Tekniques

The 4051 Applications Library Newsletter Vol. 2 No. 1

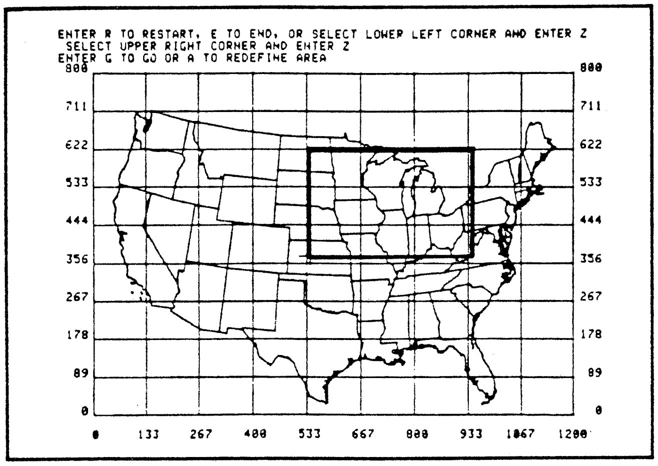
4907: Segmented Data Base Provides Fast Graphics Access

By Les Brabetz

(This is the second of three articles on using the new 4907 FILE MANAGER flexible disc mass storage unit as a powerful graphics aid.)

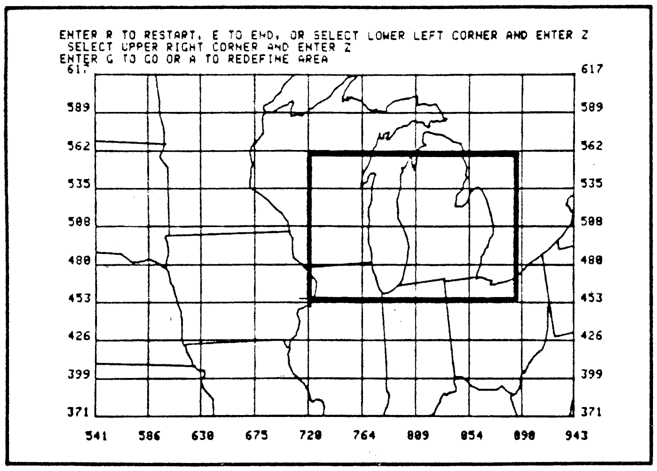
The article on segmented data bases in the previous issue of TEKniques (Vol 1 No. 10), described the use of a segmented graphics data base. The techniques discussed permit rapid display of portions of the data base rather than displaying the entire data base to view a smaller portion. Only those files (or segments of the data base) that are required to complete the display are read into memory and displayed on the screen. The ability of the 4907 FILE MANAGER to randomly access files allows the effective use of this data base technique. A file is created for each segment of a coordinate grid that overlays the entire data base. A rectangular grid was defined which varied in segment size to provide a reasonably equal density of vectors per segment. The segments generated with this grid defined some empty files, but was a much better distribution of vectors than a uniform grid.

This article describes construction of the segment files from the master data base file. Once the grid coordinates



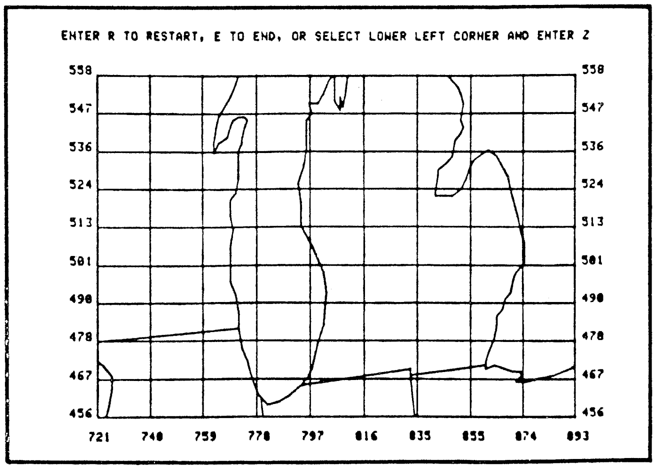
```
ENTER R TO RESTART, E TO END, OR SELECT LOWER LEFT CORNER AND ENTER Z
 SELECT UPPER RIGHT CORNER AND ENTER Z
ENTER G TO GO OR A TO REDEFINE AREA
```

Fig. 1. Default map of the United States with area selected for "zooming" (heavy rectangle).



```
ENTER R TO RESTART, E TO END, OR SELECT LOWER LEFT CORNER AND ENTER Z
 SELECT UPPER RIGHT CORNER AND ENTER Z
ENTER G TO GO OR A TO REDEFINE AREA
```

Fig. 2. Map segment selected in Fig. 1 as displayed from segmented data base. Second "zoom" requested of Lake Michigan (heavy rectangle).



```
ENTER R TO RESTART, E TO END, OR SELECT LOWER LEFT CORNER AND ENTER Z
```

Fig. 3. Lake Michigan area as requested in Fig. 2 and displayed from segmented data base.

In This Issue

are defined, there are two possible ways of building the segment files. The first method is to digitize a portion of the data base into each segment file. Any vector which crosses the segment boundaries must end exactly on the boundary, and must end at the correct angle to intercept the vector approaching from the adjacent segment; this is quite difficult. The main requirement of the segmented data base is to provide vector continuity across segment boundaries. The second method, which was used to build the demonstration data base of the U.S. Map, is to take a large data base and select vectors to transfer to separate files. The problem of boundary vector continuity is overcome by calculating an intercept point on the boundary, and adding this point to the segment file.

The original data base of the U.S. Map consists of approximately 3000 X-Y coordinate pairs. Building a segment file requires examining each point and determining which one of the following four conditions fits that point:

1. The new point and the last point are within the defined area.
2. The new point and the last point are outside the defined area.
3. The point is entering or exiting the area relative to the last point.
4. The last point was outside and the new point is outside, but the vector drawn between the points passes through the segment area.

Conditions one and two are the simplest case as the point is either stored in the segment file or discarded. Condition three requires the slope and direction of the vector to be calculated. Once these are known, the boundary intercept point may be determined. Condition four is the most elaborate solution as it requires determining the points of boundary intersection for both entrance and exit.

A rectangular definition of the segment was chosen to ease the calculation of the boundary intercept point. Once direction and slope of the vector is determined, one boundary coordinate is calculated and stored.

Determining the Boundary Intersection

Refer to the example in Fig. 4. Point P2 exceeds Y3, the maximum value of the Y range of the segment definition. A set of coordinate axes are placed on the point P1 and the quadrant of operation determined by the point relationship of P1 and P2. For this example, quadrant 1 is defined. The slope of the vector connecting P1 and P2 is used to determine which boundary will be intercepted.

