

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/224755068>

# Maple, Mathematica, and Matlab: The 3M's without the tape

Article in *Computing in Science and Engineering* · March 2005

DOI: 10.1109/MCSE.2005.18 · Source: IEEE Xplore

---

CITATIONS

8

---

READS

936

2 authors, including:



Norman Chonacky

Yale University

63 PUBLICATIONS 152 CITATIONS

SEE PROFILE



## MAPLE, MATHEMATICA, AND MATLAB:

### THE 3M'S WITHOUT THE TAPE

By Norman Chonacky and David Winch

IN THIS, THE FIRST IN A SERIES OF TECHNOLOGY REVIEWS THAT WILL APPEAR IN THE NEXT SEVERAL ISSUES OF *C&SE*, WE'LL EXPLORE THE THREE PREEMINENT PRODUCTIVITY TOOLS

for scientific and engineering computation. Although hardly identical, the packages have extensive similarities in the computational work they're intended to support. In areas in which they overlap, we'll look at how they compare, even as we contrast their scopes where they extend beyond one another. Given that you probably already use one of these tools sets, our goal is to help you decide whether one of them as now packaged is better suited to your temperament or your current technological practice than when you made your original choice. In each article, we'll examine how these packages support one of three different categories of professional work: undergraduate education, scientific and engineering computation, and technical communication and publication.

#### Out of the Gate

If computer programs can be said to have "style" or "personality," the three application packages we'll be exploring reveal themselves right from the start: the installation process. Although the user must face it only once, we believe the details of this process prefigure a consistent difference among the products.

Mathematica came on one CD-ROM, installed itself quickly and slickly, putting its icon directly on the Macintosh desktop "dock" without asking. However, before you can do anything with it, you must register the license number over the Web, thus getting a password to unlock the package for execution. This gives the feeling that Wolfram Research, the product's creator, is in control.

By contrast, Maple came on three CD-ROMs, installed on our disk of choice—though only after we deduced that the name of the chosen hard-disk partition needed to be changed—and provided immediate access to the program,

with or without registration. This gives the feeling that Maplesoft wants to unlock your imagination quickly.

Matlab also came on three CD-ROMs. The installation, however, was much less smooth—at least for us. We both had to try a few times, and the installer occasionally balked. But Matlab is a very complex product: its modular design includes innumerable toolboxes that extend the core to serve a wide variety of specialized science and engineering tasks. Further, we were installing the Mac version, which MathWorks has only relatively recently made available again, so this is not entirely surprising. Overall, Matlab gives the feeling that MathWorks' engineering is practical, uncompromised by ego.

#### Histories

In making the case for style, as well as to get clues about their goals and objectives, we begin with brief histories of these products, extracted from their own presentations about themselves. First, let's look at some pieces prefacing the 1,464-page *The Mathematica Book* (5th edition, Wolfram Media, 2003), itself a monument to the self-confidence of its author, Steven Wolfram:

Having started to use computers in 1973, Wolfram ... in 1979 began the construction of SMP—the first modern computer algebra system, which he released commercially in 1981 ... It is often said that the release of Mathematica marked the beginning of modern technical computing. Ever since the 1960s individual packages had existed for specific numerical, algebraic, graphical and other tasks ... The concept of Mathematica was a single system that could handle all [these tasks] ... Wolfram began the development of Mathematica in late 1986. The first version ... was released on June 23, 1988 ... [as] the world's only fully integrated environment for technical computing.

The application's foundation was in mathematics, but it evolved into a general tool for technical computing. From day one, it was also a commercial product that signaled the founding of Wolfram Research.

Keep in mind that Steven Wolfram was first a physicist and that Mathematica is a creature of his mind. Indeed, in reading through this history, it's difficult to separate the

product from its author. Clearly, the product is an expression of the person. The goal is to dramatically alter the way we think about, and do, science and engineering—other enterprises can join in “if they dare.”

Some of the themes of Wolfram’s driving passion also emerge here: integration and unification, changing the face of technical computing and the way we think about the world. The latter becomes even clearer in the parts of the preface that are more predictive than historical:

Wolfram’s papers on cellular automata led to a major shift in scientific thinking and laid the groundwork for a new field of science that Wolfram named “complex systems research.” ... Following the release of Mathematica Version 2 in 1991, Wolfram began to divide his time between Mathematica development and scientific research. Building on his work from the mid-1980s (on complexity and cellular automata), and now with Mathematica as a tool, ... Wolfram finally described ... in *A New Kind of Science*, released on May 14, 2002, a paradigm shift of historic importance in science.

