

USE OF PC BASED MATHEMATICS SOFTWARE IN THE UNDERGRADUATE CURRICULUM

JOSEPH M. SLAUGHTER, JAMES N. PETERSEN,
RICHARD L. ZOLLARS
*Washington State University
Pullman, WA 99164-2710*

With the advent of the personal computer, many software packages have been developed which minimize the required programming, thus allowing the user to concentrate on understanding the problem rather than on debugging. This is very desirable from an educational point of view. However, if students are required to learn the use of many different packages, any advantages gained by reducing programming requirements are offset by increasing the time spent learning how to use the various packages. In this article we will describe the experiences of one of our undergraduate students in learning and using three such programs (MathCAD^[1], Point Five^[2], TK Solver Plus^[3]) which we have examined for incorporation into our curriculum.

Obviously, there are more packages available than the three cited above, but we sought to find a single package which met all of the following criteria. First,

Joseph Slaughter is a graduate student in chemical engineering at Washington State University. He received a BS degree in chemical engineering and a BA degree in foreign languages and literatures, French, from Washington State University in 1989. His current research is in the area of bioseparations using large-scale electrophoresis.



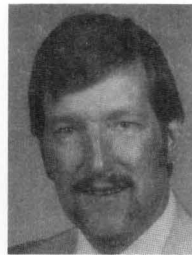
James N. Petersen is currently an associate professor of chemical engineering at Washington State University. He received his BS degree from Montana State University in 1976 and his PhD from Iowa State University in 1979. His current research interests are in the adsorption of heavy metals from aqueous streams by biological materials, and modeling and on-line optimization of biological processes.



it had to be a general purpose package, capable of all the calculations (solving sets of linear and/or nonlinear equations, iterative calculations, vector/matrix manipulations, curve fitting, simple statistics, regression) which are typically encountered by a student progressing through our curriculum. For this reason, spreadsheet programs were not included since, although many typical problems can be solved, the structure of the program is not optimal for all of the calculations (solution of ODE's, sets of nonlinear equations, *etc.*). Likewise, such packages as Matlab^[4] and GAUSS^[5], which concentrate on matrix manipulations, were not included.

Second, the program had to be capable of operation on the PC's available in our department (IBM PC/XT or AT's, or compatibles), which are also typical of those owned by the students. Thus, many of the symbolic manipulators (Macysma^[6]) were not considered.

Finally, we wanted the packages to be priced within the grasp of a typical student so that they might be purchased for use away from the academic setting. Of the three packages evaluated, both MathCAD and TK Solver Plus are available as student versions for approximately \$50, while Point Five can be site licensed for a reasonable fee. There are also a number of packages available with interfaces and structures very similar to TK Solver Plus, such as Eureka^[7] and FORMULA/ONE^[8]. Of this group, however, only TK Solver Plus was evaluated,



Richard L. Zollars is a professor of chemical engineering at Washington State University. He received his BChE (1968) from the University of Minnesota and his MS (1972) and PhD (1974) from the University of Colorado. His current research interests include adsorption, colloidal and interfacial phenomena, and bioseparations.

In this article we will describe the experiences of one of our undergraduate students in learning and using three programs (MathCAD, Point Five, and TK Solver Plus) which we have examined for incorporation into our curriculum.

due to its availability in a student version and its greater power.

All of the packages evaluated can be run on IBM PC/XT's or compatibles with no more than 512K, two 5 1/4" floppy disk drives, a graphics card, and a dot matrix printer. For this evaluation we used an IBM PS/2 Model 50Z personal computer equipped with a 80287 math co-processor. Feature-by-feature comparisons of many of these software packages have appeared^[9-11], but this type of comparison does not indicate how easily the package can be learned and used, nor its ability to easily solve typical chemical engineering problems and create a readable report (such as would be required if a student used the package to solve homework problems). Therefore, one of our senior chemical engineering students (J. Slaughter) was asked to solve typical homework problems from a number of areas (thermodynamics^[12], unit operations^[13], reactor design^[14], kinetics^[15], and numerical analysis^[16]) using MathCAD, TK Solver Plus, and Point Five. Each program was then evaluated for its utility.