If the slope of the vector connecting P1 and P2 is greater than the slope of the vector connecting P1 and P4, Y3 is used for a known boundary value. If the P1 to P2 slope were less than the P1 to P4 slope, then X4 would be used for the boundary calculation. X3 is calculated from the slope and the Y3 intercept value:

$$
\begin{aligned}
X3 &= \text{slope} * \text{distance} + \text{origin}\\
&= ((Y2 - Y1)/(X2-X1) * (Y3 - Y1)) + X1
\end{aligned}
$$

The boundary point P3 is added to the array and output to the segment file. Once the slope of the vector is determined and the boundary value known, the intercept coorinates are easily determined.

The coordinate pairs are read from the master data base file in the form of an array, and are output to the segment file in the same fashion. The output array is stored in memory until the segment boundary is crossed or until reading of the input array has been completed. If the entry array of vectors is outside of the defined area, no transfer occurs between the input and output files. The coordinate transfer begins when a point is inside the area and ends when the array point crosses the boundary.

After the first attempts were made at using this file transfer technique, two enhancements were formulated. The first was to generate four quadrant files from the main data base and then to use them for the segment file creation. This reduced the amount of time to create the segments by reducing the amount of data to scan and test against the segment area. The second enhancement was to

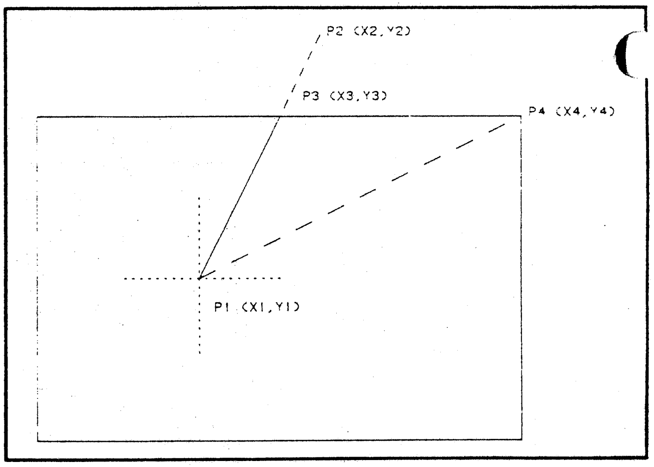


Fig. 4. Example of boundary intercept calculation for condition 3, vector exiting boundary area.

TEKniques, the 4051 Applications Library Newsletter, is published by the Information Display Group of Tektronix, Inc., Group 451, P.O. Box 500, Beaverton, Oregon 97077. It is distributed to TEKTRONIX 4051 users and members of the 4051 Application Library.

Publisher	Ken Cramer
Managing Editor	Patricia Kelley
Editor	Terence Davis
Graphic Designer	Mark Woods
Circulation	Rory Gugliotta

Volume 2 Number 1. Copyright 1978 Tektronix, Inc. Printed in the United States of America. All Rights Reserved.

attempt an equal distribution of vectors among many small segments. Careful definition of the grid for the segments is important in both creation time and display time.

Approximately four hours were required to generate the segmented data base. No operator intervention is required as input files are selected and output files defined by the segment definition program. Once created, segments can be chosen and displayed in seconds. The program used for windowing will be documented and placed in the 4051 Applications Library.

The next article regarding the segmented data base will describe the selection of the segment files for building displays.

4051 Entertains at Luncheon

Computers can be fun too! Often the strictly practical aspects of computing overpower the notion that computers can serve other purposes as well. The 4051 brought just such an idea home to the University of Wisconsin-Stevens Point Associates, a group of business, industry and university people. This group holds monthly meetings to keep communications open between industry and the Univsersity. Students get to meet potential employers, while the business community gets to see "what's new in computing technology." During two recent luncheon meetings, Dr. P.C. Holman of the University staff toured the dining room with a 4051 and a 4631 Hard Copy Unit, presenting attendees with their own personalized biorhythm chart.

Pre-luncheon publicity made reference to an article about biorhythms in the *Wall Street Journal* (Aug. 2, 1977). The article described the current resurgence of interest in biorhythm charts, a "pseudo-science" with roots in the late 19th century. At that time, Dr. Hermann Swoboda in Vienna and Dr. Wilhelm Fleiss in Berlin concluded that the physical and emotional states of human beings changed rhythmically. Alfred Teltscher, an Austrian engineer, later claimed that there is also a rhythmic change in human intellectual functioning.

Current theory is that the three cycles begin at birth and continue without change throughout a lifetime. The physical cycle is said to last 23 days, the emotional cycle 28 days, and the intellectual cycle 33 days. While there is still controversy over the validity of biorhythms, the *Wall Street Journal* pointed out, interest is booming. This is evidenced by growing sales of books and mail-order biorhythm charts.

Dr. Holman used a program that creates a set of biorhythm cycles based, as biorhythms are, on an individual's birthdate. With this program in his 4051, Dr. Holman obtained birthdates from interested attendees and let the 4051 compute and graph the individual's biorhythm cycles. A hard copy was then obtained for each individual. This event brought a broad spectrum of university and business people an experience with 4051 computing ease. It was a friendly and personal experience at computing.

While originally intended as a one-time occurrence, reaction and interest from the audience generated a repeat performance at the next monthly luncheon. There, biorhythms were accompanied with a bit of 4051-generated "space age art" as well. Audience interest remained high, and a lot of fun was had by all in this computerized "academic type of floor show."

* Editor's Note

Contest Reminder

The last issue of TEKniques announced a computer-aided design contest. Using the 4051 the program may design circuits, ship hulls, mechanical parts—anything!

Programs must be submitted on a tape cartridge, and must be accompanied by program documentation and a submittal form, and an order form for your exchange programs. Each entry will receive the usual new tape with three programs of your choice, so you can't lose. More information can be found in the last issue of TEKniques (Vol 1 No. 10).

Deadline for entries is March 31, 1978. Send your entries to: 4051 Applications Library, Tektronix, Inc., Group 451, P.O. Box 500, Beaverton, OR 97077.

Three Days to Three Minutes

A recent contribution to the 4051 Applications Library is the Lighting Intensity Program. The program was written by Florent van Vlasselaer of Tektronix Belgium, for the Europe Lighting Ltd. firm. According to the International Librarian, this program solves a problem in three minutes that took three days to do manually. See Abstract Number 51/00-3301/0.

Fast, Graphic Solution to Fourier Regressions

by Gary P. Laroff

(This is the fourth in a series of articles on applications of the Tektronix software package 4050A10 PLOT 50 Statistics Vol. 4.)

The first three articles in this series described using the 4051 to analyze data described by nonlinear functions. These articles discussed nonlinear least squares, nonlinear least squares with function minimization, and solutions to nonlinear systems of equations. In this issue we will consider those data sets that can be described by a series of sine and cosine functions.

