NeXT on Campus™

Spring 1991

In this issue:

MidasPlus brings molecular modeling to the desktop
Solving real-world problems in a *Mathematica*® lab
Experiencing non-traditional art with Fluxbase
NeXT computers: research partners at University of Washington
Interpersonal computing at ITESM
Writing custom front ends to *Mathematica*

Image 1

A revolution in education and research: *Mathematica* with the NeXTstep® advantage

Image 3

Image 2

NeXT on Campus™ is published three times a year (September, January, and May) by NeXT Computer, Inc. for the higher education community. Each issue of the journal highlights advances in instruction and research in which NeXT™ technology has played a key role, as well as NeXT-related news, feature stories, products, and resources available to NeXT users. *NeXT on Campus* welcomes your comments, suggestions, and contributions. You may send e-mail to next_on_campus@next.com or write to *NeXT on Campus*, NeXT Computer, Inc., 900 Chesapeake Drive, Redwood City, CA 94063. To receive a subscription to *NeXT on Campus*, send your request to the e-mail address or U.S. mail address above.

This issue of *NeXT on Campus*, as well as all back issues, are available electronically (in FrameMaker and PostScript formats) at the Purdue public archive site (nova.cc.purdue.edu) in the pub/next/Newsletters directory. Issues can be retrieved by using the file transfer protocol program.

Regretfully, the editorial headquarters of *NeXT on Campus* is unable to handle questions that are technical in nature or queries for information other than business concerning *NeXT on Campus*. For technical questions, please contact your local support center. If you need information on NeXT products, please call 1-800-848-NeXT.

WRITING FOR *NeXT on Campus*
If you are interested in writing for *NeXT on Campus*, please send a written proposal with your idea(s) to the e-mail or U.S. mail addresses above.

COLOPHON
NeXT on Campus is produced entirely with NeXT computers. Text is written with WriteNow® 2.0. Screen shots are captured with Scene or Grab, demonstration applications. Page composition is done with FrameMaker® 2.0. Proofs are printed with a NeXT 400 dpi Laser Printer. All text and image files are transferred via a NeXT optical disk to a full-service electronic prepress vendor.

CREDITS
Managing Editor: David Spitzler
Technical Editor: Jeff Wishnie
Designer: Joan Howard

Advisors: Jay Capela, Jean Craig-Teerlink, Conrad Geiger, David Grady, Markus Lankeit, Bruce McIntosh, Lise Pfau, Barry Silverman, Karen Steele, Jodie Stowe, Tom Suiter, Ronald Weissman, Kristofer Younger

Writers & Editors: Jayson Adams, Beth Conover, Kevin Cracraft, Richard Crandall, Barbara Gibson, Theodore Gray, Joe Hutsko, Anne Kaplan-Neher, Eric Larson, Susan Mobley, Jane Rich, Sha Xin Wei, Ann Turley, Joe Welinske, Adrienne Wong

NeXT Computer, Inc. extends its thanks to Addison-Wesley Publishing Company and Wolfram Research, Inc., and special thanks to the faculty, researchers, and students whose work appears in this issue of *NeXT on Campus*.

Spring 1991, Volume 2, Issue 3

NeXT on Campus

Table of Contents

Spring Feature

***Mathematica* brings new understanding to teaching and research** 2

Academic Projects

University of California, San Francisco
The coming of desktop molecular modeling 8

University of Nebraska, Lincoln
Students solve real-world problems in *Mathematica* lab 10

University of Iowa, Iowa City
Fluxbase: cataloging and experiencing non-traditional art with technology 12

Campus Profiles

University of Washington, Seattle
Drawing students into research 14

Instituto Tecnologico y de Estudios Superiores de Monterrey
Interpersonal computing makes its mark 18

NeXT Technology

The lure of very large numbers: Further adventures in community supercomputing 20

Creating custom front ends to *Mathematica* 22

Upgrading to Software Release 2 and the 68040 Upgrade Board 24

NeXT Resources

Academic Projects 26

NeXT Archives 29

In focus: *Mathematica* 2.0 30

Third-Party Products 32

NeXT User Groups 34

Mathematica brings new understanding to teaching and research

I. Teaching with *Mathematica*

Two debates in the mathematics community today center on training new mathematicians or users of mathematics, and the relationship between research and teaching. The common thread is how best to impart mathematical knowledge and a way of doing mathematics. Nothing is more essential than an enthusiastic and well-organized teacher. Every teacher, however, appreciates tools that are flexible, reliable, and stimulating. Traditional math classes, as taught with chalk, mimeographs, and textbooks in expository lectures face several limitations:

- Chalk works well in the hands of a skilled board-artist but is limited to the rough, nonreproducible figures that can be sketched in front of a class.
- Texts and mimeographs can present materials efficiently, but in their traditional form must balance expository motivation or clarity against depth.
- There is tension within course syllabi: should we cover examples more carefully to motivate the theory, or must we race through the definition-lemma-theorem-proof-corollary cycle? In frequently taught "service courses," such as calculus and differential equations, the syllabus has devolved to a "grab bag" of unmotivated topics—formal techniques like partial fraction expansions or series solutions to an ordinary differential equation—coupled with artificially neat exercises.
- There is discontent among both students and teachers with the lecture model where the math instructor "reads" lecture notes to a class. Students' responses are largely limited to homework exercises and exam questions.

To invigorate the study of mathematics, here are several ways that students and faculty can use *Mathematica*®—a powerful tool for teaching and learning.

Mathematica as a calculator

Why not just use a calculator? Students discover that *Mathematica* on NeXT™ computers is more flexible than a handheld symbolic calculator. Aside from its arbitrary precision arithmetic (exact arithmetic over the rationals), rich set of calculus and linear algebra operators, special functions, and revolutionary integration of sophisticated graphics and animation, *Mathematica* is a programming language designed to read like standard mathematics. This brings the computer within reach of a student or teacher who expects to focus on the mathematics—not the machine.

At the Rose-Hulman Institute of Technology, Mark Yoder, assistant professor of electrical and computer engineering, now teaches his image processing course in a laboratory of 25 NeXTstation™ computers. Yoder felt that advantages of using NeXT computers and *Mathematica* significantly outweighed the risk involved with introducing a new computing environment to his students.

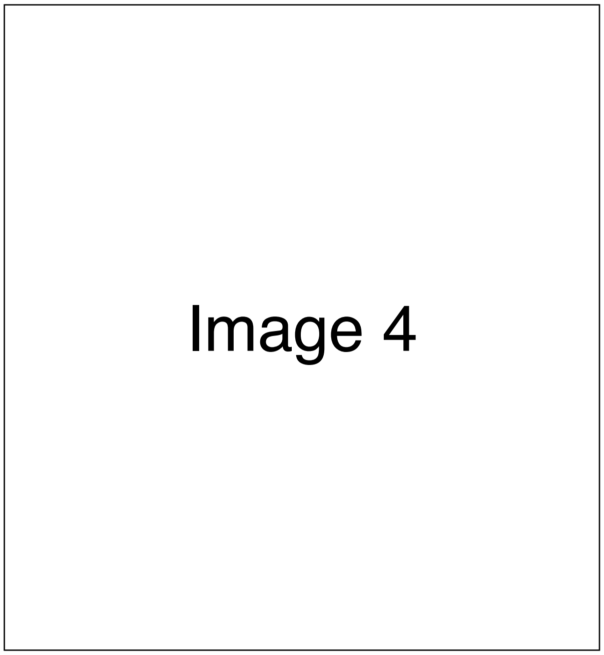
"I used to teach the image processing course in a traditional lecture setting, and the major disappointment I had with it was we couldn't easily implement a given algorithm to see what it did to an image," Yoder comments. "With *Mathematica,* I frequently introduce a new algorithm during the first half of a period and have the students implement the algorithm during the second half." Because *Mathematica* can quickly evaluate complex image processing algorithms and display the results, students can experiment and gain a better understanding of image processing.

Multimedia authoring with *Mathematica*

The superiority of *Mathematica* on NeXT computers over special purpose hardware or software becomes clear when students and faculty begin creating *Mathematica* Notebooks—electronic documents that integrate live algebraic or analytic calculations, text, graphics, animation, and sound into an easy-to-use environment. Students can interact with Notebooks by changing variables, defining functions, even writing their own animations or sound compositions; they can carry Notebooks on a disk or exchange them with other students via electronic mail. Teachers find Notebooks a flexible medium to write up labs and lecture notes for class.

> *Students can interact with Notebooks by changing variables, defining functions, even writing their own animations or sound compositions.*

At the Georgia Institute of Technology, Brian Evans, a graduate research assistant, teaches a course in signal processing. His students write up their work in *Mathematica* Notebooks. In the Analog Filter Design unit, for example, students use animation to visualize the changes in the filter



Plots for a two-pole filter $\frac{w_0^2}{s^2+\frac{w_0 s}{Q}+w_0^2}$ *with Q varying from .4 to .6*

as they vary the parameters; later they can redesign the filter and observe the results.

Relating theory with application

Mathematica enables students to encapsulate knowledge in a usable form, whereas in a traditional course this encapsulation takes the form of theorems and sometimes, memorized proofs. In Brad Osgood's core calculus course at Stanford University, for example, students are guided into reinventing the bisection and Newton's root-finding methods, then introduced to *Mathematica*'s built-in root-finder function. Armed with this *Mathematica* function, plus some knowledge of the theory behind its operation, students can apply it later in the course. Similarly, they can build an Euler method for solving ordinary differential equations before being introduced to *Mathematica*'s sophisticated adaptive Runge-Kutta method. They compare these methods by error analysis, reinforcing the differential calculus learned before, and apply these encapsulated tools and theorems to a variety of physical problems that would have been beyond the scope of a traditional calculus course.

Mathematica as a math microworld

Also at Stanford, David Stork is pioneering a course in Adaptive Pattern Recognition and Neural Networks using *Mathematica*. Since *Mathematica* supports both procedural and functional styles of programming, students who have some programming experience can implement their projects in C or Pascal-like style, though they can transfer the neural net methods that depend on vector calculus, linear algebra and some statistics directly into *Mathematica* Notebooks. It is rewarding, especially in this visually oriented discipline, for students to see the results in graphical form.

Jon Barwise, founding co-director of the Center for the Study of Language and Information at Stanford (now at Indiana University), comments in an article on the gulf between research mathematics and what mathematicians teach: "There are several computer programs designed to teach standard materials, that relieve faculty of the chore. The worst are basically automated page turners with on-line grading to help the student get through some standard text. Such programs only exacerbate the problem, increasing the distance between what the students learn and what mathematics is really like.

"There are many alternatives, however. For one, we can use the computer as a tool for experimentation. By making mathematics come alive, it can serve as a powerful source of intuition and inspiration. A program like *Mathematica*. . . provides tools with which even freshmen can explore uncharted waters, looking for patterns and then trying to understand them. Using such programs made me realize what toy examples I have always used in teaching calculus. We have in calculus a most sophisticated tool for exploration, but we seldom encourage students to use it except in a goldfish bowl fished over by countless generations of earlier students. . . .

". . . By allowing the student to carry out huge calculations no one could perform with pencil and paper, students can apply what they learn to original and significant problems. And the graphic capabilities of the modern computer allow us to visualize functions and other patterns that would have been completely inaccessible when I was a student. . . . In rethinking our curriculum. . . we have an exciting opportunity to introduce students to the joys of mathematical discovery that motivate us as mathematicians." (From "Notices of the AMS," v. 37 no. 8, 10/1990, p. 1017.)

Integrating *Mathematica* with NeXTstep applications

What's wrong with using special purpose math applications? Nothing, as long as they can be integrated into an instructor's needs. *Mathematica* is easily extensible in the NeXTstep® environment, providing a narrative structure that can be reshaped at anytime by the instructor. At the simplest level, one can execute C routines from within *Mathematica*. One of Stork's Notebooks, for example, uses a Voronoi tesselation routine that was compiled from public domain research code and called from within *Mathematica* like a standard *Mathematica* function.

There are instances when it is better to mask the full power and complexity of *Mathematica,* especially when dealing with a physical or geometric simulation. One can use NeXTstep to write highly interactive and robust custom applications backed by *Mathematica'*s computation engine. With Interface Builder,™ one can create and redesign these custom front ends easily, leaving the courseware author free to think about content and learning.

> *Mathematica is easily extensible in the NeXTstep environment, providing a narrative structure that can be reshaped at anytime by the instructor.*

For certain simulations, custom front ends can bring mathematical models to life in a way that could only have been done, more laboriously, with physical demonstrations. Best of all, students can carry these virtual labs around in a backpack and even write their own experiments. Advantages of custom front ends include capability masking, refined input methods, and refined output.

- For example: Students work with a simpler, friendlier custom interface so they don't have to learn as much *Mathematica* to study a particular problem. A student of knot theory, for example, might want to graphically assemble 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 standard Notebook front end, for example, with sliders dynamically attached to *Mathematica* functions or graphic objects, as in the knot theory example.
- Output can be refined. In place of the Notebook's top-to-bottom outline format, a custom front end can present more sophisticated forms of animation, perhaps of multiple objects with transparent overlays for a pharmaco-kinetic whole-body box model.

For more information on custom front ends, see "Creating custom front ends to *Mathematica*," on pages 22-23 of this issue, and "Integrated curriculum encourages students to become problem solvers," and "PhaseScope™ energizes the study of mathematics," from *NeXT on Campus* Winter 1991.

II. Mathematical Research with *Mathematica*

Twentieth-century mathematicians study a universe of objects—logics, prime numbers, dynamical systems, groups, knots, and spacetime singularities, to name a few—but since Gauss' day three centuries ago, they have moved away from a constructivist, "bare-handed" calculation approach. Why? Partly because of a shift in style and because the objects couldn't be computed. Theorems like "black holes radiate and small black holes radiate faster than big ones" were not discovered or proved by calculation in the common sense of the word. Until now, computers offered mathematicians essentially no help in either exploring their universe or proving their conjectures, because computer programs did not speak mathematical language; it took too much effort to teach them even the most basic mathematical knowledge and forms of reasoning. Computers and especially computer software have matured to the point where mathematicians can use computers in addition to their beloved chalk and blackboard.