MATHCAD

MathCAD is a free-format scientific scratchpad supporting 69 built-in functions and 29 symbolic operators. It comes on two disks with a well-written user's guide and a MathCAD reference booklet. After reading the first three chapters of the user's guide we felt fairly confident about setting up solutions to typical problems. When difficulties were en-

countered, help could be found either in the user's guide or with the on-line help facility. One of the strongest characteristics for use in education is its format. Graphically created symbols are used instead of function names, *i.e.*, $\sqrt{(\text{arg})}$ instead of `sqrt(arg)`, as shown in Figure 1. Also, eighteen commonly used Greek letters are available for use in the equations and text. These symbols make it easier for the user to find errors and to create a readable report.

The free-format style of the problem files makes editing quite easy, much like erasing an error on a paper scratchpad. However, moving about in MathCAD can become tedious in large files since the maximum cursor movement is either 80% of a page or to the beginning or end of the current region. Moving through the file is slowed even further if MathCAD is in its automatic calculation mode. This latter problem can be overcome by switching to the manual calculation mode. MathCAD can also write and read ASCII files so that data created from other software can be analyzed.

MathCAD solves simultaneous equations (linear and non-linear) using a solve block technique (see Figure 2). This technique is initiated by guessing a value for the unknown, entering the equation to be solved by using a "given" command, and requesting the solution by using a "find" command. If the convergence criteria is not met, MathCAD will supply a message to that effect, and the "find" command can be replaced by a "minerr" command to find the result

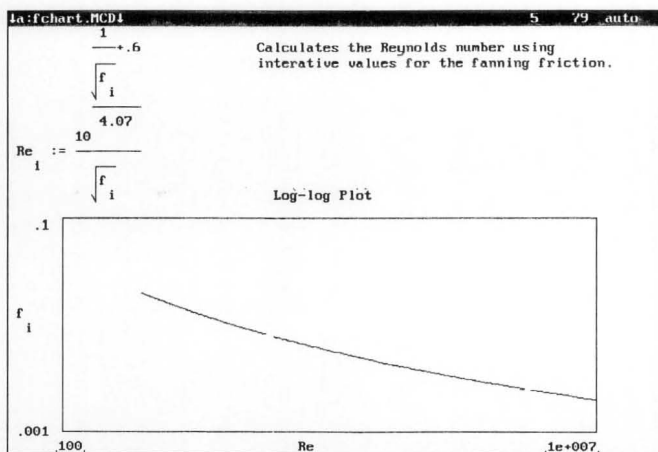


Figure 1

FIGURE 1. MathCAD screen showing features such as symbol operators and imbedded graphics.

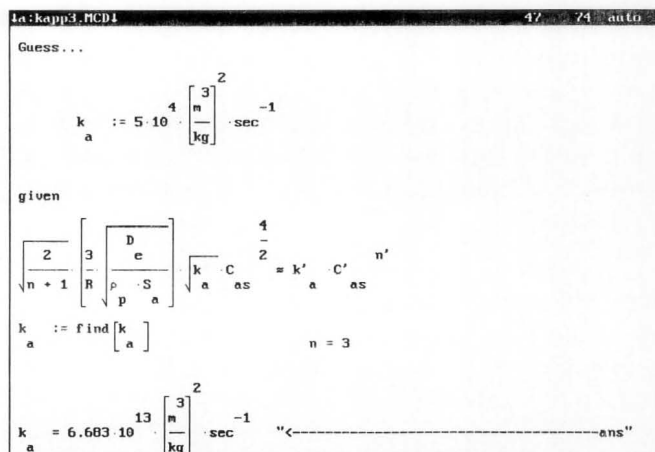


Figure 2

FIGURE 2. Example of MathCAD use of a solve block and units.

with the smallest error. The results obtained using "minerr" are good estimates which can then be used as new guess values.

