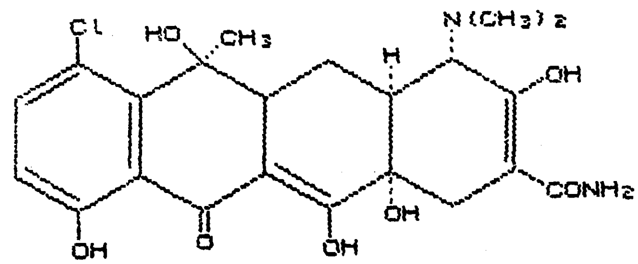
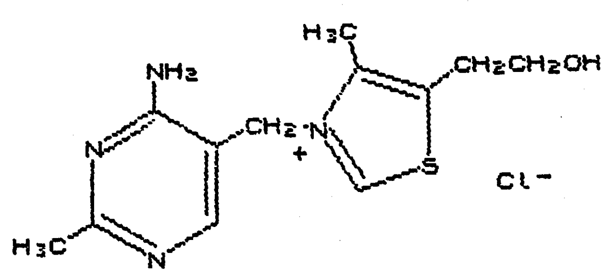
Computers in Chemical Education Newsletter

Published by the ACS Division of Chemical Education
Committee on Computers in Chemical Education
Paul Cauchon, Chairman/Canterbury School, New Milford, CT 06776
Donald Rosenthal, Editor/Department of Chemistry, Clarkson University, Potsdam, NY 13676

Volume VIII Number 2 **June 1985**



AUREOMYCIN



THIAMINE CHLORIDE

ON THE COVER

Dr. Mike Davis (Department of Organic Chemistry, La Trobe University, Bundoora, Victoria, Australia 3083) won honorable mention in the microcomputer graphics category at the Computer Graphics Contest held at the Eighth Biennial Conference on Chemical Education last summer for his figure entitled "Aureomycin". This figure was produced using a Sinclair "Spectrum" 48k microcomputer with a ZX printer. A photograph of the figure was taken and enlarged. The program is about 210 lines in length written in BASIC with an embedded assembly language routine. Dr. Davis indicates the program is general and allows an organic chemist to draw fairly complex structures quickly. The program will draw multiple ring compounds of different size and orientation containing heteroatoms, different types and angles of bonds, and substituent groups with extra graphic symbols. The printer output is directly useable in research papers and theses. The metal-coated paper used in the ZX printer photocopies very well. Dr. Davis has used enlarged negative photos of the printer output in poster sessions in Australia and New Zealand.

MESSAGE FROM THE CHAIRMAN

"INTERFACING" continues to be one of the hottest topics for discussions concerning the roles of computers in chemical education. With many chemistry teachers as yet only slightly less comfortable with the concept than are many of our students with the mole concept, there is more than enough interest to draw a crowd to any presentation that deals with linking microcomputers to lab instruments. For example, the first of the Potsdam workshops to be oversubscribed was - you guessed it - section D, "Interfacing". Numerous pending proposals seeking NSF support under the "Teacher Enhancement" program list familiarization with interfacing as their primary objective. Notwithstanding the position taken by one speaker at the Computer Secretariat conducted during the Miami ACS meeting, a theoretical chemist who "accused" us of "...perverting the proper role of computers by using them to deliver CAI instead of solving Schrodinger equations", it appears that what we are doing for our students will be significantly enhanced once interfacing becomes an everyday activity in our labs and classrooms. Before this can happen, there is a real need to identify valid applications that use interfacing for purposes beyond illustrating the technique of interfacing. Learning ABOUT computers as data collection devices is certainly a valid course objective, as is learning to use any laboratory instrument, but what seems even more important is establishing new objectives that become possible BECAUSE of this technique. Progress is being made. Ken Ratzlaff offered some ideas in the third part of his excellent series, "Scientific Applications of the Apple Game Port" in this Newsletter (Dec. '84). At the Miami meeting, John Moore outlined two intriguing experiments under development using the SERAPHIM photocell and thermistor-based kits exploring reaction rates and equilibrium. But we are a long way from having a truly satisfactory answer for the chemistry teacher who asks "Now that I've built it, what do I do with it?" At one session in Miami, the speaker displayed his heart rate on the screen of an Apple with an LED and a phototransitor. He went on to say "You can do lots of interesting experiments with this device!" Exactly what some of those things are was offered in the form of a challenge to our imagination rather than an already existing repetoire of validated applications.

But I am confident that there are a number of people reading this Newsletter who have already recognized and risen to the challenge and are doing some interesting chemistry with a "bloctronic" or a thermistor. How about putting your experiences on paper and submitting them to this Newsletter. Let's hear about how you've connected your computer to the "real world" of live data and what the impact has been on your courses. It has often been said that computers will change not only how we teach but what we teach, as well. I'm convinced that the ability to simultaneously collect, process and display data will enable us to do exciting things we never thought of as possible, in fact never thought of at all. This Newsletter is one forum for those of you who are already doing them.

This is the last issue of the Newsletter you will receive unless you renew your subscription by sending in the subscription renewal form by mid-July. Those who have already renewed their subscription will have the subscription expiration date appear on the first line of the mailing label (8609 is September 1986).

Two C. C. C. E. National Workshops have been planned for this summer. Workshop - East will be held at Clarkson University in Potsdam, NY from July 28 to August 1. Workshop - West will be held at Truckee Meadows Community College in Sparks, NV from August 11 to 14. An announcement of the Workshops - West appears elsewhere in this issue. The eastern workshops currently have over 100 registrants and several of the workshops are rapidly becoming filled. The Microcomputer Interfacing workshop has been filled for several months. If you're interested in interfacing apply to the western workshops.

We hope to organize a rather full program of contributed papers on computer uses in chemical education for the National Meeting to be held in New York City from April 6 to April 11,1986. We are interested in activities at the high school, first two years of college and advanced course level. Those of you who are willing to contribute papers or have suggestions for the high school program should contact Paul Cauchon, Canterbury School, New Milford, CT 06776, (203) 355-3103. Professor Patricia Flath, Paul Smith's College, Paul Smith, NY 12970, (518) 327-6264 will handle the first two years of college. I will help organize papers concerning the use of computers in advanced courses. (Professor Donald Rosenthal, Department of Chemistry, Clarkson University, Potsdam, NY 13676, (315) 268-2389). A symposium involving "Applications Software for Lecture and Laboratory Courses" is planned. The symposium would involve the use of substantial program packages in education, i.e. the use of word processing, electronic spread sheets, data bases, statistical, numerical methods and graphics packages in courses. I would like to hear from anyone interested in participating in the symposium. For a variety of reasons it will not be possible to use computers, monitors or video projectors. Only overhead projectors and slide projectors will be available.

Anyone having suggestions for programs at future National Meetings (fall 1986 - Anaheim, CA and beyond) should contact Paul Cauchon.

An article by Tim Eckert describing a short Basic program or subroutine which can be used to automatically scale the axes in plotting data is found in this issue. The routine requires less than 20 lines of code. If you have a short routine which is of general interest, please send it to me for publication in a future issue of this Newsletter.

LAB CHECKER
A PROGRAM TO HELP STUDENTS IMPROVE LAB REPORTS
by Paul Cauchon*

In carrying out laboratory assignments, students make many types of errors as they collect, record and process their data. To help students improve their skills, conscientious teachers devote considerable time and effort going over laboratory reports, making a variety of notes and comments. Credit is "taken off" for a variety of failures such as incorrect or missing units, an improper number of significant digits or arithmetic mistakes. Correcting a laboratory report in such a manner that the student can tell whether points were subtracted for conceptual, manipulative, or format errors is a tedious and all too frequently unrewarding task, since it only works if the student is willing to go over the corrected paper thoughtfully. Unfortunately, the elapsed time between submitting a report and getting it back seldom serves as a positive catalyst for self-evaluation. Since the goal of "correcting" papers is to improve student performance by identifying weaknesses and making suggestions, it would seem to be far better if as many errors as possible are brought to the student's attention BEFORE reports are handed in, so that the report can be dealt with in a positive manner, i.e., corrected, instead of merely noted.

At Canterbury School we have developed a series of computer programs keyed to our laboratory manuals that seem to be effective in helping students improve their laboratory reports. These programs provide students with a quick, individualized method of checking laboratory results prior to submitting their final report. A computer is normally available in the laboratory so the check can be made soon enough to repeat a step, or even the whole experiment, when results are too far off. Since each "run" requires only two or three minutes, one machine is ample for a 16 student lab section. These programs reveal what the students are doing wrong without taking off any points. It is then possible to distinguish errors in arithmetic from those in format or overall approach. Appropriate corrections can thus be made in a "real time" sense. The potential for raising laboratory grades offered by these programs is quickly apparent to the students and they need little encouragement to run them. Another result of using these programs is that the teacher has much more time available for evaluating the substance of the report.

Here is a typical dialog illustrating the program which accompanies an experiment on the determination of percent oxygen in potassium chlorate:

PLEASE ENTER EACH ITEM AS CALLED FOR.
WATCH YOUR UNITS AND SIG FIGS!!

MASS OF CRUCIBLE AND COVER? 20 g

YOU DON'T HAVE ENOUGH DIGITS.
MASS OF CRUCIBLE AND COVER? 20.00

YOU FORGOT THE UNITS.

MASS OF CRUCIBLE AND COVER? 20.00 g

MASS OF CRUCIBLE, COVER, CONTENTS BEFORE HEATING? 2.20 g

THAT SEEMS VERY LOW...ARE YOU SURE? N

MASS OF CRUCIBLE, COVER, CONTENTS BEFORE HEATING? 22.20 g

MASS OF CRUCIBLE, COVER, AND RESIDUE AFTER HEATING? 21.12 g

WHAT DO YOU GET FOR THE MASS OF POTASSIUM CHLORATE? 2.20 g

I AGREE...

WHAT DO YOU GET FOR THE MASS OF OXYGEN RELEASED? 1.12 g

HMMM...I GET SOMETHING DIFFERENT.

WANT SOME HELP? Y

SINCE OXYGEN HAS LEFT THE CRUCIBLE, ITS MASS IS EQUAL TO THE LOSS OF MASS BETWEEN YOUR SECOND AND THIRD MEASUREMENTS.