Many natural phenomena, such as acoustical, optical and electronic phenomena, are of a periodic character. Music, for instance, is composed of regular oscillations, partly a fundamental tone with a certain frequency η, and partly overtones with frequencies 2η, 3η, 4η, ... The ratio of the strength of the fundamental tone to that of the overtones is the determinant for our impression of the sound. In a loudspeaker, variations of electrical current are converted into variations in air pressure which (assuming an ideal speaker) are described by the formula:

$$y = r \sin (wt + v)$$

where y is the observed oscillation; r is the *amplitude* of the oscillation; w is the angular frequency, and is equal to 2π times the frequency, and v is a constant which defines the state of the time $t = 0$.[1]

The separation of periodic signals into a fundamental tone and overtones (harmonics) is present not only in acoustics, but also permeates many other areas. According to Fourier, every function with period $2\pi/w$ can, in general, be described by series of sines and cosines.

Program 8 of Statistics, Vol 4, Fourier Regressions, can solve for the appropriate Fourier series that fits a particular data set. The program employs two fitting techniques for x,y data pairs, a least squares fit for any x,y data set and a discrete Fourier series calculation if the data x values are in evenly-spaced ascending sequence.

Method of Calculation

The methods used to find the coefficients are the most efficient available. The least squares routine uses the Cholesky (square root) decomposition method. The Fourier method is evaluated by a fast recursion formula.

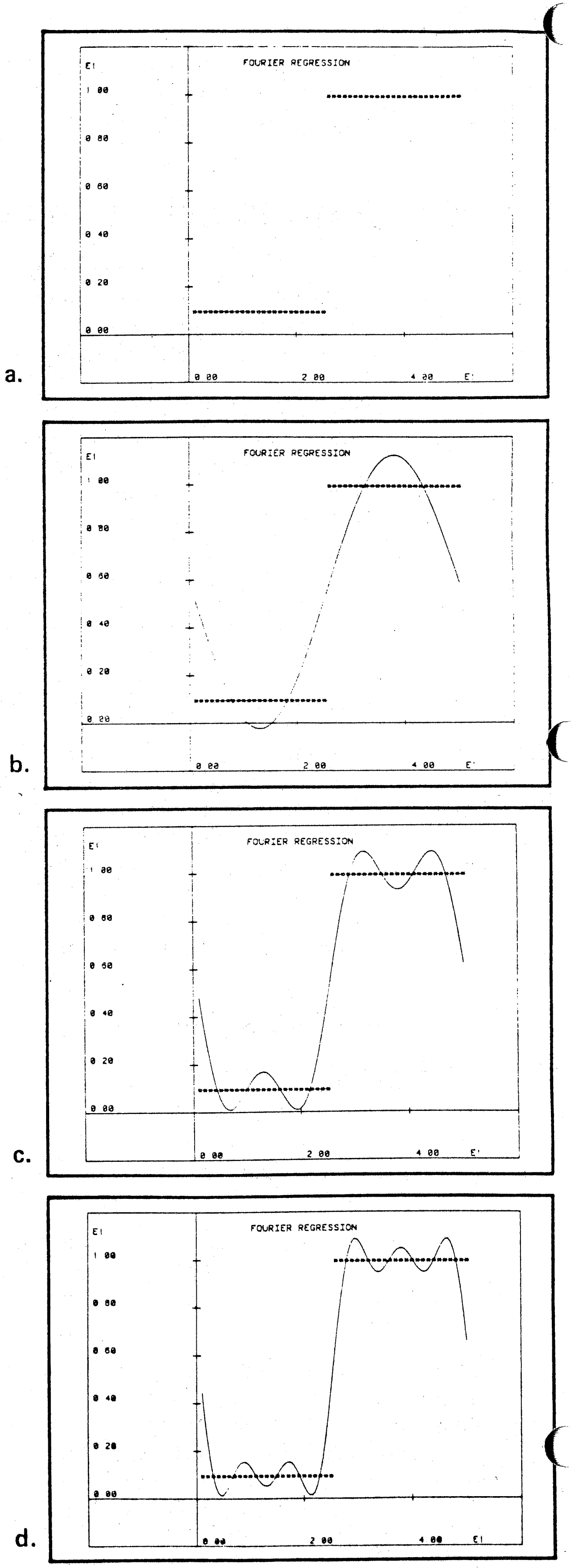
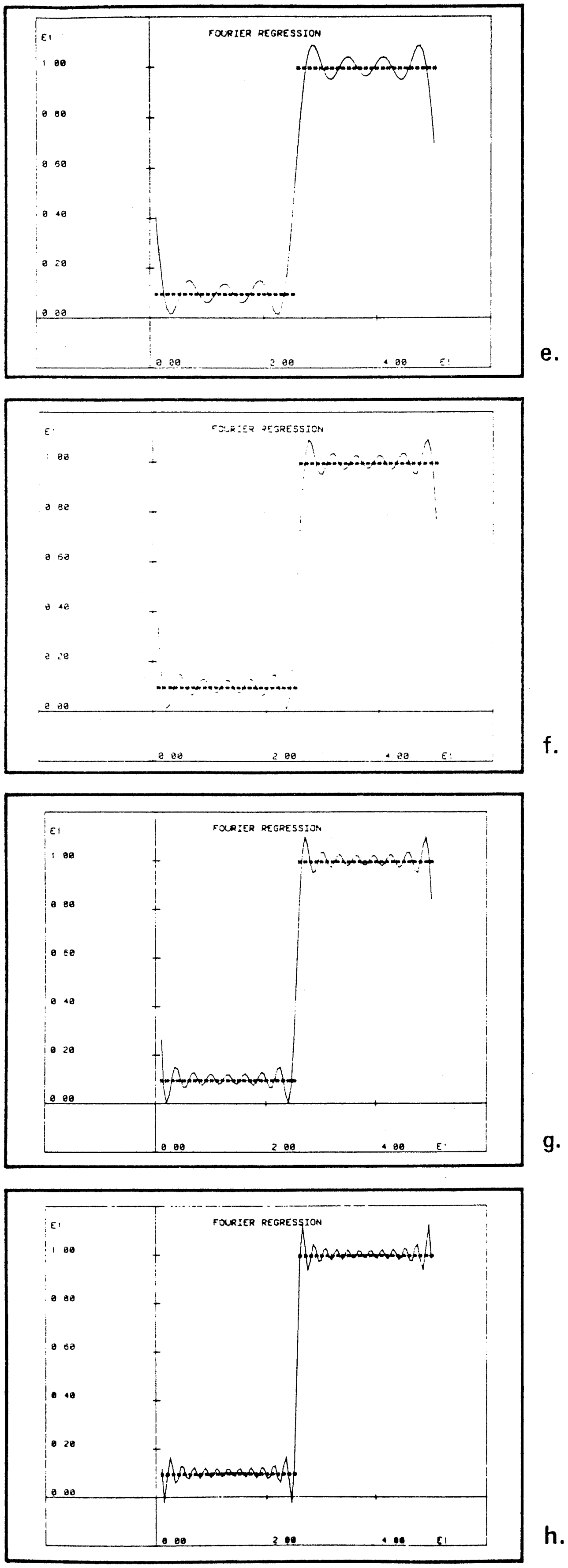


Fig. 1. a. Square wave data with b. one harmonic, c. three harmonics, d. five harmonics, e. seven harmonics, f. twelve harmonics, g. sixteen harmonics, and h. 24 harmonics of a Fourier Series.



