MATHEMATICAL POWER TOOLS

Maple, Mathematica, MATLAB, and Excel

Judith G. Mackenzie, Maurice Allen*
University of Canterbury • Christchurch, New Zealand

The concept of using computers to help solve mathematical and computational problems has a particular appeal to engineers. Prior to the appearance of personal computers, programs were coded in a computer language, often FORTRAN, for a very specific application. Later, generalized packages were developed to solve problems in particular disciplines. Some were purely calculational, to solve differential equations or to invert matrices; others had more of an engineering flavor. Such specialized engineering software was powerful, but had a narrow focus; for example, there were programs for electrical circuit design, civil engineering structural calculations, or flowsheeting programs for the design of chemical processing plants.

Software programs with a strong emphasis on calculation and numerical evaluation continue to be marketed for the personal computer. Reviews of individual packages outline the latest features; for example, Maple V, Release 3,[1,2] Mathematica 3.0,[3,4] MATLAB 5.0,[5-9] and Excel 7.0.[10] Comparisons of some of these mathematical packages for science and engineering education have been made by Seiter[11] and Pattee.[12]

Two reports on the teaching of first-year undergraduate calculus courses, using Mathematica, give favorable outcomes.[13,14] Both courses are entirely computer based, online with interactive text, and students have access to many examples. Students see calculus as a course in scientific measurement, calculation, and modeling through the use of technology. "Technology also make it possible to present the subject as a highly visual, often experimental, scientific endeavor."[15]

When using Mathematica for teaching chemical engineering concepts in process control and reaction engineering, Dorgan and McKinnon[16] found that students had mixed but generally positive reactions to its use. Several articles featuring the use of symbolic algebra computing in control engineering were published recently to foster a greater awareness of the potential offered to engineers by environments such as Mathematica and MATLAB.[17,18] Munro[17] emphasized that there are many areas where symbolic computing can offer significant improvements in the reliability and accuracy of results obtained.

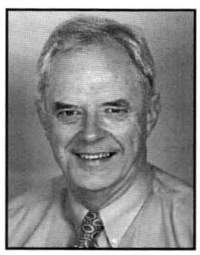
This article is concerned with these mathematical power tools and will investigate general-purpose computer applications for mathematical calculations and symbolic algebraic manipulation. Their comparison and evaluation will be from the viewpoint of an undergraduate engineer seeking to solve mathematical problems quickly and reliably and to communicate results. A direct comparison of each package with the others will be given for each of the problems posed.

MATHEMATICAL TOOLS

Maple™ performs computations that include symbolic algebra and numeric approximations, linear algebra, calculus, trigonometry, differential calculus, infinite and indefinite integration, modeling, statistics, and graphics, and produces program statements for a FORTRAN compiler. It is a symbolic manipulative language that clearly displays algebraic expressions especially useful for integration and differentiation.[19] Barker considers Maple to be more powerful than Mathematica when it comes to solving complex physics problems.[20]

Mathematica™ combines numerical calculations and symbolic manipulations into an interactive environment, coupled

Judith Mackenzie is Senior Tutor in the School of Engineering, University of Canterbury (New Zealand). She has a Master's degree in Education and is currently studying for a PhD in the Department of Chemical and Process Engineering. Her topic is the application of computers to innovative teaching in ChE education.



Maurice Allen is Associate Professor in the Department of Instrumentation and Control Engineering at Murdoch University (Perth, Western Australia). His teaching and research centers on process control, the modeling and simulation of industrial processes, and the application of computing to process engineering and teaching.

* *Address: Instrumentation and Control Engineering, Murdoch University, Perth, Western Australia*

with graphic visualization and a high-level programming language. The program is divided into a kernel, which does the computation, and the front end, which provides the user-interface and input capabilities. Mathematica equations are stored and can be imported or exported in ASCII format, favoring high portability. The Mathematica interface enables users to organize text, graphics, computer output, and pictures in a single 'notebook.' Included with Mathematica are standard functions and add-ons that allow the advanced user to perform more complex mathematical analysis.

