tick mark (point #2) and the top or bottom tick mark (point #3). The UTM X coordinate for point #3 and the UTM Y coordinate for points #1 and #2 are then input to the Graphic System. The program determines the equations for two lines parallel to the UTM axes and passing through the three control points, solves the equations simultaneously to determine the digitizer coordinates X(4), Y(4) of an interior point and uses this point with the known map scale (1:24000) to calculate the coefficients of the transformation equation.

This transformation procedure is easy and quick to use, accurate to within one digitizer unit anywhere on the tablet and will transform points at a rate of about 12 points per second.

```
100 REM DIGI/TRANSFORM 11-APR-79
110 INIT
120 DIM X(4),Y(4)
130 G=8
140 PRINT "LDigitize point # 1, 2 and 3"
150 FOR P=1 TO 3
160 INPUT @G:X(P),Y(P),Z$
170 PRINT "G";
180 NEXT P
190 PRINT "Enter (in UTM meters) the X-coordinate of point # 3"
200 PRINT "and the Y-coordinate of point # 1 and # 2"
210 INPUT U1,V1
220 M1=(Y(2)-Y(1))/(X(2)-X(1))
230 B1=Y(1)-X(1)*M1
240 M2=-(1/M1)
250 B2=Y(3)-X(3)*M2
260 X(4)=(B2-B1)/(M1-M2)
270 Y(4)=M1*X(4)+B1
280 H1=SQR((X(2)-X(1))↑2+(Y(2)-Y(1))↑2)
290 C1=(X(2)-X(1))/H1
300 S1=(Y(2)-Y(1))/H1
310 S=24000/(200*39.37)
320 A=S*C1
330 B=S*S1
340 C=U1-A*X(4)-B*Y(4)
350 D=V1-A*Y(4)+B*X(4)
360 PRINT "_A=";A,"_B=";B,"_C=";C,"_D=";D
370 PRINT "JDigitize points to be transformed.J"
380 INPUT @G:X1,Y1,Z$
390 PRINT "GDigitizer coordinates : ";X1;",";Y1;"I";
400 U1=A*X1+B*Y1+C
410 V1=A*Y1-B*X1+D
420 PRINT "UTM coordinates : ";INT(U1+0.5);",";INT(V1+0.5)
430 GO TO 380
```

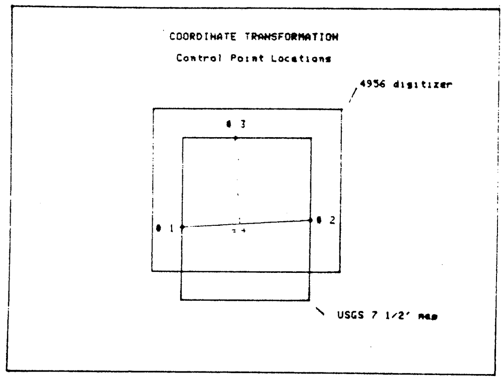
```
100 REM DIGI/TRANSFORM 11-APR-79 by Stan Jensen / CARB
110 REM Following REMarks keyed to program listing wherever possible.
120 REM   Vectors for digitizer coordinates
130 REM   GPIB address for 4956 digitizer (20" x 20")
140 REM   Control points :
150 REM    # 1 : any UTM tick mark on left map edge
160 REM    # 2 : tick mark on right edge with same UTM Y as # 1
170 REM    # 3 : any UTM tick mark on top or bottom edge
180 REM   Line # 170 signals operator
190 REM   The three control points provide one UTM X and Y :
200 REM    # 3 provides the UTM X
210 REM    # 1 and # 2 provide the UTM Y
220 REM   Slope of the line between points 1 and 2
230 REM   Y-intercept of Line 1-2
240 REM   Slope of a line ("Line 3-4") perpendicular to Line 1-2
250 REM   Y-intercept of Line 3-4
260 REM   Digitizer coordinates of the point ("# 4") with the
270 REM   UTM coordinates entered for the three control points
280 REM   Hypotenuse of a right-triangle between points 1 and 2
290 REM   Cosine of rotation from digitizer to UTM
300 REM   Sine of rotation from digitizer to UTM
310 REM * "SCALE" : converts digitizer units to UTM meters @ 1:24000
320 REM   A-term (rotation/scaling)
330 REM   B-term (rotation/scaling)
340 REM   C-term (rotation/scaling/translation)
350 REM   D-term (rotation/scaling/translation)
360 REM   Display coefficients of digitizer-to-UTM transformation
370 REM   Start of endless loop to transform points
380 REM   Digitize a point
390 REM   Signal input and display digitizer units
400 REM   Transform digitizer X and Y to UTM X
410 REM   Transform digitizer X and Y to UTM Y
420 REM   Display UTM meters (accuracy = ~3 meters @ 1:24000)
430 REM   Repeat ad nauseum


Digitize point # 1, 2 and 3
Enter (in UTM meters) the X-coordinate of point # 3
and the Y-coordinate of point # 1 and # 2
469000,3767000

A=2.92019383149
B=-0.0733894594 32
C=467888.684211
D=3757562.59424

Digitize points to be transformed.

Digitizer coordinates : 357,3124    UTM coordinates : 465403,3766997
Digitizer coordinates : 3968,2045   UTM coordinates : 476890,3767000
Digitizer coordinates : 1020,1221   UTM coordinates : 469001,3762019
```



