NeXTstep: The Ideal Platform for *Mathematica*

Executive Summary

In fields as diverse as college mathematics, engineering research, and financial modeling, the value and promise of *Mathematica*® is well known. NeXT understands and appreciates this fact: Since the introductions of both *Mathematica* and the original NeXT Computer in 1988, every NeXT™ computer delivered to users in education has included *Mathematica.*

***Mathematica* is at once a powerful and sophisticated calculator, a graphics application for three-dimensional imaging, a tool for symbolic manipulation, a high-level programming environment, and a medium for creating interactive structured documents in Notebook format.**

Hailed globally as one of the most valuable tools for research, engineers and scientists have explored and solved thousands of problems via *Mathematica* on NeXT computers. With the *Mathematica* Notebook front end, faculty and graduate students on campuses around the world use NeXT computers for courseware development in math-intensive disciplines. And most recently, users in business—particularly in financial services—are exploiting *Mathematica*'s computational capabilities for building financial models and trading applications. Only the NeXTstep™ implementation of *Mathematica* offers these complete features for such a broad range of users.

This paper describes the features and benefits enjoyed by users who run *Mathematica* on NeXTstep. For more information on NeXTstep, please consult companion papers in the NeXT White Paper Library by calling 1-800-879-NeXT.

I. Introduction: Understanding the NeXTstep Advantages

To appreciate the overall superiority of *Mathematica* running on the NeXT platform, it is important to understand that NeXTstep—NeXT's graphical user interface and application development environment—allows *Mathematica* to run with unprecedented power, flexibility, and extensibility. NeXTstep's advantages for the *Mathematica* user include:

- Concurrent window interface: *Mathematica* runs elegantly alongside other applications in NeXTstep's true multitasking UNIX® environment.
- Pure PostScript graphics: Display PostScript® provides identical imaging on the 92 dpi NeXT MegaPixel Display™ and on the NeXT 400 dpi Laser Printer, as well as other PostScript output devices.
- Interprocess communication: to create custom *Mathematica* applications in NeXTstep, (known as custom front ends to *Mathematica)*, a programmer can message *Mathematica* via the Speaker/Listener protocol to obtain computational results from *Mathematica* in a way that is completely transparent to users. One can message *Mathematica* itself or its computation engine—the kernel.
- *Mathematica* Notebooks: NeXTstep is the only workstation platform for which Notebooks are available, making NeXTstep the ideal environment for technical publishing and courseware development.
- Parallelization: with a NeXTstep application called Zilla, multiple *Mathematica* sessions may be invoked on networks of NeXT computers to allow the simultaneous solution of different parts of a large problem.
- Media processing: sounds and images may be acquired using NeXT computers and further analyzed via *Mathematica*.
- Superior processor and operating system technologies: many problems previously unassailable on less powerful desktop computers can now be addressed using NeXT's 25 MIPS computational power and robust multitasking and virtual memory.