Though mathematicians may entertain sharply different notions of calculation, a symbolic manipulation program (SMP) like *Mathematica* certainly can do small algebraic calculations that form part of the underbrush of mathematical research. Moreover, despite the injunction against relying on pictures for proofs, mathematicians often sketch ideas as figures. These sketches, whether algebraic or geometric, are guided by heuristics of experience. *Mathematica's* computable graphics, which are bound to some mathematical structure as opposed to merely graphic structure, provide an intelligent blackboard for this kind of exploration. In *Mathematica,* one works directly with mathematical types—graphs, differential equations, quadratic forms, tensors, and so on—rather than data structures or control structures that have little to do with the object of study.

Mathematica's principal strengths include: a rule-based pattern-matching, functional language; integrated manipulation of graphics, algebra, numerics, and sound; uniform kernel behavior across platforms; and, on the NeXT platform, a sophisticated document front-end called the Notebook.

Algebraic computation

For his doctoral research at Princeton University, John Sullivan (now at the Geometry Computing Project in Minnesota) used *Mathematica* for algebraic calculations that would have been tedious to perform by hand, and might have been "swept under the rug" in the final write up. In a paper on a crystalline approximation theorem for hypersurfaces, Sullivan calculated a bound on the number of unit balls that can be placed with centers on an r-sphere in dimension *n*. Sparing readers the nonessential algebra, he encodes the calculation as a few lines of *Mathematica* using Gegenbauer polynomials, which can be independently verified or studied.

Using *Mathematica* as a scriptable algebraic calculator can be useful. Mathematicians have written packages to perform algebraic computations in topology (the Kauffman-Jones polynomials, and braid groups) and elliptic curves. But *Mathematica* is also a programming language, which unlike all traditional languages and most SMPs, provides a concise representation of algebraic expressions in fairly standard mathematical syntax.

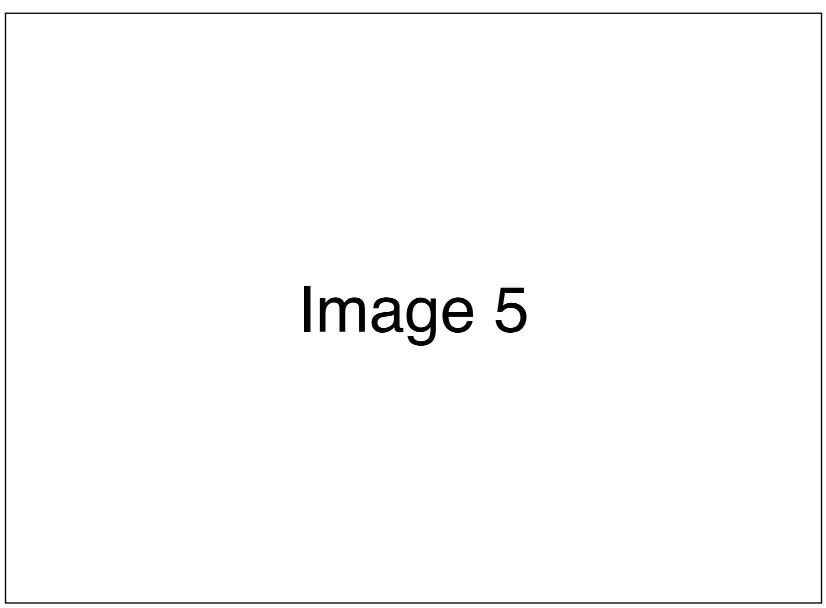
Dana Scott, at Carnegie Mellon University, wrote an entire course in projective geometry, partly to explore how feasible it was to implement mathematical ideas on a personal computer.

Scott says, "Mathematically, the most interesting outcome of the experiment of teaching this particular course was to find how easy it is to use the ideas of linear space theory, exterior products, and partial differential operators in presenting geometry, both in computations and in proofs aided by the powers of symbolic computation in *Mathematica*. The implementations of these ideas were done in a natural way—programming did not detract from the mathematics.

"The beauty of *Mathematica* is that we can not only do numerical computations such as:

```
PD[3 x + 4 y + z, 7 u + v - 25 w]
```

(which evaluates to zero showing that the point with coordinates {7,1,-25} lies on the line with coordinates {3,4,1}), but we can also allow indeterminants in the formulae that



Mathematica integrates seamlessly with other NeXTstep applications. Formulas can be exported in TeX™ form and viewed in a TeX previewer. Graphs, formulae, sounds, and text can be cut and pasted between Mathematica Notebooks and any number of applications.

give us conditions for something to happen."

To define certain functions on polynomials with indeterminate coefficients, a Degree function, for example, Scott taught *Mathematica* a few definitions (called "transformation rules" in *Mathematica*).

"The virtues to be emphasized here are that these rules work and are easy to write. It's important to be able to prototype ideas quickly, and my experience is that *Mathematica* allows that. Once the idea is made coherent and firmly demonstrated, then attention can be directed to more efficient implementations." His implementation in *Mathematica* of a generalized exterior product is essentially the standard definition:

```
PE[p___, 0, q___]      := 0;
PE[p___, l_ + m_, q___]:= PE[p,l,q] + PE[p,m,q];
PE[p___, n_ l_, q___]  :=
    n PE[p,l,q]/; DegP[n]== 0;
PE[p___, l_,l_, q___]  := 0;
PE[p___, l_,m_, q___]  :=
    -PE[p,m,l,q] /; Order[l,m] == -1;
PE[x,y] = w; PE[x,z] = -v; PE[y,z] = u;
PE[u,v] = z; PE[u,w] = -y; PE[v,w] = x;
```

Furthermore, working out explicit formulas allows the student or teacher to draw the figures corresponding to theorems using the extensive graphical capabilities of *Mathematica*.

Geometric computation

One of *Mathematica*'s chief strengths is its ability to produce powerful and mathematically useful graphics. At the University of Maryland, Alfred Gray has used *Mathematica* to study Gibbs' phenomenon. "The graphics interface is useful for pointing the way to new results. I discovered a new type of Gibbs' phenomenon that exists for Bessel functions, but not ordinary trigonometric functions. It arose when I did some graphs of Fourier Bessel series. Although Bessel functions have been studied for more than one hundred years, few graphs were made because of the computational complexity. The graphs were comparatively easy to do with *Mathematica*, and I have written an article with Mark Pinsky (Northwestern University) that gives a theoretical basis for the graphs."

Gray has also written *Mathematica* packages to compute geodesics—curves of shortest length—on two-dimensional surfaces. In his original setting, Gray collaborated with George Francis (University of Illinois, Urbana), using a Silicon Graphics computer to render the equations that were computed by *Mathematica*. In version 2.0 of *Mathematica,* using MathLink™ in the NeXT environment, he hopes to drive rendering software directly from *Mathematica* on a NeXT computer. "I feel *Mathematica*'s ability to generate graphics is its most useful feature."

At Stanford University, graduate student Sha Xin Wei has been experimenting with *Mathematica* as a tool for studying curvature-dependent evolutions of surfaces. The study of geometric variational problems traditionally has used methods from partial differential equations, geometric measure theory, and differential geometry. Until recently, the use of numerical simulations has been restricted because software has not been sophisticated enough to directly visualize the differential equations.

"With *Mathematica*," says Sha, "one doesn't need an expert programmer or a numerical analyst to quickly and flexibly explore the qualitative behavior of geometric nonlinear partial differential equations. Animation and color graphics provide a natural way to study all sorts of parametrized evolutions.

"It's important to be able to prototype ideas quickly, and my experience is that Mathematica allows that."

"One powerful application from the Geometry Computing Project is Ken Brakke's Surface Evolver program, which was written in C but used rather cumbersome data files to describe the topology and geometry of the initial surface. As soon as I got Brakke's Evolver, I wrote a *Mathematica* script that allowed me to prepare initial surfaces and view Evolver output using *Mathematica*'s powerful 3D plot functions, saving me from dealing with hordes of vertex, edge, facet lists. This was the easiest and quickest way to prepare initial data and to see the results in a geometrically meaningful fashion.

"The difference between *Mathematica* and all other graphics programs is that graphic objects can be directly linked to their geometric/algebraic structure. An example I like to use

Image 6

Mathematica can be used to present evolutions of surfaces or parametrized families of surfaces as animation sequences. Shown are (left to right): the output of Brakke's Surface Evolver, written in the C language; an isometric family of minimal surfaces, and evolution of a torus by mean curvature.

is the classical Weierstrass representation of minimal surfaces. Such a surface can be described via a complex vector-valued integral involving its Gauss map and another complex function. By simply inserting a complex rotation $e^{i\phi}$ into the Weierstrass representation, you can have *Mathematica* draw an entire smooth family of minimal surfaces out of one—it's a lovely mix of complex analysis, geometry, and animation."

Integrating *Mathematica* with other applications

Mathematicians have started to use *Mathematica* in concert with other applications. The Display PostScript® environment supports a TEX previewer, and with *Mathematica's* psfig utility, it is simple for Gray to paste *Mathematica* graphics into his TEX articles. Applications like FrameMaker® make it easy to integrate research into book-length monographs.

Gray says: "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."

Sha Xin Wei intends to use Interface Builder to construct front ends for other geometry applications and *Mathematica*. "I expect to use applications in concert with *Mathematica,* some running on other machines on a network, exchanging geometric and algebraic results with each other. We'll be able to write mathematical hyperdocuments with TEX-like displays that have directly manipulable algebra along the lines of FrameMath™ as well as embedded geometric simulations. My goal is to be able to open an article where, say, a surface is described as some minimal solution to a variational problem. I'd like to be able to select a patch on the surface using a mouse or perhaps a glove, define my own perturbation by writing a few equations with a pen, and watch it deform, all within one window—an intelligent blackboard, working in conjunction with the mathematician."

University of California, San Francisco

The coming of desktop molecular modeling

The practice of chemistry inhabits a strange twilight world between the abstract and the real. The basic building blocks of matter are hidden from the chemist's eye; yet matter itself is tangible and visible to us all. Chemical knowledge derives from the chemist's preoccupation with building models in an attempt to visualize the invisible. An exciting modeling technology that utilizes sophisticated software running on computer graphics workstations heralds a new era of visualization for the chemist. Some of the potential of this new world of molecular modeling is demonstrated by a software application called MidasPlus, created by scientists and programmers at the School of Pharmacy at the University of California, San Francisco (UCSF), under the direction of Tom Ferrin and Robert Langridge.

"Chemists have always been better at understanding molecules when they can see a three-dimensional model, look at it from different angles, and even move parts of it around," observes Tom Ferrin, adjunct professor in the Department of Pharmaceutical Chemistry and director of computing at the Computer Graphics Laboratory. "There's some understanding process that happens when they can manipulate molecular structures."

"Physical models, made from colored balls and wire, work well for small structures," he continues, "but as the models grow larger and larger, they become fragile and unwieldy. So there's a lot of advantages to modeling molecules in a computer, doing all the manipulations on the graphics display. And, if you're going to transfer that process to the computer, interactive color graphics are critical."

Inside the MIDAS touch

The Molecular Interactive Display and Simulation (MIDAS) System is a collection of programs developed by the Computer Graphics Laboratory in the early 1980s. MIDAS originally ran on three-dimensional graphics display terminals attached to host computers. The primary component is MidasPlus, an interactive graphics program for the display and manipulation of large molecules such as proteins and nucleic acids like DNA. In 1989, MidasPlus was rewritten to work with new color graphics workstations.

MidasPlus starts with a profile of a molecule acquired from other software tools that perform computations based on information gleaned from crystallization experiments. MidasPlus can show a molecule as a stick-figure image, in which lines represent atomic bonds, and the intersection of lines, atoms; as a ribbon image, in which parts of the molecule are displayed as geometric shapes, such as helices and planes or sheets; and as a conic image, in which each atom is represented as a three-dimensional ball stuck together with other balls. In all of these views, the molecule can be oriented in any position.

The stick-figure imaging can be particularly rich in information. A part of the molecule can be selected for investigation, and magnified; labels can be displayed indicating the atom type and its electrostatic charge. The distance between atoms can be measured in angstroms, and dynamically altered with the mouse. Surface representations of the atoms can be illustrated with van der Waals surfaces.

Introducing MidasPlus to the NeXT environment

When MidasPlus was rewritten, the code was designed with portability in mind. A layer of three machine-dependent software modules was inserted between the computer and the main application program; porting became an exercise in creating these three modules for each specific platform. The bulk of MidasPlus—about 85 percent of the total source code—is written in the C programming language, and transferred to the NeXT environment with little or no change. Greg Couch, a programmer in the Computer Graphics Laboratory, learned Interface Builder and the Application Kit™ to develop the machine-dependent code.

"Greg completed the port over an eight-month period, and he had several other responsibilities at the time," Ferrin says. "We estimate that it took the equivalent of four months of full-time work—and that includes learning Interface Builder and the details of NeXTstep. I'm pleased with the results." Ferrin, Couch, and the other programmers in the Computer Graphic Laboratory have been so taken with the NeXT development tools that they have made NeXT computers their development platform of choice.

"I'm impressed with Interface Builder," Ferrin says, "and we've already developed several applications that are unrelated to MidasPlus. For example, we developed a program for monitoring our network—and each of its different subnetwork segments—that interfaces with a device called an EtherMeter. One of our programmers, Conrad Huang, developed it while we tracked how long he worked on it, and it's fair to say it took half the time to develop that application on a NeXT computer than it would have in an alternative development environment."

Desktop molecular modeling on every workbench

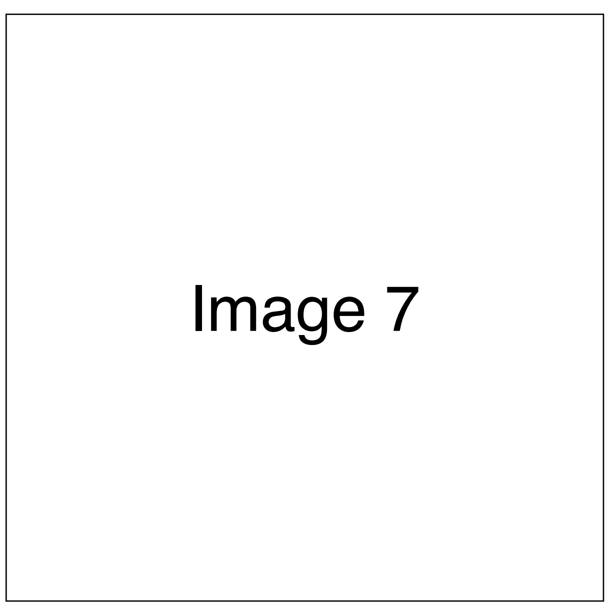
But it is the implementation of MidasPlus on the NeXT computer that has really inspired Ferrin, as he articulates a vision of affordable computer modeling for every chemist.