If we were writing a program we would normally construct a small repetitive loop that multiplies each of the cosines by its coefficient, and accumulate the products in a large sum. This involves calculating a lot of separate cosine values and wasting enormous amounts of 4051 time. The method used in the program is optimized by a routine that computes the cosine terms from adjacent cosines in our series. [2]

Using the Program

Data is entered into the program from keyboard or tape. For data that is evenly spaced along the x-axis, only the y values must be keyed in. For the best results it is important to know whether the least squares or Fourier series will be used for the regression. The program works on periodic data, so the following guidelines are appropriate for optimized results:

1. Use only one period of the data.
2. Use as many data pairs as possible.
3. For the least squares method, best results occur when y at x_{min} = y at x_{max}.
4. For the Fourier method, the y at x_{min} should *not* equal y at x_{max}.
5. The calculation time will increase with an increase in the number of data pairs and or the number of harmonics required.
6. The number of data pairs must be equal to or greater than two times the largest number of hamonics required.

Example 1: Square Wave

Trivial examples of periodic functions are sine waves and square waves. Let's consider the square wave as our data and approximate it with a Fourier series calculated with the Fourier method in lieu of the least squares technique. Figure 1a shows 50 data points, 25 at the bottom of the square wave and 25 at the top. Note that we should not have 13 points at the bottom, 25 at the top and 12 at the bottom again because of rule number four, above (y at x_{min} should not equal y at x_{max}).

Approximating the data with one harmonic yields the function in Fig. 1b. Three harmonics more closely approximates the square wave. Figures 1c, 1d, 1e, 1f, 1g and 1h show the progressive approximations using 3, 5, 7, 12, 16 and 24 harmonics. Indeed, one requires a long series of sines and cosines to simulate a square wave!

Example 2: Oscilloscope Trace

The Fourier Regression program is designed for analysis of periodic functions such as the pattern in Fig. 2, which is not unlike an oscilloscope trace. Four cycles (periods) of the wavefσrm are shown. One period, delineated by the Y axis line, should be used as program input data. Digitization can be done with a ruler and graph paper, or with the 4956 Graphics Tablet.

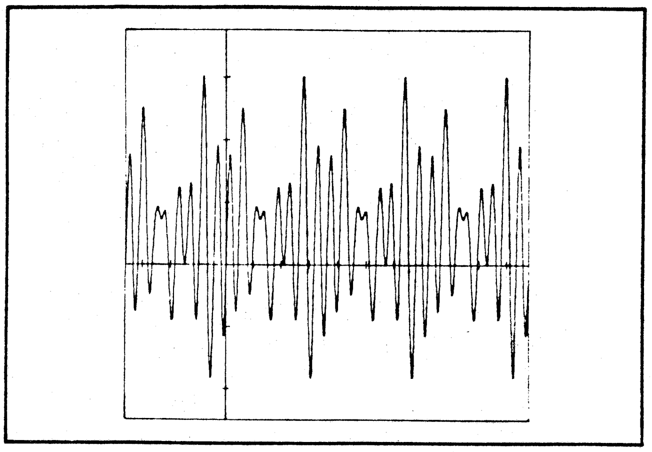


Fig. 2. Data Waveform

The data can be plotted with the PLOT DATA key on the function key overlay. The data, digitized into 360 points, appears in Fig. 3.

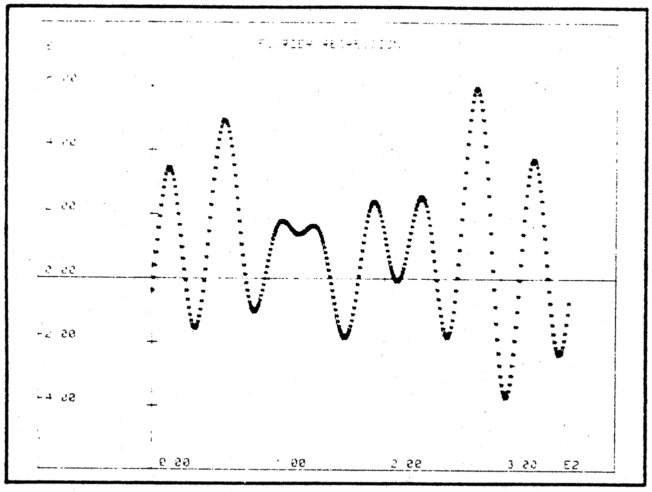
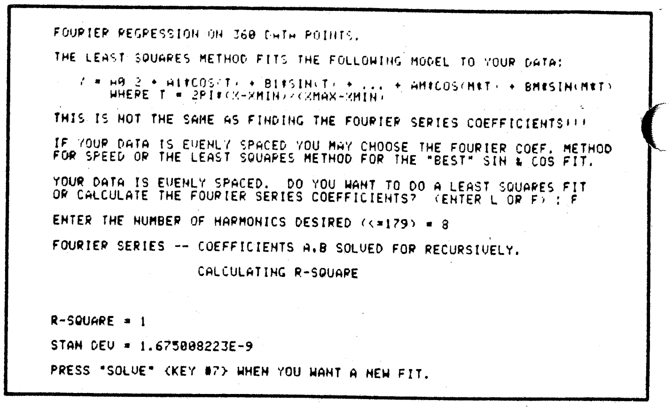


Fig. 3. Data Plot

The coefficients can be calculated by pressing the SOLVE key. If you have entered evenly spaced data from the keyboard or from tape, you must indicate your choice of methods. The Fourier Coefficients method is fastest, but the Least Squares method provides the best fit.

If the data is not evenly spaced, the Least Squares method will be used; no choice is offered.