MATLAB™ is an interactive, matrix-based system for scientific and engineering numerical computation and visualization.[21] The program operates with scalars, vectors, and matrices from expressions entered by the user. A variety of built-in functions can be used for displaying two- or three-dimensional color graphics. The basic MATLAB package may be extended with any of the different tool boxes designed for engineering specialities such as systems identification, optimization, control, spines, and Simulink. Electrical engineers like MATLAB because it is matrix-based and particularly suited for signal processing, digital communication, and control-system design.[22] A symbolic mathematics option in MATLAB uses the Maple kernel that extends its numerical capabilities to algebraic manipulation.

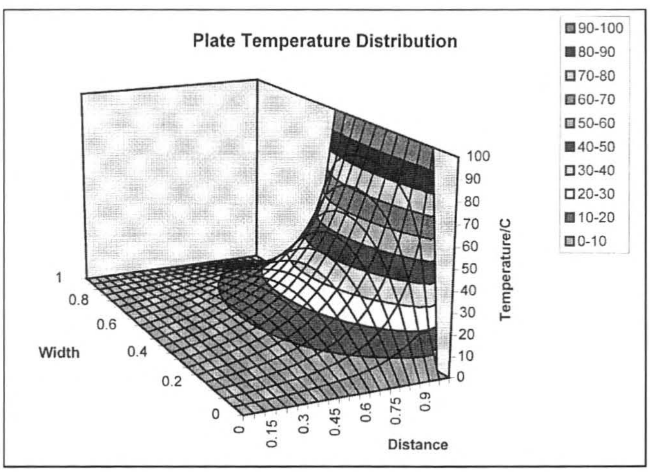


Figure 1. *Plate temperature distribution.*

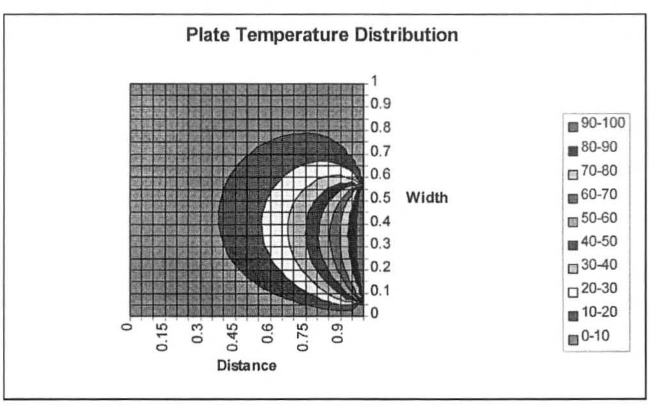


Figure 2. *Contour plot of plate temperature distribution from Excel.*

Excel™ is a popular spreadsheet with limited symbolic capability, but it is effective for small engineering calculations. The wide range of built-in mathematical and statistical functions, the ease of interactive programming, ease of re-use and modification, rapid graph generation, and on-line help make it an efficient design and prototyping tool. Although Excel was designed for business purposes, it is a practical tool for scientists and engineers.[23]

ENGINEERING PROBLEM SOLVING

To evaluate and compare the usefulness of these mathematical tools for teaching engineering problem solving, four engineering problems were solved using each of the four mathematical packages. Engineering problems considered were

- *The calculation and graphical display of a heat-transfer calculation*
- *The inversion of a large matrix as part of input-output economic analysis*
- *A root-finding calculation for control-system design using the Bode stability criterion*
- *The solution of a set of ordinary differential equations to evaluate the quality of the control-system design.*

PROBLEM 1
Two-Dimensional Heat Transfer

The two-dimensional steady-state conduction equation can be discretized on a rectangular grid to relate the temperature at any point to the temperatures at its four adjacent points.[24]

$$T_{i,j} = \frac{T_{i-1,j} + T_{i+1,j} + T_{i,j-1} + T_{i,j+1}}{4} \tag{1}$$

If the temperatures on the boundary are specified, then this equation can be used to iteratively calculate the temperatures at all points within the boundary. The rectangular configuration of a spreadsheet conveniently conforms to this formulation. Figures 1 and 2 show the temperature distribution of a square plate, all boundaries at 0°C except for a half of one edge, unsymmetrically placed, at 100°C.