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 for a small recording fee per program plus the charge for the tape cartridge. One tape will hold several programs. (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*

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. 2618.

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 NUMBER: 51/00-5901/0

Title: **Spherical Polygon Area On Earth**
Author: A. Jon Kimerling
Oregon State University
Corvallis, OR
Memory Requirement: 16K
Peripherals: Optional—4631 Hard Copy Unit
Statements: 105

The program calculates areas of spherical polygons in sq. km. (i.e., State of Oregon) from latitude/longitude values along their perimeter. Latitude longitude values are consecutively entered from tape or the keyboard. The coordinate method of area calculation used is based on adding or subtracting spherical triangles, with 90°N a common vertex for all triangles. Napier's Rules are employed to find values of angles in each spherical triangle, values that are used to compute spherical excess and then triangle area. These areas are then either added or subtracted from the cumulative area for the polygon depending on the difference in longitude between the two points which, with 90°N, form each spherical triangle.

```
SPHERICAL POLYGON AREA PROGRAM

ARE LAT./LONG. VALUES ENTERED FROM
TAPE CASETTE OR THE KEYBOARD? (T OR K)  K

ENTER LATITUDE AND LONGITUDE OF BOUNDING POINTS
USE -LAT FOR S.LAT., AND -LONG FOR W.LONG
CONVERT DEG.,MIN.,SEC. TO DECIMAL DEGREES
ENTER 999,999 WHEN THROUGH

ENTER PT. # 1  89,0
 89 0
CORRECT?(Y OR N) Y
ENTER PT. # 2  89,1
 89 1
CORRECT?(Y OR N) Y
ENTER PT. # 3  90,1
 90 1
CORRECT?(Y OR N) Y
ENTER PT. # 4  90,0
 90 0
CORRECT?(Y OR N) Y
ENTER PT. # 5  999,999
 999 999
CORRECT?(Y OR N) Y

TOTAL PTS. = 4

AREA = 107.413952105 SQ.KM.

CALCULATE ANOTHER AREA? (Y OR N) N

GOOD BYE
```

ABSTRACT NUMBER: 51/00-6005/0

Title: **TEKNIQUES**
Author: Captain S. K. Sanford
Aberdeen Proving Ground, MD
Memory Requirement: 24K
Peripherals: (2) 4924 Digital Cartridge Tape Drives
Optional—4631 Hard Copy Unit
—4641 Printer
Statements: 800