MathCAD performs iterative calculations using vector notation and an iteration counter (range variable), as shown in Figure 3. Only one equation may appear inside this iteration loop, but multiple functions may be used. In Figure 3, for example, a user-defined function called "gradf(s,y)" is defined by using the built-in derivative function. Other user-defined functions [H(x,y), s(x,y), and λ(x,y)] are subsequently defined using "gradf", and all of the functions are combined into a single equation which is used in the calculation.

Graphs can be created very easily in MathCAD and can be imbedded into the report. Graphs can be formatted for size, type (linear, semi-log, or log-log), and number of subdivisions for each axis using six different symbols with or without a connecting line. One graphing feature that we found desirable was the ability to define the x-axis with more than one variable (neither Point Five nor TK Solver Plus were able to do this), as shown in Figure 4. MathCAD also creates tables, but they were limited to fifty elements per column and it takes practice to create presentable tables.

MathCAD supports a wide array of vector and matrix operations and functions, although they are limited in certain aspects. The largest array that we were able to create using the data editor was limited to fifty elements (newer versions can create 100-element arrays on a 10-by-10 matrix). The absolute size limit for arrays created by iterations and/or equations, is 8000 elements. We did not feel that this was a significant limitation since problems larger than this would typically not appear in classroom assignments. A more significant problem occurred when intermediate results in a calculation produced a 1-by-1 array—a value which should be used as a scalar. MathCAD only recognizes this value as an array so that further computations, such as the multiplication of another array by this "scaler," were prohibited.

Another feature that we found to be very useful was the ability of MathCAD to use units. MathCAD uses four base units: L (length), M (mass), T (time), and Q (charge). Any of these can be redefined if necessary (we found it convenient to redefine the base unit for charge as a base unit for temperature). These four base units are then used to derive other units such as force, energy, velocity, acceleration, etc.. During calculations, MathCAD converts all units into

the base units, checks for compatibility and gives the results in the base units. The user is able to change to any defined unit by simply entering the desired unit. Not only does this feature provide a means of detecting errors, but it also makes the report much more readable (see Figure 2). MathCAD provides three files (for mks, cgs, and US customary units) that contain most of the desired unit conversions.

Perhaps MathCAD's strongest point is the readability of its output, as shown in Figures 1 through 4. The graphical representation of operators, ability to mix text, plots, and calculations, and the inclusion of units on the calculations makes the output appear much as it would if the user were using a paper and pencil. Indeed, one does not need to know much about MathCAD in order to be able to read the output. The only noticeable idiosyncrasy in the output is the use of the symbols "!=" and "= ". MathCAD re-

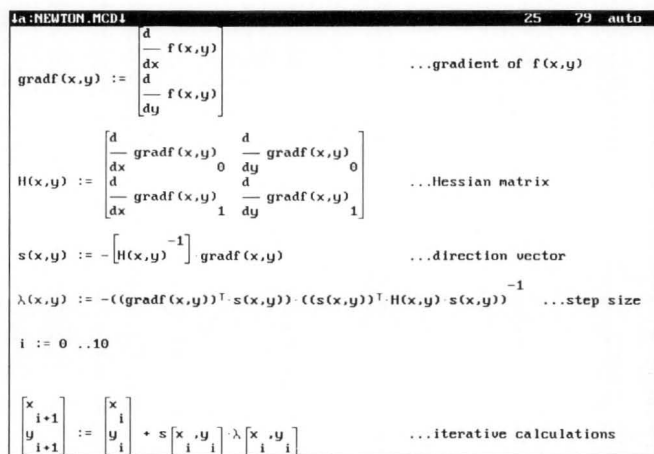


Figure 3

FIGURE 3. Iterative calculation in MathCAD.