The results and graphs were generated in Excel in about 40 minutes. The programming capabilities of MATLAB and Mathematica were used for the same calculation, but more than twice the time was required for the programming. The temperature results were easily and effectively graphed in Excel, MATLAB, and Mathematica. Graphical representation facilitates the problem solving and verification process, and color variation helps students to visualize the problem solution.

PROBLEM 2
Energy Analysis

Energy analysis is a tool to determine how much of an energy resource is required to enable a given good or service to be produced and delivered to its consumer, enabling a physical description of the operation of a real-world process to be formulated. Input-output analysis can be used for analyzing the energy and environmental consequences of consumption. Peet[25] gives an example of energy analysis using 80 sectors.

Mathematically, the problem was that the data was available in the matrix form

$$X = AX + Y \tag{2}$$

where X and Y are vectors of system inputs and outputs, and A is the technical coefficient matrix. But the system was to be analyzed in the form

$$X = (I - A)^{-1}.Y \tag{3}$$

where I is the identity matrix. Computationally, an 80-by-80 matrix was required to be subtracted from the identity matrix and inverted. The matrix was easily inverted with Mathematica and MATLAB, but Excel was unable to handle such a large matrix. It was important, however, to have the data in the correct format before importing.

The data was saved as plain ASCII text and imported into Mathematica with

mymatrix=ReadList["c:\excel\energy.dat", Number, RecordList -> True]

and into MATLAB with

load c:\excel\energy.dat

Maple has a "read" function for importing data, but the large matrix data was unable to be imported into Maple.

After solving a large system of linear equations, an estimation of the condition of the computed solution is important for verification of numerical accuracy. Maple and MATLAB have functions to estimate the condition number of the inverted matrix to provide this verification.[26] The condition number of the matrix in the example above was 9.8, indicating that the inversion was relatively accurate and resulting in the loss of only one decimal place of significance.

PROBLEM 3
Root-Finding for Control System Design

An ideal proportional-integral-derivative controller, having the Laplace transform,

$$\overline{m} = K_c\left(1 + \frac{1}{T_i s} + T_d s\right)\overline{e} \tag{4}$$

where m and e are the valve position and error, K_c is the proportional sensitivity, and T_i and T_d are the integral and derivative times, was to be designed for the fourth-order process (see Figure 3)

$$\overline{c} = \frac{1}{(T_a s + 1)(T_b s + 1)(T_c s + 1)(T_e s + 1)}\overline{m} \tag{5}$$

where T_a, T_b, T_c, and T_e are time constants. The Bode stability criterion[27] requires solution of the equation

$$F(x) = \pi - \tan^{-1}(T_a x) - \tan^{-1}(T_b x) - \tan^{-1}(T_c x) - \tan^{-1}(T_e x) + \tan^{-1}\left(T_d x - \frac{1}{T_i x}\right) = 0 \tag{6}$$

for the angular frequency, x. This numerical approach, coupled with the calculation of the amplitude ratio, replaces the well-known graphical procedure[27] using the Bode diagram graph paper.

This root-finding problem was successfully and quickly solved by each of the four applications. The dialogue with Mathematica, with the output indented, is shown in Figure 4.

Excel was used in two ways for this problem. First, a Newton root-finding method was derived by hand, taking about 50 minutes to derive and verify the derivatives. Each

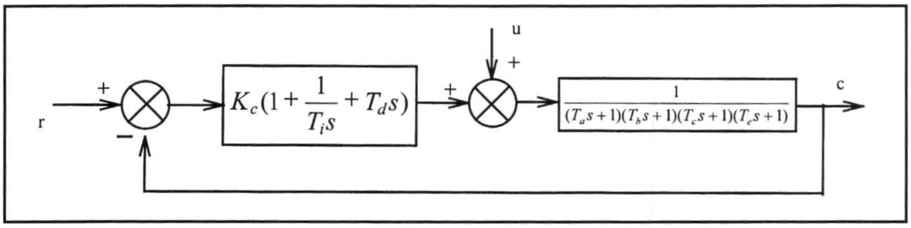


Figure 3. Control system block diagram.

```
F := P - ArcTan[x Ta] - ArcTan[x Tb] - ArcTan[x Tc]
  - ArcTan[x Te] + ArcTan[x Td - RTi / x]
Ta = 1; Tb = 2; Tc = 3; Te = 4; Ti = 7;
RTi = If[Ti > 0.0, 1/Ti, 0.0];
Td = Ti / 4; P = π;
FindRoot[F == 0, {x, 0.5}]
Plot F Degree, x, 0.002, 2 ,
   AxesLabel -> "x", "y"
  x→ 0.716372
```