WHAT DO YOU GET FOR THE MASS OF OXYGEN RELEASED? 1.08 g

THAT'S WHAT I GET.

WHAT DO YOU GET FOR THE PERCENT OXYGEN IN POTASSIUM CHLORATE? 49.1%

I AGREE...

WELL, YOUR CALCULATIONS ARE DONE CORRECTLY BUT YOUR ANSWER IS OUTSIDE OF THE ACCEPTABLE RANGE...BETTER TRY THE EXPERIMENT AGAIN.

During the past year, the programs were standardized through restructuring in order to develop LAB CHECKER, a utility program organized to support most quantitative laboratory experiments. It is set up to ask for the measured data, and check each entry for proper significant figures, units, and reasonableness. It then asks for the calculated results, and checks format as before but now it also checks for accuracy to see if the data entered were properly processed.

Before LAB CHECKER can be used for a particular experiment, certain items of information unique to that experiment must be coded at appropriate places in the program. For each item of data, there must be a label (text to be displayed when the item is requested), units, number of digits after the decimal point, lowest reasonable value, and highest reasonable value. Entering a "0" for any of these will simply bypass the check for that characteristic. For each calculated result one must enter the same information, except that in place of a "reasonableness check", the error tolerance must be specified, along with the explanatory text to be provided when a result is incorrect. In addition, algorithms for calculating each result must be entered in the program. Exactly what must be entered, how, and where is all explained in the documentation. It takes only about an hour to prepare LAB CHECKER for a new experiment.

LAB CHECKER is written in BASIC and runs on the Apple II family of computers. However, it is easily translated onto any other computer which can be programmed in BASIC. For a copy of the Apple II program with supporting documentation, send me a disk and a large, self-addressed envelope with $.90 worth of stamps...or pick up a copy at the software exchange at the Seventh C.C.C.E. National Computer Workshops or at the SWAP SHOP at CHEM ED '85 in Montclair.

*Canterbury School
New Milford, CT 06776

AN EFFICIENT ALGORITHM FOR AUTOMATIC SCALING OF AXES

by Timothy Eckert*

Often data generated by laboratory experiments are more easily analyzed when plotted on a graph. In fact, plotting programs are commonly included in computer programs which process data. A necessary part of such programs is the scaling of the axes to provide convenient minima, maxima and step sizes. Some plotting programs require the user to manually provide the desired minima, maxima and step sizes for the axes, a bothersome task. Other programs automatically scale axes, but either use the largest and smallest coordinates of the data points for the maxima and minima or else always set the minimum at zero. The former technique leads to axes scaled with awkward numbers, while the latter precludes graphing data points with negative coordinates. Moreover even if all the points have positive coordinates but are clustered relatively far from zero, an axis minimum at zero prevents differentiation of the points.

The algorithm in my program, SCALING, automatically sets the minima and maxima of the axes at positive or negative values not only to accomodate points with negative coordinates but also to allow greater distinction among points clustered far from zero. To demonstrate the versatility of this program the table below shows the scaling output for six types of data ranges (for simplicity, in one dimension only).

data minimum	data maximum	axis minimum	axis maximum	step size
0	0.9	0	1	0.1
1.2	14	0	16	2
900	1001	860	1020	20
-100	9	-140	20	20
-9	-6	-9.5	-4.5	0.5
-3	0	-3.5	0	0.5

SCALING is written in BASIC and is listed below. Its algorithm is <u>not</u> designed to stand alone, but instead to be readily incorporated into the user's own plotting program. In its present form it receives the input of the data minimum and maximum and outputs the appropriate axis minimum, maximum and step size.

Of course no algorithm can provide ideal, automatic scaling for all purposes. For example, if one wants to extrapolate a line far from the data points, perhaps to locate a distant axis intercept, this program which focuses on data points fails. An option for manual scaling axes should always accompany any algorithm for automatic scaling.

<u>PROGRAM</u>

```
  5  REM ALGORITHM FOR SCALING AXES
 10  PRINT "DATA MINIMUM, DATA MAXIMUM"
 20  INPUT N,X
 30  If X<>0 THEN 50
 40  X = -.1 ∧9
 50  M = (1+SGN(X))*N/2+(SGN(X)-1)*X/2
 60  W = (1+SGN(X))*X/2+(SGN(X)-1)*N/2
 70  D = W*(1-(0.5∧INT(LOG(W/(W-M))/LOG(2))))
 80  G = 10∧(INT((LOG((W-D)/25))/LOG(10))
 90  S = G*5*2∧INT(LOG((W-D)/(25*G))/LOG(2))
100  B = S*(INT(W/S)+1)
110  E = B+S*INT((D-B)/S)
120  F = (1+SGN(X))*E/2+(SGN(X)-1)*B/2
130  C = (1+SGN(X))*B/2+(SGN(X)-1)*E/2
140  PRINT "AXIS MINIMUM";F
150  PRINT "AXIS MAXIMUM";C
160  PRINT "STEP SIZE";S
170  END
```

<u>DOCUMENTATION</u>

This listing is the fully general form of the algorithm.

If the data maximum cannot equal 0, lines 30-40 may be deleted.

Lines 50, 60, 120, 130 invert parameters if the data maximum is negative. But if the data minimum cannot be negative, these lines may be deleted, and M, W, E, B may be replaced by N, X, F, C, respectively, throughout the program.

Lines 80-90 establish step size as the product of 1, 2, 4 or 5 and a power of 10 to insure 5-10 axis divisions. If more divisions are desired, the number 25 in lines 80-90 may be increased.

Lines 100-110 establish the axis maximum and minimum.

Some versions of BASIC may lead to an occasional deviation in output associated with arithmetic round-off errors in statements like line 90.

*Chemistry Department
Fredonia State University College
Fredonia, NY 14063

APL—A TOOL OF THOUGHT

B.D. Joshi*

INTRODUCTION

APL stands for 'A Programming Language'. It was invented by Iverson as a notation for direct implementation and testing of ideas on a computer in an interactive environment.[1] Since then it has evolved into a general purpose programming language. Recently at least four versions of APL have become available for IBM PC.[2] This development, in conjunction with the popularity of IBM PC, has opened up a new dimension in personal computing, especially for mathematics and science oriented educational computing.

On a beautiful sunny day of May 1971 an aggressive IBM salesman left a shiny APL printing-terminal and an APL manual, for any one to play with, in Geneseo's then brand new Greene Science Building. The terminal was connected to an IBM main-frame computer at SUNY Binghamton. During that memorable summer I caught the "APL bug", and I am still thrilled by that summer's experience. For the next six years I did all of my scientific computing in APL. From 1978 to 1983 I taught a 3-credit hour APL programming course at Geneseo to a total of about 250 students. Last year I acquired IBM's APL for my IBM PC, and have been developing interactive user transparent problem solving packages in APL for use in our physical chemistry and related courses.

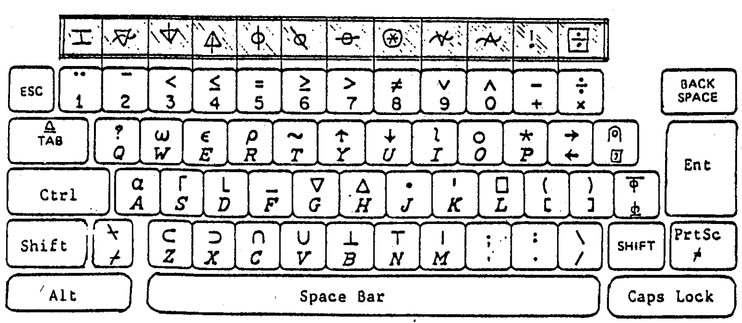
NATURE OF APL

Among general purpose programming languages currently avaäilable, APL stands in a class by itself.[3] There are three landmark papers that elucidate the nature of APL.[4-6] These along with five other basic articles dealing with APL have recently been reprinted as a single paperback book.[7] For beginning students of APL there are many textbooks and manuals available. The best known in this category is the book by Gilman and Rose.[8] In addition, there is a quarterly magazine devoted to a special-interest-group, consisting of persons interested in various aspects of APL, SIGAPL, has been organized within the Association for Computing Machinery.

Much of the terminology and symbolism used in APL is borrowed from the general domain of mathematics. It uses the precise concepts of scalars, vectors, matrices, and multi-dimensional objects for data structures. It also uses the well defined concepts of functions and operators for "data processing". In APL, *functions* modify the *data* and *operators* modify *functions*. Two types of *functions* are used in APL. A monadic function requires the data as its right argumant, and a dyadic function requires two pieces of data, presented as a left argument and a right argument. APL *operators* modify only the *dyadic functions*. There are at least 23 monadic and 53 dyadic intrinsic functions, often referred to as the primitive functions, available to APL programmers. In addition there are at least 5 general purpose operators used for modifying most of the primitive dyadic functions. Recursive programming is also supported in APL.

Single character symbols are used to represent APL *functions*. APL *operators* are represented by symbols containing one to three characters. The APL character set on IBM PC includes upper and lower case English Alphabets, some Greek Letters, and some special characters. The APL keyboard for IBM PC is shown in Fig. 1. It takes a while for new users of APL to get used to the APL character set.

APL Keyboard
(IBM PC)



Note: 1. Using 'Alt' in conjunction with the numeric keys in the top row produces characters shown in the shaded area, i. e., to generate a '⊖' hold the 'Alt' key down and press the '7' key.
2. Using 'Alt' in conjunction with an 'Alpha' key produces the corresponding lower case alphabet, i. e., to generate a 'q' hold the 'Alt' key down and press the 'Q' key.

Figure 1

APL is an array oriented language. Problems to be solved using the full power and capabilities of APL require a thought process for algorithm development not shared by traditional programming languages like FORTRAN, BASIC, and Pascal. I believe that this is one of the reasons why some people who have used FORTRAN, BASIC, and/or Pascal find if difficult to use APL. Since APL is an interpreter-based language, each line of the code is interpreted and executed separately. Thus, the strategies used for writing efficient APL programs are somewhat different from those used for writing programs in compiler based languages. Some APL programmers push this idea to an extreme by writing complete one-line programs which may be difficult for beginners to comprehend.