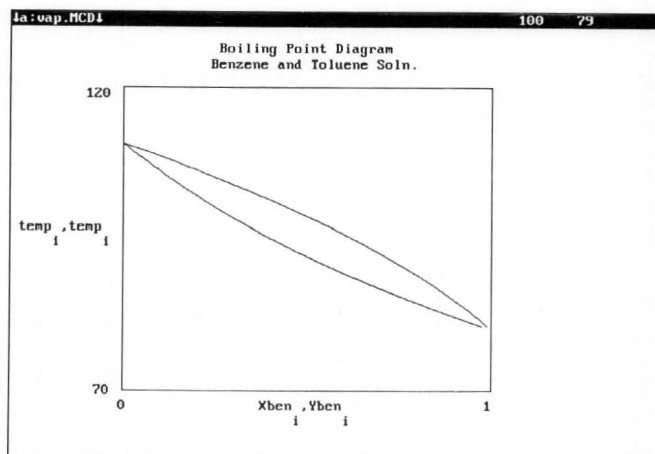


Figure 4

FIGURE 4. Example of boiling point diagram created by MathCAD with multiple definition of the ordinate.

quires that all variables be defined first (using the “:=” symbol) before a value can be calculated (using the “=” symbol). This arrangement requires extra keystrokes since any computed variable must be entered twice: once to define the variable and once to compute the value.

MathCAD has also recently become available in a student’s version. This version contains all of the power of the full version, but is limited to only 120 lines of output (two pages). We found this limitation on the output file size to be very restrictive for all but the most elementary of problems.

TK SOLVER PLUS

TK Solver Plus is a structured equation solver supporting 68 built-in functions and 13 relational and arithmetic operators. It comes on six disks with three manuals, a well-written (and much-needed) reference manual, an introduction to TK Solver Plus, and application notes. Three of the disks contain 103 models and functions which are separated into 13 categories ranging from finance to matrix manipulations. As with MathCAD, a student version of TK Solver Plus is available which contains everything included in the full version with the exception of the application notes and three library disks. TK Solver Plus is also able to read and write ASCII, DIF, and WKS files.

Of the three programs that we evaluated, we found TK Solver Plus to be the most difficult to learn. It was necessary to complete the introduction manual before feeling comfortable enough to start working. In part, this is due to the manner in which TK Solver Plus is organized. It is divided into nine main worksheets and several subsheets, with each sheet containing only a specific type of information. The variable sheet is used to enter values for the variables and to view the results. The rule sheet

```
(4r) Rule: "This program finds the time needed for the batch reaction des 25+/F9"

```

VARIABLE SHEET				
St Input	Name	Output	Unit	Comment
0	Xao			Initial conversion of A
.65	Y			Conversion of A
	Cao	.00298507	lbmol/ft ³	Initial concentration of A
	Cbo	.00268657	lbmol/ft ³	Initial concentration of B
'ra	fn			Reaction rate function
6	n			Number of steps used in SIMPSON
	rate	8097.7945		Value of internal
	time	24.472521	min	Time needed for batch reaction

```

S Rule
"this program finds the time needed for the batch reaction described in
"prob. 4-17 in ELEMENTS OF CHEMICAL REACTION ENGINEERING, Fogler(1986)
"it uses the library model SIMPSON to solve the integral
"find values for Cao and Cbo
Cao=Xao/V
Cbo=Ybo/V
"Solve integral of fn from Xao to Y using Simpson's rule
call Simpson(fn,Xao,X,Y,rate)
time=Xao/V*rate
F1 Help F2 Cancel F5 Edit F9 Solve / Commands = Sheets ; Window switch

```

Figure 5

FIGURE 5. Variable sheet and rule sheet from TK Solver Plus using the direct solver mode.

In an attempt to give some impression of the individual strengths and weaknesses of these various packages, we rated them on . . . ease of learning, ease of use, matrix operations, equation solving capability, versatility, use of units, generation of graphs and tables, etc.

defines the rules to be used, *i.e.*, specifies the relationships between variables, and initiates user-defined functions. Other sheets are used to create user-defined functions and algorithms, lists, tables, and graphs, to define unit conversions, and to format the sheets and variables.

Editing and moving about in TK Solver Plus was not difficult. Convenient window commands are used to move from one sheet to another, and a split screen can be used to display two sheets at a time (see Figure 5). Errors were marked in the status column on each sheet, along with a brief description and an indication of the line on which the error appeared.