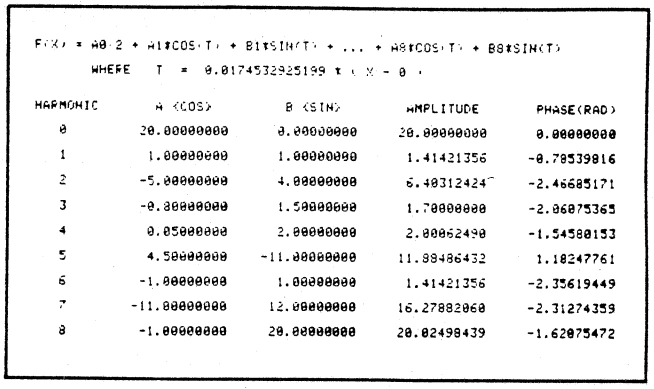
The display will indicate the maximum number of acceptable harmonics, and you will be asked to enter the number desired. When you have entered the number, equations are formed and solved. The R-Square and Standard Deviation are calculated and displayed as in Fig. 4.



```
FOURIER REGRESSION ON 360 DATA POINTS.

THE LEAST SQUARES METHOD FITS THE FOLLOWING MODEL TO YOUR DATA:

   Y = A0/2 + A1*COS(T) + B1*SIN(T) + ... + AM*COS(M*T) + BM*SIN(M*T)
      WHERE T = 2PI*(X-XMIN)/(XMAX-XMIN)

THIS IS NOT THE SAME AS FINDING THE FOURIER SERIES COEFFICIENTS!!!

IF YOUR DATA IS EVENLY SPACED YOU MAY CHOOSE THE FOURIER COEF. METHOD
FOR SPEED OR THE LEAST SQUARES METHOD FOR THE "BEST" SIN & COS FIT.

YOUR DATA IS EVENLY SPACED.  DO YOU WANT TO DO A LEAST SQUARES FIT
OR CALCULATE THE FOURIER SERIES COEFFICIENTS?  (ENTER L OR F) : F

ENTER THE NUMBER OF HARMONICS DESIRED (<=179) = 8

FOURIER SERIES -- COEFFICIENTS A,B SOLVED FOR RECURSIVELY.

                 CALCULATING R-SQUARE


R-SQUARE = 1

STAN DEV = 1.675008223E-9

PRESS "SOLVE" (KEY #7) WHEN YOU WANT A NEW FIT.
```

Fig. 4. Choosing Calculation and Number of Harmonics

The results are displayed in tabular format, as shown in Fig. 5 by pressing the PRINT PARAMETERS key.



```
F(X) = A0/2 + A1*COS(T) + B1*SIN(T) + ... + A8*COS(T) + B8*SIN(T)
     WHERE   T  =  0.0174532925199 * ( X - 0 )

HARMONIC        A (COS)         B (SIN)        AMPLITUDE       PHASE(RAD)
   0         20.00000000      0.00000000     20.00000000      0.00000000
   1          1.00000000      1.00000000      1.41421356     -0.78539816
   2         -5.00000000      4.00000000      6.40312424     -2.46685171
   3         -0.30000000      1.50000000      1.70000000     -2.06075365
   4          0.05000000      2.00000000      2.00062490     -1.54580153
   5          4.50000000    -11.00000000     11.88486432      1.18247761
   6         -1.00000000      1.00000000      1.41421356     -2.35619449
   7        -11.00000000     12.00000000     16.27882060     -2.31274359
   8         -1.00000000     20.00000000     20.02498439     -1.62075472
```

Fig. 5. Fourier Regression Results

When the PLOT FUNCTION key is pressed, the minimum and maximum values are displayed, the data is plotted and the function is overlayed, as in Fig. 6.

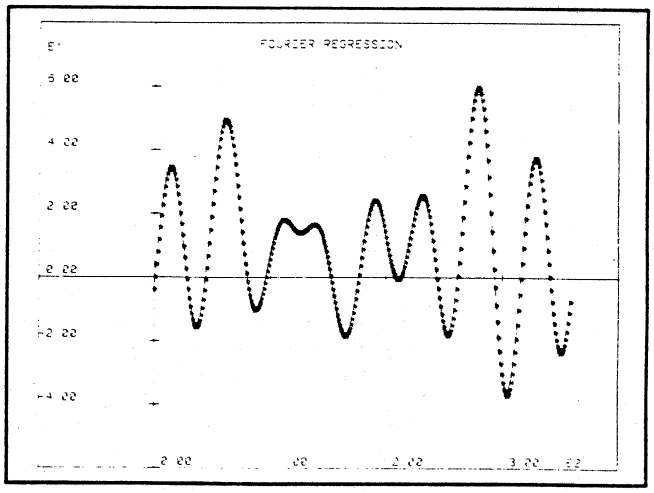


Fig. 6. Data and Resultant Function Plot

These are only two examples of analysis and graphic representation of data made easy through use of Statistics Vol. 4.

[1] *Numerical Methods* by Germund Dahlquist and Ake Bjorck. Englewood Cliffs, Prentice-Hall, 1974, p. 405.

[2] Acton, F.S., *Numerical Methods That Work*, New York: Harper and Row, 1970, p. 11.

BOOKS: Science Texts use BASIC

The following are summaries of two texts that emphasize the importance of the computer as a classroom learning aid in the areas of physical chemistry and physics. Both texts employ the BASIC language to provide instruction in the fundamentals of programming as well as "canned programs" for problem solving in student studies. Programming is approached with a specific purpose and audience in mind. Programming is approached not as an entity but rather as a useful tool to be *applied* in physical sciences. These books were brought to the attention of the 4051 Applications Library staff by Dr. P.C. Holman and his students at the University of Wisconsin-Stevens Point. They recently modified the textbook programs to run on the 4051, and the publishers granted permission to include these excerpts in the library. See Abstract Number 51/00-5201/0.

Problem Solving in Physical Chemistry

Problem Solving in Physical Chemistry introduces the physical chemistry student to the fundamentals of computer programming in an eminently practical way. The student armed with these fundamentals can then use the "canned programs" provided in the text for course studies. In addition, the student can create and modify other programs for expanded problem-solving activities. There is no presumption of previous experience in programming. The intent is that the student will rapidly be able to master the fundamentals of programming in BASIC. This allows approaching the computer as a valuable tool, a vital part of the classroom whole.

The first three chapters of the book concern writing simple computer programs, running these programs on the computer, and debugging and editing these programs to achieve satisfactory computer output. The sample programs and programming assignments are based in real concepts and calculations from beginning physical chemistry. Detailed descriptions are given for using computer commands, for using program storage devices, and for organizing and debugging computer programs. Step-by-step instructions are provided throughout.

The second section presents five "canned" programs which are frequently used in physical chemistry. These programs are intended to be permanent residents of a computer program library. Each of five chapters describes a single program; use of each of the canned programs is illustrated with a real physical chemistry problem. Exercises are also taken directly from physical chemistry topics. The PLOT program plots any continuous function via teletypewriter; AREA carries out numerical integration by use of Simpson's rule; LINEQ is a linear least squares fit of X-Y data points; POLEQ is a polynomial least squares fit of X-Y data points; and ROOTS finds all of the roots of most polynomials.

The third major section in this book contains approximately 200 different computer-oriented problems. These are divided into six chapters according to major topics within physical chemistry: Gas Calculations, Molecular Energies, Classical Thermodynamics, Statistical Thermodynamics, Kinetics, and Quantum Mechanics. Each chapter is divided into approximately ten subtopics. The concepts and equations from each subtopic are summarized; several computer problems follow each subtopic. Each of the problem chapters has been developed to be independent of the other chapters. The problem chapters may be used or covered in any order. Student time requirements are kept to a minimum.