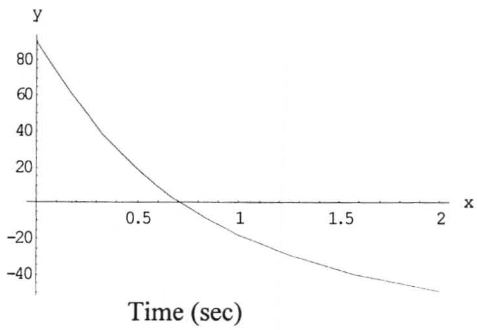


Figure 4. Mathematica output to confirm location of the angular frequency root.

successive iteration was a row of the spreadsheet. Alternatively, when the "goal"-seeking command was used, only 5 minutes were required, similar to the time required for any of the other applications. But in the full-design procedure, the root-finding had to be repeated three times, with only minor modification. Each of the applications supports the construction of user-defined functions, similar to functions, procedure, or subroutines in procedural programming languages.

```
<<Calculus`LaplaceTransform`
Ta=1; Tb=2; Tc=3; Te=4; Ti=7;
RTi=If[Ti>0.0,1/Ti,0.0]
Td=Ti/4; Nf=1; u=0; r=1; Kc=5;
G=Kc(1+RTi/s+Td s)
Df=(Ta s+1)(Tb s+1)(Tc s+1)(Te s+1)
c=FullSimplify[(Nf u+Nf G r)/s/(Df+G Nf)]
InverseLaplaceTransform[c,s,t]
Plot[%,{t,0,60}, Frame->True,
    FrameLabel->{"Time (sec)","Response"}]
```

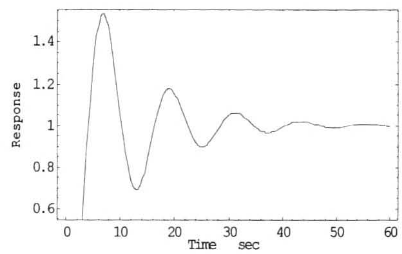


Figure 5. *Time response in Mathematica—set-point response.*

```
Ta=1; Tb=2;Tc=3;Te=4;Ti=7;Td=Ti/4;Kc=3
a=Kc*(1-Td*(Te+Tc)/Tc/Te)/Te;
b=Kc*(1/Ti-1/Te+Td/Te/Te);
A=[0,0,Kc*Td/Tc/Te,a,b;1/Ta,-1/Ta,0,0,0;0,-1/Tb,
-1/Tb,0,0;0,0,1/Tc,-1/Tc,0;0,0,0,1/Te,-1/Te]
eig(A)
for i=1:250
  t=0.2*(i-1);
  p=expm(A*t)*[1 1 1 1 1]';
  pa(i)=p(1); pb(i)=p(2); pc(i)=p(3);
  pd(i)=p(4); pe(i)=p(5); x(i)=t;
end
plot(x,pa,x,pb,x,pc,x,pd,x,pe),
xlabel ('Time (sec)'), ylabel ('Response');
```

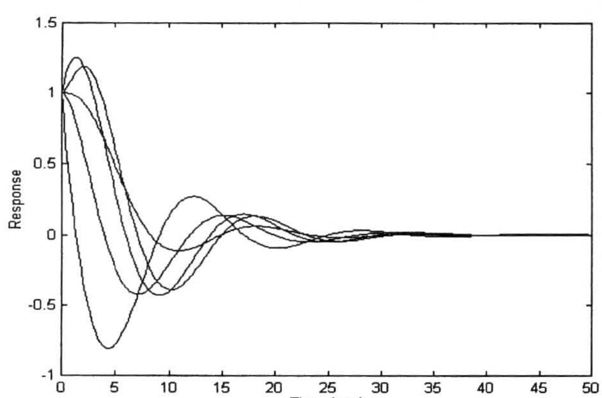


Figure 6. *Control system initial condition response.*

PROBLEM 4
Differential Equation Solution for Time Response

The control system design in Problem 3 had the form shown in Figure 3. Problem 4 was concerned with the response of this system to a change in set point, r, or to a disturbance, u, in order to confirm the control-system design.