"Until now, if you wanted a 3D color graphics workstation, the least expensive and smallest computer you could buy would be one of the Silicon Graphics IRIS machines," he explains. "A reasonable configuration costs $20,000, and you can easily spend $40,000 if you add high-performance graphics, high-capacity disk drives, tape drives, and stereo viewing peripherals. You can't afford to put one of those on every chemist's workbench; yet clearly, the capabilities of computer modeling would be beneficial to virtually every chemist. NeXT computers provide the potential to put interactive molecular modeling on every chemist's bench. We like to call 'it Desktop Molecular modeling'."

The productivity benefits are obvious. Currently, even in a computer-rich environment like the UCSF Pharmaceutical Chemistry department, researchers must share IRIS workstations for modeling work. With a workstation on every chemist's bench, waiting for computer time would become a thing of the past. In addition, a multi-window workstation environment, such as that provided by NeXTstep, will let chemists access all of the tools of a multimedia working environment—word processing, database query, graphics, and now interactive molecular modeling. It's the technological capabilities available in higher education today that lend credence to Ferrin's vision of the future.

"I'm excited about NeXT's commitment to higher education," he says. "Tools like MidasPlus and the NeXT computer are becoming more pervasive in our department. As we train tomorrow's scientists to use these tools, they will radiate out to pharmaceutical companies and other research institutions and convince others that they could be more productive with these tools."

For more information, contact:
Professor Tom Ferrin
tef@cgl.ucsf.edu



In the MidasPlus window, stick-figure molecular models can be dragged and rotated in three dimensions. The molecule shown here is the electron transport protein flavodoxin. It is displayed with the ligand flavin mononucleotide enclosed by a van der Waals surface. (The coordinates are from the 3fxn file from the Brookhaven Data Bank at Brookhaven National Laboratory. Reference: F.C. Bernstein, et. al., Journal of Molecular Biology, vol. 112, pp. 535-542, 1977.)

University of Nebraska, Lincoln

Students solve real-world problems in *Mathematica* lab

The complexity of science and technology requires today's students to have mathematical knowledge that ranges from basic calculus, differential equations, and linear algebra, to more advanced topics. These students—as tomorrow's engineers, scientists, and researchers—will also be expected to take advantage of sophisticated computers and software tools in their investigations. The convergence of these needs—mathematical perspicacity and computer expertise—resulted in the development of a unique mathematics laboratory and curriculum at the University of Nebraska, Lincoln.

"The workstation is the desktop computer of the future and is a tool that many of our students will be required to use professionally," observes Tom Shores, professor of mathematics at the university. "We concluded that mathematics classes are the perfect environment to teach these mathematical and technological skills."

Shores, fascinated with computers since the personal computer revolution of the 1970s, reasoned from his own experience. With prize money from a teaching award, he bought a personal computer and began to experiment. He found the computer was as valuable a tool for word processing as for numerical computation and analysis. He used his computer to conduct mathematical investigations and to express the results of his investigations.

The dream of a multi-window workstation

Shores' vision called for the creation of a mathematics lab of computer hardware and software that could be used for both the study and expression of mathematics. Shores' colleague, associate professor of mathematics Steven Dunbar, explains, "Tom's vision was of a multi-window workstation environment, with each window running a different application." Users would explore mathematics in one window, then draft results in another.

With help from Dunbar and others, Shores submitted a proposal to the National Science Foundation (NSF) for a grant to fund a laboratory of workstations for the study of mathematics, particularly for the curriculum of advanced mathematics classes. As part of that effort, Shores developed criteria for the kind of computer he needed: a networked workstation featuring a 32-bit CPU, a math processor, a minimum of 4 megabytes of RAM, and the necessary mathematics and word processing software. A UNIX®-based operating system and color graphics were considered important options. With these criteria and pricing considerations in mind, Shores evaluated a variety of platforms, and his grant proposal eventually recommended NeXT systems.

"The NeXT platform is a completely integrated system that satis-fied all our requirements," Shores says. "But most important, NeXT is a UNIX computer that comes bundled with *Mathematica*," saving hundreds of dollars on each machine.

One reviewer of the proposal, Shores recalls, remarked that using NeXT computers "could be of service to the mathematics community." Shores himself had not used a NeXT computer, but had researched the technology extensively, and was firmly convinced of its, and *Mathematica*'s, capabilities. The NSF acted upon the reviewer's recommendation: a NeXT mathematics lab was "a strong idea and worth trying," and Shores received the grant. With matching funds and a lab facility from the university, Shores built a lab of NeXTcube™ computers.

The lab has been used for two classes: an introduction to differential equations, taught by Dunbar, and an introduction to linear algebra taught by Shores. Students in both classes attend a one-hour orientation session in which they are introduced to the NeXTstep user interface, WriteNow,® and *Mathematica*. Shores and Dunbar hired four students to help manage the laboratory; one unexpected but welcome bonus has been the administration of the NeXT computer network. "I'm pleased with the simplicity of set up and maintenance," says Shores.

One platform, two teaching approaches

These two faculty members use the lab differently. They see their approaches to computer use in two ways: the computer as a tutor and the computer as a professional tool.

Dunbar's use of the computer as a tutor consists of lecture lab demonstrations and homework projects. In a lecture lab demo, Dunbar uses a *Mathematica* Notebook to show his class how to work through a particular problem. The flexibility of the software, combined with feedback from students, encourages him to modify a lecture in the midst of giving it. In a recent exercise, for example, he used a set of equations involving heating and cooling to solve a murder in a mystery story, graphing the results in *Mathematica*. One student suggested a different approach and Dunbar was able to incorporate it on the spot.

Dunbar assigns mathematics projects as homework assignments. He designs a complex mathematical problem associated with a physical situation and then breaks the problem down into steps. Students solve the equations and graph the results in *Mathematica,* then prepare results and draw conclusions in WriteNow. One assignment had students calculate the heating and cooling of a large office building, taking into account the effects of the outside temperature and the building's internal heating.

"In our lab, we can assign students problems that, in difficulty and interest, are leagues ahead of what they could do by hand," comments Dunbar. "With much of the 'symbol-pushing' done by *Mathematica,* students have to focus on the intrinsic meaning of the mathematics they are doing."

Shores' use of the computer as a professional tool resembles what Dunbar does with his homework projects, but involves more real-world examples. Shores assigns problems that students have to investigate in the computer lab; for example, he recently asked students to decide among five different materials which would conduct heat most efficiently. It's a complex problem in linear algebra that must be solved five times, once for each material. He challenged his students to think of themselves as engineers at fictitious companies. They were required to define the problem, find the solution, and submit a report detailing their conclusions.

"We expect our students to express mathematical thoughts in clear sentences," Shores remarks. "We're training students how to work with computer systems and how to write reports, which closely resembles the work they'll do in their professions."

A future made in heaven

Both men are excited about the future of the mathematics laboratory.

"Besides increasing the number of computers, I want to expand the teaching materials we have available on our NeXT computers," says Dunbar. "For example, I'd like to have a comprehensive collection of Notebooks for differential equations, and teach a class without textbooks: just lectures, demos, and homework projects."

"Any post-calculus class is fair game to be taught with the lab; all we need to do is to get other instructors hooked on the machines," concludes Shores. "The interface is so straightforward that instructors can use the lab in their classes without being experts. I'd like all mathematics classes to include a lab component."

For more information, contact:
Professor Steven Dunbar
srd@mathcml.unl.edu

Mathematica Notebooks on NeXT computers

Mathematica Notebooks are structured electronic documents that can contain executable *Mathematica* code, nonexecutable text, graphics, sounds, and animations.

The basic unit within a Notebook is called a cell. Each cell holds one type of data—*Mathematica* code, text, graphics, or sounds. Cells can be grouped or nested hierarchically to create a structured, organized presentation of mathematical concepts. Text can be presented in any number of typefaces, sizes, and styles, making Notebooks easy to read and understand.

Notebooks are interactive. Cells containing *Mathematica* code can be edited and evaluated by the *Mathematica* kernel. Sounds can be played; graphics can be sized and copied into other documents; and graphics and sound cells can be combined to produce animations.

Because Notebooks seamlessly combine executable *Mathematica* code with graphics, sounds, animations, and well-formatted text, they have become the de facto format for *Mathematica*-based lecture notes, homework assignments, and exams.

For more information on how *Mathematica* Notebooks are used in teaching, see the article, "*Mathematica* brings news understanding to teaching and research," on page 2.

University of Iowa, Iowa City

Fluxbase: cataloging and experiencing non-traditional art with technology

For years, curators, librarians, and archivists have used computers to catalog and index information on works of art. This type of system operates like a basic card catalog file system: each artwork in the collection is represented by a record. Each record contains information fields such as the work's title and creator, when the work was created, its owner, and exhibit dates. This provides the minimum one needs to manage, exhibit, and interpret, for example, a contemporary portrait or a sculpture; but it is inadequate for managing a subset of contemporary art known as Fluxus and other works of art that are considered nontraditional.

The Fluxus movement attempted to break down the structure of traditional art as a submissive medium. Its primary goal was to make art more accessible—both its components and its interpretation. Frequently, everyday objects such as marble, or postcards were incorporated within Fluxus works.

To store and retrieve information on Fluxus works with conventional databases is difficult. Separating these works into standard artistic categories makes searches difficult because one has to guess in which category a librarian has placed the work. Also, because the titles of Fluxus works often have little to do with the composition or presentation of the works, searching for related works is difficult. Furthermore, many Fluxus artists are uncomfortable with the way their works have been categorized in the past and are unwilling to have them entered into traditional databases.

Unconventional wisdom

In April 1989, the National Endowment for the Arts sponsored a conference to address the inadequacies of using existing artwork information systems for Fluxus. The conference was organized by Estera Milman, founding director of The University of Iowa's Alternative Traditions in the Contemporary Arts, in collaboration with Franklin Furnace, NYC. Attending institutions included the Museum of Modern Art, NYC, and the Getty Center for the History of Art and the Humanities.

"Fluxus art was designed to incorporate, as well as mingle with, everyday life," says Milman. "You interpret aspects of Fluxus by playing with its objects—its event-kits." At the conference Milman used a work titled "Flux Year Box 2" to illustrate the event-kit concept. Crafted from a 10-inch hinged-lid square wooden box about 2 inches deep, more than 17 artists—among them Yoko Ono and George Macunias—collaborated to create "Flux Year Box 2" in 1969. Now, due to its appraised value, people cannot interact with "Flux Year Box 2" with their own hands, as the artists intended. This frustration was the inspiration behind the NeXT "Flux Year Box 2" simulation developed at

Image 8

The "Flux Year Box 2" simulation lets users manipulate objects from the original work, created in 1969. Among the objects are a postcard, a matchbox, an envelope, a puzzle, marbles, notes, and a film strip.

the University of Iowa.

Simulating reality: objects of art

“Our design model was simple,” says Joan Huntley, project leader of the NeXT “Flux Year Box 2” simulation. “We wanted to give people the feeling that they were actually working with the real Fluxus artwork—opening it and playing with the objects inside.”

“Joan’s goal of imitating real objects was a primary reason we developed the project on NeXT computers,” says Mike Partridge, creator of Fluxbase, “the underlying program. Fluxus works are generally collections of physical objects. Because the NeXTstep development environment is object-oriented,” says Partridge, “it’s inherently easier to use to simulate physical objects.”

To achieve a highly realistic effect, Huntley used a digital video camera to capture images of the actual “Flux Year Box 2.” Once all its elements were captured, Partridge converted the images to .tiff format, and displayed “them on a NeXT computer. Handling the digital images was easy,” says Partridge, “because the NeXT Application Kit contains image manipulation objects that treat each picture as a graphic object file.”

Partridge then built the virtual “Flux Year Box 2.” Using Interface Builder, he created the primary backgrounds—one of “Flux Year Box 2” closed, [in the screen shot at left], and another with its lid opened. Inside the real box are a number of small compartments, “or bins.” Contained within each bin is an individual artist’s object. To simulate the real box, Partridge placed each scanned image into its corresponding bin. He created a bin index to keep track of each object and its location. This index was especially handy during testing, enabling Partridge to quickly return each object to its original bin.

“After a couple of weeks, when I was familiar enough with NeXTstep, I added invisible buttons to each object,” says Partridge. When the user clicks an object’s “hot spot,” it activates an animated effect, such as opening a box of matches, or flipping over a playing card. “The basic, functioning prototype took about four weeks to create.”

Open and shut case

How does one experience the “Flux Year Box 2” exhibit? “The NeXT display is so large that ‘Flux Year Box 2’ and its contents are displayed at nearly actual size,” says Huntley. Among the objects are illustrated monogram cards, an envelope, a box of matches, and a film loop. Using the mouse, the user can drag an object from its bin, then explore it. The user can remove the box of matches, for example, and open it to reveal the underlying message. Or the user can open the envelope to remove the note it contains. By clicking the note’s corners, the user can unfold the note, read it, then refold it. According to Huntley, figuring out how to operate the objects is part of the experience and part of the fun. “The computer adds a new dimension to discovering and interpreting the artist’s intended message or effect.”

Today’s reality, tomorrow’s virtuality

Partridge is currently experimenting with NeXT’s Sound Kit™ Music Kit™ to take advantage of NeXT’s digital signal processor (DSP) to incorporate recorded music, and historical oral interviews with the artists.

“We’d like the artists to be able to explain their exhibits—or how their element interacts with another artist’s work,” says Huntley. To simulate the collaborative aspect of Fluxus works, Huntley wants to network a number of virtual simulations. “One example of a collaborative Fluxus artwork is an illustration that a group of artists mailed among themselves, each artist adding his or her own artistic component.”

The project team is readying a newer, richer version of Fluxbase for an upcoming exhibit in the spring of 1992 at the Franklin Furnace Museum in New York City. When the show opens, the team hopes to demonstrate their most exciting enhancement: a Mattel Power Glove interface. Wearing the Power Glove, the user interacts with the “Flux Year Box 2” by motioning at the NeXT display. “It’s like introducing an intermediate virtual reality into the Fluxus art world,” says Partridge. “We want people to come as close as possible to experiencing Fluxus artworks first hand.”