The final section contains selected answers to the problems and exercises. These can be particularly useful to the student as test cases for his newly-written computer programs.

This textbook is intended to be a first exposure to computers and to computer programming, to be a source of physical chemistry problems which will supplement and in some cases replace problems which are found in physical chemistry texts. It is hoped that chemistry departments all over will develop additional canned programs and additional computer problems to meet the needs of individual courses in physical chemistry lecture and laboratory. This is the intent of the emphasis on integrating computer usage with the learning and application of physical chemistry.

Roskos, Roland R. *Problem Solving in Physical Chemistry*. San Francisco: West Publishing Company, 1975.

Computers, BASIC, and Physics

Like *Problem Solving in Physical Chemistry*, this textbook emphasizes the importance of the computer as a teaching aid in the sciences. In this case, the science is physics. The emphasis is again on achieving understanding of the computer and computer programming as a powerful problem-solving tool.

The book attempts to provide an understanding of computers that is thorough enough to be useful, but not so detailed as to be oppressive to the beginner. BASIC is used to teach programming throughout the book, as it allows the student to rapidly and easily acquire a working knowledge of programming.

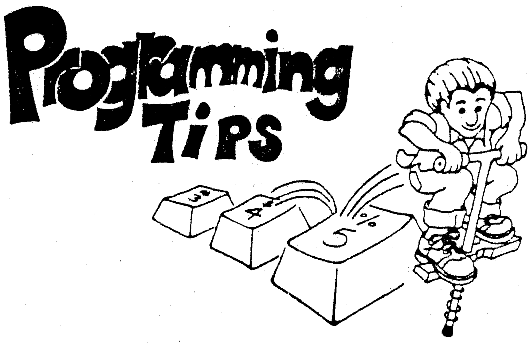
Computers, BASIC, and Physics addresses a dichotomy that typically exists between education and the real world. It is "relatively rare to find a computer as an intrinsic part of the classroom. Yet it is in the classroom where the computer is likely to have the most far-reaching consequences." This last statement can be made because of the recognition that computers already play a dominant role in most areas of the physical sciences. Therefore, it is a natural extension for the student to learn the uses of the computer as the powerful tool that it is in the scientist's world.

Part I of this text concerns the computational mode of computer usage. This mode allows the student to "solve problems in both quantity and of a complexity which significantly increases his experience and 'feeling' for physical situations." This mode allows the study of topics in physics that would otherwise require great mathematical sophistication. Topics include Finite Differences, Integration, Differential Equations, Matrices, and other related matters of physics.

Part II discusses the simulation mode. In this mode, a mathematical model of some physical process is used by the computer to simulate the process. This mode too has "exciting possibilities in the physics laboratory."

Emphasis in the book is on working *with* the computer. This can be at a time-shared terminal or on a small stand-alone system such as the 4051. Hands-on experience is "most valuable and should be the goal wherever possible." Though this book deals with applications in physics, the underlying idea is that the computer can be a valuable integral part of all course work. "It is a tool, and a very powerful one, which should be used wherever appropriate."

Peckham, Herbert D. *Computers, BASIC, and Physics.* **Menlo Park, CA: Addison-Wesly Publishing Company, 1971.**



Packing Two Integers Into One 4051 Word

There are times when it is necessary to keep two pieces of numeric information associated. This is usually done by using two arrays and keying on the index to the relationship. An alternative is to combine the two integers into one word and subsequently into one array. The net result of the alternative method is a savings of approximately 50% on memory usage.

The following technique is used to combine the numeric information into one array.

> Assume that you are using a string named N$ to hold employee names of varying lengths. There are N8 employee names in N$. Each employee has been assigned a number which is three digits and the numbers range from 000 to 999.
>
> Therefore, an array (N7) is set up to hold numbers that contain both the employee number and an index into N$ showing where the employee's name begins. The format is:
>
> N7(I)=XXX.YYY
>
> where XXX is the index into N$ and YYY is the employee number.

```
500 F3=1
510 PRINT "Enter number of employee names ";
520 INPUT N8
530 PRINT "Enter average length of names ";
540 INPUT M
550 DIM N$(M*N8)
560 N$=""
570 DIM N7(N8+1)
580 FOR I=1 TO N8
590 PRINT "JENTER EMPLOYEE NAME: ";
600 INPUT A$
610 N$=REP(A$,F3,0)
620 PRINT "JENTER EMPLOYEE NUMBER: ";
630 INPUT N7(I)
640 REM SHIFT EMPLOYEE # AND ADD IN INDEX
650 N7(I)=N7(I)/1000+F3
660 REM UPDATE INDEX POINTER
670 F3=F3+LEN(A$)
680 NEXT I
690 REM DO LAST INDEX
700 N7(N8+1)=F3
710 RETURN
```

Finding the employee name:

N is the desired employee number.

```
1000 REM RETRIEVE EMPLOYEE NAME FROM N$
1010 REM BREAK OUT THE INDEX
1020 PRINT "INDICATE EMPLOYEE NUMBER YOU WISH ";
1030 INPUT N
1040 FOR I=1 TO N8
1050 F1=INT(N7(I))
1060 REM BREAK OUT EMPLOYEE #
1070 REM NOTE ROUND-OFF COMPENSATION
1080 F2=INT((N7(I)-F1+5.0E-5)*1000)
1090 IF F2<>N THEN 1180
1100 REM FIND INDEX OF NEXT NAME
1110 F3=INT(N7(I+1))
1120 REM FIND LENGTH OF NAME
1130 F3=F3-F1
1140 REM NOW EXTRACT NAME
1150 A$=SEG(N$,F1,F3)
1160 REM PRINT IT
1170 PRINT "JJ",A$
1180 NEXT I
1190 RETURN
```

A few points to consider:

1. Be sure that there are N8 + 1 members in the array N7. The extra entry shows the beginning position for the next name to be placed in N$. In addition, this entry must be used to determine the length of the last name in N$.
2. When the data is to be saved on tape, use the following technique:

```
FIND T
WRITE#33:N8,M,N7
WRITE#33:N$
```

where T is the data file you plan to use.