Are you ready for Mathematica? Then you’d better fasten your seat belt!

Compare and contrast this with a piece on Maple’s history, extracted from the Fall 2001 issue of *The Maple Reporter*, the Waterloo Maple newsletter:

From its humble beginnings as a research project at a university to the position of an influential technology in education, research, and industry, Maple has fundamentally changed the way we do math ... [reflecting] how the mathematical community itself has evolved. ... In the ’80s, Maple was a pure programming language optimized for mathematical applications. In the early ’90s Maple V Releases 1 through 4 saw ... development of the Maple user interface ... to display mathematics in high resolution ... [and] dynamically interact with 3D plots ... [facilitating] student use. In the late ’90s, Release 5 introduced Web publishing ... enabling Maple documents [to be] translated to professional HTML pages for rapid publication on the Web...Maple 6 and 7 establish a new standard ... [of] connectivity with other software. The Maple Application and Student Centers ... [help] users take advantage of ... a living concept continually enriched and expanded by the energy and creativity of its user base.

From its birth, also in mathematics, Maple was conceived as a tool for expression as well as computation. Its developers were fostering a mathematical and scientific community—anticipating, if not driving, the evolution of science

---

## EDITOR IN CHIEF

**Norman Chonacky**  
norman.chonacky@yale.edu

---

## ASSOCIATE EDITORS IN CHIEF

Denis Donnelly, Siena College  
donnelly@siena.edu

Douglass E. Post, Los Alamos Nat’l Lab.  
post@lanl.gov

John Rundle, Univ. of California at Davis  
rundle@physics.ucdavis.edu

Francis Sullivan, IDA Ctr. for Computing Sciences  
fran@super.org

---

## EDITORIAL BOARD MEMBERS

Klaus-Jürgen Bathe, Mass. Inst. of Technology, kjb@mit.edu

Antony Beris, Univ. of Delaware, beris@che.udel.edu

Michael W. Berry, Univ. of Tennessee, berry@cs.utk.edu

John Blondin, North Carolina State Univ., john\_blondin@ncsu.edu

David M. Ceperley, Univ. of Illinois, ceperley@uiuc.edu

Michael J. Creutz, Brookhaven Nat’l Lab., creutz@bnl.gov

George Cybenko, Dartmouth College, gvc@dartmouth.edu

Jack Dongarra, Univ. of Tennessee, dongarra@cs.utk.edu

Rudolf Eigenmann, Purdue Univ., eigenman@ecn.purdue.edu

David Eisenbud, Mathematical Sciences Research Inst., de@msri.org

William J. Feiereisen, Los Alamos Nat’l Lab, bill@feiereisen.net

Sharon Glotzer, Univ. of Michigan, sglotzer@umich.edu

Anthony C. Hearn, RAND, hearn@rand.org

Charles J. Holland, Office of the Defense Dept.,  
charles.holland@osd.mil

M.Y. Hussaini, Florida State Univ., myh@cse.fsu.edu

David Kuck, KAI Software, Intel, david.kuck@intel.com

David P. Landau, Univ. of Georgia, dlandau@hal.physast.uga.edu

B. Vincent McKoy, California Inst. of Technology,  
mckoy@its.caltech.edu

Jill P. Mesirov, Whitehead/MIT Ctr. for Genome Research,  
mesirov@genome.wi.mit.edu

Yoichi Muraoka, Waseda Univ., muraoka@muraoka.info.waseda.ac.jp

Kevin J. Northover, Open Text, k.northover@computer.org

Andrew M. Odlyzko, Univ. of Minnesota, odlyzko@umn.edu

Charles Peskin, Courant Inst. of Mathematical Sciences,  
peskin@cims.nyu.edu

Constantine Polychronopoulos, Univ. of Illinois, cdp@csrui.uiuc.edu

William H. Press, Los Alamos Nat’l Lab., wpress@lanl.gov

John Rice, Purdue Univ., jrr@cs.purdue.edu

Ahmed Sameh, Purdue Univ., sameh@cs.purdue.edu

Henrik Schmidt, MIT, henrik@keel.mit.edu

Donald G. Truhlar, Univ. of Minnesota, truhlar@chem.umn.edu

Margaret H. Wright, Bell Lab., mhw@bell-labs.com

and engineering into a collaborative, group-centered enterprise. Within a “communitarian” ethos, Maple aims to be accessible to students as well as researchers. The details of its evolution are amplified in a more personal account taken from an interview with Keith Geddes, cofounder of Waterloo Maple, also in the Fall 2001 issue of *The Maple Reporter*:

In the autumn of 1980, I called a meeting of various Computer Science faculty of the University of Waterloo to discuss the idea of how we could get ... a system like MACSYMA ... It was at this meeting, in November 1980, that the Maple system was conceived. ... By the first week in December we had an operational Maple system, and it evolved from there. ... By January of 1981, we started using Maple in a course, “Introduction to Symbolic Computation” ... Mathematicians started to use the system for research in 1982. By 1983, researchers at various institutions

***Compromises intended to make Maple  
as accessible as possible...have guided  
its development strategy.***

were using [it] in various other disciplines, including computer science, engineering, physics, and economics ... and there were about 50 installations ... By 1987 there were 300 installations worldwide ... In April of 1988, we incorporated Waterloo Maple Software ... to market, distribute, and support Maple ... and by 1990 the installed base grew to 2000.

Maple is a creature of academics and a product of the academe. Its developers gave away the first distributions, spawned by interest at conferences, for roughly the cost of the media. What is perhaps not so obvious from this historical account, is how much of an afterthought commercialization was. Geddes explicates this in his interview:

Until 1990 or so, it was never about making a commercially successful product. Our vision was about creating a powerful system for symbolic mathematics.

From this history, we conclude that Maple’s development, direction, and driving force have always been communal and pragmatic rather than individualistic. Its perspective has

been evolutionary as much as revolutionary. Compromises intended to make Maple as accessible as possible to the greatest number of users have guided its development strategy both before and since commercialization. Given that the trajectory of Maple’s development and commercialization were critically determined by the culture of its time, we wonder if this “communitarian” enterprise would perhaps have followed the open-source software development path entirely if begun today.

From the MathWorks Web site ([www.mathworks.com](http://www.mathworks.com)) we get this brief account of its history:

Matlab began life in 1984 as a response [to a] recognized need among engineers and scientists for more ... productive computation environments [than] those provided by languages such as Fortran and C. [It is] a high-performance technical computing environment ... combining comprehensive math and graphics functions with a powerful high-level language ... Our goal [was and] is to change the world by accelerating the pace of discovery, innovation, development, and learning in engineering and science ... [with a] computing environment for technical computation, visualization, simulation, design, and implementation.

MathWorks’ founders combined their expertise in mathematics, engineering, and computer science to develop Matlab; this is reflected in the emergence, through the years, of various “tool boxes” designed to make applications easy to develop for a wide range of disciplines. In this sense, we get the feeling that Matlab invites and facilitates collaboration between, and conjoining of, various disciplines. These values are reflected in MathWorks’ mission statement:

At the MathWorks, we express who we are as an organization through our guiding principle, our mission, and our core values. Developed over time, each represents a philosophy or goal that is intrinsically important to the organization ... Our guiding principle is “Do the Right Thing,” ... meaning doing what is best for our staff members, customers, business partners, and communities for the long term... It also means measuring our success ... by how consistently we act according to this principle. Our mission and core values express what “doing the right thing” means in our day-to-day work ... Our [technological mission] is to change the world by accelerating the pace of discovery, innovation, development, and learning in engineering and science ... [We want our] core values, ... essential and enduring principles .... unaffected by the trends of the day or changing market conditions, ... [to be] understood and practiced throughout the organization.

The site lists the “core values” as continuous improvement and the pursuit of excellence, respect and investment (for individuals), rational workplace, learning and self-improvement, and credibility and integrity. MathWorks’ approach to product development and work environment is consistent with those in many of the most technologically advanced industries. This suggests that Matlab is going to feel very much like, and fulfill the expectations of, a professional engineering design package.

## Design Objectives

Taking cues from their histories and advertising copy as representative, we deduced several design objectives for each of these products. Here, from the Mathematica Web site ([www.wolfram.com](http://www.wolfram.com)) and the book’s preface:

The visionary concept of Mathematica was to create ... a single system that could handle all the various aspects of technical computing in a coherent and unified way. A key ... [was] a new kind of symbolic computer language that could ... manipulate the very wide range of objects involved in technical computing using only a fairly small number of basic primitives.

At a technical level, Mathematica ... is one of the largest single-application programs ever developed ... Among its core innovations are its interconnected algorithm knowledge base and its concepts of symbolic programming and document-centered interfaces.