". . . not all versions of Mathematica are created equal."
—Byte, February 1991

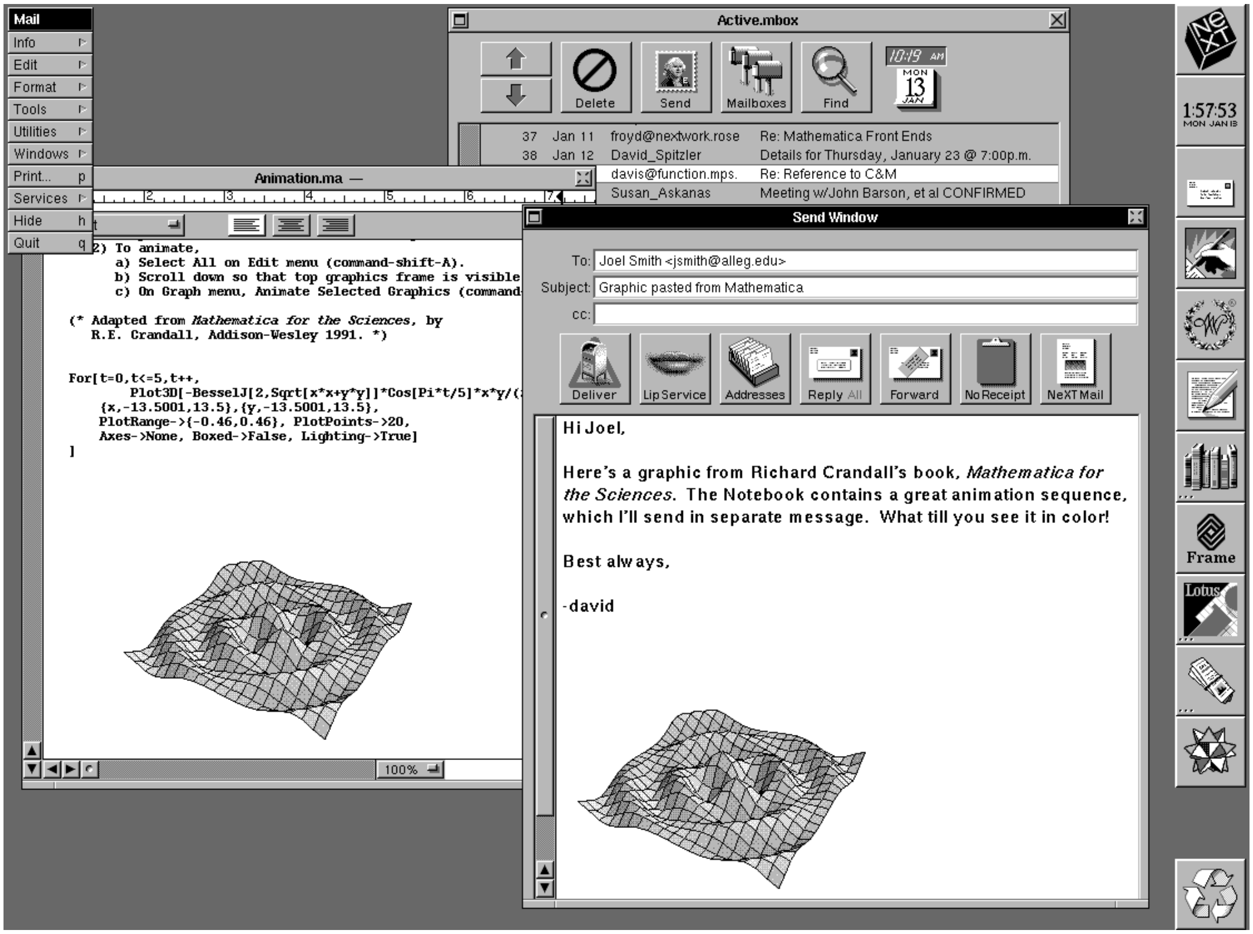

Figure 1 *Multimedia cut-and-paste facilities available in all NeXTstep applications. A graphic image is cut from Mathematica and pasted into a NeXTmail Send Window so it may be sent to a colleague. Sounds can be pasted as easily.*

II. Concurrent Window Interface

The user running *Mathematica* starts with the considerable advantage of a standard NeXTstep application: Not only are the menus, windows, and dialog panels extensive and consistent with all other NeXTstep applications, but all text and graphics employ the industry-standard PostScript imaging model. This means the user may cut and paste information, including text, graphics, sounds, and animation sequences, readily between *Mathematica* windows and other application windows—capabilities lacking in Sun platforms and MS-DOS and Windows environments.

- An application called CircuitBuilder, for example, generates explicit circuit equations into one of its windows. These often-formidable equations can be pasted immediately into the window of a concurrently executing *Mathematica* application. When the equations are solved via *Mathematica*, the user then can benefit from a range of features shared by all NeXTstep applications: the ability to graph frequency response curves, or work out circuit parameters numerically and generate tables, or perform statistical analyses.

In addition, the NeXT user may keep *Mathematica* running—yet hidden—in the application dock, ready for recall at any time. Because of the refined and well-tested efficiency of the Mach UNIX multitasking operating system, *Mathematica* requires virtually no overhead until the user calls for it; *Mathematica* can be launched at the beginning of a login session and always be in a state of readiness without consuming memory or CPU cycles.

III. Pure PostScript Graphics

A large, high-resolution graphics display is required for *Mathematica* users. The NeXT MegaPixel Display—a two-page monitor, the minimum standard for all NeXT computers—is ample space to view multiple *Mathematica* sessions and other tasks;

and the PostScript imaging model built into NeXTstep, provides the highest quality graphics available today. Most other desktop platforms do not offer these capabilities.

"[On an MS-DOS platform] Graphics, when they finally appeared, took over the entire screen, making it impossible to examine graphs side-by-side with the associated mathematical expressions."

—PC Week, August 12, 1991

A Unified Imaging Model

Having co-developed Display PostScript with Adobe Systems, Inc., NeXT employs PostScript for all imaging, whether to a screen, a printer, an imagesetter, or a fax machine. Unlike other platforms, NeXTstep running *Mathematica* does not present the user with two imaging models, one for screen display and one for printing. Furthermore, *Mathematica* renders graphic images in PostScript across all platforms. This allows, for example, seamless integration of *Mathematica* images generated on other vendors' platforms and brought into the NeXT environment.

As described in the previous section, graphics—both color and gray-scale—may also be pasted between NeXTstep applications. Color graphics are managed just as easily as gray-scale graphics, since color is an integral part of the PostScript model; and more complete color support is provided by PostScript Level II, that is implemented in NeXTstep Release 3. Release 3 uses new features in PostScript Level II to provide device-independent color matching—so the user is always assured of consistent output regardless of which printer happens to be used.

- A consistent graphics model across color and gray-scale and across different output devices provides a distinct benefit in areas such as chaos research. Whether the problem involves bifurcation, fractals, or phase transitions, the user generally confronts tremendous complexity. This complexity often manifests itself as a fine graphical detail, hence the advantage of high-resolution display and output.
- An example of the how important color an be involves three-dimensional plotting of data, whereby a fourth dimension is indicated by the use of color. Data points (x,y,z,w) where w is a fourth parameter, can be visualized by allowing w to be represented by a range of hues, say from red to violet. A related example involves plotting amplitude and phase of a complex function using a colored surface. In this case phase might be represented by a hue.