For more information, contact:
Joan Huntley
jhuntley@memex.weeg.uiowa.edu
Michael Partridge
partridg@umaxc.weeg.uiowa.edu

University of Washington, Seattle
Drawing students into research

The University of Washington in Seattle is a top contender for federal grants and contracts. Its academic and research programs are broad, attracting preeminent scholars and scientists, and a student body of nearly 35,000. In this atmosphere, drawing students into professional-level research is a priority.

NeXT technology is an on-campus research partner with a dual role: it is both a professional investigation tool and a teaching tool. Ron Johnson, head of computing and communications describes the way interdisciplinary programs at the university have taken advantage of NeXT computers: "We have been early adopters of TCP/IP-based protocols, a distributed computing environment, and deploying a large-scale network computing model. We have an environment that's the ideal incubator for using something like NeXT computers."

Mathematical modeling in the zoology department

"Biology students often have terrible allergies to mathematics," says Garret Odell, professor of zoology. A two-part course using *Mathematica* on NeXT computers is how he solves this problem. "I'm surprised at how quickly students are able to do productive work—after three or four hours of exploring." His first course, a weekly lecture with a three-hour formal lab, set to be offered every fall quarter, covers the fundamentals of mathematics, using *Mathematica* for all the calculations and formula manipulation.

Odell wants to equip zoology students with the tools of applied mathematics to make models of various biological phenomena. Projects include ladybug control of aphid infestations and molecular interaction within fruit-fly genes. Odell points out that "it takes fairly sophisticated mathematical models to express the way aphid populations grow with time, say in a field of peas, and the way ladybugs wander around, eat aphids, and modulate their wanderings according to the number of aphids. It takes parabolic partial differential equations. That is, it requires mathematics that are tougher than in chemical engineering. In a chemical reactor, for example, molecules bounce around randomly whereas bugs can wander at will in a field. You have to build in functions that express their responses to the surroundings."

Ecologists have been quick to grasp the importance of making a model to distill results into a mathematical forecast. "We had a student who worked out a model that predicted how the population of spotted owls will change as time goes on. This model has played a role in courtroom proceedings arguing that the owls need vast tracts of virgin forest, not the tiny pockets the logging industry would leave them."

Graduate students comprise two-thirds of the class. Odell sees "virtually all of them incorporating what they learn into their own research, so they're drawing pictures of their data, analyzing it, fitting it with various functions." He adds that they do this faster on NeXT computers than on other workstations. "I'm astonished at the quality of the interface, Interface Builder, and Digital Librarian.™ Next quarter we start the advanced models course, and we'll do more significant work than we could before because now we have students who can do all the mechanical things and understand the mathematics. It's gratifying to see that happen."

NDSP, ChaosPlot, and SndManager

NeXT computers do double duty in the School of Music. "Under the direction of Douglas Keefe, a physicist living in the School of Music and having a great time," NeXT computers support a cross-disciplinary investigation into acoustics, with applications for music, physics, and in the biomedical field.

Robert Ling has written several Objective-C language software applications, including NDSP, ChaosPlot, and SndManager. NDSP is a package of signal processing tools, including large-scale FFT (~256k), multi-waveform editing and viewing, short-time Fourier analysis displayed in 2D and 3D graphics, plus built-in sounds and peak detection. ChaosPlot employs a graphical user interface (GUI) to manipulate plots, while SndManager is a general utility program for sound files.

> *"These models can also be used to construct synthetic instruments that have no conventional acoustic counterpart."*

Keefe uses a suite of programs designed for experimental chaos dynamics analysis. Starting with multiphonics—unusual sounds that wind instruments produce—he records a sound file using a microphone. Then he makes a two-dimensional representation of the file to reconstruct the phase space of the dynamical system. "We can infer the complexity of the dynamics involved in producing those multiphonics, musical sounds in which multiple pitches are heard. They're used in contemporary music, in classical music and jazz, and extend the conventional range of the instrument. Dimensional analysis of the reconstructed phase space shows that some multiphonics possess a strange attractor. The same software is used to analyze acoustic emissions found in the ear. We've also found evidence for complex dynamics in these sounds produced within the ear. That tells us something about how the inner ear works."

Students from the music school and other disciplines use the NDSP program with the NeXT computers available in the School of Music computer center and in the research lab. "Students can record musical instrument sounds and use the software to study the acoustics," says Keefe. "They can do it interactively—play the instruments, record the sounds, and analyze it right there. This is an effective tool for instruction because it takes students away from the textbook, and they're not working with 'canned' sounds that someone else produced."

Current research deals with time-domain simulation of musical instruments using *Mathematica*. The aim is to model the physics of the instrument and the function of the player. The models are converted into sound files. "In this kind of modeling, everything has a direct physical significance. You can change the simulated trumpet mouthpiece on your model and play it again. If it's a correct simulation, it responds like the real instrument. These models can also be used to construct synthetic instruments that have no conventional acoustic counterpart."

Baby talk in electrical engineering

"Two-way training for speech recognition takes its inspiration from the way babies learn to speak," says Les Atlas about the project in automatic speech recognition he super-"vises in the Interactive Systems Design Lab. When babies pick up speech and hearing skills, what tends to happen is they react to what their mother says and make the mother say things in a certain way; it's a two-way process."

A second source of inspiration comes out of the computational learning theory that suggests unlearnable problems may become learnable when queries are used. According to Atlas, "What this means in speech recognition is that the performance limits of the usual train-by-example system could be improved by having the recognizer synthesize extra sounds unknown to it, and then get more labels from listeners." A speech synthesizer, written by Atlas' group, is used to conduct experiments. A NeXT computer was a natural because "the audio was integrated and the user interface facilitated programming work."

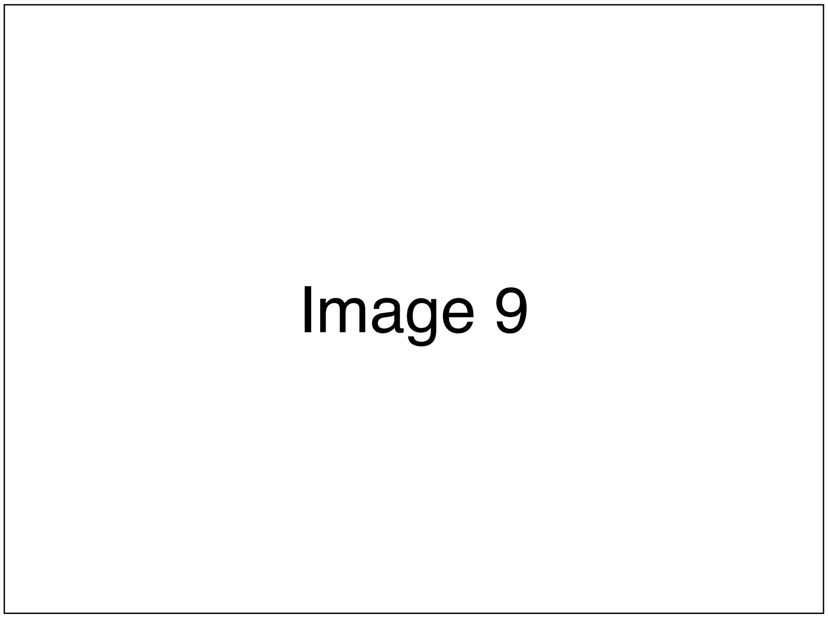
Atlas augments one-way training systems with two-way training to make them work better. "In one-way training, the recognizer gets good information, but what's missing is the idea of the recognizer itself being able to come back and ask for more words. In two-way training we'll partially train and then come back, synthesize some sounds, and label those sounds. Maybe they won't even sound like speech, but that's still information for the recognizer, for the system doing it. It will put a sound out through the speakers and we'll label it. For example, we're trying to make the computer recognize the differences among 'e,' 'b,' and 'c.' Then it will come back and make some new sounds, say "euie," and we'll label that. One possible label is 'none of the above.'"

Atlas sees two-way training using speech as an "acid test and a tough problem, but we're hoping the idea will have applications in image recognition and in various other pattern recognition applications. Passive learning in computers is restrictive, and this two-way learning is an example of an active process where the computer will query to get more information. The ability to do that could improve the performance of many pattern recognitions and other algorithms."

Neural network modeling in the primate center

Understanding the primate motor system has been a long-term project at the Regional Primate Research Center. Recent developments use NeXT computers to construct dynamic neural network models which simulate the neural mechanisms that generate patterns of activity in primate motor systems.

"The overall goal," says project manager Eberhard Fetz, is to understand how neural networks work. We do this by putting together networks that involve certain simplified neurons. These neurons act like nerve cells in the brain which are synaptically interconnected; the synaptic connection strengths can be changed in such a way as to perform a sensorimotor behavior. For example, in a step-tracking task, the monkey has to track a step change in target position by generating a step change in force. In the animal we've recorded activity of cells that generate the step change in force and we can derive networks that simulate his behavior. The advantage of modeling is it provides complete network solutions that perform the same behavior and specify the connections and activations of all the units."



The NextNet application lets researchers view the state of an entire neural network. Individual neural units can be stimulated and studied. Weights can be edited using spreedsheet-like weight editors.

Two programs were developed by Larry Shupe under Fetz. One is a command-line oriented program that runs neural network simulations in the background. It uses a variation of the back-propagation algorithm modified to handle time varying inputs and outputs instead of the usual spatial patterns, and allows recurrent as well as feed-forward connections.

"With the Genome Machine, the interface is intuitively obvious; it speaks the language of people involved in genetics and medicine."

The second program combines the first with a NeXTstep interface, allowing anyone to interact with a trained network and check how well it has learned its task by graphically viewing the current and past activities of the simulated neurons. A stimulation window lets the user impose an activity pattern on any unit and shows how this affects the activities of other units. Units can also be lesioned to test their function in the network. Connection strengths (weights) between units are displayed in a drawing window and can be edited in a spreadsheet window. Edited weights are then loaded back into the network to test the effects of weight changes.

Graduate students rotate through the lab and Fetz finds the NeXT computer "student-friendly. It's a great way for them to get a feel for the behaviors of neural systems without having to slog through the experimental details of animal recording."

Fetz says they are ready for color systems so they can "use the new features, such as video interactions, that are possible with the NeXTdimension™ board. We would like to import real-time data to the NeXT to train the network to emulate the monkey's behavior. In other words, the animal would be the system that the network has to simulate."

Genetics with the Genome Machine

"NeXT is the best platform for software development," asserts David Adler, project manager of the Genome Machine, a front end to multiple databases that is destined to become a genetic access and analysis tool of enormous value. The Genome Machine resides on a NeXT computer so it can take advantage of the multitasking environment, ease of application development, and Display PostScript imaging system.

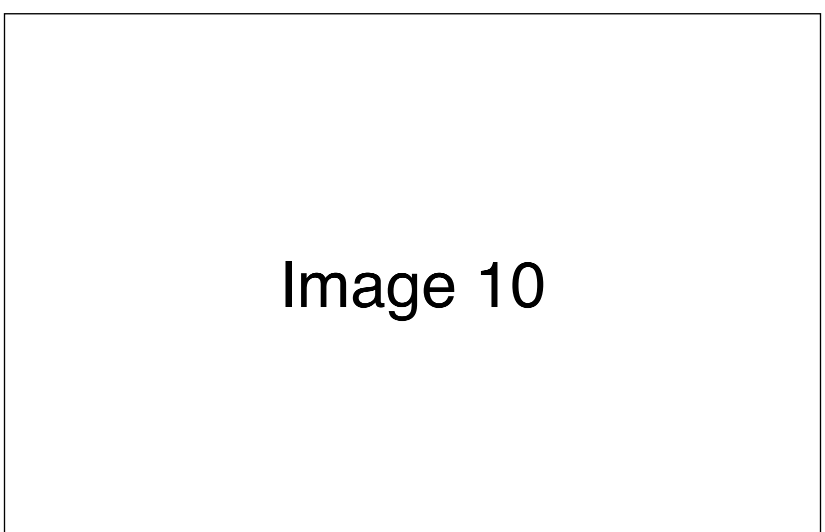
The Human Genome Project is an international research effort to read and record all the genetic information encoded on human DNA. "The problem," worries Adler, "is that there are five to ten genetic-type databases available, either over the network or on a dial-in basis. Each database has a different interface, and they're usually not user friendly. The idea I had was to have a single graphical user interface that speaks the language of the specialty of the person who wants the information—and lets them access all ten databases using the same front end. With the Genome Machine, the interface is intuitively obvious; it speaks the language of people involved in genetics and medicine."

The Genome Machine's graphical user interface uses chromosome schematics (idiograms) to access and display information. Pointing to a region on a chromosome idiogram displays genes and diseases localized on the selected region, scrolling linear physical and genetic maps and related sequence data. Graphics manipulation of chromosome schematics is accomplished with the idiogram Engine. Abnormal chromosome descriptions can be entered graphically or by using International Standard Cytogenetic Nomenclature. Queries are done by entering known information in a displayed field; all other fields and images are updated correspondingly. In addition to human DNA, idiograms and database access for several other organisms will be available, including the mouse and fruit fly.

Adler also sees the Genome Machine developing into educational software, possibly including real-time animations, targeted for introductory biology courses in high school and college, or introductory molecular biology or cell biology courses.

Open-door policy in the general-access NeXT lab

Students and faculty carry out both coursework and research in the university's NeXT general-access lab, established in 1989 under the direction of Sheryl Burgstahler, manager of Desktop Computing Services.



Modeling the NeXT lab on existing microcomputer labs, Burgstahler's staff created a lab with 10 NeXTcube computers, two servers, and two NeXT laser printers— accessible 24 hours a day, seven days a week. "We have an open-door policy," she says. "People come in on a space-available basis. There isn't a lot of bureaucracy in getting access to the machines." Anyone in the university can use the lab; no one has a private account unless they tap into a mainframe.

Easy access to NeXT resources has resulted in a lab that's heavily in demand. Users log on by using the "me" account—saving their work via file transfer to a large mainframe for permanent storage. When users log out, the file systems are cleared. Courses in biology, physics, and music share the lab. And since the lab is the site for four courses on NeXT technology, it attracts a large cross-section of university students, staff, and faculty.