Although not part of the testimony, it appears that a Mathematica design criterion is elegance. Consistent with this conclusion is that pragmatism must not be a core value at Wolfram Research. A description of the company’s “character,” appearing on its Web site, reinforces this notion:

Wolfram Research’s leadership ... stems from its ability to set the direction for new technology. The Wolfram Group is characterized by an individualist approach, a “no compromises” attitude to design, welcoming of innovation, a deep respect for the ... user base and users’ suggestions, and constant search for good general approaches rather than quick fixes.

Ever the revolutionary, systematic unifier, and grand enterpriser, Wolfram will take us where humans have never gone before. We should thus expect more sophisticated, novel, integrated, elegant (and possibly more proprietary and expensive) software from Mathematica in the future.

## EDITORIAL OFFICE

### COMPUTING in SCIENCE & ENGINEERING

10662 Los Vaqueros Circle, PO Box 3014

Los Alamitos, CA 90720

phone +1 714 821 8380; fax +1 714 821 4010;

[www.computer.org/cise/](http://www.computer.org/cise/)

## DEPARTMENT EDITORS

**Book & Web Reviews:** Bruce Boghosian, Tufts Univ., [bruce.boghosian@tufts.edu](mailto:bruce.boghosian@tufts.edu)

**Computing Prescriptions:** Isabel Beichl, Nat’l Inst. of Standards and Tech., [isabel.beichl@nist.gov](mailto:isabel.beichl@nist.gov), and Julian Noble, Univ. of Virginia, [jvn@virginia.edu](mailto:jvn@virginia.edu)

**Computer Simulations:** Dietrich Stauffer, Univ. of Köln, [stauffer@thp.uni-koeln.de](mailto:stauffer@thp.uni-koeln.de)

**Education:** Denis Donnelly, Siena College, [donnely@siena.edu](mailto:donnely@siena.edu)

**Scientific Programming:** Paul Dubois, Lawrence Livermore Nat’l Labs, [dubois1@llnl.gov](mailto:dubois1@llnl.gov), and George K. Thiruvathukal, [gkt@nimkathana.com](mailto:gkt@nimkathana.com)

**Technology News & Reviews:** Norman Chonacky, Columbia Univ., [norman.chonacky@yale.edu](mailto:norman.chonacky@yale.edu)

**Visualization Corner:** Jim X. Chen, George Mason Univ., [jchen@cs.gmu.edu](mailto:jchen@cs.gmu.edu), and R. Bowen Loftin, Old Dominion Univ., [bloftin@odu.edu](mailto:bloftin@odu.edu)

**Your Homework Assignment:** Dianne P. O’Leary, Univ. of Maryland, [oleary@cs.umd.edu](mailto:oleary@cs.umd.edu)

## STAFF

**Senior Editor:** Jenny Ferrero, [jferrero@computer.org](mailto:jferrero@computer.org)

**Group Managing Editor:** Gene Smarte

**Staff Editors:** Kathy Clark-Fisher, Rebecca L. Deuel, and Steve Woods

**Contributing Editors:** Cheryl Baltes and Joan Taylor

**Production Editor:** Monette Velasco

**Magazine Assistant:** Hazel Kosky, [cise@computer.org](mailto:cise@computer.org)

**Technical Illustrations:** Alex Torres

**Publisher:** Angela Burgess, [aburgess@computer.org](mailto:aburgess@computer.org)

**Assistant Publisher:** Dick Price

**Advertising Coordinator:** Marian Anderson

**Marketing Manager:** Georgann Carter

**Business Development Manager:** Sandra Brown

## AIP STAFF

Jeff Bebee, Circulation Director, [jbebee@aip.org](mailto:jbebee@aip.org)

Charles Day, Editorial Liaison, [cday@aip.org](mailto:cday@aip.org)

## IEEE ANTENNAS AND PROPAGATION SOCIETY LIAISON

Don Wilton, Univ. of Houston, [wilton@uh.edu](mailto:wilton@uh.edu)

## IEEE SIGNAL PROCESSING SOCIETY LIAISON

Elias S. Manolakis, Northeastern Univ., [elias@neu.edu](mailto:elias@neu.edu)