IV. Custom Front Ends—Integrating *Mathematica* With NeXTstep

A primary advantage of the NeXT platform is the ability to create custom NeXTstep front ends to *Mathematica*—separate code yet integrated applications that take advantage of *Mathematica's* power while the complexity of its syntax is hidden from the user. The ability to create custom front ends using *Mathematica* as a core engine is significantly faster and easier to do in NeXTstep than in other development environments. NeXTstep provides all the tools necessary for users to begin writing custom front ends: a complete library of interface objects and a powerful program construction tool, Interface Builder™.

Interface Builder is NeXT's tool for graphical interface development and project management. Interface Builder improves programmer productivity with its complete environment for laying out, constructing, and testing interfaces and managing the objects comprising an application.

The NeXTstep Application Kit™—NeXT's class library of more than 100 optimized objects—allows interapplication communication via NeXT's Speaker/Listener protocol. The Application Kit also includes a set of core objects that provide the framework required by any application, and a powerful set of interface and support objects that provide advanced functionality well beyond the windowing toolkits found in other workstation platforms.

Users will find a variety of commercially available objects that allow *Mathematica* to be called from within Interface Builder. Or programmers can write their own code to message *Mathematica* via Speaker/Listener. However one chooses to create a custom front end, the *Mathematica* user can write sophisticated applications in NeXTstep that call on *Mathematica* to handle calculations in a transparent way.

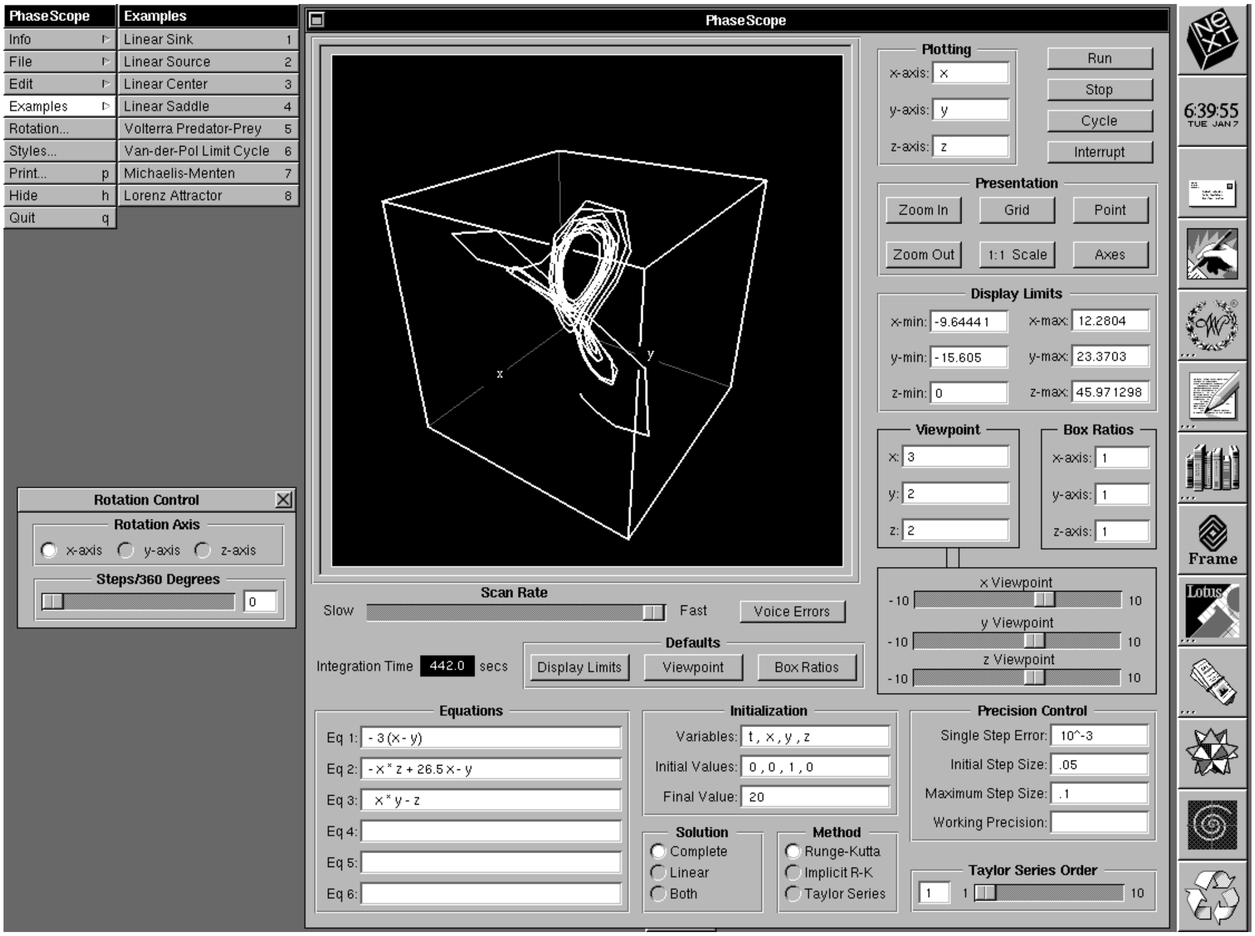

Figure 2 *PhaseScope is a custom front end that messages Mathematica's kernel. Here, the Lorenz Attractor is integrated by Mathematica and displayed by PhaseScope.*