3. To retrieve the data:

```
FIND T
READ@33:N8,M
DIM N7(N8+1).N$(M*N8)
READ@33:N7
READ@33:N$
```

4. The techniques used in 2 and 3 above will allow your programs to be independent of the addition of names into the employee list.
5. Given an employee number in J and a starting point for the employee name in N$ as the variable K, N7(I) is calculated as follows:

$$N7(I) = K + J/1000$$

Dashed Line Algorithm

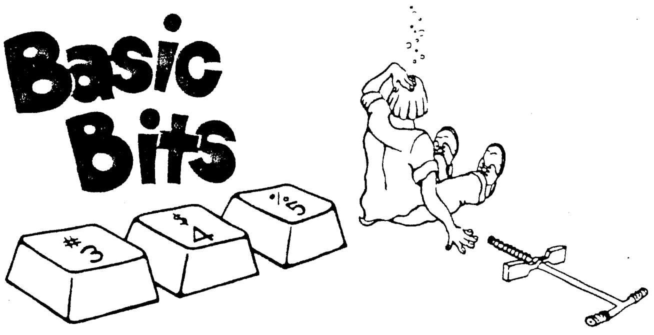
by Ken Cramer

The following dashed line algorithm is an enhanced version of the algorithm in the "Introduction to Graphic Programming in BASIC" manual, page 7-13. The following algorithm is 29 lines (versus 43) and runs in 70% of the time. The major change from the original is a restructuring of the drawing loop to minimize repetitive computations and logical comparisons. V1, V2, V3, and V4 are the viewpoint parameters; W1, W2, W3, and W4 are window parameters from the main program. D9 is the output device (32 for screen); X1, Y1 is the start point of the line; X2, Y2 is the end point.

```
1000 REM     DASHED LINE SUBROUTINE
1010 MOVE @D9:X1,Y1
1020 REM S1 IS NUMBER OF HORZONTAL USER DATA UNITS PER GDU
1030 S1=(W2-W1)/(V2-V1)
1040 REM S2 IS NUMBER OF VERTICAL USER DATA UNITS PER GDU
1050 S2=(W4-W3)/(V4-V3)
1060 REM FIND HORIZONTAL DISTANCE IN GDU'S
1070 D1=(X2-X1)/S1
1080 REM FIND VERTICAL DISTANCE IN GDU'S
1090 D2=(Y2-Y1)/S2
1100 REM D IS DISTANCE BETWEEN POINTS IN GDU'S
1110 D=SQR(D1↑2+D2↑2)
1120 REM DESIRED DASH LENGTH IN GDU'S IS L--COULD BE SET IN MAIN PROG
1130 L=1
1140 IF ABS(D2)<L AND ABS(D1)<L THEN 1260
1150 REM FIND HORIZONTAL COMPONENT OF DASH IN USER UNITS
1160 U1=L*(D1/D)*S1
1170 REM FIND VERTICAL COMPONENT OF DASH IN USER UNITS
1180 U2=L*(D2/D)*S2
1190 REM  MAX FUNTION USED FOR NEAR VERTICAL OR HORIZONTAL LINES
1200 FOR I=0 TO X2-X1-U1*2 MAX Y2-Y1-U2*2 STEP 2*(U1 MAX U2)
1210 RDRAW @D9:U1,U2
1220 RMOVE @D9:U1,U2
1230 NEXT I
1240 REM  DRAW TO END ACTUAL END POINT TO BEGIN NEXT LINE
1250 DRAW @D9:X2,Y2
1260 X1=X2
1270 Y1=Y2
1280 RETURN
```

The following expansion of the above algorithm gives five line styles depending on the value of L1 supplied by the main program. Different styles and drawing speeds can be attained by changing the variable L in line 1130.

```
1000 REM     MULTIPLE DASH PATTERN--DASHED LINE SUBROUTINE
1010 MOVE @D9:X1,Y1
1020 REM S1 IS NUMBER OF HORZONTAL USER DATA UNITS PER GDU
1030 S1=(W2-W1)/(V2-V1)
1040 REM S2 IS NUMBER OF VERTICAL USER DATA UNITS PER GDU
1050 S2=(W4-W3)/(V4-V3)
1060 REM FIND HORIZONTAL DISTANCE IN GDU'S
1070 D1=(X2-X1)/S1
1080 REM FIND VERTICAL DISTANCE IN GDU'S
1090 D2=(Y2-Y1)/S2
1100 REM D IS DISTANCE BETWEEN POINTS IN GDU'S
1110 D=SQR(D1↑2+D2↑2)
1120 REM DESIRED DASH LENGTH IN GDU'S IS L--COULD BE SET IN MAIN PROG
1130 L=1
1140 IF ABS(D2)<L AND ABS(D1)<L THEN 1600
1150 REM FIND HORIZONTAL COMPONENT OF DASH IN USER UNITS
1160 U1=L*(D1/D)*S1
1170 REM FIND VERTICAL COMPONENT OF DASH IN USER UNITS
1180 U2=L*(D2/D)*S2
1190 U3=2*U1
1200 U4=2*U2
1210 REM L1 IS THE VARIABLE FOR SELECTING THE DASH PATTERN
1220 GO TO L1 OF 1260.1320,1370.1440.1520.1590
1230 GO TO 1590
1240 REM  MAX FUNTION USED FOR NEAR VERTICAL OR HORIZONTAL LINES
1250 REM         STANDARD DASHED LINE
1260 FOR I=0 TO X2-X1-U1*2 MAX Y2-Y1-U2*2 STEP 2*(U1 MAX U2)
1270 RDRAW @D9:U1,U2
1280 RMOVE @D9:U1,U2
1290 NEXT I
1300 GO TO 1590
1310 REM         DOTTED LINE
1320 FOR I=0 TO X2-X1-U1 MAX Y2-Y1-U2 STEP U1 MAX U2
1330 RDRAW @D9:0,0
1340 RMOVE @D9:U1,U2
1350 NEXT I
1360 GO TO 1600
1370 REM         DOT-DASHED LINE
1380 FOR I=0 TO X2-X1-U1*3 MAX Y2-Y1-U2*3 STEP 3*(U1 MAX U2)
1390 RDRAW @D9:U3,U4
1400 RMOVE @D9:U1,U2
1410 NEXT I
1420 GO TO 1600
1430 REM      LONG DASHED LINE
1440 FOR I=0 TO X2-X1-U1*3 MAX Y2-Y1-U2*3 STEP 3*(U1 MAX U2)
1450 RDRAW @D9:U1,U2
1460 RMOVE @D9:U1,U2
1470 RDRAW @D9:0,0
1480 RMOVE @D9:U1,U2
1490 NEXT I
1500 GO TO 1600
1510 REM     LONG/SHORT DASHED LINE
1520 FOR I=0 TO X2-X1-U1*5 MAX Y2-Y1-U2*5 STEP 5*(U1 MAX U2)
1530 RDRAW @D9:U3,U4
1540 RMOVE @D9:U1,U2
1550 RDRAW @D9:U1,U2
1560 RMOVE @D9:U1,U2
1570 NEXT I
1580 REM  DRAW TO END ACTUAL END POINT TO BEGIN NEXT LINE
1590 DRAW @D9:X2,Y2
1600 X1=X2
1610 Y1=Y2
1620 RETURN
```



SPACE Function

The SPAce function returns the maximum number of bytes required to store a current program in external ASCII format. However, the BASIC interpreter arrives at this number by multiplying the number of program lines by 72. Many programs use considerably less than 72 characters per line resulting in excess tape space reserved. An alternative is to divide the maximum bytes returned for a more realistic MARking:

FINd n
MARk 1, SPA/2

Refer to the SPACE function under the MEMORY MANAGEMENT section in your 4051 Graphic System Reference Manual for further information.