In APL no declaration of any type regarding the nature and size of variables is required. All input/output is generally format free. However, if needed, formatted output can be easily generated. The main burden on an APL programmer is that of developing creative, efficient, reliable, and robust algorithms for solving problems at hand.

Because of the tremendous flexibility provided by the available intrinsic functions of APL, usually no two APL programmers will create the same algorithm to solve a given problem. If no accompanying documentation for the algorithm exists, it may be difficult for an APL programmer to understand another person's program. Large APL programs are generally developed in modular form. The individual modules, called by the generic name, *functions* , are coded and tested separately. There exist strict rules for passing data from one module to the other. Data can also be shared by various modules in "global" form. All variables in APL are considered to be global unless they are localized in a given module. Variable-name conflicts among modules is easily avoided. This is accomplished by identifying all variables specific to a given module and localizing them in that module.

Science and mathematics oriented students seem to find it easy to learn and use APL. Students with no prior computing experience seem to learn APL easily, and generally faster, than those who have had "experience" with BASIC, FORTRAN, and/or other similar languages. Some formal background in college level algebra seems to be helpful in learning and mastering some of the primitive functions of APL. It takes a spirit of "adventure" and "discovery" and an easily available computer, i.e., a personal computer, to learn and practice APL. I have enjoyed learning, using, and teaching APL. The most remarkable thing about APL is that 14 years after first being exposed to it, I still enjoy learning, using, and teaching this unusual, and still evolving, language.

IBM's APL FOR IBM PC

To implement IBM's APL on an IBM PC you need (1) a graphic's monitor and card, (2) a 8087 chip - the numeric co-processor, and (3) a graphics printer. The APL software provided by the vendor includes the basic language interpreter, and two types of facilities called (a) application workspaces, and (b) auxiliary processors. There are five applications workspaces to help the user with

- printer management,
- full-screen editing of user defined functions,
- file management,
- asynchronous communications, and
- music samples.

There are six auxiliary processors to manipulate various devices of the PC:

- AP80 for graphics-printer control,
- AP100 for BIOS/DOS interrupt handling,
- AP205 for management of screen display,
- AP210 for DOS file management,
- AP232 for asynchronous communications, and
- AP440 for music generator.

The package comes with a 318 page reference manual.[10]

PROGRAMS DEVELOPED AT GENESEO

Two general purpose programs have so far been developed for use by Geneseo students on an IBM PC for data analysis and problem solving.[11] Each program contains about 16 different modules.

GSLINE: This program performs a linear regression of F(y) on G(x), given a set of (X, y) data and algebraic forms (user specifiable) of F(y) and G(x).[12]

NEWTON: This program solves <u>any</u> algebraic equation in one unknown,

$$F(a, b, c, \ldots ; y) = 0,$$

where a, b, c, etc are known parameters, and y is the variable of interest.

Figures 2 and 3 show two examples of the use of NEWTON in solving physical chemistry probblems. The information displayed within square brackets is supplied by the user.

Figure 2 shows a solution of an equation,

$$y^3 + Ky^2 - y(k + cK) - kK = 0,$$

encountered in a more accurate calculation of the hydrogen ion concentration, y, in an aqueous solution of a weak acid. Here c is the molar concentration of a monoprotic acid with dissociation constant K, and k is the ionic product of water.

Figure 3 shows the use of NEWTON for solving the van der Waals equation for the molar volume of a gas.

```
                    Solve for y: F(a, b, c, ... ; y) = 0
                   DATA:  Type the following information:

 F(a, b, c, ... ;y)=[ (y*3) + (KXyXy) - (yXk+cXK) + kXK                 ]
 Derivative: dF/dy =[ (3XyXy) + (2XKXy) - k+cXK                          ]
 Initial guess: y0 =[ (KXc)*0.35              ]  Convergence Threshold[ 1E-14 ]

                PARAMETERS                            ITERATIONS
     ==============================       ==================================
     K=[ 1.00E-18 ]    [          ]        1    1.59882E-007
     k=[ 1.00E-14 ]    [          ]        2    1.22571E-007
     c=[ 0.100    ]    [          ]        3    1.05014E-007
       [          ]    [          ]        4    1.00338E-007
       [          ]    [          ]        5    1.00002E-007
     ==============================        6    1.00001E-007
                                           7    1.00000E-007
 THE ROOT IS:    1.00000E-007
 Converged in:    7 Iterations
 F(y) at the root =   0.00000E000
 |===================================|    ==================================
 |Ent: To continue          Esc: To exit|
 |Tab/Cursor Keys(←↑↓→): To locate field|
 |     & type/change data in it         |
 |===================================|
 c-BJoshi/84
```

Figure 2

```
                    Solve for y: F(a, b, c, ... ; y) = 0
                   DATA:  Type the following information:

 F(a, b, c, ... ;y)=[ ((P+a÷y*2)Xy-b) - RXT                              ]
 Derivative: dF/dy =[ P - (a÷y*2)X1-2Xb÷y                                ]
 Initial guess: y0 =[ RXT÷P                   ]   Convergence Threshold[  1E-8 ]

                PARAMETERS                            ITERATIONS
     ==============================       ==================================
     P=[ 2X101325 ]    [          ]        1    1.10877E-002
     a=[ .364     ]    [          ]        2    1.10877E-002
     b=[ 4.27E-5  ]    [          ]        3    1.10877E-002
     R=[ 8.314    ]    [          ]
     T=[ 273.15   ]    [          ]
     ==============================

 THE ROOT IS:    1.10877E-002
 Converged in:    3 Iterations
 F(y) at the root =   0.00000E000
 |===================================|    ==================================
 |Ent: To continue          Esc: To exit|
 |Tab/Cursor Keys(←↑↓→): To locate field|
 |     & type/change data in it         |
 |===================================|
 c-BJoshi/84
```

Figure 3

*Department of Chemistry
State University of Arts and Science
Geneseo, NY 14454

BOOK REVIEWS

With this issue, it is a pleasure to welcome two new reviewers to the column, Richard Uriarte and Mike Moffitt. Mike recieved his Ph.D. from Ohio State in Biochemistry. He taught Chemistry and Computer Science for six years at Stockton State College in New Jersey before joining the staff at Texas A&M University Agricultural Extension Service, where he is currently employed as a computer systems engineer. Mike words with a LISA and an Apple IIe at home and uses various BIM and IBM clones at work.

Rich also received his Ph.D. from Ohio State, but he is an inorganic chemist. He is now an Associate Professor of Chemistry at St. Peter's College in New Jersey, teaching Inorganic and General Chemistry. Apparently Rich is spending much of his time on his new Mac, but he is also currently working on a computer workbook for possible use with a freshman chemistry text. We will be looking forward to future contributions from both Rich and Mike.

Since the last issue of the Newsletter, several readers have written to volunteer to serve as reviewers, and we will be hearing from them in future issues. If there are other readers who wish to review for this column, please write to me: Dr. Harry E. Pence, Department of Chemistry; SUNY-Oneonta, Oneonta, NY 13820, for further information and assistance. As always letters of comment and suggestions of books for review are also welcome.

THE APPLE MACINTOSH™ BOOK
by Cary Lu
Microsoft Press, Bellevue, WA
1984, 383 pages, paperbound, $18.95

Reviewed by Richard J. Uriarte*
and Michael J. Moffitt**

The Macintosh™ is the kind of computer that generates strong reactions. Depending on your attitudes, it's either a liberating marvel because it requires so little actual thought to operate, or a case of techno-pablum, again because it requires so little thought to operate. Most of the books about the Macintosh™ have been written by people clearly in the prior camp; they seem to read like extended travel brochures. Perhaps this is because they were written after brief exposure to the Macintosh™ (and its mass of promotional hype), rather than after in-depth experience. The Apple Macintosh™ Book, by Cary Lu of Microsoft Press, suffers from much less problem of focus, even though it came out near the release date for the Macintosh™, probably because the folks at Microsoft had Macintosh™ to work with for over two years before the rest of us.

This graphically stunning book seems to be several books in one. The first part of the book details how to set up a Macintosh™ out of the box -- useless to anyone who doesn't have a Macintosh™ , and equally useless to anyone who already does. It does have nice pictures, though. Other chapters in the first part of the book also seem to suffer from this "not enough if you already know -- too much if you don't" problem, as Lu runs through simple operations and use of Apple-provided software. Short chapters dealing with two Microsoft products, Multiplan™ and Chart™ , are sound, though very cursory. The chapter on word processing is interesting in that it compares the free MacWrite™ with Microsoft Word™ . It is interesting that almost a full year after the publication of The Apple Macintosh Book Microsoft Word™ is finally available on the Macintosh™ . A section on business applications was a bit premature, since only recently have accounting programs begun to appear.

The chapter on programming languages gives some indication of the frustration that Macintosh™ owners have felt. Of five different packages discussed by Lu, only one is current - MacPascal™. As of March, 1985, Apple had not released MacBASIC™, MacLOGO™, or the Assembler/Debugger, and Microsoft had already made extensive improvement of its BASIC, resulting in a 2.0 version much more elegant than that discussed by Lu.

The second part of the book gives a great deal of insight into how the Macintosh™ actually works; more insight, in fact, than is available from Apple. There are separate chapters on the hardware design, the video screen, the keyboard, the mouse, disk drives, I/O ports, printers, and modems. Much of this information will be of interest to both experienced and inexperienced users, but detail is not quite on a level sufficient to engage the serious hardware hacker. For example: "The best way to connect a non-Macintosh™ printer is to find someone who has done it already and copy the technique exactly (page 207)." The section concludes with general comments on software design, advice on maintenance "getting help", and gives a fairly detailed comparison between the Macintosh™ and IBM PC (though IBM fans won't have gotten this far, I fear).

The final section of the book claims to "contain material for people with specialized needs or interests (page 281)." It does give some interesting suggestions on how to trace artwork, as well as advice on how to modify the Macintosh™ for use in a moving vehicle (bolt it down.) It also gives simple advice for use by the handicapped. There is a detailed discussion of communications and data-moving concepts, all of which are a little out of place because they are so much in the abstract. The Technical Topics chapter is the gem of the entire book because it contains exceedingly arcane information such as pinouts for the mouse connector and the wiring for an adapter cable to RS-232C devices. These alone could justify the purchase of the book for the really experienced MacUser.