Instituto Tecnologico y de Estudios Superiores de Monterrey

Interpersonal computing makes its mark

Mexico's Instituto Tecnologico y de Estudios Superiores de Monterrey (ITESM) was founded in 1943 by a group of Monterrey businessmen looking to provide the community with a source of high-quality technical education. Today, ITESM offers undergraduate and graduate courses at 26 campuses in 25 cities and serves 40,000 students from all over Mexico. When considering the school's numerous campuses and wealth of offerings—26 undergraduate and 30 graduate degree programs at its Monterrey campus alone—it is no surprise that ITESM graduates make up a high percentage of Mexico's business and professional leaders.

The first step toward NeXTstep

In 1988, ITESM became the first Latin American university to purchase NeXT computers. ITESM purchased nine NeXTcube computers for evaluation, exclusively to introduce faculty to the technology. David Trevino, computer services director, explains why ITESM chose NeXT: "We had been working with PCs and Macintosh® computers and found them good for word processing, but we needed a technology that went further. We saw workstation technology as having a major impact on both the academic and nonacademic worlds, and we evaluated NeXT technology as the best workstation offering in terms of both hardware and software quality."

Enthusiastic initial response to NeXT technology led the school to order 50 NeXTstation computers. Of these, 31 were set up in a student lab, networked to four servers. The others were distributed to the mathematics, administrative sciences, business, economics, and architecture departments for faculty development.

The school's early emphasis on faculty development is based on the belief that faculty involvement fosters widespread use of the technology. Trevino says, "We think it's important to give the faculty high involvement with and understanding of the technology, if students are truly to benefit from that technology."

Interpersonal Computing and NeXTmail bring remote campuses closer

The NeXT computers' electronic mail, networking, and multimedia capabilities were of particular interest to ITESM. Trevino explains, "We are highly distributed, with campuses all over Mexico. We needed electronic communications that could handle information in a natural way. With the multimedia mail application, we feel that NeXT computers can help us improve communications between our remote campuses."

ITESM already produces and offers courses to remote sites via a satellite TV network. One goal for the NeXT systems is to facilitate the interaction with remote students, for example, homework assignments can be delivered and returned using NeXTmail.™ Some faculty are also interested in developing computer-based lessons. These lessons can also be delivered via NeXTmail to remote locations across the country.

Interpretative Structural Modeling on NeXT

ITESM instructors are already using the NeXT computer as a development platform for Interpretative Structural Modeling (ISM). ISM is a decision-making methodology that was developed in the early 1970s as part of a larger decision-making approach called Interactive Management. Trevino describes: ISM is "a technique that deals with group decision-making dynamics by balancing anthropological, logical, and technological issues. It's a computer-based tool that helps groups interact and organize ideas. With the approach we're taking, one computer is used to aid an entire group."

Trevino continues, "ISM is not new, it's been around for nearly 20 years and has been used by groups, including the United States Department of Defense. Previous versions of the computer-based tools had user interfaces that were difficult to use. The NeXT user interface makes the ISM technique much more accessible. Also, the potential of the power of a NeXT computer is important. For example, ISM sessions can produce 100 to 200 ideas. ISM requires a great deal of computing power to help people make inferences and then produce results."

"We are highly distributed, with campuses all over Mexico. We needed electronic communications that could handle information in a natural way."

At the Monterrey campus, a new Center for Interactive Planning and Design is being set up, which will feature a NeXT computer and NeXT-based ISM software. This facility will be used to conduct workgroups for systems design students within the school's systems engineering program.

Taking advantage of multimedia

NeXT computers' multimedia capabilities are having an impact in the architecture department. Department faculty members were intrigued by the potential of the NeXTstation as a multimedia platform and began their work by developing a prototype lesson with MediaStation.™ Called Renaissance Castles in France, the prototype was so successful that three instructors are now working on a much larger multimedia database for a History of Architecture course. Trevino explains, "We are pushing computers for architecture and the humanities. We needed a platform that could integrate images and sound to make difficult concepts less intimidating to students."

"We also plan to make these multimedia materials available on 15 NeXT computers in the library. In class, the instructor will use a NeXT and a projector to present materials and students will go to the library after class to review and work with the same materials."

The computer of choice for computing instruction

The NeXT computers are also making a mark in ITESM's technical courses. The 31-system lab is being used for the Operating Systems and Compiler Design and Implementation courses, which serve a total of 250 students. Trevino explains, "We have two departments, basic computing and computer science. The basic computing courses are large, so our goal is to have a machine for every 10 students."

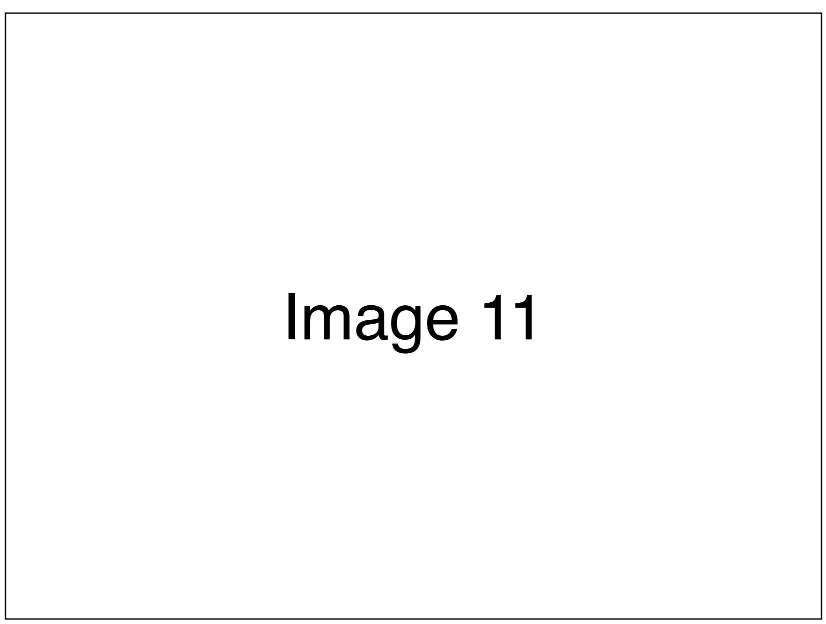
A course on advanced programming tools offered by the computer science department is also taking advantage of the NeXT lab. This class is smaller—20 students—and students are using Interface Builder to develop basic applications and to work with object-oriented environments and windowing systems.

Interpersonal computing at ITESM

Though there is considerable excitement at ITESM about many of the tools available for NeXT computers—Interface Builder, MediaStation, and *Mathematica* in particular—according to Trevino, what really sold the school on NeXT technology was its interpersonal computing capabilities. He says, "We have thousands of personal computers on campus, about 350 of them networked. We felt we needed to move toward a computer technology that has networking and distributed computing features already bundled in the machine. We think Mr. Jobs' interpersonal computing concept is what's needed in the '90s and beyond."

For more information, contact:
David Trevino
david@mtecv2.mty.itesm.mx

Architecture faculty have developed a multimedia database to supplement teaching materials for a History of Architecture course. The database presents images and historical background on Renaissance Castles in France.



The lure of very large numbers: Further adventures in community supercomputing

by Richard E. Crandall
Director, Scientific Computation Group
NeXT Computer, Inc.

A supercomputer is a computer that is only one generation behind the requirements of the users.

—J. Worlton to Neil Lincoln, *Supercomputing Review,* October 1989

The community supercomputer I described in *NeXT on Campus* Summer 1990, is now known as “Zilla” and, as promised implicitly in that article, is now a demonstration application with some striking features and economies. Zilla’s refinements are due to the labors of Josh Doenias of the Scientific Computation Group, aid from the Operating Systems Group, and community spirit and feedback from the many individuals who allow their machines to join the supercomputing runs.

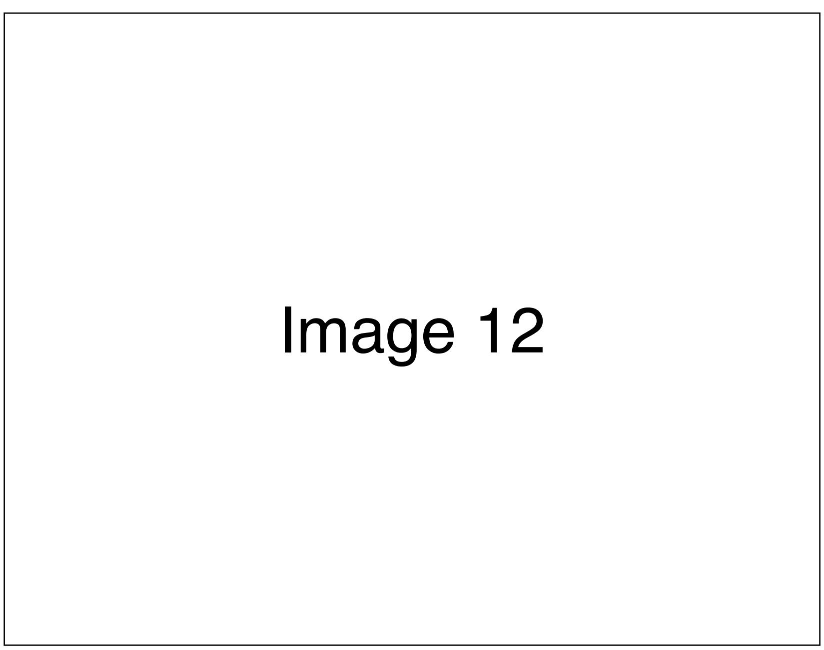
I like to think Zilla represents an expandable version of the kind of machine to which Worlton refers in the quote above. If a Zilla user (user of the application per se) has enough volunteered computers, and a sufficiently parallelizable computation with moderate transverse throughput, that user may be able to pull, say, possibly to within *one-half* a generation behind the ultimate requirements.

Another perspective: Though we have obtained genuinely new numeric results from Zilla, we have not yet tapped the available power. It appears possible to advance the state of the art, in certain numeric domains, by properly harnessing Zilla power. We have machines in Massachusetts appended to the machines in California. The east coast machines are in their own application window (see figure below). In real time, the Massachusetts machines send the same kind of data back to California as are generated by California machines themselves. In fact, all data from all parts of the country will usually be deposited in a common directory. There’s no telling how far these Internet schemes can be taken. It is possible even to add other types of machines, like Crays, to Zilla configurations. You may launch separate Zilla applications, each having its own collection of windows, each window replete with volunteered machines.

Problems involving large numbers

To date, problems involving large numbers have been a primary test of Zilla power, partly because one can be rigorous about whether a result is known, difficult, or correct. Because of the purity, enormity, and infi-nitely frozen aspect of very large numbers, they represent a collective “lure” for certain mentalities, and I for one have not been able to escape this pull. I should also add that we vary Zilla’s diet. With David Springer of the Graphics Group, we have successfully “Zillafied” Pixar’s RenderMan, by generating a color animated movie, one frame per volunteered machine every few hours. Also, we continue to explore music, business, database, and general science problems to feed to Zilla.

For very large numbers we take advantage of some outstanding fea-



The Zilla application acting something like a Cray supercomputer. The application is running in California, juggling about 100 machines, working on a factoring problem. Note the five Boston machines (upper right). A lock icon means the machine’s user does not bestow permission. A white bulb on a gray screen means the Zilla process is running; a dark bulb on a white screen means the Zilla process has politely backed away because a human is using the machine and Zilla has suspended itself. The four hidden windows (lower left) are more collections of machines running/pending, some collections containing yet dozens of more machines.

tures of NeXT computers: two megaflop floating point (68040 speed), massive virtual memory (effectively several tens of megabytes, sometimes >100 megabytes), and efficient file/communication facilities. The idea is to have computers generate answers to tough numeric questions and feed these answers to destination files, usually in one "research directory."

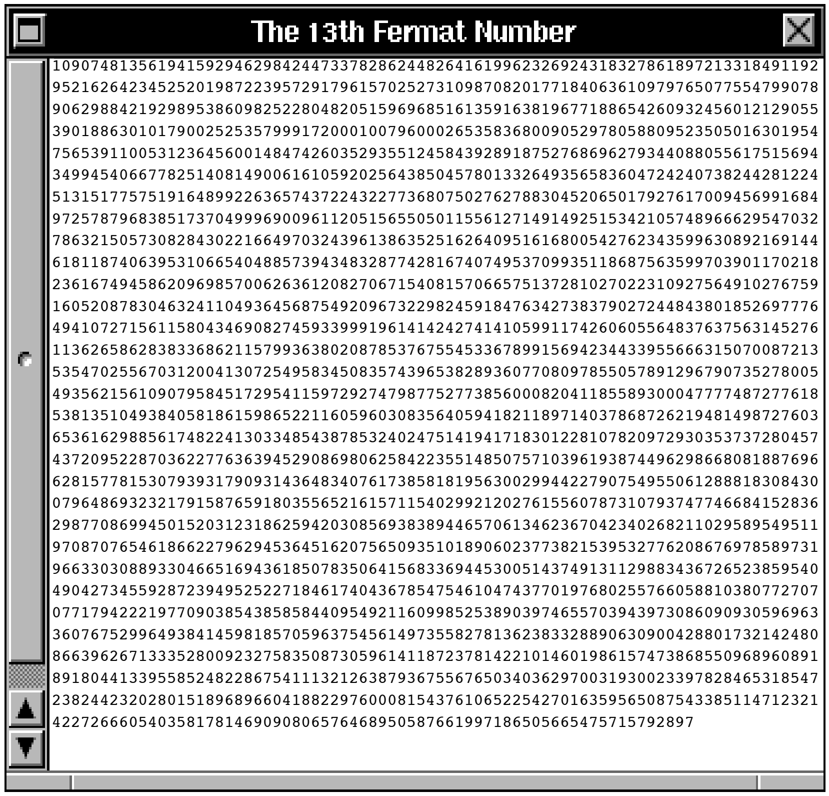
Some problems we solved

In *NeXT on Campus* Summer 1990, I mentioned that Zilla had settled Pierre Fermat's "Last Theorem" for exponents up to one million. Since then we have done more calculations, determining for example the (we believe) highest known settled exponent; that is,

$$x^{8388019} + y^{8388019} = z^{8388019}$$

has no solutions in positive integers (x,y,z). It appears now that about 1 ZU-month would be required to settle all exponents up to 2^{22}, or about 4 million. ("ZU" is a Zilla Unit, or 100 NeXTstation equivalents.)