Differences between the nature of the mathematical applications became more evident when solving this problem. Maple and Mathematica provided the simplest solution. The relationships defining the closed-loop transfer function were defined and solved automatically for the Laplace transform of the process variable, c. The inverse Laplace transform function[28] available in both Maple and Mathematica gave the required time response within an elapsed time of 5 minutes. The Mathematica notebook dialogue and response are shown in Figure 5.

A finite difference approach could have been taken in MATLAB and one of its differential equation solvers (ODE23 or ODE45) used. But since the problem was linear, the closed-loop transfer function was rewritten in state space form, and the matrix exponential function was used to calculate the time response.

The MATLAB m-file to set up the process matrix A, calculate its eigenvalues, and derive and plot the time response is shown in Figure 6. The algebra required 100 minutes, setting up the MATLAB calculation an additional 10 minutes, and the actual calculation and plotting about 10 seconds.

Excel did not have differential equation support. A solution was obtained by deriving the differential equation for the controller and each of the first-order elements from its transfer function and using a finite difference approximation to provide a recursive relationship (see Figure 7). Setting up

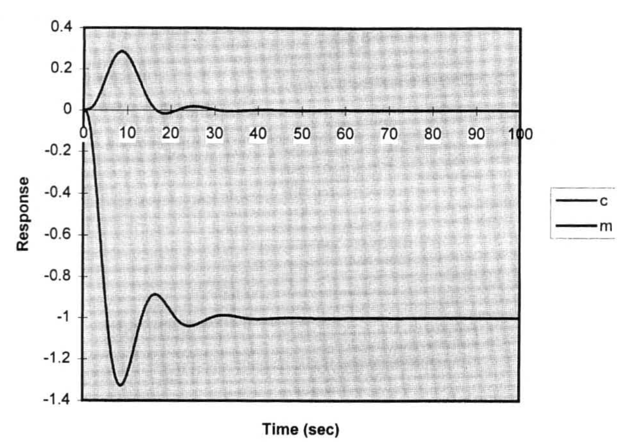


Figure 7. *Excel graph of time response to a unit step in the load, u.*

the spreadsheet took about 60 minutes and its calculation several seconds. Modeling the time response in this way enabled variables to be changed, giving an almost simultaneous change in the graph.

EVALUATION

Attributes of the mathematical packages were rated on a one-to-five-point scale (one being the worst and five the best) to assess their scope, intuitiveness, ease of use, graphics, and fitness for engineering applications; the results are shown in Table 1. All four of the packages are powerful problem-solving tools. In this evaluation, Mathematica was ranked ahead of MATLAB, with Maple following a close third and Excel fourth. But such comparisons are subjective and the differences between the packages were small.

TABLE 1
Evaluation of the Mathematical Problem-Solving Tools

	Maple	Mathematica	MATLAB	Excel
Scope	4	5	5	3
Intuitiveness	4	5	5	4
Ease of use	3	5	4	3
Graphics	3	4	4	5
Fitness for engineering applications	4	5	5	3
Symbolic manipulation	5	5	4	0
Totals	23	29	27	18

CONCLUSIONS

The best application in a particular instance depends heavily on the nature of the problem. Maple had the advantage of giving a symbolic analytical solution, but did not have the numerical capabilities of Mathematica, MATLAB, or Excel. Each of the three "M"s dealt well with symbolic manipulation and graphics. Excel displayed the most flexible graphics with, for example, the capacity to easily rotate three-dimensional plots. The Mathematica notebook provided an excellent interactive feature for documentation, report writing, and teaching. The advantages of a particular application are lost if extensive work by hand is required to express the problem appropriately for that application.

Our opinion is that engineers need to be skilled in at least one spreadsheet such as Excel, a programming language, and at least one of the other mathematical packages. If an engineer is heavily involved in matrix manipulation and linear systems, then MATLAB has advantages, especially if its extensive optional tool boxes are relevant. Equally, Mathematica has distinct advantages in its use of a natural language, the "notebook" feature, and user interface. Maple was the most difficult package to learn and program, but was useful for verification of mathematical analysis.