All in all, this is a professionally produced work. It suffers a bit from trying to please two vastly different audiences (those who have seen the MacWay, and those who haven't ...yet), but it remains the most thorough and objective source on information yet available.

*St. Peter's College, Jersey City, NJ 07306
**Texas A&M University, College Station, TX 77843-2468

VISICALC FOR SCIENCE AND ENGINEERING
by Stanley R. Trost and Charles Pomernacki
SYBEX, Berkeley, CA
1983, paperbound, $13.95

Reviewed by Harry E. Pence
and Celene DiFrancia*

Although VisiCalc is best known as a spreadsheet program for business applications, it can also be extremely useful in some chemical problems. Especially in analytical chemistry laboratories, it is used to do repetitive caluclations, organize and correlate data, develop reports, and identify data trends. In fact, one of the reviewers (C.D.) has done exactly this type of work in an industrial laboratory during the summer and vacations for the past year.

Trost and Pomernacki have written this book to demonstrate some of the possible scientific uses of VisiCalc. The authors preseume that the reader already has a working knowledge of VisiCalc, although a brief summary of the necessary commands is provided in an appendix. This assumption should create few difficulties since there are several books available that describe VisiCalc and other similar spreadsheet software.

The major emphasis of the book is on creating design tables or spreadsheets that can be manipulated with routine VisiCalc commands. The sample spreadsheets discussed are intended to both serve as a library of useful applications as well as demonstrate how the techniques can be extended to other situations.

The authors have chosen to draw their examples from a widely divergent group of scientific fields, including not only statistics, communications, heat flow, optics, and simple mechanics, but also electrical, solar, mechanical, and civil engineering. Over fifty examples are provided, and in each case the approach is the same. First a specific situation is outlined, then the appropriate data is provided, and finally instructions are given for a sample problem. The instructions are clear, although a little dry. No unexpected results were discovered in running a number of these examples.

It is unfortunate that so few of the problems selected are directly related to chemical applications. More important, the focus on specific situations tends to somewhat blur the generalities that might identify broader implications of the method. Reading this book will help to develop a general appreciation of the power of VisiCalc and similar spreadsheet analysis software, but it is left to the individual to determine how these principles might relate to chemical problems.

The chemist who wishes to pursue the possibilities of spreadsheet analysis further is left with few alternatives. A brief bibliography is included, but few of the books listed seem likely to deal with chemical problems. In short, the reader must accept the fact that further work must be done independently.

Obviously, this book leaves something to be desired for chemists, but as the old saying goes, it may be the only game in town. Despite the lack of chemical applications, the book can suggest possible chemical uses of VisiCalc. For some readers, the book may be a worth-while source of information and ideas, but others may choose to work on their own or wait until a more directly applicable book becomes available.

*Department of Chemistry
SUNY-Oneonta, Oneonta, NY 13820

MODULA-2 FOR PASCAL PROGRAMMERS
by Richard Gleaves
Springer-Verlag, New York, NY
1984, 145 pages, softcover, $16.95

Reviewed by Brian Pankuch*

Modula-2 is Niklaus Wirth's newest programming language. Pascal, his first language written in the late sixties, was designed principally for teaching structured programming. Somehwat to Wirth's surprise, people began using Pascal as the language of choice in many programming applications. As might be expected, Pascal does not contain many of the ingredients needed to make it a general purpose language. Modula-2 is a general purpose language designed primarily for writing software systems. It introduces changes which allow simpler programming with improved readability and efficiency. The scope of the language is greater than Pascal, since it has been extended to include system design and machine-level programming.

As indicated by the title, this book assumes a knowledge of Pascal. It is divided into three main parts. The first, "New Concepts", was the hardest for me to understand since, of course, it was the newest. All parts of the book are written with many short programming examples, but in this section they were not sufficient for me to develop a good feeling for why each one of the innovations was being made. If I had Modular-2 available, I'd play with the short programs and make a few changes in them to see what happens. This should make the concepts more concrete and understandable.

Modules are similar to units in Pascal. The objects imported and exported to modules must be more carefully defined and are made visible only where needed. This limits the scope of imported objects and makes a module or program that uses the module less likely to give unexpected results.

Modules also seem as if they might be easier to maintain, since it is possible to change some modules without recompiling others in the system. In Pascal, changes in a unit that is called by other units requires each of these units to be recompiled and replaced in the library. It is less than clear to me exactly when this is necessary in Modula-2, but it does appear that the library would be recompiled and restructured less often.

Easier access to machine level programming is important in speeding up a program. Finding where the program is bogged down and optimizing that portion of the program in machine language generally produces significant speed-ups in program execution. This is especially noticeable in graphics and applications involving extensive calculations.

The second part of the book, "Differences from Pascal", was much easier for me to follow. Virtually all of the changes made sense to me. Most of these improved program clarity or ease of use. There were several specific examples that struck me as I was reading. Modula-2 is case sensitive, and so using upper or lower case makes a difference. For instance in Pascal, FIRSTNAME, firstname, or FirstName would all be the same variable; in Modula-2 each is different. The last version, FirstName, is the preferred form. All characters in a name are significant, not just the first eight as in most versions of Pascal. All reserved words (those defined in Modula-2) must be capitalized. The result is that many techniques Pascal programmers have normally used to make their programs more readable and legible are now required in order for the program to compile.

IF, FOR, and WHILE have their own END instead of being used with BEGIN-END pairs. Comments can be nested without having the compiler write strange error messages or commenting out critical parts of a program. Strings and characters can be used more interchangeably. String utilities are provided that include all those used is UCSD Pascal, so the result is not only more flexibility in using strings, but also more freedom.

A new type, CARDINAL, has been added. It is an unsigned integer and is used in place of the INTEGER, which is now used only when a VARIABLE can be negative. This would appear to allow the compiler to do more checking if the programmer will initially think out the variable possibilities.

The CASE and variant RECORDS now include an ELSE statement, which will catch unspecified case values. This is handy when using a case selector which is of an unallowed type but isn't specified in the CASE label list. Other changes, such as putting vertical bars in front of CASE label, make the program more readable.

The different types CARDINAL, REAL, and INTEGER are not permitted in the same expression, so it is necessary to decide on the type of the answer and then change all types to explicitly agree with this one type. Fortunately, utilities are included which allow one to change explicitly from one type to another.

There are several other helpful changes. ELSIF allows modifications to the IF THEN statements, so that the FOR statement can have a step value other than 1 (which is still the default value). Procedures can be called before they are declared. File handling seems much more sensible and more specific, with more checking and probably less possibility for error.

The last section contains the utilities. These seem to give much more control over handling errors as well as string handling that is, as noted earlier, very similar to UCSD capabilities. All the input-output procedures for text file or terminal access seem well thought out with more error checking than I'm used to, although in actual use they may not perform as well as expected.

Overall, the book offers many hints and warnings about how to use or not misuse the various aspects of Modula-2. Those who have had previous experience with Pascal should probably be writing simple programs in a few hours with this book. I would recommend beginning by going through the first and third parts very rapidly to gain a general overview. After that, it would be best to concentrate on part two, actually write some simple things, and play with some of the examples. Learning the syntax should occur quickly since it is very similar to Pascal. For the newer concepts I think it would take me a lot more than a few hours and probably more than what is covered in this book to understand them thoroughly.

For those who are thinking of getting involved with Modula-2, I'd recommend reading Gleaves' book before investing in a Modula-2 compiler. The book explains better than anything else I've seen what Modula-2 is all about. Although it takes some work, it is certainly a readable text.

*Chemistry Department
Union County College
Cranford, NJ 07016

THREE DEGREES ABOVE ZERO: BELL LABS IN THE INFORMATION AGE

by Jeremy Bernstein
Charles Scribner's Sons, New York, NY
1984, 241 pages, hardcover, $17.95

Reviewed by Brian Pankuch*

This book is an intriguing mixture of the history of technology and science with personal insights about some of the scientists who have made important discoveries. Bernstein provides extensive background material about many Bell scientists, beginning with their early research interests and continuing to the current work they are doing at Bell Labs. I was struck by the diversity of these individual stories. While some scientists seemed to move directly into their main research, others knew what they wanted to do but found it difficult to pursue their chosen field at the appropriate level until arriving at Bell. A great deal is said about academic freedom in higher education, but one can't help but be impressed by the freedom many of these scientists feel they have. The combination of basic research, technology, development and the freedom to pursue personal interests is not only satisfying to the individual but also a very successful approach. Seven Nobel Prizes in Physics alone have been won by Bell Lab scientists.

The four main parts of the book are mostly concerned with the solid state and fiber optic areas, but considerable chemistry and material science seep in. Many sections of the book offer impressive descriptions of the latest technology. Bell is testing fiber-optics which transmit 420 million bits/sec. Using these techniques, a thirty volume encylcopedia could be transmitted in one second, making a mistake in only a single letter.

Among the computer sagas is the story of Belle, a very capable chess playing computer who beats 99 out of 100 human players and is providing insight into how people think when they play chess. It seems that part of what grandmasters do is to see the entire chess board as a single entity. That strikes me as being a bit like thinking of NaCl as a chemical with many properties, i.e., a whole entity. It isn't just a group of letters that are laboriously put together following a bunch of rules to make a compound. In otherwords, this research may give us insight into the learning process and how it can be improved.

The stories of some of the major discoveries are especially interesting. Some researchers identified their major interests quickly; others seem to be continually searching for new areas, whether in the area of computer identification of human speech or the photovoltaic effect in silicon. The latter discovery occurred by chance when Russel Ohl placed a cooling fan in front of a bench light, chopping the light. The voltage shown on an oscillosocpe followed the chopped light. Instead of just moving the light, he began to study the effect, and fifteen years later the first solar battery resulted from the research begun in this way.

Many other discoveries resulted from the same combination of inspiration and hard work. Shockley and Bardeen were able to work on the transistor because of a technological breakthrough. They combined new physical insights with an improved mathematical theory to produce greater understanding and further technical breakthroughs.