The program prompts for documentation fields shown on the standard Tektronix documentation form, stores variable data, produces 4631 Hard Copy or 4641 Printer copy. The document may be updated or displayed at a later time from the work tape.

```
                 *** TEKTRONIX PROGRAM DOCUMENTOR ***

   OPTIONS
   *******
                 1             DIRECTORY
                 2             GENERATE NEW DOCUMENT
                 3             EDIT OLD DOCUMENT
                 4             DISPLAY (CRT) DOCUMENT
                 5             PRINT DOCUMENT

                 10            QUIT
                 ********************************
                 * DOCUMENT ON FILE 1 ON UNIT 2 *
                 * WORKSPACE ON FILE 1 ON UNIT 3 *
   ↓ SELECT KEY  ********************************

TEKTRONIX                                                 4051
                                     APPLICATIONS LIBRARY PROGRAM
=================================================================
TITLE                               |ABSTRACT NUMBER
                                    |
TEKNIQUES                           |
------------------+-----------------+----------------------------
ORIGINAL DATE     |REVISION DATE    |MEMORY REQUIREMENT
AUGUST 1978       |OCTOBER 1978     |24K
------------------+-----------------+----------------------------
AUTHOR                              |PERIPHERALS
S.K. SANFORD, CAPTAIN, USA          |(2) 4924
ARTERIEL TESTING DIRECTORATE        |LINE PRINTER (RS-232)
ABERDEEN PROVING GROUND             |DATA COMMUNICATIONS INTERFACE
APG, MD  21005  (301-278-4596)      |
=================================================================
ABSTRACT
DOCUMENT 4051 PROGRAMS IN STANDARD TEKTRONIX FORMAT.
```

ABSTRACT NUMBER: 51/00-8028/0

Title: **Change and List Program Variables**
Author: S. Schicktanz
Technical University
Munich, Germany
Memory Requirement: 8K
Peripherals: Optional—4631 Hard Copy Unit
Statements: 147

The program allows listing or changing the names of the variables of an ASCII program from tape. Listing the program is also available. When changing variable names, input is tested for validity and correct type; errors are indicated by an appropriate message. The changed program can then be output to the original tape file.

The program can be used either with the menu or the User-Definable Keys. The user is prompted for necessary input information by use of the POInter-statement. It is not necessary to terminate input with the RETURN key.

```
LIST OF VARIABLES:

A,
A$,
B,
B$,
C,
C$,
D, D0, D1, D2,
E, E0,
F, F1,
G, G1,
H,
I,
K,
L, L0,
M, M0,
N,
N$,
O,
P,
Q,
R, R0,
S,
T,
U,
Y,
Z1,

O-output new program
N-new start
V-list variables
P-  "  program
C-change variables

Variable to be changed:  I; New name: K0
Variable to be changed:  6
invalid name! : C; New name: C1
Variable to be changed:  D4; New name: F$: wrong type!
Variable to be changed:  D$; New name: F$
Variable to be changed:  H$; New name: PR
invalid name! : P$
Variable to be changed:
```

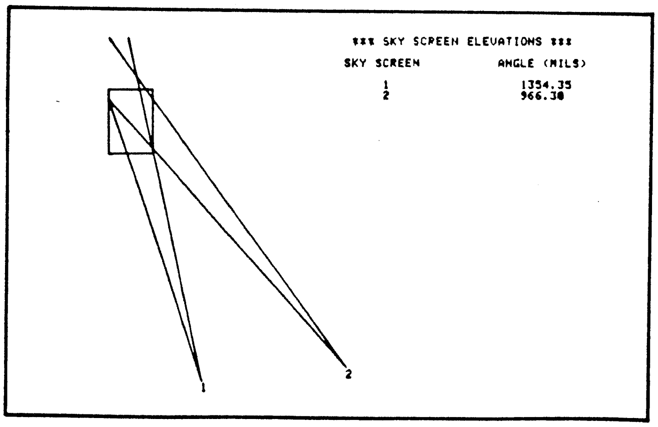
ABSTRACT NUMBER: 51/00-9535/0

Title: **Sky Screen**
Author: Captain S. K. Sanford
Aberdeen Proving Ground, MD
Memory Requirement: 8K
Peripherals: Optional—4631 Hard Copy Unit
Statements: 103

The program computes the elevation angle (in mils) for a number of sky screen devices to provide approximately maximum internal target window area coverage with no internal dead spaces.