The strongest attribute of TK Solver Plus is equation solving. If it cannot solve an equation directly, it automatically goes into its iteration mode. We were able to solve twenty linear equations with twenty unknowns, using two guesses, in a matter of seconds. If TK Solver Plus could not converge using the guesses provided, it would leave the last calculated values as the new guess values, and we only had to re-initiate the solve command until convergence was reached. On those occasions where convergence could not be obtained, TK Solver Plus would indicate this by displaying a convergence value in a highlighted window and leaving the output column blank.

Another strong attribute was the capability of TK Solver Plus to let the user create functions and algorithms. The functions and algorithms may be defined as rule, list, or procedural functions. Rule functions are those which call other functions or define equations. Procedural functions create a procedure which the variables follow in order to obtain a result (this is where understanding how to program in TK Solver Plus is important, and we found that reviewing the models in the library disks was helpful). List functions are used in creating tables, interpolating, and displaying intermediate values during iterative calculations.

Creating tables and graphs is done on the table and plot sheets, respectively, and were fairly easy to generate. The graphs and tables could each be stored and transferred from one file to another. The only way to print a graph, however, was to first view it

and then print it using the print screen key or command. Three types of graphs may be created: line, pie, and bar. A line graph could produce as many plots on one graph as lists entered, using fifteen different colors (for line only plots) and as many symbols as can be created using a standard IBM keyboard and the approximately 249 ASCII characters, as extended by IBM (including fifteen Greek letters). An example of a line graph illustrating the path taken in the optimization problem can be found in Figure 6. Plotting two lines, each having points at different values of the independent variable (as in Figure 4) was quite complex using TK Solver Plus and involved concatenating two separate lists into a single list so that the resulting list could be plotted.

One feature that we felt could use some improvement was the unit sheet and the unit column on the variable sheet. TK Solver Plus does not recognize units either in an equation or while performing calculations and so will not give a warning of incompatible units, as MathCAD does. On a few occasions we entered incorrect units but did not find our mistakes until after printing the report.

TK Solver Plus does not contain matrix operations within its basic package, although they are provided on the library disks or by creating your own algorithms. This is due to the algorithms TK Solver Plus uses when solving sets of equations which do not rely on matrix operations. In those situations where matrix operations were desired, we found that the calculations could be performed, but were generally more trouble than they were worth.

Printed output from TK Solver Plus is obtained by separately printing each sheet, graph, and/or table. We found these reports difficult to read and follow, especially for the beginner or non-user. Comment statements, as shown on Figure 5, did help the readability, but the reader *must* be familiar with TK Solver Plus to understand and follow the report as a whole.

POINT FIVE

Point Five is a mathematical scratchpad containing over 150 built-in functions divided into six categories: arithmetic, statistics, finance, matrix operations, data transformation, and procedure functions. This wide range of functions indicates that it is meant to be used by both engineering and non-engineering students alike. Point Five comes with one systems disk (or can be installed with two master disks), a well-written user's guide, an introductory tutorial manual, and a quick reference card. The tu-

torial was easy to follow and took only about one hour to complete. After completing the tutorial, we felt comfortable with the scratchpad and had very little trouble creating the desired models. Point Five has very nicely- organized help files for quick, on-screen help. Like the other programs, Point Five is able to read and write ASCII and DIF files, thus permitting it to interact with other software.

Point Five consists of two windows on one split screen (see Figure 7). The lower window is the scratchpad where formulas are entered and edited. The upper window is where the results are scrolled. Variables can be entered and edited either directly on the scratchpad or by using the data editor. Using the data editor makes analyzing and manipulating data quite easy; one simply enters a set of data, computes the results, and then enters a new set of data without having to change the scratchpad.