## CS MAGAZINE OPERATIONS COMMITTEE

Bill Schilit (chair), Jean Bacon, Pradip Bose, Doris L. Carver, Norman Chonacky, George Cybenko, John C. Dill, Frank E. Ferrante, Robert E. Filman, Forouzan Golshani, David Alan Grier, Rajesh Gupta, Warren Harrison, James Hendler, M. Satyanarayanan

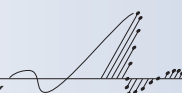
## CS PUBLICATIONS BOARD

Michael R. Williams (chair), Michael R. Blaha, Mark Christensen, Roger U. Fujii, Sorel Reisman, Jon Rokne, Bill Schilit, Linda Shafer, Steven L. Tanimoto, Anand Tripathi



IEEE Antennas &  
Propagation Society

IEEE  
Signal Processing Society





To contrast, here are some Maple design objectives, again extracted from *The Maple Reporter*:

The primary goal [was] to design a computer algebra system that would be accessible to large numbers of researchers in mathematics, engineering, and science, and to large numbers of students for educational purposes ... transforming the way engineers and scientists use technology.

Maple's creators planned to employ "a special architecture and highly intuitive, refined command set," allowing rapid development of algorithms and fast porting to various operating systems. Among fundamental criteria, the company listed an insistence on space efficiency and time efficiency in its efforts to create and manage supportive resources for users, "demonstrating our commitment to the second part

***Given Matlab's long list of components,  
MathWorks seems to prize modularity.***

of our vision: making it easy for more people to use Maple."

Given this design ethos, we thought it would be interesting to compare these earlier design criteria to Maple's features as they exist today. As described in the promotional material on Maplesoft's Web site ([www.maplesoft.com](http://www.maplesoft.com)), current capabilities and resources include

- a context-intelligent right-click operation menu and an extensive selection of user-interface features obviating the need to memorize commands;
- interactive visualization to let users explore data efficiently and intuitively;
- extensive connectivity features, including Microsoft Excel, Matlab, TCP/IP, and OpenMaple;
- programming language features, intuitive syntax, easy debugging, and a rich feature set to support complex applications; and
- free resources, as embodied by the Maple Application Center.

Based on this, we should expect Maple's future releases to be more usable, understandable, publishable, and possibly more economical.

MatLab is fundamentally a matrix-oriented product—MatLab is short for "matrix laboratory." Aside from its "do the right thing" unifying principle and firm belief that "a right solution" always exists, MathWorks leaves us to infer its design objectives from the de facto evidence presented by its product.

Given Matlab's long list of components, MathWorks seems to prize modularity. Matlab's core—a computational engine and I/O facilities—handles common functions for a wide variety of toolboxes, which enable developers in numerous disciplinary areas to create applications without (re)building entire systems. This approach's pragmatism and economy are consistent with an engineer's mentality, which prizes these values. Accordingly, we expect that Matlab will continue to spawn specialty toolboxes, and that the company will nurture the collaborative development of specialized applications with industrial and academic partners.

From this examination of history and design principles, we conclude that all parties are designing their products generally to meet their stated explicit (and implicit) objectives. To the extent that these coincide with their proclivities and needs, users should find one or another product congenial. Of course, all companies are subject to market pressures in a competitive world, so we can't assume—however strong their insurances to the contrary—that some market shift might not present compelling reasons for any of them to reinvent themselves.

As an example of how the market can influence a product's development objectives, consider spreadsheets. Who could have guessed that a product originally intended for the business world would be used for science and engineering simulations? We never used VisiCalc, but we understand that it was the progenitor of the general spreadsheet application program. Indeed, it was popular enough to get people to use the rather primitive Control Program/Monitor (CP/M) operating system. It didn't take long for Microsoft to translate the VisiCalc concept into Excel for the DOS and Mac platforms, both of which were more advanced than CP/M.

The spreadsheet's paradigmatic rows and columns look like transplants from the accountant's desk, but one of us (Chonacky) remembers seeing his first simulations computed using a spreadsheet—at a physics meeting, of all places—and being amazed at the obvious simplicity of programming simulations this way. Although looking at a spreadsheet, he was seeing the electrostatic equipotentials for a point-charge distribution. Who'd have thought it?

It's likely that the science and engineering marketplace led to the expansion of this accountant's tool to include a pack-

age of scientific and mathematical functional routines, as well as graphics and curve fitting, as parts of the standard Excel application.