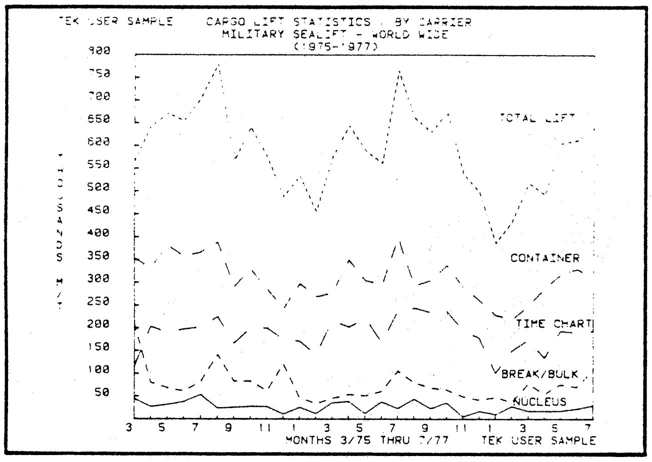
4051 Applications Library Program Abstracts

Documentation and program listings of these programs may be ordered for $15.00 each. Programs will be put on tape for an additional $2.00 handling charge per program and a $26.00 charge for the tape cartridge. (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.)

Please use the Applications Library Order Form. Order forms are included in the Membership Packet and are available from your local Tektronix Sales Engineer. Orders outside of the U.S. must be ordered through the local Tektronix sales office.

ABSTRACT NUMBER: 51/00-9517/0

Title: **Time Series Graphing**
Author: Mallory M. Green
Military Sealift Command/U.S. Navy
Memory Requirements: 32K bytes
Peripherals: 4952 Joystick
4662 Plotter Optional
Statements: 740



This program is designed to plot and graph time series data. The time series is limited to 35 units of time, such as days, months or years per graph. Fourteen User-Definable Keys are utilized to give the user a flexible program with a variety of options. Graphs may be drawn to both the screen and the 4662 Plotter. Data can be easily entered, displayed, changed and/or saved to tape. The time series plots include:

A one-to-three line title
Solid and dashed data set lines
Data set labels
Ticked and labeled axes

Additional options allow:

Scale changing for units of measure on Y-axis
Extension of X-axis time series units
Graph classification labeling
Superimposing grids on graph

ABSTRACT NUMBER: 51/00-3301/0

Title: **Lighting Intensity Distribution**
Author: Florent van Vlasselaer
Tektronix Belgium
Memory Requirements: 16K
Peripherals: 4631 and/or 4662 optional
Statements: 231 plus a 30 x 10 table

The program calculates the lighting intensity distribution over a user-specified area. The output is in the form of a grid showing the intensity in the individual squares. Total intensity and average intensity per square are also given.

The user gives the dimensions of the area, position and height of the lamp, its type and strength and the point in the area at which the lamp is directed.

Using a table for that type of lamp, the program calculates the distribution and outputs it in the form of a grid. The grid is labeled and the total intensity and average intensity per square is given. Output on the plotter is optional.

The program was conceived specifically for football fields but can be used for any rectangular area.

Only the table of values for projector type 'D' are given here; the values for projector type 'C' must be implemented by the user. This type uses a much larger matrix and would need a 32K 4051.

ABSTRACT NUMBER: 51/00-5201/0

Title: **Textbook Problems Adapted to 4051 BASIC: Problem Solving in Physical Chemistry**
by Roland R. Roskos
Computers, BASIC, and Physics
by Herbert D. Peckham
Adapted by: Dr. P.C. Holman
Memory Requirements: 8K
Peripherals: 4631 Hard Copy Unit Optional
Statements: 1135 and 2141 respectively

Two textbooks emphasize the use of computers for problem solving in sciences of chemistry and physics. Both employ BASIC as the language. Dr. Holman and students have modified programs from the books to run on the 4051. Programs can be used to solve common types of problems in these branches of the sciences. Save student time normally used keying in programs for use in studies.

12 programs from chemistry; 34 programs from physics. Only listings are available (written and/or tape). Documentation available in the source textbooks. Both books are summarized elsewhere in this issue. Permission to include materials from these texts was graciously granted by the publisher of each.

TEKniques

The 4051 Applications Library Newsletter
VOL. 1 Index (10 Issues)

A complete listing of titles by major subject areas.

Programming Tips (continued)

Numeric Quicker Sort on 4907 Disc Unit	No. 9, p. 5
Plot Arrays Faster and More Efficiently	No. 5, p. 3
Pointing To Areas On the Screen	No. 5, p. 2
Renumbering Programs	No. 7, p. 4
Saving Memory—FOR/NEXT Loop Exit	No. 9, p. 5
Speed It Up With Strings	No. 9, p. 4
Speeding Up The 4051	No. 1, p. 2
Speeding Up Trigonometric Functions	No. 10, p. 9
Storing a Graph On Tape From the 4051	No. 6, p. 1
Truncation Utility	No. 7, p. 5
Using the Pointer Command in Interactive Programs	No. 7, p. 4

BASIC BITS

Add a Byte to Binary Strings	No. 8, p. 7
Cartridge Respooling	No. 8, p. 7
Clean and Cool	No. 9, p. 8
Correct Tape Inserted	No. 10, p. 15
Drawing Upside Down or Backwards	No. 8, p. 7
For The Novice	No. 7, p. 3
GPIB	No. 8, p. 7
MARking and Re-MARking	No. 7, p. 8
Recover Your Memory	No. 10, p. 15
Save Your Memory	No. 10, p. 15
Speeding Up Proceedings	No. 10, p. 15

Publications

Computer Graphics Book Available	No. 8, p. 1
4051 Graphic System Publications	No. 10, p. 13
GPIB Hardware Support Manual	No. 8, p. 4
GPIB Manual Now Available	No. 6, p. 3
IEEE Specs Available	No. 1, p. 7
Intro to Graphic Manual Valuable	No. 10, p. 9
Opportunity to Receive Recognition	No. 9, p. 5
Wang 2200 to 4051 and HP9830 to 4051 Conversion Guides Now Available	No. 4, p. 8

TEKniques

Applications Newsletter	No. 2, p. 4
Let Us Hear From You	No. 1, p. 2
More Than One Copy	No. 10, p. 7
Publication Deadlines	No. 10, p. 7
Survey Provokes Suggestions, Criticism	No. 10, p. 4
There's An Asterisk On Your Mailing Label	No. 10, p. 7
What It's All About	No. 7, p. 2

Tektronix
COMMITTED TO EXCELLENCE

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

ADDRESS CORRECTION REQUESTED