Custom Front Ends: Aiding Ease of Use and Understanding

There are certain instances when it is better to hide the power and complexity of *Mathematica*. For many simulations, custom NeXTstep front ends can bring mathematical models to life in a way that could only have been done, more laboriously, with physical demonstrations. In this way users can work with a simpler, friendlier custom-developed interface so they do not have to learn as much *Mathematica* to study a particular problem. And faculty can determine how many parameters of a complex model a student must deal with at any given time. A model can be simplified by hiding parameters within a relatively simple interface; more parameters can be displayed as the student's knowledge progresses to a more complex level.

A student of knot theory might want to assemble graphically or dissect a knot while seeing the associated polynomial invariants constructed in parallel, without worrying about *Mathematica* syntax. With complete control over input, custom interfaces can be more interactive than the *Mathematica*'s standard Notebook front end—with sliders, for example, dynamically attached to *Mathematica* functions or graphic objects, as in the knot theory example. Front ends can even be designed as laboratory learning games or quizzes or tests.

When creating a custom front end, programmers have the choice of calling on the *Mathematica* application itself or its executable kernel. Gourmet[1], for example, an experimental supercalculator messages the *Mathematica* kernel by sending it problems and displaying the answers in its own window. Other front ends and related tools created for academic applications include:

---

1. By Prof. Richard E. Crandall of Reed College and NeXT's Chief Scientist. Described in "Adventures in supercalculator design." *NeXT on Campus*™, fall 1990.

- **PhaseScope**[1] is a front end that messages *Mathematica*'s kernel. Providing a complete user interface and graphics display, PhaseScope qualitatively analyzes the stability characteristics of dynamical systems. PhaseScope was written to be used as a learning tool to support topics in a typical course in differential equations. PhaseScope can also be used as a general purpose modeling tool, where the user can check the qualitative behavior of assumptions before proceeding to a detailed analysis. Although some knowledge of differential equations is required to understand fully the ways PhaseScope can be used, no such knowledge is required to use it exploratatively.
- **Function and Sons**[2] is an example of a front end that messages the *Mathematica* application. Function and Sons displays graphs of a function and its first and second derivatives, in random order, and asks the user to identify which is which. A wrong answer queries the user to try again or to view the correct results.
- **MathTools**[3] is a collection of Objective C classes that can be used with Interface Builder. These classes have methods for sending commands to *Mathematica* as well as for receiving and displaying textual and graphical results from *Mathematica*. This makes it easy to build graphical front ends to existing *Mathematica* packages and Notebooks and to have other applications call on *Mathematica* to perform calculations and graphing.
- **TaylorSeries**[4] is another example of a front end that messages the *Mathematica* application. TaylorSeries provides a graphical means of comparing a function with one or more of its Taylor series approximations. The user first enters a function to be approximated, and then chooses Calculate Approximation which messages *Mathematica*. The user then has the option of graphing a single or multiple approximations, which are also generated by *Mathematica*.

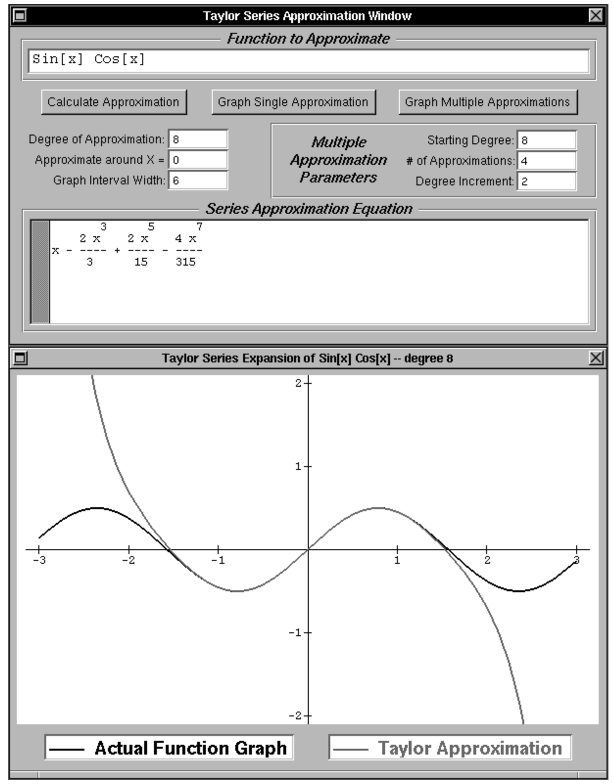

Figure 3 *A custom front end to the Mathematica application, TaylorSeries requires Mathematica syntax to be entered into its Function to Approximate field in order to solve the Series Approximation Equation.*

Front Ends for Financial Applications

The success of financial analysts and traders often depends on analyzing large volumes of data, often in real time, using sophisticated models. Financial engineers are finding faster ways to prototype applications by using *Mathematica* running on NeXTstep. Examples of such prototyping include analyzing the path of interest rates over time, calculating the cost of a commodity at a point in the future, or asset valuation and risk management.