The two best features of Point Five are the built-in functions and the matrix operations. The variety of functions helps limit the amount of programming

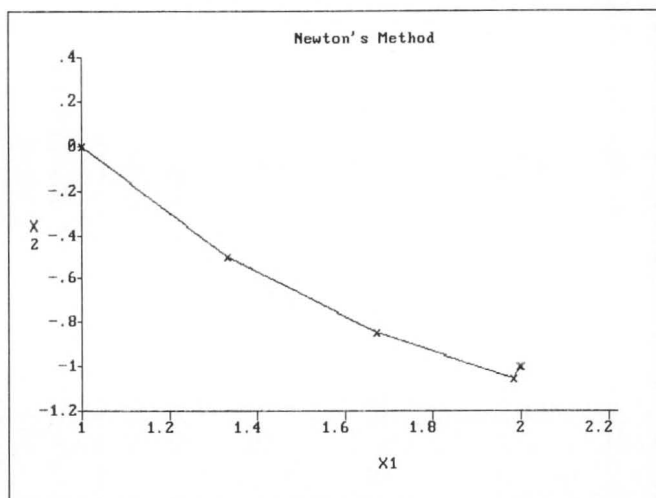


Figure 6

FIGURE 6. Line graph created by TK Solver Plus for optimization problem.

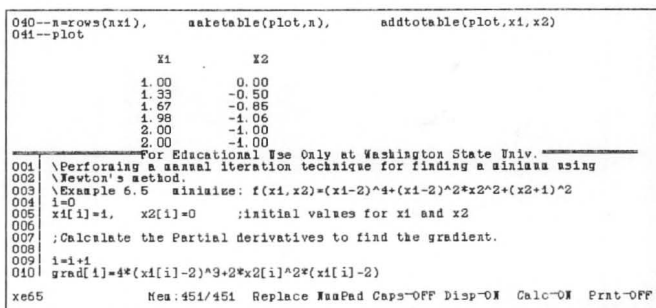


Figure 7

FIGURE 7. Screen from Point Five illustrating an interactive calculation.

required to create various models, which makes working with Point Five easier for the beginner. We found the matrix functions to be the best of the three software packages that we evaluated. We were able to enter an 89-by-89 matrix with the data editor, invert it, and then check the results by multiplying the inverted matrix by the original matrix, all on one line. The results were quite accurate.

We were also able to solve simultaneous linear equations, such as the optimization problem, by using an augmented matrix and one function, `SIMEQ(var.)`. Graphs and tables are quickly and easily created in Point Five. The graphing abilities are limited, however, to only scatter plots, line graphs, and bar graphs. Line graphs are limited to six lines, and the graphics are only CGA resolution (see Figure 8). Also, plots cannot be stored in separate files and can only be printed by doing a screen print.

Unlike TK Solver Plus and MathCAD, Point Five is not an equation solver; it is more like a powerful programmable calculator. This difference is particularly noticeable in that Point Five requires that all equations be in their explicit form. For example, we were not able to solve for the Fanning friction, f , in the equation

$$1 / \sqrt{f} = 4.07 \cdot \log(N_{Re} \cdot \sqrt{f}) - 0.6$$

used in Figure 1 since it could not be reduced to an explicit form.

Iterative calculations are performed in two ways: by line, using a "FOR...DO..." statement, or in a marked block by using the EXECUTE N TIMES command. Point Five limits the use of marked blocks to one per file, thus limiting the use of the EXECUTE command.

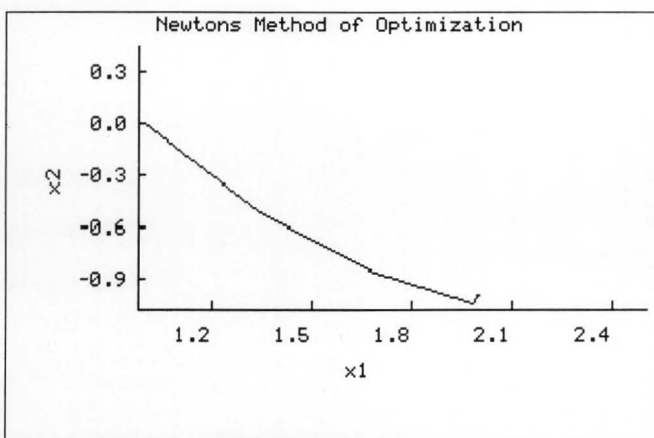


Figure 8

FIGURE 8. Line graph created by Point Five for optimization problem.