On 5 January 1991, Zilla discovered a previously unknown factor of F_{13}, the thirteenth Fermat Number. This monster is $2^{2^{13}} + 1$, shown above to convey an idea of scale.



Zilla found a new factor of this number (see text).

The factor of F_{13} discovered was

```
2663848877152141313
```

This 19-digit factor is not large by today's standards, except that F_{13} is so large that a 19-digit factor might be as hard to find as, say, a 30-digit factor of a much smaller number.

For F_{13} we used the Elliptic Curve Method (ECM), which is a parallel algorithm par excellence. Each machine had to have a copy of F_{13} and various related numbers, which is yet another example of how virtual memory capability comes into play. These algorithms were used to factor a number from Richard Brent of the Australian National University.

$13^{101} + 1 =$

```
32231356442557642935643947501553246771104068082374234947558458292280063097893682282057713618219768342498230722014
```

which was split completely into:

```
2 * 7 * 3327037444864439 *
7425107270430419 *
932061553127902722185 3 *
145608610441138474973193809 51 *
686686445425016030757228727087
```

The final, 30-digit factor was, as explained before, roughly as hard to find as was the new factor of F_{13}.

Running Zilla

When you construct and run a Zilla factoring configuration, one important "lure" of large numbers will speak to you: You can always find numbers on which supercomputers will grind helplessly. Thus, very large number arithmetic can be thought of as the basis of a test suite that will scale upward "forever." As computers get faster, or something like Zilla gets more expansive, one may still use number theoretical algorithms as arbitrarily harsh tests of new configurations.

The numerical results should not overshadow the single most satisfying aspect of Zilla. I sometimes call it the "resource recycling" aspect: Zilla can do what computers were originally invented to do—things humans cannot do—such as working all night, every night.

Creating custom front ends to *Mathematica*

by Jayson L. Adams
Strategic Developer Engineer
NeXT Computer, Inc.

On the NeXT platform, *Mathematica* consists of a kernel, the program that executes *Mathematica* language constructs, and a front end, a graphical user interface to the kernel. Any application can act as a front end to the *Mathematica* kernel. For example, Gourmet—a supercalculator created by Richard Crandall of NeXT's Scientific Computation Group—employs the kernel as the primary computational engine. The combination of NeXTstep's Interface Builder and a variety of methods for communicating with the kernel (MathPalette objects, Speaker/Listener, the Subprocess object, and MathLink) makes building front ends easier on NeXT machines than on any other platform.

Interface Builder

Building a custom front end begins with constructing a user interface. NeXTstep's Interface Builder application simplifies this task: Interface Builder lets you assemble an interface graphically, rather than writing hundreds of lines of code to create user interface objects. Interface Builder supplies several palettes of objects such as windows, buttons, and text fields, which you drag into your application and modify as needed (size, font, etc.). You can test the interface without leaving Interface Builder and without writing any code. Interface Builder makes prototyping and refining user interfaces easy, and gives programmers more time to concentrate on the engine that drives their application.

MathPalette

Of all the options for connecting to *Mathematica,* Objective Technologies, Inc. provides the simplest solution with a product called MathPalette.™ MathPalette consists of a palette containing the OTProcessor, OTAssignField and OTMathView objects. You can drag these objects into your application just as you would buttons or text fields or any other palette object. (You can also access these objects directly from Objective-C language code.)

OTProcessors pass instructions from your application to the *Mathematica* kernel and process the results. OTAssignFields let users set the values of *Mathematica* variables. Finally, the OTMathView displays *Mathematica*-generated Encapsulated PostScript® images (prepending the PostScript definitions supplied by *Mathematica's* psfix utility).

Speaker/Listener

Speaker and Listener objects, supplied by NeXTstep's Application Kit, let applications send messages to one another. Speaker objects, as their name implies, send messages and Listener objects receive them. Every application has at least one Speaker and one Listener, and as long as you know the messages an application understands, you can communicate with it.

To have your application send a message to *Mathematica,* you must first locate your application's speaker:

```
speaker = [NXApp appSpeaker];
```

Then your application's speaker must connect to *Mathematica:*

```
[speaker setSendPort:
   NXPortFromName("Mathematica",
                  NULL)];
```

Now your application's speaker can send messages to *Mathematica:*

```
[speaker msgProcessString:"4+5\n"
         into:&mathematicaOutput
         processType:0
         ok:&flag];
```

This message instructs *Mathematica* to feed the character string "4 + 5\n" to a kernel and set `outString` to point to the result, in this case a string similar to "Out[1] = 9." ("\n" means new-line, equivalent to pressing the Enter key within a Notebook.) *Mathematica* ignores the `processType` parameter—your application should pass zero as its value.

Note that while all speakers understand a certain set of messages, the `msgProcessString:into:processType:ok:` message is not one of them. To communicate with *Mathematica,* your application needs a custom speaker created by the msgwrap utility. Also note that if you use Speaker/Listener you communicate with the *Mathematica* front end—not the kernel itself—and you incur the overhead of invoking the front end (generating a default notebook window as a side effect). In exchange, you get a simple way to send commands to *Mathematica.*

UNIX pipes and the Subprocess object

UNIX pipes are a standard form of interprocess communication allowing any two applications to exchange information. Forming a pipe connection to *Mathematica* requires code more complicated than its Speaker/Listener equivalent. Luckily, the "Subprocess object" (located in the /NextDeveloper/Examples/Subprocess folder) eliminates this complexity by providing an object-oriented interface to UNIX pipes.

To employ Subprocess object communication, your application must create a Subprocess object that talks to the *Mathematica* kernel:

```
mathProcess = [[Subprocess alloc]
               init:"/NextApps/Mathematica.app/Kernel/math"];
```

Next, your application tells the Subprocess object which object should receive the kernel's output:

```
[mathProcess setDelegate:
                ouputReceiver];
```

At some point you send a character string to the kernel:

```
[mathProcess send:"4+5\n"];
```

`mathProcess` messages `outputReceiver` when the kernel responds with its character string result. This design lets your application continue to process button presses and other events while the kernel performs its calculations, unlike Speaker/Listener which forces your application to wait until *Mathematica* replies. The Subprocess object talks directly to the kernel, avoiding the front end, and it lets you invoke multiple *Mathematica* kernels vs. Speaker/Listener's single kernel.

MathLink

Both the Speaker/Listener and UNIX pipes techniques require your application to interpret the kernel's output, a potentially tedious and error prone process. If you used *Mathematica* to factor integers, for example, you might send a command such as:

```
FactorInteger[123456]
```

and receive the following kernel output:

```
Out[29] = {{2, 6}, {3, 1},
          {643, 1}}
```

To use this data, your application would have to parse the output characters into integers—not impossible but not simple either. You also have no reliable method of detecting errors the *Mathematica* kernel may encounter while executing your commands. MathLink attempts to solve these problems by enabling applications and *Mathematica* to exchange expressions rather than character strings. For example, the above transaction using MathLink would resemble the following:

```
MLPutFunction(p, "FactorInteger",
              1);
MLPutInteger(p, 123456);
```

`p` represents a connection to the *Mathematica* kernel. The first line sends the function FactorInteger to *Mathematica* and tells it to expect one argument. The second line sends that argument. Since you know *Mathematica* will return a list in this case, you ask for it:

```
if (!MLGetList(p, &length)) {
    determineError(p);
}
```

`length` tells you how many elements comprise the list. You can use this function again to retrieve each sublist, and use the `MLGetInteger` function to retrieve each sublist's elements. The function's return value indicates whether or not an error has occurred. If an error has occurred, you can write a function—called here `determineError()`—that queries p for the error condition. For a more thorough discussion of MathLink, refer to The MathLink Communication Standard manual. MathLink appears in the 2.0 version of the *Mathematica* kernel.

Conclusion

When designing your front end, you must select the *Mathematica* communication option that best fits your needs. The wide range of choices and their relative ease-of-use, combined with Interface Builder, allow *Mathematica* front end development requiring less effort than must be expended on any other platform. NeXTstep empowers developers to use *Mathematica* as their application's computational engine, which should result in innovative products.

For further reading

"Adventures in supercalculator design," Richard E. Crandall, *NeXT on Campus* Fall 1990

Mathematica, A System for Doing Mathematics by Computer, Second Edition, Stephen Wolfram, Addison-Wesley Publishing Company

Mathematica for the Sciences, Richard E. Crandall, Addison-Wesley Publishing Company

The Mathematica Journal, Addison-Wesley Publishing Company

"The MathLink Communication Standard," Arkady Borkovsky, Wolfram Research, Inc.

"The NeXTstep Advantage: Application Development with NeXTstep," NeXT Computer, Inc.

"NeXT System Reference Manual: Concepts & Reference," NeXT Computer, Inc.

Upgrading to Software Release 2 and the 68040 Upgrade Board

Eric A. Larson
Technical Support Engineer
NeXT Computer, Inc.

NeXT has released two upgrade products that provide original NeXTcube owners with dramatically improved performance and functionality: Software Release 2 and the NeXT 68040 Upgrade Boards. Users who want only the improved performance and new features of Release 2 may purchase the software separately. NeXT offers Release 2 on a license-only basis. This means that a Release 1.0 user may upgrade an additional computer by using the Release 2 upgrade optical disk (from a friend or colleague), thereby paying only for the license and not the optical disk.

Ordering upgrades

To order software and hardware upgrades, contact your campus NeXT reseller. The reseller will provide you with the necessary pricing and technical information. If you don't know who your reseller is, call 1-800-848-NeXT or (415) 424-8500 to find out. You may also call these numbers to ask questions about upgrading or to request literature on other NeXT products.

SOFTWARE RELEASE 2

What's included?

There are two configurations of the NeXT system software: Software Release 2 and Software Release 2 Extended. Software Release 2 is pre-installed on NeXT computers equipped with 105 MB and 200 MB hard disk drives. Release 2 includes end-user applications. Software Release 2 Extended is pre-installed on NeXT computers equipped with 400 MB, 660 MB, and 1.4 GB hard disk drives. It includes a complete set of developer tools, all software in Release 2, plus additional applications. The Extended configuration is included in all Release 2 upgrades and is shipped on an optical disk. The following manuals are also included: *Upgrading to Release 2, Setup and Tutorials, Release Notes, NeXT User's Reference, NeXT Applications,* and *Network and System Administration*. The software upgrade also includes a pair of keyboard tilt feet and a software redemption coupon (see below). See sidebar at right for what's included with the two releases.

Upgrading your system software

There are two ways to upgrade from Release 1.0 to Release 2. NeXT provides an application called Upgrade2.0 that allows you to preserve existing files and configurations, such as your NetInfo™ database files. You may also rebuild the disk from scratch, using the BuildDisk application in /NextAdmin. BuildDisk destroys all files on the disk so you must back up the files you want to keep.

The Upgrade2.0 application lets you preserve *Mathematica* and Allegro® CLCommon LISP. The 1.0 versions of these applications both run under Release 2 but will not be included on the upgrade optical disk (see below). The 1.0 version of Allegro CL 1.0 runs only on the 68030 logic board (it will not run on the 68040 board) and requires software patches to run properly. Please contact your authorized NeXT support center for information about patching Allegro CL 1.0.

As part of the upgrade, all owners of 68040-based NeXT computers are entitled to receive *Mathematica* 2.0 from Wolfram Research Inc., SYBASE® SQL Server™ from Sybase, Inc., and Allegro CL from Franz Inc., all previously bundled with Release 1.0. Updated versions for Release 2 were not complete at the time of this writing, so you will not find them on your Release 2 disk. A redemption coupon is included so you can obtain these applications free of charge when they are available. The applications will be shipped on 2.88 MB floppy disks. This offer applies only if your software redemption coupon is postmarked on or before December 31, 1991.

Release 2.1 Update Set

The current version of Release 2 is Release 2.1. The primary objective of this release is to support NeXT's color products. Users of NeXT monochrome computers may also benefit from running the new release due to several bug fixes. The Release 2.1 Update Set is available only on 2.88 MB floppy disks for users of either Release 2.0 and Release 2.0 Extended. Since no licensing fees are associated with the software, it may be copied by an unlimited number of users.

Release 1 and Release 2 NetInfo Compatibility

Release 1 and Release 2 NetInfo databases are functionally compatible. On a network, you can mix Release 1 and Release 2 NetInfo servers. Release 1 NetInfo data files are upwardly compatible to Release 2. You can move NetInfo data files from a Release 1 to a Release 2 system, but not in the other direction.

Expanded Programming Language Support

The C-language compilers included in Software Release 2 Extended support ANSI-C, Objective-C and C++ languages. The NeXT Application Kit continues to be Objective-C based: Interface Builder still generates template code in Objective-C, and palette objects are Objective-C objects. However, with Release 2, you can combine Interface Builder nib files, C++ and Objective-C language code in one program.

C++ support in Release 2 is provided by the GNU G++ compiler which has been extended to recognize Objective-C language constructs. The compiler is based on G++ version 1.36.4, which implements version 2.0 of the C++ language, as specified by AT&T. The "NeXT C++ compiler produces native code" (or machine code) rather than C code.

The C++ compiler consists of the compiler driver (/bin/cc++), the compiler proper (/lib/cc1++), a post-linker (/lib/collect), a symbol demangler (/bin/g++-filt), an enhanced symbol table tool (/bin/nm++), and an enhanced profiler (/usr/ucb/gprof++). In addition, NeXT's gdb debugger provides support for debugging C++.

Support for C++ is intended to give developers with a significant investment in C++ object classes a migration path to the NeXT development environment. NeXT recommends using the Objective-C language for applications you develop from scratch.

HARDWARE UPGRADE
What's included?

The 68040 Upgrade Board for the NeXTcube comes without RAM. You transfer the RAM from your 68030 board to the 68040 board. A RAM removal tool and antistatic strap are provided for this. You also receive the *68040 Processor Upgrade Board* manual and Release 2 Extended. If you do not want to do the upgrade yourself, a coupon is provided that you can present to your authorized NeXT service center. The service center will perform the upgrade for you for no charge.

New input/output ports on the 68040 upgrade board

The 68040 hardware upgrade board has different external SCSI connectors and serial port pinouts than the 68030 logic board. The external SCSI port on the 68040 board uses an SCSI-2 type connector. To connect an external SCSI-1 device to the 68040 board you need an SCSI-1 to SCSI-2 adapter cable, which is available from NeXT. For pinout information about the serial ports, search on "zs" in Digital Librarian under the UNIX Manual Pages target.