## Uses

In this review series, we're not taking on a systematic study of Mathematica, Maple, and Matlab users' current practices. Companies closely guard the comprehensive profiles of their products' installation bases as "marketing secrets." Thus, we must be content with generalized corporate-advertising claims and anecdotal examples drawn from small user samples identified to satisfy diversity profiles. The latter we will leave to subsequent Tech News reviews in this series; but the former we can draw from several places.

## Mathematica

From the *Mathematica Book's* preface, we get the following:

At first, Mathematica's impact was felt mainly in the physical sciences, engineering and mathematics. But over the years, [it] has become important in a remarkably wide range of fields ... physical, biological, and social sciences, and other fields outside the sciences, ... all of the Fortune 50 companies, all 15 major departments of the US government, and all the [world's] 50 largest universities.

And from the Mathematica Web site:

In engineering, Mathematica ... [is a] tool for development and production. In commerce, [it] has played a significant role in ... financial modeling, as well as ... general planning and analysis. [It] has also emerged as an important tool in computer science and software development: its language component is widely used as a research, prototyping, and interface environment.

Mathematica is clearly a big player with big players. In light of assertions about its widespread use in industry and government, Mathematica seems to have gone the way of the spreadsheet, though in the opposite sense—that is, a tool invented primarily to make the conduct of science and engineering more rational has found its way, via the market, into other areas and applications.

A search of Amazon.com's book catalog, using "Mathematica in the title, author, or subject" as the criteria, returned 593 items. A quick inspection of the titles reveals that most of these are about pure or applied mathematics, rather than applications of mathematics to other disciplines. This suggests that a substantial number of the users are acade-

mics, and their applications are pedagogical. On the other hand, titles wouldn't reveal commercial users, who are unlikely to publish the results of their proprietary application-development work as books.

Whether deserved or not, but perhaps as a consequence of its grand vision, Wolfram Research is widely perceived, among academics, as unwilling to concede to pricing that smaller players—modest universities and small colleges—can afford. Small businesses might similarly feel this pinch, but we have no input from them regarding pricing pressures at this time. To the extent that this perception is correct, however, the academic users are probably disproportionately from large research universities. This grand vision might also be responsible for driving up Mathematica's cost—an approach that seems to favor commercial users, for whom

***Mathematica is clearly  
a big player with big players.***

productivity, rather than cost, tends to be the primary criterion for software decisions, even in small companies.

## Maple

Much of what we can conclude about Maple's users comes from the software's development history. Maple seems consistently inclined toward its "family of origin" in the academic community. In fairness, however, it's also worthwhile to look at how Maplesoft presents its product in its public-relations material:

Maple 9.5 is the environment of choice for scientific and engineering problem-solving, mathematical exploration, data visualization, and technical authoring. [It] transforms the way technical professionals and students use technology, ... integrating the power of ... numerics with ... symbolics [with] full computing support for any activity involving mathematics ... it allows [the user] to encapsulate and manage technical knowledge. In industry, [it] facilitates true collaborative knowledge management. In the classroom, [it] frees the student and teacher to focus on concepts ... A rich user environment ... shortens the learning curve and maximizes productivity gains.

This suggests that different messages are being aimed at two

different communities: academics and industry users. Indeed, it was interesting to examine two separate brochures that Maplesoft had prepared for that very purpose. Note the similar but individually tailored “summary statements” from these two brochures. For industry:

Highly integrated product suite delivers a flexible platform to compute, deploy, and share technical information.

For academe:

Highly integrated product suite and an extensive set of free resources increase the effectiveness and efficiency of any academic activity that depends on mathematical information.

***We don't intend to try to prove,  
and will not assert, that one  
package is better than another.***

The company offers two different descriptions of the possible uses for Maple's three principle components. For industry:

MapleNet enables enterprise-wide collaboration through sharing results data and analytical knowledge over the Web. Professional toolboxes offer domain-specific tools to transform the core [Maple] ... to discipline-specific applications. Resources/services [afford] ... Web resources.

For academe:

MapleNet is a full environment for online, e-learning, providing an ... interactive Web-based environment ... for [educational] content creation [and] deployment. Maple T.A. ... for Web-based testing and assessment includes access control, interactive grade book, algorithmic question generation, and intelligent assessment of mathematical responses. The Maple Application Center offers ... free downloadable resources ... simple classroom demos, curricula for entire courses ... in calculus, differential equations, classical mechanics, linear algebra, statistics, etc.

Maplesoft appears willing to support its “dual use” product

with supplementary resources that are specifically suited to industry and academe. The capabilities its products share, conditioned by their common Maple core, are mathematical computation, visualization, and integrated communications. Moreover, the emphasis seems to be on serving broad constituents who require common mathematical tools rather than on transforming the way that users view their respective worlds.