1. By Prof. Michael J. Mezzino, Jr. of University of Houston - Clear Lake. Described in "PhaseScope energizes the study of mathematics," *NeXT on Campus*, winter 1991.
2. By David J. Fischer, a junior applied optics major at Rose-Hulman Institute of Technology.
3. By Profs. Charles Fleming and Judy Halchin, application developers for Educational Computing Services at Allegheny College.
4. By Mike Allard, a senior computer science major at Rose-Hulman, and Tony Zamora, a Ph.D. candidate in computer science at Indiana University

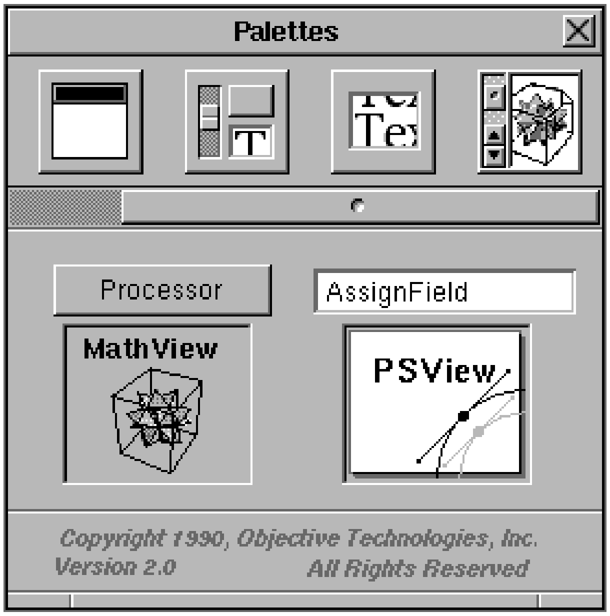

Figure 4 *As a loadable palette for Interface Builder, MathPalette (icon in upper right-hand corner) provides objects (pictured are OTProcessor and OTAssignField) that can be dragged and dropped into custom front ends to Mathematica.*

"The developers on our staff are not UNIX experts, but with NeXTstep they were developing powerful applications in UNIX before they even knew the definition of the word 'shell.'

". . . we are heavy users of Mathematica for prototyping all of our financial algorithms. Using a custom palette. . . called MathPalette, we use most of the functionality of Mathematica (including 3D graphics). . . . Our statistician—who never programmed in his life—delivers functioning prototype systems to our traders with no intervention from developers!"
—Hadar Pedhazur, Vice President, Equities Technology, UBS Securities, Inc. Open Systems Advisor, July 17, 1991

The NeXTstep user working with *Mathematica* enjoys a clear advantage for these classes of applications: NeXTstep allows programmers to build mission critical custom applications rapidly that take full advantage of *Mathematica*'s power. This allows programmers to keep a traders' customers in the know by quickly prototyping the effects of a changing market. As in the world of college mathematics, front ends to *Mathematica* are simple to build, yet for traders these front ends are an integral part of doing business.

To build custom financial applications, financial engineers use the NeXTstep development environment, *Mathematica,* and commercial objects and tools that have been created specifically for the financial market. One such tool is MathPalette—a loadable palette of Interface Builder-compatible objects.

- **MathPalette**[1] consists of a palette containing OTProcessor, OTAssignField, and OTMathView objects. OTProcessor manages communications with the *Mathematica* kernel, supplying information about its status, and returning responses to user requests. OTAssignField allows the user to set values the of *Mathematica* variables. OTMathView displays *Mathematica*-generated Encapsulated PostScript images.

V. *Mathematica* Notebooks for Publishing and Courseware Development

It is without dispute that the *Mathematica* Notebook has created a revolution in the world of mathematics. The Notebook is a powerful working environment that allows the user to create interactive documents that can contain executable *Mathematica* code, nonexecutable text, graphics, sounds, and animation sequences. The uses for Notebooks are infinite, ranging from developing courseware to technical publishing to simply distributing mathematical ideas.

While the *Mathematica* Notebook is not unique to the NeXT environment, it runs only on a few platforms. And the Notebook implementation on the

1. Available from Objective Technologies, Inc. Other palettes available from OTI are Chooser, GraphPalette, OTProvide, and SmartFields.