Managers at every level at Bell Labs have very strong backgrounds in science and technology but very different attitudes toward their jobs. Addison White comments on his position at Bell "...the jobs that management people should be doing are fairly dull. So the higher my responsibility became, the duller the work was." Another manager, Arno Penzias, Nobel Prize winner in Physics and Director of Research, has a different slant. "The nice thing about each of the administrative jobs that I have had is that the previous occupant retired, so that I was able to redefine the job. In each case I became convinced that I had the best conceivable job, not just in Bell Laboratories but probably in the world of science ... Each new job turned out to be marvelous."

Some of those interviewed expressed concern that the break-up of AT&T may create more pressure for short term results and less emphasis on basic fundamental research which has a longer term payback. As might be expected, some individuals are mostly concerned with continued funding of personal interests, but others have a general anxiety that the strange magic that makes Bell Labs what it is may not continue. With the freedom at Bell and a Director of Research with an attitude like that expressed by Penzias, one feels that Bell Labs will continue to do very well in the future.

This gives some of the flavor of the book, good science and technology laced with interesting personal anecdotes. In many ways, these scientists and engineers come across like the rest of us. They have a great deal of uncertainty as to the future and are seeking balance and satisfaction in their work. The difference is that much of their work is literally at the frontiers of current science. Bernstein's book not only suggests that this fantastic group of people will continue their balancing act in the future but gives the reader some very positive feelings about what science and technology are creating.

*Chemistry Department
Union County College
Cranford, NJ 07016

BASIC COMPUTER PROGRAMS IN SCIENCE AND ENGINEERING

by Jules H. Gilder
Hayden Book Co., Rochelle Park, NJ
1980, 256 pages, softcover, $8.95

Reviewed by Harry E. Pence

There are a wide variety of computer books with titles that indicate that they are designed for scientists and engineers, and the purposes of such books vary equally widely. Glider has written this book to fulfill two major goals; it is intended to be a sourcebook of mathematical, statistical, and electronic engineering programs and to provide examples that may be used as models by those just learning to write computer programs. It accomplishes both of these ends rather well.

This book contains 114 scientific programs designed to be run on an Apple II microcomputer using Applesoft II BASIC. A machine language sub-routine to clear the screen appears in each program, but otherwise there appears to be little that is limited to either a special version of BASIC or a specific computer. There should be no difficulty adapting any of these programs to other systems. A set of two diskettes which includes all of the programs may be purchased with the book. Most of the programs are relatively short (average length about 40 lines), but those who plan to use the book extensively may find this option to be worth considering.

The library of programs provided can be roughly divided into two groups. The first group, consisting of 49 programs, deals with general mathematical principles and data analysis. These programs involve techniques such as matrix manipulation, various least squares treatments, calculations with complex numbers, Gaussian elimination, Simpson's Rule, solution of simultaneous equations, and the Newton-Raphson Method.

The other group of programs are concerned with calculations in electricity and electronics. This group begins by solving a number of potentially useful basic electricity problems dealing with simple wiring, resistor configurations, and power transformers. These are followed by basic electronics programs that perform similar calculations. The remainder, and probably the real meat of the book, deals with electronic circuit design. Over fifty programs relate to the design of basic circuits, various types of active and passive filters, attenuator pads, and communications circuits. These programs may be especially useful to not only the hobbyist but also the professional electronics technician.

In each case, the author provides a listing of the program, a print-out from a typical run of the program, and a very brief explanation of the procedure involved. The background provided is adequate for someone already familiar with the technique but probably not for most novices. The abbreviated explanations, lack of exercises, and the absence of even a brief bibliography will significantly limit the use of the book as a text.

The programs themselves seem to be error-free and straight-forward. There is little error trapping, and mistakes usually cause an exit from the program. Since many of the programs are rather short, this doesn't constitute a serious liability. The book does offer an extensive library of potentially useful programs and so does seem to fulfill the author's stated purposes reasonably well.

The usefulness of this book will vary significantly from reader to reader. It could be valuable for someone doing electronics work, but it would probably have serious liabilities if used as a textbook, and many chemists may find that the majority of the programs are not related to their normal activities. Under the circumstances, no general recommendation is possible, but each reader must decide whether or not the book will satisfy individual needs.

*Department of Chemistry
SUNY-Oneonta, Oneonta, NY 13820

STRIDE MICRO
A MICROCOMPUTER REVIEW
by Brian Pankuch*

Stride Micro
5905 Energy Way
Reno, Nevada 89502 (702) 322-6868

Summary of standard model:
Price $2900
Memory 256 K
Disk drive - One of 640 K capacity, 68000 processor, Liaison Operating System, p-system 4.2 w/LAN (local area network), multiuser software.

Other needed equipment
Terminal $700 and up
Language: Pascal, BASIC $225 and up.

(Stride Micro was formerly Sage, and they changed their name to avoid legal barriers with other companies using the same name.)

Well I finally did it about a year ago. Got tired of borrowing a free machine from work and was finding more and more uses for a personal computer. I started by asking myself what I wanted and needed in a system. Most of my time is spent writing programs in Pascal. Some time is spent on reports for work and on reviews such as this. The p-system can be used as a primitive word processor. (Five keystrokes and the editor is in the word processing mode with the advantage that you use all the usual editor commands to do editing whether for a program or a report. The system does a few extra useful things for you such as keeping within margins when you add or delete text.) My need was a system which had Pascal and the p-system. No problem since you can buy both for both systems, but on most it costs extra. On the Stride the p-system comes with your purchase.

Another big factor was the large size programs I was writing, over 100 K in most cases. So I needed more space on the diskettes for storing parts of programs. Pascal is a compiled language; this means a program needs to be compiled each time I want to see a result on the screen. A very fast CPU will both speed up compile time and the rate at which the little lights (pixels) that make up the graphics on the screen can be turned on/off. The Stride uses the Motorola 68000 CPU whose speed allows the Stride to compare with a DEC VAX which is a much larger and more expensive machine.

The least expensive Stride gives you 256 K RAM expandable to 512 K (each K allows storage of 1000 characters) with a single disk drive. What a disk drive! It holds 640 K.

So I did it. Got a Stride with one disk drive. Instead of the standard 256 K memory I got an additional 256 K for a total of 512 K ($500 extra). The base price for the 256 K Stride is $2900. If that sounds too good to be true, it is. At this point you have a computer with the Liaison (p-system) operating system. You still need a terminal. Yes a terminal, not a monitor, and a language.

The Stride should work with any terminal configured properly. If it isn't one being actively supported by Stride then you better be sure your dealer is willing to configure it for you. Otherwise, it is a bit like pulling yourself up by your bootstraps. The terminal adds about $700 unless you already have your own terminal.

You have a wide choice of tools, i.e., languages and operating systems. Application programs are available, but you simply won't have anywhere near the choice of programs that are available for the Apple or IBM. Then how many word processors or spreadsheets do you really need?

For languages you have a choice of 68000 Assembly, APL, BASIC, C, CBASIC, COBOL, Forth, FORTRAN, LISP, Modula-2, Mumps, Pascal. They vary in price from $225 and up. If you wish to purchase additional operating systems you again have wide choice - CP/M-68K, RM-COS, UNIX V, Idris, HyperFORTH, PDOS, BOS, Mirage, MOSYS, TRIPOS, Mumps.

You might wonder if you really need 512 K RAM? No you don't need it, but if you have a second terminal available you can have two users on one Stride with the 512 K. The software for two or more users is part of the standard package, but a second disk drive would be an idea. I haven't used this feature, but I understand it can be difficult to hook up.

Have you heard of the 'ram disk'? This allows you to put aside part of the computer's memory as a 'ram disk'. On my machine I set aside 385 K of computer memory for a 385 K 'ram disk' that exists only in the computer memory. The 'ram disk' is noiseless and blindingly fast. Although it requires two transfers to get all the information from a real 640 K diskette drive to the 'ram disk' then to another diskette (as in making back up copies) I find I rarely need to transfer the entire 640 K and one transfer is sufficient.

How about the speed of the overall machine? I was using a 16 bit Terak which is about 5 to 6 times faster than an Apple on most operations and raced the Terak against the Stride. A 32 K program I had written was put into the editor then compiled, linked and run. The Stride was over 7 times faster, which makes it a lot faster than an Apple. When you're compiling and recompiling large programs this is a time saver. When you are doing a large number of calculations or graphics the speed is really appreciated.

Not so many years ago it was popular to compare your car's 0-60 mph time to others to see who had the fastest car. Times change but not people. Now I'm the kid with the biggest, fastest computer in the neighborhood. People always want to know the speed, amount of RAM memory and the amount of storage on the disk drive. The Stride is a winner on all three.

For my uses, primarily programming, the system works fine. Some of the disadvantages I have found over the last year come from support. When you have a problem the first step is to ask a friend. No one I know has a Stride. Next step is the manuals. On a scale of 1-10 (with 10 the best), I'd give the manuals a 3. Supposedly new manuals are available which can be read by someone who didn't invent the Stride, but I haven't used them. The next step would be your local dealer. There have been three different dealers in the last year in my area and not one is able to answer questions. Next, the Stride offices. In the course of writing this review it occurred to me that people teaching chemistry would be interested in word processors which allowed writing chemical formulas. This means sub and superscripts in the word processor. First I called the Boston office, they thought it would be possible. Trying to get it more definite, I called the Reno office; they didn't think it could be done.

So if you buy something, make sure you see it work. Not to be too hard on Stride, their people are always courteous and will usually tell you frankly whether they know an answer or not. It is a problem for all companies. By the time an employee knows the answers to the tough questions, they are too valuable to be answering the same questions over and over again. Understanding the problem may help your blood pressure, but this still leaves questions unanswered. For instance, it took me over two weeks to find a way to transfer programs from the Terak to the Stride. (Once transferred they worked fine. It was the transfer process which was the problem, not the p-system or Pascal.)

Another problem I find is that my screen frequently freezes. You have to reboot and when you do that you lose everything in RAM memory including 'ram disk'. This means you have to back up to the diskette often. This p-system seems less robust and more fragile than those on the Terak or Apple.

I have only talked about the smallest machine. For a price you can get up to 3000 K (3 meg) of memory and can add hard disks from 10 to 448 meg. You can add a lot of users to machines of this size. Stride is a real mover and seems likely to keep up with the latest hardware and software innovations. Two other factors which sold me on buying a Stride were a 20% educational discount, and a deal where you could trade in your old machine toward a new one getting the present value of the old machine toward the new one. I was disappointed to find in my last contact with Stride that they are no longer honoring this last commitment.