Disk support in Software Release 2

Release 2 supports floppy drives that read both UNIX and MS-DOS® formatted floppy disks. If you own the original NeXT computer with no floppy, NeXT recommends you purchase an SCSI floppy drive from a third-party vendor. (See "Third-Party Products," on pages 32-33.)

Release 2 also supports most third-party SCSI hard drives. As long as the drive meets a few basic specifications, it will be recognized by the Workspace Manager.™ (For more information, see NeXTanswer hardware.586.)

Software Release 2 contains:

End User

Workspace Manager™
Edit
Digital Librarian
NeXTmail™
Preferences
Preview for PostScript®
PrintManager
Webster's Ninth New Collegiate Dictionary®
Webster's Collegiate® Thesaurus
WriteNow® Word Processor
DataViz/Bridge™ (DataViz)

Developer Tools

VT100™ Terminal Emulator

System Administration Applications

MailManager
NetInfoManager
NetManager
PrinterTester
UserManager
Installer

Software Release 2 Extended contains all of Software Release 2 plus:

End User

Oxford® Dictionary of Quotations
William Shakespeare: The Complete Works
TEX™ Document Processing System (Radical Eye Software)

Developer Tools

Interface Builder™
The NeXT Compiler for the Objective-C Language
C++ Language Compiler
Objective-C Language Class Definitions
56001 DSP Tools
GNU Emacs
GNU Debugger
BUG-56™ Debugger (Ariel)
Malloc Debugger
AppInspector™
PostScript Tools
Application Kit™
Music Kit™
Sound Kit™
On-line Technical Documentation

Academic Projects

The purpose of this section is to facilitate communication among NeXT users in higher education. Many of the projects are under development. If a project's status is not listed, please contact the project manager for more information.

This is a partial list of projects. Due to space constraints, projects listed in previous issues of *NeXT on Campus* do not appear here.

Business

Decision Support System (DSS)
Researchers use neural networks to apply problem optimization to real-world situations. A prototype marketing simulation is under development. Real-world information from on-line services, real-time conferencing, multimedia communication, and neural net-based optical character recognition will be combined with a network of computers to create the simulation. *Status: under development*.
Kwangbo Shim
Researcher
Ecole des Mines, St-Etienne, France
shim@grasp2.univ-lyon1.fr

Communication

Television Station Scheduler
An application for creating and editing public television programming schedules. It displays information about special programming, nationwide programming, satellite feeds, and local programming with an intuitive graphical interface. It also incorporates updates fed via e-mail from PBS headquarters in Washington, DC.
Fouad Habboub
Assistant, Operations TV/Computer Services Manager, WBGU TV
Bowling Green State University
fhabbou@andy.bgsu.edu

Computer Science

Computer Engineering Design
A senior-level course. Students are taught object-oriented programing and apply their knowledge to create applications for electrical engineering courses. Applications developed by students include packages for discrete simulation and antenna analysis. *Status: ongoing*.
John R. Glover
Director of Engineering Computing
University of Houston
(713) 749-1820
glover@uh.edu

NeXT Boxer A next generation Logo designed primarily for use by late primary and middle school children. Boxer's interface consists of boxes within boxes. A box may contain text, graphics, programs, and other boxes. Primitives can be extended to include animation, audio and video. *Status: being coded*.
Jennifer Woodward
Graduate Student in Educational Technology
San Francisco State University
jenniwoo@sutro.sfsu.edu

Data Storage and Retrieval

ICPSR Guide to Resources and Services The Inter-university Consortium for Political and Social Research (ICPSR) is a data archive and disseminating organization that produces a guide to more than 2,500 datasets used in research and instruction. Abstracts from the guide are accessed with Digital Librarian for rapid search and retrieval. *Status: ongoing*.
Ann Janda
Manager of Information Services
ann_janda@nwu.edu
Bill Parod
Programmer/Analyst
bill_parod@nwu.edu
Northwestern University

Marbles A complex relational database organizing and retrieval mechanism. Documents are organized into hierarchical clusters of 3D spheres called marbles. When the system is queried with a natural language statement, the brightness of the marbles is adjusted so that the clusters most strongly selected by the query are the brightest. Double-clicking on a marble reveals another cluster of documents. Double-clicking on a document opens it.
Status: being coded.
Scott Deerwester
University of Chicago
scott@sage.uchicago.edu

Southwestern Ontario Heritage Landscape Inventory A multimedia system to categorize and document information on heritage landscapes. Photographic images are faxed directly from the heritage sites to the inventory system.
Nancy Pollock-Ellwand
Assistant Professor of Landscape Architecture
University of Guelph
(519) 824-4120 ext. 6577

The Writing Library A database of writing-related questions and solutions. Students enter questions about writing style, grammar, and mechanics into a browser. Other students scan the questions and suggest solutions. Questions and solutions are organized into easy-to-search categories. *Status: ongoing*.
Joel Smith
Directory of Educational Computing
Allegheny College
js01@music.alleg.edu

Economics

Economic Instruction A series of *Mathematica* Notebooks illustrating the principles involved in applying mathematical methods to economic problems. The Notebooks accompany an economics textbook.
Status: prototyped.
Hal Varian
Professor of Economics
University of Michigan
hal_varian@um.cc.umich.edu

Engineering

Aero 485: Nonlinear Dynamics
An aerospace engineering course in chaotic dynamical systems. Students generate phase plane portraits, bifurcation diagrams, Poincare maps, and fractal images of iterated function systems and determine Lyapunov Exponents, calculate fractal dimension, and generate Poincare Maps while studying nonlinear dynamical systems.

Rotational Dynamics A set of

Mathematica Notebooks and packages for computing rotational dynamics. Routines include Euler angles, direction cosines, Euler parameters and inverse transformation matrices. The work is based on *Optimal Spacecraft Rotational Maneuvers*, by John L. Junkins and James D. Turner, Elsevier, 1986.
Status: under development.

Dynamics Toolbox An application created with Interface Builder that incorporates *Mathematica* and C routines to generate Poincare Plots, determine Lyapunov exponents, fractal dimension, and Fourier and wavelet transforms. It will include several integrators, and 2D and 3D plotters. The toolbox will be used as a research tool for investigating non-linear dynamical systems.
Status: under development.
John L. Junkins
Professor of Aerospace Engineering
Texas A&M University
junkins@atlantis.tamu.edu
William L. Kealey
System Administrator
Department of Aerospace Engineering
kealey@intrepid.tamu.edu

Construction Engineering A suite of applications that combine project scheduling, estimating, accounting, and expert systems.
Status: being coded.
Thomas Froese
Stanford University
froese@cive.stanford.edu

English

English 100 A course designed to teach college writing skills to freshmen. Instructors developed ten lessons for the class; each lesson is a NeXT application that leads students through an explanation of a particular issue (i.e. plagiarism) or skill (i.e. providing proper support for personal opinions). Students use a built-in notebook feature to answer questions and complete assignments. *Status: ongoing*.
Susan Smith
Assistant Professor of English
Allegheny College
Meadville, PA 16335

Human-Machine Interface

The Computer as Social Actor
An experiment to learn how voice affects human-machine interaction. A computer tutors human subjects by providing information and administering tests. Subjects may protest their scores to the computer, claiming the computer provided misleading information. During different runs of the experiment, human sounding voices, computer sounding voices, or no voices at all will be used for the tutoring and protest sessions. Subjects fill out a questionnaire to determine their reactions to the computer. Future versions of the experiment will incorporate voice input.
Clifford Nass
Assistant Professor of Communication
Stanford University
nass@suwatson.stanford.edu

Visual Object-oriented C (VOC) A visual programming language that is a super-set of the Objective-C language. Programmers define and connect icons to construct applications. Connections between icons can be changed while the application is executing. VOC is used to develop human-machine interface, to improve code reusability, and for rapid application development.
Suichi Tashiro
Researcher in Computer Science
Electrotechnical Laboratory, Japan
tashiro@etl.go.jp

Language

Speech Synthesis System
Researchers are developing a text-to-speech object for NeXT computers. Pronunciation is based on dictionary look-up, with letter-to-sound for words not found. The initial system uses proprietary segment rhythm and intonation algorithms, and offers some speed and accent control. Related research includes Touch ‘n’ Talk, a device to help visually disabled persons access computers, and automatic speech and lip synchronization for computer animated speaking faces. Trillium Sound Research Inc., of Calgary, plans to commercialize the research.
David Hill
Professor of Computer Science
University of Calgary
hill@cpsc.ucalgary.edu

Mathematics

Courseware for Mathematics and Physics A suite of classical mechanics simulations that integrate *Mathematica* Notebooks, custom interfaces, and C-language mathematics libraries to help students move from introductory courses in mathematics and physics to advanced courses. *Status: ongoing*.
Charles G. Flemming
Educational Computing Services
Allegheny College
cfcgf@ux1.cts.eiu.edu

Multimedia

VideoEdit An application to control up to 32 video devices using VideoMedia’s V-LAN transmitter and receivers. It functions as a video editor and frame-by-frame animation controller. It can also edit Edit Display Lists. Future plans include integration of video capabilities of the NeXTdimension system.
Status: under development.
Harold Brokaw
Associate Director, Center for Performing Arts and Technology
University of Michigan
hal@cpat.umich.edu

Music

Frame Suite Programs for granular

Academic Projects

(continued)

synthesis. *frame* uses granular synthesis to generate IRCAM and NeXT .snd files. Other programs in the suite generate script files for *frame*. The suite is used along with NeXTCmix and the sound and music kits to teach a two semester undergraduate/graduate course on synthesis and programming. Wright uses these tools for composition and scored two operas, *The Trojan Conflict* and *Dr. Franklin*.

Maurice Wright
Chairman, Department of Music Composition
Temple University
(215) 787-8013

Philosophy

Artificial Morality An environment for users to construct moral and amoral (rational) agents. Agents can be tested by playing mixed-motive games such as the Prisoner's Dilemma and Chicken.

Peter Danielson
Associate Professor of Philosophy
University of British Columbia
danielsn@unixg.ubc.ca

Physics

X-ray Signal Acquisition Satellite
A NeXT computer is used to prototype a satellite-based X-ray detector. The satellite's signal processing capabilities are to be based on the Motorola DSP56001 chip, the same DSP built into NeXT computers.

Nobuyuki Kawai
Research Scientist
The Institute of Physical and Chemical Research, Japan
nkawai@rkna50.riken.go.jp

Publishing

Caligula: Emperor of Rome* and *Roman Imperial Grand Strategy
Two history books written using FrameMaker on a NeXT computer. Camera-ready copy for *Roman Imperial Strategy* was created using the NeXT 400 dpi laser printer.

Arther Ferrill
Professor of History
University of Washington
ferrill@u.washington.edu

Signal Processing

Automated Lip Contour Tracing
Read My Lips This project uses video capture, contour tracing, and modeling techniques (involving parameters such as mouth opening area and lip height and width) to determine the visemes (visual equivalent of phonemes) produced by speaking lips.

Wigner-Ville Distribution A graphical interface and a statistical package to analyze biological acoustical and electrical signals using the Wigner-Ville distribution.

John Nicol
Systems Analyst in Audiology and Speech Sciences
University of British Columbia
jnicol@mtsg.ubc.ca

Symbolic Analysis of Signals and Systems 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. Accompanying *Mathematica* Notebooks teach piecewise convolution and the z-transform.

Brian L. Evans
Graduate Research Assistant in Electrical Engineering
Georgia Institute of Technology
evans@eedsp.gatech.edu

Virtual Reality

Fractal Boxes A 3D virtual reality representation of a database structure of global problems related to the environment, economy, and politics. The structure is a hierarchy of boxes, each assigned with text, images, and sounds for a specific topic. Users look at the problem structure from different angles, enter boxes, and view the substructure of the problems. Each box has associated sounds such as news related to the problems, spoken information, music, or environmental sounds that help users keep their orientation within the structure. The auditory portion of the reality is created with a NeXT computer. A Silicon Graphics® computer provides the visuals. *Fractal Boxes* will be presented as part of the *Earth Station* exhibit at the Ars Electronica Festival for Art Technology, and Society, in Linz, Austria.

Gottfried Mayer-Kress
Visiting Assistant Professor of Mathematics
University of California, Santa Cruz
gmk@ucscc.ucsc.edu

If you would like your project(s) included in future sections of Academic Projects, please submit the following information:

Project title and discipline
Project manager's full name and title
Institution/Organization
Department
Address, phone, and e-mail address
Number of systems in department
Brief description of project
Project status

Please send to:

next_on_campus@next.com or
NeXT on Campus/Academic Projects
NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063

NeXT Archives

This section lists of some of the newest public-domain software, documentation, programming examples, and other resources available for NeXT users at public archive sites. All resources are accessible using the file transfer protocol program, and most are available via e-mail.

Due to space constraints, resources listed in previous issues of *NeXT on Campus* may not appear in this list.