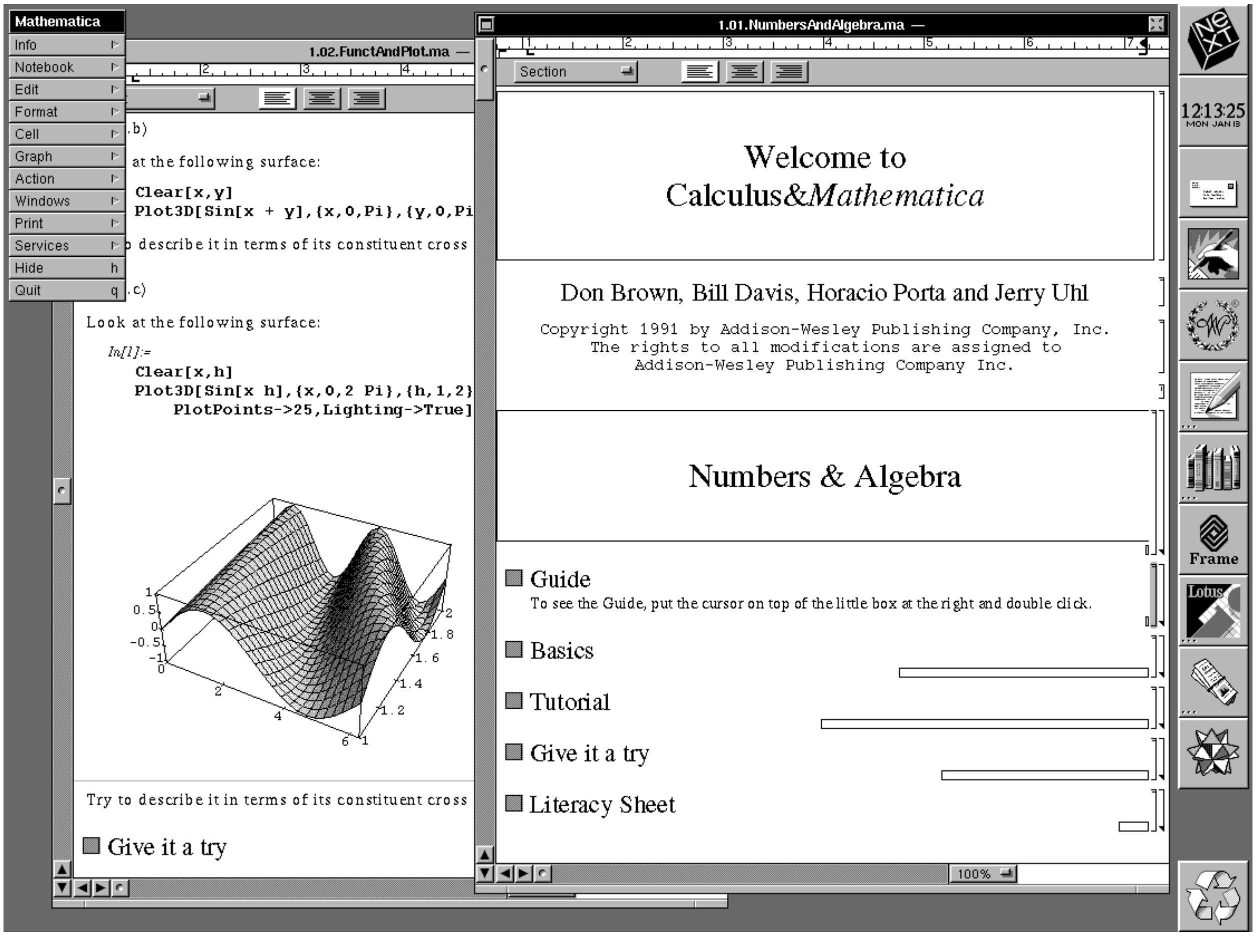

Figure 5 *Calculus&Mathematica courseware developed using Mathematica Notebooks exclusively. The course is for structured or partly supervised teaching and learning, or self-paced independent study.*

NeXT platform is the only UNIX workstation version available today. All other workstation platforms must run *Mathematica* from a traditional, cumbersome UNIX-shell command line interface.

The basic working unit within a Notebook is called a cell. Each cell holds one type of data, and cells can be grouped hierarchically to create organized mathematical concepts, making Notebooks easy to navigate.

While Notebooks—as standalone documents—are identical across all platforms that support them, the NeXT user running *Mathematica* has the considerable advantage of a well-rounded computing environment. Multitasking, virtual memory, multimedia cut-and-paste facilities, and Display PostScript all make the NeXT environment superior for Notebook development and use.

Courseware Development

Mathematica is a relatively new product, introduced at the same time as the original NeXT Computer, in 1988. The earliest adopters of *Mathematica* were mathematicians, and it is no surprise that many of the courses initially developed are in calculus. Today, development is not limited to any mathematical subdiscipline or even to mathematics itself. Courses in differential equations, linear algebra, signal processing, physics, engineering, econometrics, image processing, among many others are being developed using Notebooks.