The aiming angle in mils of each sky screen is displayed, as well as a drawing of the sky screens (numbered points) as they will acquire the target window (box).

The program is tutorial from tape.

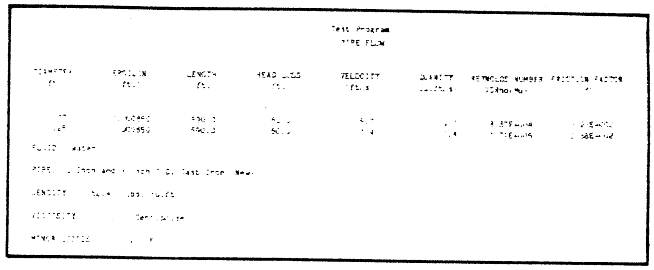



```
*** SKY SCREEN: TARGET WINDOW ACQUISITION ***
AUTOCOPY? (Y/N): Y
(TYPE  0  TO QUIT)
ENTER NUMBER OF SKY SCREENS: 2
ENTER WINDOW WIDTH, HEIGHT, AND CENTER ALTITUDE: 300 500 2000
ENTER DISPLACEMENT, ALTITUDE, AND FAN ANGLE FOR
   SKY SCREENS FROM CLOSEST TO FARTHEST:
FOR SKY SCREEN 1: 500 0 100
FOR SKY SCREEN 2: 1500 100 100
```

ABSTRACT NUMBER: 51/00-1201/0

Title: **Pipe Flow**
Author: H. E. Sherer
Mining Services, Inc.
Tuscaloosa, AL 35404
Memory Requirement: 16K
Peripherals: 4641 Printer
Statements: 174

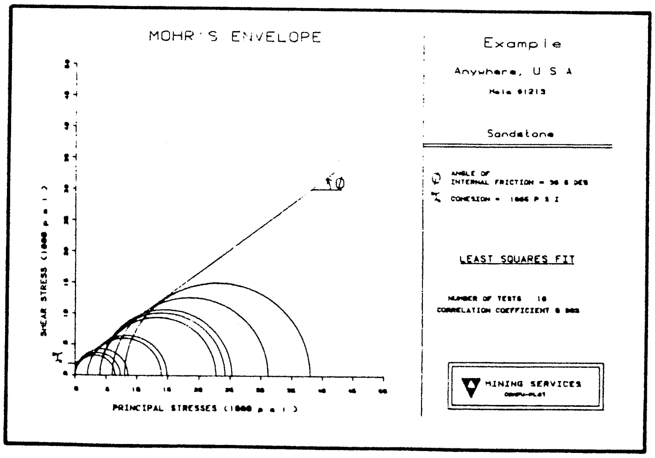
The program uses the 'Colebrook' Formulas of turbulant pipe flow to solve for (1) Head Loss, (2) Flow or, (3) Pipe Diameter for fluids flowing in pipes. Minor losses may be taken into account, along with various fluid parameters.



ABSTRACT NUMBER: 51/00-5301/0

Title: **Solution and Plot of Mohr's Envelope**
Author: H. E. Sherer
Mining Services, Inc.
Tuscaloosa, AL 35404
Memory Requirement: 32K
Peripherals: 4641 Printer
4662 Plotter
Statements: 495

The program uses an iterative search and least squares fit to solve Mohr's envelope. This program will accommodate up to 300 uniaxial and triaxial tests, will compute the correlation coefficient, and plot the results.



Tektronix
COMMITTED TO EXCELLENCE

TEKTRONIX, INC.
Information Display Group
Applications Library
Group 451
P.O. Box 500
Beaverton, Oregon 97005

STELLA MCMORRAN
60-307

IF YOU HAVE MOVED
PLEASE CALL 3618

ADDRESS CORRECTION REQUESTED

4050 Series Applications Libraries

Africa, Europe, Middle East

4050 Series Applications Library
Tektronix International, Inc.
European Marketing Centre
P.O. Box 827
1180 AV Amstelveen
The Netherlands

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)

Ms. Bev Brandon, 73-312
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