Yet there are indications, both in this advertising strategy and in the product-evolution history, that Maple is primarily an academician's tool whose user community Maplesoft would like to expand to businesses. This is not because it lacks computational sophistication or power, but because it grew out of the minds of academic mathematicians and has been tuned to their needs.

Querying Amazon.com's book catalog, using “Maple in the title, author, or subject” as the search criteria, returned 668 items. A quick inspection of titles reveals that roughly two-thirds of these were unrelated to Maple software, referring instead to author names and maple trees. The roughly 200 others were divided between pure or applied mathematics and mathematics related to other academic disciplines. There were proportionately more of the latter than there were for Mathematica.

### **Matlab**

Based on the engineering flavor embedded in the design strategy, company values, and, most significantly, the toolbox components' character (many of which embody laboratory or system design capabilities), we expected Matlab to be more widely represented in this book catalog search than the other two packages.

Querying Amazon's book catalog using “Matlab in the title, author, or subject” as the search criteria, we located 366 items. A quick inspection of the titles reveals that these cover a wider range of disciplines than either Mathematica or Maple. Many of the titles were for engineering disciplines and applications. It is also notable that a significant number of the toolboxes are available only for Windows rather than for Macintosh or Linux—consistent with the fact that Windows machines are ubiquitous in the engineering community.

### **New Approach for Technology Reviews**

We're hoping this series of reviews will serve as a practical instrument for scientists and engineers to use in selecting application software. Given that this is a learning process for both the writers and readers, let's move context to the forefront: we know that knowledge is context-dependent, so un-



## A New Kind of Review

**W**e're dedicating multiple Technology Reviews to the same cluster of products for several reasons. First, we want to see how you, our readers, respond to various content and presentation styles. Since the merger of our precursors—*AIP Computers in Physics* and *IEEE Computational Science and Engineering*—we've conducted readers' surveys, but to get more detailed knowledge of our readership, we want to engage you in a process of feedback and modification. Focusing multiple reviews on these technologies will provide a concrete set of points using a single context for a continuing dialogue on specific issues that readers raise. Running the reviews in sequential issues will offer time and multiple opportunities for readers to respond and for us to react with modifications.

Maple, Mathematica, and Matlab are so widely used that a justifiably detailed and comprehensive discussion of them would be too lengthy for a single issue. Presenting them in this manner will also let us test a new approach to the technology review format—one of increased depth, wedded closely to the user's experience, based on illustrative case studies, and introducing a metric of contrast that relates the critique to some standard of comparison.

Our goal is to increase the Technology Reviews' substance and practical utility to *CiSE* readers. For this, we'll need your attention, help, and feedback. We thus invite you to submit comments and critiques on these articles' content and style. Next issue, we'll have details on several ways to respond to us, along with the first review installation—on undergraduate educational use. Don't miss it!

Understanding and interpreting these packages' capabilities will necessarily be related to how and where they are applied—their *use context*. Accordingly, we'll try to explicitly define the uses we have in mind when formulating each critique.

Please remember that this review series is an experiment. We'll try to keep in mind that our critiques must connect practically to a wide variety of users and uses. But only you can evaluate whether they advance your understanding of the products and improve your ability to judge their capabilities and suitability. We hope at least some of you will provide feedback to help us accomplish these ends.

Given these productivity applications' enormous scope and potential, readers will interpret critical reviews in a wide variety of ways. To reduce the ambiguity of interpretation, we'll rely on specific use contexts to illustrate the capabilities we address in each segment. Specifically, each installment will focus on the productivity packages' technical capabilities in one of three use categories: undergraduate education; science and engineering research computation; and science and engineering communication and publication.

The reviews will headline case studies—practical examples of how the products have been or might be used—from which we'll try to build some general characteristics, patterns, or "morals" that might be helpful for others in making their own evaluations. In choosing illustrative cases, we'll attempt to fairly represent a span of use settings—industry, research laboratory, and academe.

We don't intend to try to prove, and will not assert, that one package is better than another. Rather, our comparative-review methodology is intended to present a basis for each reader to create a personal metric for comparing the packages.

### What You Will See, What You Might Expect

Each article in this series will approach the use of these pack-

ages from one of three user perspectives: an instructor of undergraduate science and engineering students; an R&D person looking to perform scientific or engineering computations; and an instructor, researcher, or student who needs to communicate technical work to others *in situ* or otherwise publish science and engineering material.