"Mathematica without Notebooks is a superb mathematics processor, but Mathematica with Notebooks is an unparalleled system for producing mathematics courseware."
—The Mathematica Journal, Volume 1, Issue 1, Summer 1990

"The Notebook file format is common across all the platforms that support it. . . Notebooks created on a NeXT can be displayed and run on a Macintosh." —Personal Computer World, March 1991

Traditional courses in mathematics relied on faculty lecturing on theory and illustrating theory in pencil-and-paper assignments for students. These exercises, for many students, became drudgery.

Notebooks offer a compelling alternative method of learning and instruction. When a Notebook is used to its potential, it becomes a medium where each calculation, graph, animation, or sound is a living piece of data and can be reexamined, reevaluated, or corrected to manipulate an infinite number of functions, parameters, or variables—all within a single document. Furthermore, since *Mathematica* handles rich text the same way a wordprocessor does, each numeric or graphic example may also contain an explanation of the theory and importance that are necessary for proper understanding. And those students who understand concepts presented visually better than concepts presented symbolically, benefit significantly from *Mathematica*'s ability to display results instantaneously using two- and three-dimensional plots.

Courses have been developed worldwide using *Mathematica* Notebooks, many on NeXT computers. Among the courses available are:

- **Calculus&*Mathematica*,**[1] a one-year calculus course in a laboratory setting in which faculty seldom lecture. Lessons are in 60 Notebooks that students complete at a rate of 2 per week. Notebooks are interchangeable between the platforms that support them, allowing the course to be taught on NeXT and other platforms. Calculus&*Mathematica* is currently being taught at 20 institutions in North America.
- ***Calculus Using Mathematica*,**[2] a complete text with companion *Mathematica* Notebooks. The curriculum emphasizes the role of calculus in scientific and mathematical applications.
- **Symbolic Analysis of Signals and Systems,**[3] an extensive *Mathematica* package for digital signal processing, including: Laplace and Fourier transforms (for continuous-time signals) and z, discrete Fourier, and discrete-time Fourier transforms (for discrete-time signals). All transforms work for two-sided and multidimensional functions and show students each step of the transformation process. The accompanying *Mathematica* Notebooks teach piecewise convolution and the z-transform.

1. By Profs. Don Brown, Horacio Porta, and Jerry Uhl of University of Illinois, and Bill Davis of Ohio State University. Available from Addison-Wesley Publishing Co.
2. By Prof. Keith Stroyan of University of Iowa. Available from Academic Press, Inc.

Publishing

Although *Mathematica* is not a page layout program, the Notebook front end allows users to create attractive documents with consistent text formatting commands, multiple fonts, and margin rulers. Flexible cell spacing and text leading commands give the user control over paragraph formatting.

For more sophisticated publication-ready technical documents, the combination of the TEX™ document processing environment—including TEXView™, a PostScript previewer—and *Mathematica* make NeXT an optimal technical publishing environment. Moreover, both TEX and *Mathematica* are free of charge for NeXT educational users, and TEX is included on systems for all of NeXT's users.

Alfred Gray, professor of mathematics at the University of Maryland College Park sums up document processing for the NeXT user: "A NeXT workstation is the most convenient place to use *Mathematica*. First, the ability to combine text and graphics in one file makes working with *Mathematica* on a NeXT much easier than on other workstations. Even if the other workstations have windows, they do not have Notebooks. Secondly, editing is much easier. I can work three times faster on a NeXT than on another workstation. Thirdly, it is easy to transfer information between *Mathematica* and word processors. When I write a research article that requires graphics, I work with TEX and *Mathematica* simultaneously. I generate the graphics with *Mathematica,* write the article with TEX, and put the graphics in the TEX document."

3. By Brian Evans of Georgia Institute of Technology

VI. Parallelization

Although distributed network computation in itself is not new, the NeXT environment provides easier access to supercomputing power for the user who needs more cycles than what a few machines on a desktop can provide, as well as the user who does not have the means to access supercomputer time. A NeXTstep application called Zilla[1], available with all NeXT computers, provides a graphical interface for supercomputing via a network of NeXT computers. Through this interface, the network can be used to create a virtual supercomputer by distributing computations automatically among two or more machines. Zilla can reach Cray-level supercomputer speeds on certain problems by distributing pieces of the problems to individual computers, running separately and often at different times.

For these innovative computational abilities Zilla received the *Smithsonian / Computerworld Award for Best Scientific Software of 1991*.

Zilla makes it easy to parallelize *Mathematica* problems. And since *Mathematica* is comprised of two distinct parts—the front end and the kernel, each capable of running separately, even on different machines—the user can, for example, launch several *Mathematica* kernels on many computers using Zilla, each kernel having a unique set of parameters, multiplying the power of *Mathematica* many-fold. A network of 100 NeXTstation™ computers, for example—a quantity defined as one Zilla Unit—performs like a typical supercomputer for a tiny fraction of the cost and with none of the software conversion and support expenses associated with traditional supercomputing.

VII. Media Processing and Media Integration

NeXT computers support standard multimedia data formats across most applications, including sounds and images, in some instances, acquired on a real-time basis. *Mathematica* provides an excellent way to experiment with media processing, as *Mathematica* closely relates sounds and graphics: both can be generated from an equation.