In February 1985 Stride announced a graphics board for $400 which sounds fast and versatile. I haven't seen it yet.

In summary, if your needs are similar to mine, I would seriously look at this system. Be very sure there is strong local dealer support. Get everything set up and running the way you want before walking out with the machine. Actually that's a good idea with any purchase, isn't it?

*Department of Chemistry
Union County College
Cranford, NJ 07016

WORKSHOPS, MEETINGS, CONFERENCES & COURSES

Please send information to Donald Rosenthal, Editor. Describe the program, include location sponsoring group, dates, costs and who to contact for further details (name, address, and phone number). Information should be sent as far in advance as possible.

July 28 - August 1: Seventh C.C.C.E. National Computer Workshops - East at Clarkson University, Potsdam, NY 13676
Six intensive workshops are planned. See detailed information elsewhere in this Newsletter. Contact: Donald Rosenthal at the above address (315-268-6647).

July 29 - August 2: World Conference on Computers in Education in Norfolk, VA.
Contact Gerald Engel, WCCE/85, Department of Computer Science, Christopher Newport College, Newport News, VA 23606.

August 2 - 5: CHEM ED '85 at Montclair State College, Upper Montclair, NJ.
Workshops, demonstrations, presentations, poster sessions and computer sessions. Individual presentations demonstrating effective use of currently available software and hardware which can be used in the classroom. Approaches to problem solving as illustrated by TK! Solver (Alan Smith) and George (Dick Cornelius). Computer Assisted Testing (Darell Beach). Workshops on laboratory instrument interfacing; and spread sheets, word processors and other aids to the chemistry teacher. A C.C.C.E./SERAPHIM software evaluation center will operate throughout the conference. Contact CHEMED '85, C/O Dorothy Lehmkuhl,56 Normal Avenue, Upper Montclair, NJ 07043 for further details.

August 11 - 15: Seventh C.C.C.E. National Computer Workshops - West at Truckee Meadows Community College, 700 Dandini Blvd., Reno, NV 86512
Five intensive workshops are planned. See detailed information elsewhere in this Newsletter. Contact John Clevenger at the above address for additional details (Phone 702-673-7221).

August 14: SERAPHIM - C.C.C.E. High School Teacher Workshop at Pocono Environmental Education Center, Dingmans Ferry, PA 18328
This workshop has two basic objectives- To illustrate how microcomputers can be of use to a chemistry teacher and to provide an opportunity to preview representative samples of commercially available software. Jim Nelson (148 Schuyler Road, Springfield, PA 19064. Phone 215-544-6866) is the workshop leader. Contact Jack Padalino (RD #1, Box 268 at the above PEEC address for additional information. Phone 717-828-2319).

August 21: SERAPHIM - C.C.C.E. High School Teacher Workshop at Pocono Environmental Education Center, Dingmans Ferry, PA 18328
See the August 14 listing.

August 22 - 24: Personal Computer and STD Computer Interfacing for Scientific Instrument Automation, Washington, DC
A hands-on workshop with each participant wiring and testing interfaces. Directed by Mr. David E. Larsen and Dr. Paul E. Field. $450 for 3 days. For more information contact Dr. Linda Leffel, C.E.C., V.P.I. State University, Blacksburg, VA 24061. (703) 961-4848

Workshop repeated in Greensboro, NC September 19 - 21.

August 23 - 28: IUPAC-sponsored 8th International Conference on Chemical Education in Tokyo
Write J. T. Shimozawa, Chemical Society of Japan, 1-5 Kanda-Surugadia, Chiyoda-ku, Tokyo, 101 Japan.

September 8 - 13: 190th ACS National Meeting in Chicago, IL Symposia, general papers and exhibits. Those wishing to present papers must submit four copies of an abstract with the original on an ACS abstract form by May 1 to William F. Coleman, Department of Chemistry, Wellesley College, Wellesley, MA 02181 (617-235-0320, ext. 3129)

October 12: SERAPHIM - C.C.C.E. High School Teacher Workshop at Science Education Center, Emporia State University, 1200 Commercial St., Emporia, KS 66801.
See August 14 listing for details. Ken Hartman and Don Murphy are workshop Directors. Contact Dr. George R. Davis at above address for additional information. (Phone 316-342-7502)

October 18 - 19: First Eastern Small College Computing Conference at the University of Scranton and the Hilton at Lackawana Station.
Designed to promote a free exchange of information among small college personnel concerned with the use of computers in the academic environment. Intended to span all academic disciplines. Topics include software, simulation, course management, curriculum development, computer literacy, the computing laboratory and micro/mini/mainframe. Contact Professors Meinke and Beidler, ESCCC, University of Scranton, Scranton, PA 18510.

November 1: SERAPHIM - C.C.C.E. High School Teacher Workshop at SUNY A & T, Alfred, NY 14802
See August 14 listing for details. John Anderson (Chemistry Department at the above address) is workshop Director.

November 23: SERAPHIM - C.C.C.E. High School Teacher Workshop at Averill Park High School, Averill, NY 12018
See August 14 listing for details. Dennis Clancy is workshop leader. Contact Jim Rogers at above address for additional information. (Phone 316-342-7502)

Software QUERIES and REPLIES

Software QUERIES AND REPLIES should be sent to Ken Loach, Department of Chemistry, SUNY at Plattsburgh, NY 12901. (818) 564-2230. Hardware QUERIES and REPLIES should be sent to Jim Beatty, Chemistry Department, Ripon College, Ripon, WI 54971. (414) 748-8123.

WHO DONE IT?

WHO DONE IT? information should be sent to the appropriate section editor (Hardware or Software - see QUERIES).

WHO-123 (June '85)

The Education Committee of the ACS Division of Chemical Information is developing a general introduction to computer searching for a chemical literature course or for use in any of several subject areas (e.g., analytical, inorganic, organic and physical chemistry). These computer searching curricular modules will be available to all chemistry departments. Those who wish to receive this material as it is produced should write to Mrs. Arleen N. Somerville, University of Rochester and ask for the Computer Searching Instructional Modules. (D.R.)

WHO-124 (June '85)

The following papers were among those presented at the 19th ACS Great Lakes Regional Meeting at Purdue University, West Lafayette, IN on June 10-12, 1985.

S.L. Burden, N. Kastelein, P. Klunzinger (Chem. Dept., Taylor Univ., Upland, IN 46989): "Low-Cost Robotized Solution Preparation System for the Undergraduate Chemistry Laboratory".

T.W. Adams (Chem. Dept., Purdue Univ., West Lafayette, IN 47907): "Microcomputers, Project SERAPHIM and the Chemistry Teacher".

D.A. Neufelder (Chem. Dept., North High School, Columbus, IN 47201): "Computer in Lab and Classroom".

M.A. Kercher, G.M. Bodner (Chem. Dept., Purdue Univ., West Lafayette, IN 47907): "Authoring Languages for Writing CAI Programs".

K.S. Aniliker, J.D. Herron (Chem. Dept., Purdue Univ., West Lafayette, IN 47907): "Computer-Based Itembank Utilizing Commercially Available Data Management and Word Processing Software".

W.H. Stanton, Jr., R. Latham, J.D. Herron, B.S. Freiser (Chem. Dept., Purdue Univ., West Lafayette, IN 47907): "TK! Solver in Analytical Chemistry".

J.S. Martin (Chem. Dept., Univ. of Alberta, Edmonton, AL, Canada T6G 2G2): "Dialogue in Computer Assisted Instruction".

D. Malone (Chem. Dept., Clarke College, Dubuque IA 52001): "Microcomputer-Based Analytical Laboratory Program".

L.M. Julien, F.D. Williams (Dept. of Chemistry and Chemical Engineering, Michigan Technological Univ., Houghton, MI 49931): "Microcomputer Programming in First-Year General Chemistry".

G. Rhodes (Chem. Dept., Univ. of Southern Maine, Portland, ME 04103): "Protein Graphics on the Commodore 64 Microcomputer".

G. Wiggins (Chemical Library, Indiana Univ., Bloomington, IN 47409): "Teaching the Use of CAS Online at Indiana University".

S.G. Smith, L.L. Jones (School of Chemical Science, Univ. of Illinois, 601 S. Matthews, Urbana, IL 61801): "Computer-Assisted Videodisc Instruction in Chemistry".

V.L. Maffel, R.E. Pressler (Eli Lilley Co., Tippecanoe Labs., P. O. Box 685, Lafayette, IN 47902): "Clarifying Chemical Data Using Computer Graphics". (K.L.)

WHO-125 (June '85)

The following papers were among those presented at the 189th American Chemical Society Meeting, April 28-May 3, 1985 at Miami Beach.

J.W. Moore, E.A. Moore (Eastern Michigan Univ., Ypsilanti, MI 48197): "Project SERAPHIM's Workshop Leader Training Program".

W.R. Robinson (Purdue Univ., West Lafayette, IN 47907): "Use of Spreadsheet Programs in Chemistry".

P.L. Samuel (Boston Univ., Boston, MA 02215): "Strategy, Graphics and Problem Solving: Design of CAI Software".

J.W. Judkins, M. Sullivan, J.J. Lagowski (Univ. of Texas, Austin, TX 78712): "Computer-Simulated Laboratory - A New Concept in Computer Aided Laboratory Instruction".

J.W. Judkins, J.J. Lagowski (address above): "An Example of the Integration of CAI with the Undergraduate Laboratory".

R.J. Hanrahan (Univ. of Florida, Gainesville, FL 32611): "Introducing Use of Microcomputers in Undergraduate Physical Chemistry".

M.D. Johnston, Jr. (Univ. of South Florida, Tampa, FL 33620): "Nonlinear Regression and Multi-Step Chemical Equilibria".

R.D. Hester, R.A. Campbell (Univ. of Southern Mississippi, Hattiesburgh, MS 39401): "Inexpensive, Portable Data Acquisition System Used in Friction Reduction Studies".

L.M. Julien (Michigan Technological Univ., Houghton, MI 49931): "Interactive Simulated Calorimetry".

A. Duffield, P.J. Mitchell (Loughborough Univ. of Technology, Loughborough, England): "Use of a Commercially Available Microcomputer for Electrochemical Experimentation".