The best tool depends on individual needs, and the time spent learning the applications will reap the benefits of these powerful mathematical tools. How to incorporate them into our graduate and undergraduate courses is a key issue for engineering educators.

REFERENCES

1. Hutton, J., and J. Hutton, "The Maple Computer Algebra System: A Review," *J. of Appl. Econometrics,* **10**(3), 329 (1995)
2. Foster, K., "Major Math Updates," *IEEE Spectrum,* **33**(8), 64 (1996)
3. Vaughn, S.M., "Review of Mathematica," *Human and Ecological Risk Assess.,* **3**(3), 343 (1997)
4. Studt, T., "Mathematica 3.0 Adds Equation Editor for Presentation Graphics," *R&D Mag.,* **38**(8), 33 (1996)
5. Gramley, R., "MATLAB 5.0," *IEEE Comp. Sci. & Eng.,* **4**(1), 89 (1997)
6. Conrad, A., "Mathematical Modeling Program Receives Major Upgrade," *Microwaves & RF* (1997)
7. Foster, K.R., "Matrices and Much, Much More—MATLAB," *IEEE Spectrum,* **34**(2), 14 (1997)
8. Schaufelberger, W., "The Student Edition of MATLAB," *Automatica,* **33**(5), 1005 (1997)
9. Waynant, R., and R. Fuller, Eds., "Software and the Web.," *Circuits and Devices,* May (1997)
10. Strassberg, D., "Open Excel Inside a 32-Bit Graphing Package," *EDN* **42**(19), 13 (1997)
11. Seiter, C., "Mathematics Analysis," *Macworld,* June (1992)
12. Pattee, H.A., "Selecting Computer Mathematics," *Mech. Eng.,* September (1995)
13. Park, K., and K.J. Travers, "A Comparative Study of a Computer-Based and a Standard College Freshman Calculus Course," unpublished report (1995)
14. Noss, R., "Reading the Sines. Report on Teaching and Learning with Technology Programme: Project 15," Imperial College, London, and The University of Leeds (1995)
15. Uhl, J.J., "Calculus and Mathematica" *U.M.E. Trends,* **6**(6), 15 (1995)
16. Dorgan, J.R., and J.T. McKinnon, "Mathematica in the ChE Curriculum," *Chem. Eng. Ed.,* **30**(2), 136 (1996)
17. Munro, N., "Symbolic Algebra Tools for Control Teaching," *Comp. and Cont. Eng. J.,* 58, April (1997)
18. Barker, H.A., and M. Zhuang, "Control System Analysis Using Mathematica and a Graphical User Interface," *Comp. and Control Eng. J.,* 64, April (1997)
19. Heck, A., *Introduction to Maple,* Springer-Verlag, New York, NY (1993)
20. Barker, D., "Another Hard-Core Problem Solver," *Byte,* 266, May (1991)
21. Etter, D.M., *Engineering Problem Solving with MATLAB,* Prentice-Hall, Englewood Cliffs, NJ (1993)
22. Orsak, G.C., and D.M. Etter, "Using the Internet and MATLAB," *IEEE Signal Processing Magazine,* **12**(6), 23 (1995)
23. Orvis, W.J., *Excel for Scientists and Engineers,* 2nd ed., Sybex, San Francisco, CA (1996)
24. Gerald, C.F., and P.O. Wheatley, *Applied Numerical Methods,* Addison-Wesley, Massachusetts (1984)
25. Peet, N.J., *The Use of Input-Output Methods to Evaluate the Energy Requirements and Environmental Consequences of Economic Activity: A Practical Introduction,* Chemical and Process Engineering, University of Canterbury (1991)
26. Kahaner, D., C. Moler, and S. Nash, *Numerical Methods and Software,* Prentice-Hall, Englewood Cliffs, NJ (1989)
27. Coughanowr, D.R., and L.B. Koppel, *Process Systems Analysis and Control,* McGraw-Hill, New York, NY (1965)
28. Abell, M.E.L., and J.P. Braselton, *Mathematica by Example,* AP Professional, Boston, MA (1994) ❐