1. Developed by Prof. Richard E. Crandall of Reed College and NeXT's Chief Scientist. Described in *NeXT on Campus*, summer 1990 and spring 1991 issues.

Generating the sounds of mathematics via *Mathematica* is greatly enhanced by the CD-quality sound capabilities that are part of every NeXT computer. Although it has been possible for many years to hear mathematical functions, this usually required complex programming. Running *Mathematica* on NeXTstep, sounds can even be pasted into a Notebook from other NeXT sound applications and then can be converted into *Mathematica* expression form and analyzed or manipulated.

- Assuming that a user has a sound file and wants to create a spectrogram or a sonogram. All the user needs to do is to read the sound file into *Mathematica,* perform the appropriate Fast Fourier Transforms (FFT) using built-in *Mathematica* functions Fourier[] and InverseFourier[], and then graph the results. For image processing, the user can read in an image file, and easily apply two-dimensional FFTs, antialiasing, or deblurring routines via *Mathematica*.

The potential for media-oriented transformations is substantial, for *Mathematica's* new sound functions are relevant to a broad range of disciplines. Just as *scientific visualization* is changing the way we look at scientific results, so, too, the combination of NeXT and *Mathematica* may popularize *scientific auralization*, allowing us to listen to our results. The advantages to musicians and acoustic researchers is clear, but physicists and engineers also need these capabilities to analyze complex signals. NeXT computers provide facilities, out of the box, to manage media of virtually all types. With *Mathematica*, the right tools to analyze these media are readily at hand.

VIII. Superior Speed and Memory

"The new features [in Mathematica 2.0] . . . will tax many PCs to their limit." — InfoWorld, September 30, 1991

The claim of superior speed on a NeXT computer is by no means artificial. Users have found that the 68040-based NeXTstation, for example, runs *Mathematica* calculations about an order of magnitude faster than a Macintosh IIci. Perhaps equally important for the physical sciences and financial modeling, where large data files are common, all NeXT computers enjoy excellent virtual memory performance. When the user installs *Mathematica* on a NeXT computer, no additional memory is required to launch and run the application. Virtual memory management is so seamless that, depending on disk configuration, the user can run sessions that require tens of megabytes of memory. Writing a *Mathematica* recursion for which memory requirements run to infinity will not damage other processes running under NeXTstep. NeXTstep is graceful and foregiving in the way it handles the exhaustion of resources.

- An example of the power of speed-with-memory: The NeXT Scientific Computation Group was presented with a problem, a nonlinear system in ten unknowns arising from a chromatographic model, that a number of other platforms failed to solve. NeXT's Scientific Computation Group solved this problem symbolically on a NeXTstation running *Mathematica*. During the successful computations virtual memory ranged up to 20 megabytes. Most other desktop platforms would not have provided the required computational speed or memory necessary for this problem.

IX. Conclusion

That *Mathematica* and NeXT products together make a great combination is no accident. Indeed, *Mathematica* was effectively introduced to the world when NeXT announced its first computers in 1988, and NeXT is commited to continuing to provide the best platform for using *Mathematica*. Among the benefits of the great integration of *Mathematica* and NeXT hardware and software that users enjoy today:

- An innovative Notebook front end to *Mathematica*—a tool that is fundamentally changing the way faculty teach numerically-intensive subjects, and the way students visualize mathematical relations.
- NeXTstep is the best all-around development environment for creating custom front ends to *Mathematica* for mission critical tasks.
- A platform that allows completed applications—both commercial and custom—to work together
- NeXT's optimized system performance which maximize throughput for *Mathematica* users.

Power, customization, ease of use, and ease of development are the technology goals shared by Wolfram Research, Inc. and NeXT Computer, Inc. Together, NeXT and *Mathematica* provide users with an environment of unparalleled richness—whether in finance, engineering, or education.

Additional Resources

- Blachman, Nancy. *Mathematica: A Practical Approach,* Prentice Hall, 1992
- Crandall, Richard E. *Mathematica for the Sciences,* Addison-Wesley, 1991
- Gray, Theodore W., and Glynn, Jerry. *Exploring Mathematics with Mathematica,* Addison-Wesley, 1991
- Maeder, Roman. *Programming in Mathematica,* Addison-Wesley 1991
- Smith, Cameron. *The Mathematica Graphics Guidebook,* Addison-Wesley, 1991
- Vardi, Ilan. *Computational Recreations with Mathematica,* Addison-Wesley, 1991
- Wagon, Stan. *Doing Mathematics with Mathematica,* W.H. Freeman, 1991
- Wellin, Paul, editor. *Mathematica in Education,* Sonoma State University
- Wolfram, Stephen. *Mathematica: A System for Doing Mathematics By Computer,* Addison-Wesley, 1991
- *The Mathematica Journal,* Miller Freeman, 1990, 1991, 1992
- *NeXT on Campus* NeXT Computer, Inc., 1990, 1991