Readable reports could be made in Point Five with the use of comment statements. Turning the DISPLAY function off permits only comment statements, blank lines in a model, and results to be printed. Turning DISPLAY on prints each line executed. When using a marked block to perform iterative calculations, with DISPLAY on, long reports may be generated as each result computed during every iteration is displayed. We were able to avoid this problem by entering each line once to obtain a printout of the procedure and then executing the block, but this was only working around a weakness in Point Five. Unlike either MathCAD or TK Solver Plus, Point Five has no capability for working with units.

DISCUSSION AND CONCLUSION

Of the five example problems we selected, both MathCAD and TK Solver Plus were able to solve all five. Point Five was able to solve four of the five, but could not solve the friction factor problem due to the implicit equation used. TK Solver Plus appeared to be the fastest of the three packages, although all were rapid enough for the example problems so as not to be an inconvenience.

Comparing the three packages is very difficult since they are completely different, not only in structure but in emphasis of capabilities. Keeping in mind that these programs were evaluated for use in an engineering curriculum, we found Point Five to be a very useful tool for performing quick, on-the-spot calculations and data manipulations. Its library of functions was impressive and very useful for creating educational models and statistical and financial analysis. It was very easy to use and a good tool for those who have little experience with personal computers, but the limitations encountered for non-linear equation solving reduced its utility.

TK Solver Plus is a very powerful equation solver with the capability of a programming language. It is probably the most powerful of the three programs, but is also the hardest to learn. For quick, on-the-spot calculations (especially matrix operations) it is almost not worth the trouble. However, once the user becomes familiar with TK Solver Plus, its abilities are practically unlimited. Its rigid structure makes learning the program somewhat easier, but it also makes printed reports hard to read.

We found that we preferred going to MathCAD when attempting to solve a problem because of MathCAD's versatility, its ease of use, and the read-

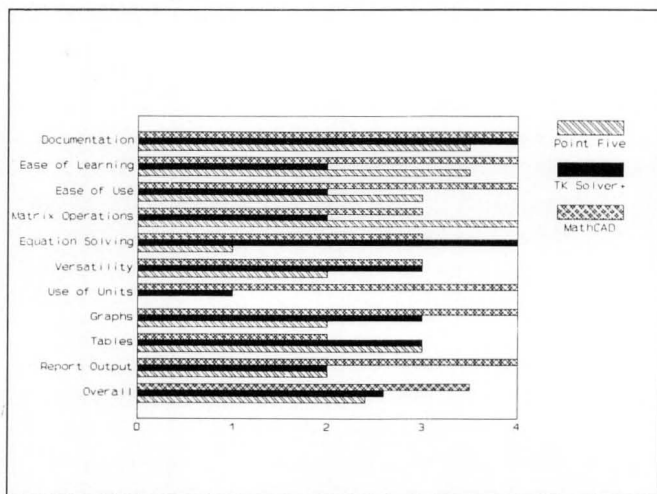


Figure 9

FIGURE 9. Comparison of Point Five, TK Solver Plus, and MathCAD. Scale: 0=unsatisfactory, 1=poor, 2=fair, 3=good, 4=excellent.

ability of its reports. We were able to efficiently perform matrix operations, create user functions, and solve series of simultaneous equations, without having to know many programming techniques. We enjoyed the feature of creating and editing graphs without having to leave the scratchpad every time we wanted to examine a plot – a feature especially helpful when trying to find roots. But one of the most impressive features is that the printed results from MathCAD are very readable. The free-format use of units and graphic abilities make it easier for the user and non-user to read and understand a MathCAD file.

In an attempt to give some impression of the individual strengths and weaknesses of these various packages, we rated them on the accompanying documentation, ease of learning, ease of use, matrix operations, equation solving capability, versatility, use of units, generation of graphs, generation of tables, readability of output, and overall impression on a scale from 0 (unsatisfactory) to 4 (excellent). While each of the programs that we evaluated has particular strengths and weaknesses, as indicated in Figure 9, MathCAD was our selection as the program best suited for general use in the chemical engineering curriculum.