From each perspective, in turn, we'll examine several capabilities. The first of these are functional capabilities—that is, what kinds of tasks can we perform in each of the three use areas? Pressed by considerable course loads, for example, the undergraduate instructor might need to create and revise presentation materials quickly. This instructor might also need to visualize very difficult (to visualize) concepts for highly heterogeneous student groups. Alternatively, the scientist might require facile management capabilities in order to analyze data sets from a matrix of experiments. The engineer might need high computational accuracy and the ability to explore large parameter spaces for optimization. Students and instructors, in particular, will often wish to place their material on the Web, but any of these individuals should be able to create unambiguous output for communication and publication.

Another issue of concern is the accessibility and clarity of the materials (program code, metadata, notation, provenance, and so on) produced using these tools. For example, instructors assigning homework in a given product will need to understand the program code their students write. Researchers might also need transparency in computational programs to allow other R&D team members to scrutinize the original calculations from their own viewpoints. The question of provenance is also important when working in teams—who wrote what and when—to ensure version concurrence. Instructors and researchers creating tutorial materials need great flexibility in pro-

ducing notation and in formatting their narratives so that their materials' appearance is never slave to the technology used to render it.

Sometimes we fail to appreciate the availability, accessibility, and quality of reference material and support until the package is on our desktops, but these affect different types of users in different ways. For example, undergraduate instructors charged primarily with their students' conceptual development can't devote much time to teaching them how to use tools. Research scientists' and developers' time is expensive, so technical support of productivity software is key to its cost-effectiveness. Most people producing publications work under deadlines, which means a publication's value might depend on its timeliness. This implies a cost-benefit computation of a different sort, but again, the accessibility and quality of product support and reference materials have an impact.

In future reviews, we'll also assess issues such as the three packages' performance, interoperability, and learning curves, which are important to varying degrees to the three user types. Case studies will provide a concrete basis for discussion and evaluation: we'll include either actual examples of comparable work done with all packages or reasonably realistic examples that illustrate the variety of uses we've outlined. To the extent possible, we will use examples drawn from actual users' work. At very least, we'll seek out actual users to critique any illustrative examples we create.

In analyzing and generalizing the case-study results, we'll try to consistently address a range of important questions, such as:

- What is the price of going with a single integrated package rather than adopting a piecemeal approach?
- What differences in the packages' historical genesis and development might indicate future evolution patterns?
- To what extent are the packages equivalent?
- Are there significant differences between versions of Mathematica, Maple, and Matlab for the Macintosh, Windows, and Linux platforms?
- Are their differences in the distribution of uses according to discipline or workplace?

Is it possible to construct cost-benefit analyses of these three products that are of use to the several types of users? We'll find out if we hear from you, so keep those cards and letters coming.

CSE

**Norman Chonacky** is a research fellow in the Center for United Nations Studies in the Political Science Department at Yale University. His research interests include cognitive processes in research and education, environmental sensors and the management of data derived from sensor arrays, the physico-chemical behavior of material in environmental systems, and applied optics. Chonacky received a PhD in physics from the University of Wisconsin, Madison. He is a member of the IEEE Computer Society, the American Association of Physics Teachers (AAPT), American Physical Society (APS), and American Association for the Advancement of Science (AAAS). Contact him at [norman.chonacky@yale.edu](mailto:norman.chonacky@yale.edu).

**David Winch** is emeritus professor of physics at Kalamazoo College. His research interests are focused on educational technologies. Winch received a PhD in physics from Clarkson University. He is coauthor of *Physics: Cinema Classics* (ZTEK, videodisc/CD 1993, DVD/CD 2004). Contact him at [dmwinch@kitcarson.net](mailto:dmwinch@kitcarson.net).

## Special Offer for IEEE Members!

### 26 Management Courses!

From renowned sources such as the American Management Association (AMA) and Peter Drucker. Courses include...

- ▶ **AMA – Negotiate to Win**
- ▶ **AMA – Managing Employee Conflict**
- ▶ **Peter Drucker – Permanent Cost Control**
- ▶ **Peter Drucker – Knowledge Worker Productivity**
- ▶ **The Conference Board – How to Build High-Performance Teams**

And more! Low-cost access for IEEE members. For details, visit...

[www.computer.org/DistanceLearning](http://www.computer.org/DistanceLearning)