R.E. Dessy (Virginia Polytechnic Inst. and State Univ., Blacksburg, VA 24061): "Electronic Laboratory".

R. Langridge (Pharmaceutical Chemistry Dept., Univ. of California, San Francisco, CA 94143): "Molecular Graphics: Computer Assisted Insight and Reasoning in Three Dimensions".

J.W. Moore, E.A. Moore (Eastern Michigan Univ., Ypsilanti, MI 48197): "Mainframes Here; Minis There, Micros Everywhere! Computers in the Chemistry Curriculum".

W.M. Butler (Univ. of Michigan, Ann Arbor, MI 48109): "Computer Aided Instruction of the Future". (K.L.)

WHO-126 (June '85)

The following papers were among those presented at the 19th ACS Middle Atlantic Regional Meeting at Monmouth College, NJ:

J.M. Berthoumieux, I.A. LaCava (Chemical Engineering Dept., CUNY City College, New York, NY 10031): "Real Time Complex Dynamic Simulation in Chemical Process Control Education".

T.R. Stouch, P.C. Jurs (Chemistry Dept., Penn State Univ., University Park, PA 16802): "Computerized Definition, Display and Comparison of Molecular Shape, Volume and Surface Area." (K.L.)

WHO-127 (June '85)

The American Biology Teacher, January 1985 is an issue with the theme "Software Development in Biology Education", which may be of interest to users of chemical CAI. It includes papers on simulation, the SUMIT Courseware Project, interactive videodisc applications and cluster-analysis of experimental data (with Apple II Basic source code)". (K.L.)

WHO-128 (June '85)

CET Research Group (P. O. Box 2029, Norman, OK 73070; (405) 360-5464) are offering three utilities for data-analysis: NNLSQ for non-linear least-squares function fitting; SEQS for solution of up to 20 simultaneous non-linear equations: French Curve for cubic-spline fitting of curves to data sets of up to 600 x,y data points. The standard utilities are for the Apple II. NNLSQ is also available or under development for CP/M, PDP11 and HP-85 systems. (K.L.)

WHO-129 (June '85)

C.R. Rhyner (College of Enrivonmental Sciences, Univ. of Wisconsin, Green Bay, WI 54301) has applied a VIC-20 and an E and L Analog Interface Designer (AID-1) to the generalized I/O and analysis of laboratory data. It is suitable for high-school or college teaching. (K.L.)

WHO-130 (June '85)

M.C. Cheney and E.J. McHale (Corporate Research, The Foxboro, Co., D-330 Bldg. 01-2, Foxboro, MA 02035; for details send SASE (#10) to M.J. McHale) have designed and used a simple I/O interface for a Commodore 64, with a large number of analog and binary inputs and outputs. It has been applied successfully as a general purpose data-logger and instrument controller. (K.L.)

WHO-131 (June '85)

There are two articles of general interest in American Laboratory, May 1985.

J.R. Purvis (III) has reviewed the applications of the General Purpose Interface Bus (equivalent to the IEEE-488, ANSI MC1.1 and IEC 625-1 buses) as a broadly-applicable interfacing device for instruments and computers.

R. Eisberg and V. Elings describes the application of the TECHFONT System (Santa Barbara Technology) and an IBM PC to the creation and word-processing of text containing special technical symbols. Up to 128 characters (in addition to the 128 characters of the ASCII standard set) can be created and printed. (K.L.)

WHO-132 (June '85)

Here are two books on CAI techniques:

D. Berentes: Apple Logo, A Complete Illustrated Handbook, 1984, Tab Books, Blue Ridge Summit, PA 17214.

R. Landa: Creating Courseware, A Beginner's Guide, 1984, Harper and Row, New York, NY 10022.

and here are two books on interfacing techniques:

B.E. Cline, An Introduction to Automated Data Acquisition, 1983, Petrocelli Books (Van Nostrand-Reinhold, NY 10020).

M.D. Seyer, RS-232 Made Easy: Connecting Computers, Printers, Terminals and Modems, 1984, Prentice-Hall, NJ 07632. (K.L.)

WHO-133 (June '85)

Flexible IEEE-488 Interfacing. Keithley Instruments is offering a new 193 Digital Multimeter with a build in IEEE-488 Interface. The intrument is programmable and has the ability to store up to 500 readings. It is a 6 1/2 digit DDM and has the capability to record up to 1000 readings per second at 3 1/2 digits. The specifications are too detailed to give justice to here. It has thermocouple and platinum resistance thermometer linearization capabilities built in. Programmable parameters include: Range, Function, Zero, DB, Reference, Trigger, Delay, Scaling, Calibration, Status, Self Test, and Output Format. It is intended to be a high performance flexible amplifier and A/D converter for data collection. Cost is $2,100. (J.B.)

WHO-134 (June '85)

Blackbox Problems. Almost anyone who works with computers soon faces the problem of connecting one piece of computer hardware to another. You may meet this problem when you wish to connect your microcomputer to a printer, another micro or your mainframe. You find that your output is EIA RS-232C and the input on the device you wish to connect is Centronics parallel, IEEE-488, 20 ma current loop, etc. Unless you buy components specifically made for your computer, you may find that even if you have the same type output-input on two of your devices they are not compatible. Some microcomputer manufacturers design components so that only their components will work without modification. RS-232-C frequently gives you gender and cross over problems for pin 2 does not connect to pin 2, etc. For those who may wish to purchase line drivers, RS-232-C/current loop converters, port expanders, patch cables with ability to identify configuration and books to solve many problems, I recommend the Black Box Catalog obtainable on request by writing:

Black Box Corporation, P. O. Box 12800, Pittsburg, PA 15241-99809

The catalog contains a useful technical section and glossary. If you know what you want, you often can obtain a protected and documented black box for about the same cost as purchasing the components separately. (J.B.)

NEW AUTHORING TOOLS FOR THE PC
by George Gerhold*

Last fall IBM released a new version of the PILOT authoring language under the name IBM PILOT. PILOT is an authoring language which has been implemented on a number of microcomputers. With a minimal amount of editing, it is practical to move Apple Super PILOT programs to PC machines and, in that process, to speed them up by an order of magnitude.

What features justify the term "more powerful?" Speed and string space are two immediately obvious items. User defined characters are essential in chemistry, and IBM PILOT offers 128 without removing any of the standard 96. All characters are accessible to both the student and the author. Byte-addressable files make data retrieval and record storage more flexible. Auto-type conversion between strings and numbers frees the author from the bother of programming around that distinction. This version also removes restrictions on the use of variables; subscripted variables and expressions can be used where earlier implementations required simple variables.

A whole range of supplementary items for use with IBM PILOT is being developed at Western Washington University as part of a three-year contract with IBM. Some of these are routines which can be made part of PILOT programs immediately. Routines to display compressed graphics (ten full screens per second are flashed on or wiped on in a variety of styles), a routine to allow use of DOS commands (thus making PILOT a front-end for other language), a routine to allow simple text-editing of multi-line student responses, and a routine to control a video disk are already available. Some of the supplements are editors to make the use of IBM PILOT easier. A text editor which contains an imbedded character editor and which displays the new characters is available for creating and editing PILOT programs. A sprite editor for definition of objects for animation is being tested. Sprites are defined by cutting regions from a graphic display; sizes range from 32 x 32 to 64 x 128 pixels. The graphic displays for sprite definition and for other purposes are created using any of a number of standard graphic editors (e.g., PC Paint, Dr. Halo, PCPG, Pencept, Koala Pad); we supply the importing information and software. A frame editor, which will speed the process of screen design, will be available soon. Some of the supplements are additions to the PILOT language. A detailed list would make sense only to the experienced PILOT user.

There is another class of items which make the authoring process easier. Examples include training materials, documentation, and exemplary code which can be reused. A six-hour on-line tutorial on the PILOT language and a two-hour on-line tutorial on screen design are available. Documentation for video disk interfaces and for use of various graphic editors is in preparation. Sections of code which can be imbedded in authors' programs will be available; examples include implementation of various character sets, record keeping sequences, user-controlled escape and help sequences, and templates for standard instructional dialogue sequences.

Authoring systems are attractive to some authors. Unfortunately, many authoring systems are so restrictive that the ambitious author soon outgrows their capabilities. If there is not an underlying language, the author must start from scratch. We will be producing an authoring system which generates PILOT code. This will be an umbrella which includes many of the above editors and which allows the author to move back and forth easily between programming and using the automated authoring system.

The combination of all of these items will lead to an authoring tool which will make practical the development of extremely sophisticated and effective instructional materials.

*Associate Dean
College of Arts and Sciences
Western Washington University
Bellingham, WA 98225

NATIONAL COMPUTER WORKSHOPS—WEST REGISTRATION FORM

Name ______________________

Address ______________________

____

____

Phone ______________________

WORKSHOPS (turn over for details)

1st Choice ______________________

2nd Choice ______________________

☐ Check here if you wish to present a computer related poster session, exhibit software or provide public domain software to share.

REGISTRATION FEE
$130.00

ADDITIONAL EXPENSES (see fold to right)
Send your room reservations directly to:
Nendels Motor Inn
55 E. Nugget Avenue
Sparks, NV 89431
(702) 358-6900
by July 28, 1985

Send this form and a check or money order in U.S. funds (payable to: Board of Regents) for the REGISTRATION FEE ONLY to:

Division of Arts & Sciences
Truckee Meadows Community College
7000 Dandini Blvd.
Reno, NV 89512

SEVENTH C.C.C.E. NATIONAL COMPUTER WORKSHOPS WEST

August 11-15, 1985

Sunday—Do your own thing!

2-7 p.m.	Registration and sign-in-at Nendels
7 p.m.	General Introductory Meeting-at Nendels
7:30 p.m.	Social and Get Acquainted Hour

Monday-Wednesday-Hit it hard!

9-12 a.m.	Workshop Sessions
Noon Lunch on Campus	
1-4:30 p.m.	Workshop Sessions
4:30-6 p.m.	Dinner on the town (of your own choosing)
6-8 p.m.	General Meetings and State-of-the-Art Presentations by Guest Presentors
8 p.m.-??	Seeing the sights in Reno

Thursday-Easing out of it!