ACKNOWLEDGEMENT

This work was supported by a Teaching Development Grant from the College of Engineering and Architecture, Washington State University.

REFERENCES

1. MathCAD, MathSoft, Inc., One Kendall Square, Building 200, Cambridge, MA 02139
2. Point Five, Pacific Coast Software Inc., 887 NW Grant Ave., Corvallis, OR 97330
3. TK Solver Plus, Universal Technical Systems, Inc., 1220

- Rock Street, Rockford, IL 61101
4. Matlab, The Math Works, Inc., 20 N. Main Street, Suite 250, Sherborn, MA 01770
5. GAUSS, Aptech Systems, Inc., 26250 196th Place SE, Kent, WA 98042
6. Macsyma, Symbolics Inc., 8 New England Executive Park East, Burlington, MA 01803
7. Eureka, Borland International, 4585 Scotts Valley Drive, Scotts Valley, CA 95066
8. FORMULA/ONE, Soft-Sense, 12 Rockaway Lane, Arlington, MA 02174
9. Shacham, M., and M.B. Cutlip, *Chem. Eng. Ed.*, **22**(1), 18 (1988)
10. Heller, M., *Personal Engineering & Instrumentation News*, **5**(9), 39 (Sept. 1988)
11. Smith, A.L., *Academic Computing*, **2**(3), 36 (1987)
12. Himmelblau, D.M., *Basic Principles and Calculations in Chemical Engineering*, 4th ed., Prentice-Hall, Inc., Englewood Cliffs, NJ, p. 591 (1982)
13. McCabe, W.L., J.C. Smith, and P. Harriott, *Unit Operations of Chemical Engineering*, McGraw-Hill Book Company, New York, p. 85 (1985)
14. Fogler, H.S., *Elements of Chemical Reaction Engineering*, Prentice-Hall, Inc., Englewood Cliffs, NJ, p. 164 (1986)
15. Fogler, H.S., *Elements of Chemical Reaction Engineering*, Prentice-Hall, Inc., Englewood Cliffs, NJ, p. 572 (1986)
16. Edgar, T.F., and D.M. Himmelblau, *Optimization of Chemical Processes*, McGraw-Hill Book Company, New York, p. 208 (1988) □

POLYMERIZATION PILOT-PLANT

Continued from page 39.

tive. Our efforts reflect our belief that it is time to introduce more sophisticated control algorithms (which, at present, are usually encountered at the graduate-student level) at the undergraduate level, combined with more complicated real processes.

ACKNOWLEDGEMENTS

Financial support from the Natural Sciences and Engineering Research Council (NSERC) of Canada, the Institute of Polymer Research (IPR), University of Waterloo, the Manufacturing Research Corporation of Ontario (MRCO), and the Computer-Aided Process Engineering (CAPE) Laboratory at the University of Waterloo, is gratefully acknowledged.

Many thanks also go to the "brave students": Mike Nakagawa, Gary Jubien, Mel deSouza, Howard Leung, Jerry Lin, Cam Phan, and Mike Kuindersma.

REFERENCES

1. Stephanopoulos, G., *Chemical Process Control: An Introduction to Theory and Practice*, Prentice Hall, New Jersey (1984)
2. Hoogendoorn, K., and R. Shaw, "Control of Polymerization Processes," *Proc. IFAC PRP-4 Automation*, Chent Belgium, Pergamon Press, p. 623 (1980)
3. MacGregor, J.F., A. Penlidis, and A.E. Hamielec, "Control of Polymerization Reactors: A Review," *Poly. Proc. Eng.*, **2**, 179 (1984)
4. Richards, J.R., and P.D. Schnelle, Jr., "Perspectives on Industrial Reactor Control," *Chem. Eng. Prog.*, **84**, 32 (1988)
5. Davidson, R.S., "An Intelligent Temperature Controller for Jacketed Reactors," *Am. Cont. Conf.*, **2**, 1380; *CEP*, **82**, 18 (1986)
6. Kern, A.G., "Simplify Batch Temperature Control," *Chem. Eng.*, pg 61, March 28 (1988) □