9-12 a.m.	Workshop Sessions
Noon Lunch on Campus	
1 p.m.-??	Closing up Workshop Session Activities

Accommodations

Guests are encouraged to make reservations with Nendels Motor Inn who will provide single/double occupancy rooms for $28.00 per night. For those of you bringing families, there is no charge for children under twelve; a $6.00 charge for each additional person over twelve. A reservation card has been enclosed for you to make reservations directly with Nendels.

If you will be arriving via Reno Cannon International Airport free shuttle service is provided by Nendels from the airport to the Motor Inn.

Nendels Motor Inn has a swimming pool, cafe and lounge for its guests and arrangements can be made for free shuttle service to a few selected local attractions.

Nendels will also provide a shuttle service from the Motor Inn twice a day to the College for workshop sessions.

Meals

Breakfast and dinner may be purchased at very reasonable rates at the many restaurants and casinos in the Reno area. Lunch will be available for a nominal fee at the community college.

Workshop Registration

Enrollment in each workshop will be limited. The registration fee must accompany the Registration Form in order to reserve a place. You will be enrolled in only one of the workshops. Be sure to indicate which workshop you wish to register for on the Registration Form. Select a second choice, if you have one. Registrants will be notified if a workshop is full. The registration fee is refundable (less a $5.00 processing charge) upon request received prior to July 22.

Location

Reno is located in western Nevada at the eastern border of the beautiful Sierra Nevada Mountains. While Reno is famous for its casinos, it also offers cultural attractions such as symphony, ballet and theatre productions. Lake Tahoe, a popular resort area, is less than an hour away by car. Camping, hiking and fishing areas are also plentiful in the nearby mountains.

Workshops

Chairman of Organizing Committee: M. Lynn James, Department of Chemistry, University of Northern Colorado, Greeley, CO 80639 (303) 265-9242 or 268-2389.

Chairman of Local Arrangements Committee: John Clevenger, Department of Chemistry, Truckee Meadows Community College, 7000 Dandini Blvd., Reno, NV 89512 (702) 673-7221.

A. GETTING STARTED

Director: Pat McIntyre
Microcomputer Resource Center
Saint Martin's College
Lacey, WA 98503
(206) 438-4334

This workshop is designed to provide the background and hands-on experiences needed to start using a microcomputer as an instructional tool. The workshop will cover the following topics: computer care and maintenance, an introduction to programming (BASIC and PILOT), use of the computer for instructional support, and the fundamentals of hardware and software selection

Participants will learn:

- how to handle software packages
- a working computer vocabulary
- enough BASIC to interpret and modify programs
- how to write simple instructional programs in PILOT
- some of the advantages and limitations of the microcomputer environment
- how a microcomputer can make their life a little easier

No previous computer experience will be assumed.

B. COMPUTER GRAPHICS FUNDAMENTALS

Director: Jess (J.W.) Schilling
Department of Chemistry
Trinity University
San Antonio, TX 76284
(512) 736-7381

This workshop will be self-paced and participants with adequate background will be able to move forward rapidly or skip those parts with which they are already familiar. Topics covered in part I will include: an introduction to high resolution graphics on the Apple II, the use of shape tables, creating text, and simple animation by page switching. Part II will deal with scaling, windows and viewports, clipping, and an x-y plotting program. Part III will be concerned with using external graphics devices, the Koala pad, and a molecular drawing program. Part IV is especially designed for those with strong backgrounds and will cover three dimensional transformations, projections into the graphic surface, and examples of useful graphics software. Participants will be expected to have a knowledge of a higher level language, preferably BASIC. Familiarity with the Apple II series of computers is helpful though not required but experience with some microcomputer is essential. Textbook: "Apple Graphics Activities Handbook" by Harold J. Bailey and J. Edward Kerlin, Robert J. Brady Publishing Company, Bowie MD 20715.

C. MICROCOMPUTER INTERFACING

Director: James O. Currie, Jr.
Department of Chemistry
Pacific University
Forest Grove, OR 97116
(503) 375-6151

In this workshop participants will learn how the Apple II computer can be used to acquire data and to control experiments. During the course, approximately 50% of the time will be spent in the laboratory where participants will have an opportunity to work through a number of experiments, including interfacing to the Apple II bus and using the game ports and simple transducers to demonstrate the principles of data acquisition and control. In the lecture portion of the course simple gates, I/O software, analog to digital and digital to analog converters will be discussed, along with other state-of-the-art topics. No prerequisites are required but some knowledge of BASIC is helpful. Textbook: "Apple Interfacing" by Titus, Larsen, and Titus, Howard Sams, Bobbs Merrill Publishing Company, Indianapolis, IN 46206.

D. DESIGN AND DEVELOPMENT OF CAI IN CHEMISTRY

Directors: Stanley Smith
Department of Chemistry
University of Illinois
Urbana, IL 61801
(217) 333-3839

Ruth Chabay
Department of Psychology
Stanford University
Stanford, CA 94305

The CAI workshop will be a hands-on workshop in which participants will design and program short instructional segments on a topic and in a style of their choice. Group discussions will cover examples of CAI lessons, common problems and pitfalls, authoring tools and aids, licensing and copyright issues, programming techniques, and instructional design issues. Participants will work in small groups (2-3 people) on designing and programming actual instructional sequences. All programming will be done on Apple II series computers. Prerequisite will be a minimal familiarity with BASIC and an operating system, but participants need not be expert programmers.

E. EVALUATION AND IMPLEMENTATION OF SOFTWARE IN CHEMICAL EDUCATION

Director: M. Lynn James
Department of Chemistry
University of Northern Colorado
Greeley, CO 80639
(303) 351-2559

This workshop will concentrate on a consideration of the types of software available for use in chemical education including examples of good and bad software, things to look for in evaluating software, and methods of implementing this software into the chemistry classroom and laboratory. A great variety of software for chemistry will be available including both commercial software and materials available through project SERAPHIM. Illustrations of Drill and Practice, Tutorial, Simulation, Educational Games, Problem Solving, Material Generation, Data Management, and Lab Data Checking types of software will be reviewed. No prerequisites will be required but some familiarity with the microcomputer will be helpful.

COMPUTERS IN CHEMICAL EDUCATION NEWSLETTER

SUBSCRIPTION RENEWAL FORM

The mailing list will be updated beginning with the September 1985 Newsletter. If you wish to renew or initiate your subscription please fill out this form completely and return it with your remittance to:

Dr. Donald Rosenthal
Department of Chemistry
Clarkson University
Potsdam, NY 13676

Please check one:
U.S.A. __ 1 year for $2.50 __ 2 years for $4.50
All other countries __ 1 yr for $5.00 __ 2 yrs for $9.00
Payment MUST accompany this form. Please make a check or money order payable in U. S. funds to Computers in Chemical Education Newsletter.

Back issues from 1981 to the present may be obtained at $1.50 per issue. Issues appear in (1) March, (2)June, (3)September and (4)December.
Number of back issues ordered ___ $1.50/ issue $_____
Specify which issues ____________________

____

NAME ____________________

ADDRESS ____________________

____

____

CITY ____________________ STATE____ ZIP__________

PHONE NUMBERS Work (   )__________ Home (   )__________

Courses which you teach ____________________

Name of school or professional affiliation, if not indicated in the above address.

____

Please provide all information requested on the other side of this sheet.

OVER PLEASE

1. Are you a member of the ACS? __ Yes __ No
   Division of Chemical Education __ Yes __ No
   Division of Computers in Chemistry __ Yes __ No

2. Brief description of hardware and software which you use.

3. Do you have telecommunications capabilities? __ Yes __ No

4. What communicates with what? Indicate any special software or hardware.

5. Do you have access to any networks? __ Yes __ No
   If so, please identify. ____________________

6. Areas of Computer Activity and Interest

   Leave the space provided below blank if you have no present interest or activity. 1 means a little, 2 means moderate, 3 considerable, and 4 with a consuming passion.
   Use the space on the right to identify the courses in which the software or hardware is used. Identify the programming language(s), computer and any specialized hardware or instruments used.

	Degree of Interest	Activity	Description of Use
CAI Drill and Practice	______	______	______
CATC (Comp. Assisted Test Construction)	______	______	______
CMI (Comp. Managed Instruction)	______	______	______
Simulation	______	______	______
Numerical Methods and Statistics	______	______	______
Graphics	______	______	______
Software development other than above	______	______	______
Interfacing	______	______	______
Laboratory Automation	______	______	______
Other (specify)	______	______	______

	Personal Use	Administrative Use	Use in Instruction
Word Processing	____	____	____
Software Used ____________________			
Electronic Spreadsheets	____	____	____
Software Used ____________________			
Data Bases	____	____	____
Software Used ____________________			

PROJECT SERAPHIM

National Science Foundation: Science Education

Department of Chemistry
Eastern Michigan University
Ypsilanti MI 48197
(313) 487-0368

Project SERAPHIM is an NSF-sponsored clearinghouse for instructional microcomputer information in chemistry. It began operation in 1982 and has rapidly grown into a unique organization that spans a range of activities:

- Materials Clearinghouse--gathering and distributing materials that are currently available in chemistry; maintaining large databases of microcomputer users and software reviewers;
- Training Programs--training teachers to use existing and to develop new materials;
- Research and Development--holding "think tank" meetings to try to chart the course of the future of microcomputers; awarding Fellowships to carry out both research and development.

If you would like to receive either more general information about Project SERAPHIM or materials that Project SERAPHIM distributes, indicate so below:

___ Please put my name on your mailing list to receive Project SERAPHIM News and other information about the Project. (Please fill in blank below.)

___ I have a friend or colleague who would like to be on your mailing list to receive Project SERAPHIM News and other information about the Project. (Please fill in blank below.)

___ Please send me a copy of the current Project SERAPHIM Catalogue, listing all materials available from the Project.

___ Please send me a copy of the current Project SERAPHIM Software List, containing information about all of the instructional computer programs that we are aware of from all sources--commercial, noncommercial, or from this Project. Cost: $5, prepaid only. Send with check payable to Project SERAPHIM.

My name: ____________________ Friend's name: ____________________

Address: ____________________ Address: ____________________

Project SERAPHIM, NSF Science Education

John W. Moore, Director
J. J. Lagowski, Co-Director
Elizabeth A. Moore, Project Manager