FTP Archive Sites

Site	*Focus*	*Hostname*	*IP Address*
Electrotechnical Laboratory, Japan	General	etlport.etl.go.jp	192.31.197.99
Oregon State University	General	cs.orst.edu	128.193.32.1
Princeton University	Music	princeton.edu	128.112.128.1
Purdue University	General	nova.cc.purdue.edu	128.210.7.22
University of Illinois	*Mathematica*	ftp.ncsa.uiuc.edu	128.174.20.50

E-mail Archive Sites

Purdue	General	archive-server@cc.purdue.edu	na

Resources

Applications

Name	*Site*	*Pathname*	*Description*
AutoScore	Purdue	pub/next/2.0-release/binaries	Automatically plays music after an idle time
Bessie	Purdue	pub/next/sounds	An interactive tutorial on FM synthesis
Boinkout	Purdue	pub/next/2.0-release/binaries	A souped-up break-out game with color and sound
DraftApp	Purdue	pub/next/2.0-release/binaries	A charting and drafting tool
IComm	Purdue	pub/next/2.0-release/binaries	Telecommunications software with vt100 emulation and zmodem file transfer protocol
Reversi	Purdue	pub/next/2.0-release/binaries	The classic board game
X11R4.pkg	Oregon	pub/next/binaries	The X-Windows system and libraries
Zen	Purdue	pub/next/2.0-release/binaries	A NeXTstep ELK-Scheme environment

Demos

CreateBeta	Oregon	pub/next/demos	A beta version of Stone Design's draw program
DataPhileBeta	Oregon	pub/next/demos	A beta verion of Stone Design's database program
Diagram!	Purdue	pub/next/demos	A general purpose chart and diagramming tool
PowerGlove	Purdue	pub/next/demos	A paint program that uses the Mattel PowerGlove for input

Utilities and Programming Examples

greyboard	Purdue	pub/next/2.0-release/source	A net-wide scratch pad for conferences
NeXTmille	Purdue	pub/next/2.0-release/source	The Mille Bourne card game

Newsletters

BaNG	Purdue	pub/next/Newsletters	Newsletter of the Bay Area NUG
BuZZNUG	Purdue	pub/next/Newsletters	BuZZings and the NeXT User Journal
NeXTVieW	Purdue	pub/next/Newsletters	University of British Columbia journal
NeXus	Electro	pub/NeXT/documents/newsletters	Newsletter of the NeXT User Society
OnCampus	Purdue	pub/next/Newsletters	*NeXT on Campus*
rmNUG	Purdue	pub/next/Newsletters	Newsletter of the Rocky Mountain NUG
SCaNeWS	Purdue	pub/next/Newsletters	Newsletter of the Southern California NUG
Tao	Purdue	pub/next/Newsletters	Robert Lin's tabloid about NeXT

Miscellaneous

AppKit	Purdue	pub/next/2.0-release/docs	Release 2.0 AppKit class heirarchy
Fonts-2.0-sw	Purdue	pub/next/misc	A collection of shareware PostScript fonts
NeXTAnswers	Purdue	pub/next/docs/NextAnswers	Answers to technical questions
Dvorak Keymapping	Purdue	pub/next/2.0-release/misc	A keymapping for Dvorak keyboards

In focus: *Mathematica* 2.0

Theodore W. Gray and Jane O. Rich
Wolfram Research, Inc.

Mathematica, a general system for doing mathematics by computer, was created by Wolfram Research, Inc. to provide scientists and mathematicians with a comprehensive mathematical productivity tool. In the two years since its introduction, *Mathematica* has become a valued tool for instructors, researchers, and students at academic institutions worldwide, as wells as for technical professionals in the corporate world.

Mathematica consists of two parts: the kernel and the front end. The kernel is the mathematical engine that carries out calculations. The front end is the graphical user interface that lets users interact with the kernel. Together, the kernel and the front end allow users to evaluate numerical expressions, including arbitrary precision evaluation of most common and special functions; and to manipulate symbolic expressions for, say, integration, differentiation, series expansion, and pattern matching. *Mathematica* includes powerful graphics features, allowing users to make two- and three-dimensional plots as well as arbitrary combinations of graphics primitives, and a high-level programming language that lets users extend *Mathematica* to handle a variety of problems. The *Mathematica* front end allows users to create interactive documents (called Notebooks) that combine *Mathematica* input and output with text, graphics, and sounds. Notebooks created on NeXT computers are fully compatible with Notebooks created on other platforms.

In keeping with a mutual commitment to education, Wolfram Research and NeXT Computer have collaborated by bundling *Mathematica* with every NeXT computer sold to education users in North America.

Mathematica 2.0

Mathematica 2.0 was announced in January, and will ship on the NeXT platform in the second quarter of 1991. *Mathematica* 2.0 includes major improvements over *Mathematica* 1.2 in both the mathematical kernel and the Notebook front end.

The kernel

The *Mathematica* 2.0 kernel has been redesigned to be faster and more efficient than *Mathematica* 1.2. The kernel has been streamlined and many internal functions and algorithms have been rewritten to increase speed and efficiency. Here are some of the most significant improvements in *Mathematica* 2.0:

The sounds of mathematics. Although it has been possible to hear mathematical functions for many years, it has generally involved complicated and inconvenient programming, and been limited to simple numerical functions. People wishing to hear Airy functions have often been frustrated. With *Mathematica,* almost any function can be easily played.

The new sound functions, `Play[]` and `ListPlay[]` work much like the graphic functions `Plot[]` and `ListPlot[]`. The `Play[]` function converts any periodic function into a sound. For example, `Play[Sin[263 2 Pi t], {t, 0, 1}]` plays one second of middle-C. The `ListPlay[]` function turns any list of numbers, for instance the digits of a rational number, into a sound. Functions can be combined to produce harmonies and it is possible to produce stereo sound by giving `Play[]` a list of two functions.

The benefits of *Mathematica'*s sound functions apply across a range of disciplines. The advantages to music and acoustics researchers is obvious, but physicists and engineers also use their ears to analyze complex signals (from the mechanic diagnosing engine troubles by listening, to physicists listening to audio translations of charged particle waves in the Martian atmosphere). Until now, mathematicians were left out.

Functions. The number of functions in *Mathematica* has increased from 560 to 834. The most requested function—numerical differential equation solving—is a feature of *Mathematica* 2.0. Linear optimization, another popular request, is also supported.

Programming language. *Mathematica*'s internal programming language has been extended to include block structure and variable scoping functions. These functions allow the creation of larger and more efficient *Mathematica* programs. New debugging features are also supported.

Numerical expression compiler. Invoked automatically by many functions, the numeric compiler in *Mathematica* 2.0 makes it possible to evaluate complex numerical functions up to 20 times faster than in *Mathematica* 1.2. The compiler may also be invoked manually.

MathLink. The MathLink communication standard makes it easy for *Mathematica* programs to call functions inside external C programs and for C programs to call functions within *Mathematica* 2.0. People writing NeXTstep applications, even applications not directly related to mathematics, can use *Mathematica* to perform a variety of numerical, symbolic, graphical, or audio tasks.

Debugging. The *Mathematica* programming language has been enhanced to make debugging large programs easier. *Mathematica* has two new debugging commands: `Trace` and `Dialogue`. Using `Trace`, programmers can obtain a list of any assignment or transformation that occurs during the evaluation of an expression. By placing `Dialogue` commands in their expressions, programmers can arbitrarily halt evaluation and inspect the values of variables and functions.

The front end

The *Mathematica* 2.0 front end is a full implementation of the Notebook front end. The front end has been enhanced with new capabilities such as improved sound and animation control. Here are some of the new features:

Sound. Sounds generated by the *Mathematica* kernel `Play[]` or `List-Play[]` commands can be displayed in a Notebook. The sounds are contained in Sound Cells, which display a sound's waveform and can be manipulated much like graphics. Sounds pasted into a Notebook from other NeXT sound applications can be converted into *Mathematica* expression form and analyzed or manipulated. Using the Animate command, sounds and graphics can be combined into synchronized animations. Sounds can also be cut and pasted between the *Mathematica* 2.0 front end and other NeXTstep applications.

Debugging. The *Mathematica* 2.0 front end provides support for the new debugging features of the kernel. In particular, when users halt a cell evaluation by typing Command-"period" they are given the option of inspecting variables before continuing or aborting the calculation.

Electronic publishing. Though *Mathematica* is not a page layout program, improvements in the front end allow users to write interactive electronic documents and generate attractive printed documents. The 2.0 front end supports common text formatting commands, multiple fonts, and margin rulers. New cell spacing and text leading commands give the user control over paragraph formatting. The first large scale electronic document made with *Mathematica* 2.0 is the book *Exploring Mathematics with Mathematica* by Theodore Gray and Jerry Glynn. (See "Third-Party Products" on pages 32-33 for more information about this book.)

Mathematica 2.0 is available, free of charge, with every NeXT computer delivered to education users in North America. *Mathematica* comes with on-line documentation and a documentation kit for educational customers. NeXT users who are not in education can purchase *Mathematica* 2.0 directly from Wolfram Research, Inc. by calling (217) 398-0700.

The Mathematica 2.0 front end can display sounds as either a mathematical expression or a wave-

Image 13

Third-Party Products

This section includes a sampling of software and hardware available for NeXT computers. For more information about these products, please contact vendors directly. For information on additional software and peripherals for NeXT computers, please call 1-800-848-NeXT.

Software

Analytical Tools

Improv™ The spreadsheet redefined with formulas in English, rearrangeable data, graphics, sound, and 1-2-3 compatibility. Lotus Development Corp. 1-800-577-8500

Connectivity

3270Vision™ An integrated NeXTstep application that connects NeXT computers to IBM® mainframes over SNA and TCP/IP networks.
3270Vision Coax A package consisting of 3270Vision and Coax Gateway software and an internal Coax Adapter. It connects NeXT computers to IBM 3174 or 3274 controllers and emulates an SNA Distributed Function Terminal. Conextions, Inc. (508) 475-5411

FloppyWorks Software to enable NeXT-compatible floppy drives to read and write Macintosh 1.44 MB formatted files. FloppyWorks supports Macintosh-compatible hard and removable media drives. Digital Instrumentation Technology, Inc. (505) 662-1459

Microphone 3.0 Telecommunications package designed specifically for microcomputers with a graphical interface. Software Ventures Corp. (415) 644-3232

Worldtalk/400™ This product family provides a ready-to-deploy solution for organizations that need an enterprise-wide messaging system. It interconnects many existing messaging applications, including NeXTmail. Touch Communications, Inc.(408) 374-2500

X.25/Daemon-S Serial port X.25 software that provides a gateway to public and private X.25 networks. Morning Star Technologies, Inc. 1-800-558-7827

Information Management

OCR Servant–Limited Edition
Optical character recognition software that ties in seamlessly with NeXT's "Services" feature. HSD Microcomputer. (415) 964-1400

OnDuty™ An integrated set of utilities for office and personal tasks. Features include a phone directory and dialer; time log and contact information log; and a calendar and appointment book. Digital Instrumentation Technology, Inc.
(505) 662-1459

OMEN III An integrated investment information systems for stockbrokers and other investment professionals. Microstat Development Corp. (604) 228-1612

PaperSight™ **and Papersight Light**
Document image management solutions that provide organizational features and facilitate communication. A Kurzweil optical character recognition option is available.
Visual Understanding Systems, Inc. (412) 687-3800

Virginia Tech InfoStation 1.0
A hypermedia information access system for library automation, including on-line catalog search and retrieval of audio, textual, and graphic data. VTLS, Inc. (703) 231-3605

Who's Calling?™ **2.0** A fully automated telephone tracking system with multiuser access and voice response. Adamation, Inc.
(415) 452-5252

Languages & Tools

Allegro CL Common LISP A full-featured LISP programming environment, fully compatible with Software Release 2 and 68040-based NeXT computers. Franz, Inc. (415) 548-3600

[OT Palettes: 2.0]™ A collection of custom object palettes to extend the development power of Interface Builder. Available palettes: **SmartField** provides an advanced set of text fields for data entry and data validation. **Chooser** provides a scrolling chooser. **MathPalette** objects facilitate front-end development by providing communication between applications and the
Mathematica kernel. **GraphPalette**™ provides multiple-range, two-dimensional plotting. Objective Technologies, Inc. (212) 227-6767

Presentation

Diagram!™ A general-purpose diagraming and charting tool. Lighthouse Design, Ltd. 1-800-366-2279

Publishing

The Font Company Type Library
More than 1,500 different typefaces, all in PostScript Type 1 format.
The Font Company. (602) 998-9711

FrameMaker 2.0 A technical publishing software package that includes word processing, graphics, page layout, equation editing, and book-building tools. Frame Technology Corp. (408) 433-3311

Illustrator 3.0 A graphic design and illustration program for generating high-quality artwork. Adobe Systems, Inc. (415) 961-4400

***Mathematica* 2.0** A comprehensive software application for mathematical computation. Wolfram Research, Inc. (217) 398-0700

Redmark™ Groupware for multiple proofs and critiques of all types of documents. Features include standard proofreading symbols, drawing tools, text, voice recordings, and imported graphics.
Epitome, Inc. (615) 675-0910

TopDraw™ **2.0** An object-oriented drawing package with sophisticated illustration and layout capabilities. Media Logic, Inc. (213) 453-7744

WordPerfect® A full-featured word processor. WordPerfect Corp. (801) 225-5000

System Software

co-Xist The X windows environment with Motif. co-Xist lets users run NeXTstep applications and X windows applications side-by-side. Pencom, Inc. (512) 343-1111

SoftPC 2.0™ A complete software emulation of an IBM PC, AT-class compatibility and performance. Insignia Solutions, Inc.
(408) 522-7600

Sound & Music

SoundWorks 2.0 Software for viewing, editing, and manipulating digital audio or other sampled data. Metaresearch, Inc. (503) 238-5728

Utilities

LockScreen A screen saver/lock that eliminates screen burn and provides security for labs or offices. imp software. (415) 949-0240

Hardware

Ariel QuintProcessor™ An add-in board with five 27 MHz M56001 DSP chips and on-board RAM.
Ariel Corp. (201) 249-2900

CubeFloppy 2.9 A floppy drive that reads and writes 3.5-inch disks (720 KB, 1.44 MB, and 2.88 MB). Each drive is shipped with a SCSI cable, FloppyWorks software, full documentation a one-year warranty. Digital Instrumentation Technology, Inc. (505) 662-1459

Dazzl Model 16/12 Analog to Digital Converter An add-in board with 16 single-ended or 8 differential channels, a maximum sampling rate of 200 KHz, and a centronics-compatible parallel port. Dazzl, Inc. (309) 674-9317

Digital Eye Color An 8-bit color video digitizer, which accepts signals in NTSC and S-VHS formats. Metaresearch, Inc. (503) 238-5728

Eclipse Mass Storage The Eclipse line includes DAT backup drives, removable hard drives, CD-ROM drives, and hard disks ranging from 120 MB to 1 GB. Microtech International, Inc. (203) 468-6223

GeoKit Objects that provide map rendering and graphical informational services (GIS). The kit includes a dynamic map sheet, a user-extensible geodetic projection (26 built-in projections), and a graticule object. Deltos Fleet Computing.
(214) 540-2301

Interfax Modem A 9600 bps Group 3 fax modem with a 2400 bps MNP 5, Hayes-compatible data modem. Abaton. (415) 683-2870

Makedisc CD-ROM formatting software for creating a CD-ROM disc image. Young Minds, Inc.
(714) 335-1350

Metro Mass Storage DAT backup drives, CD-ROM drives, and hard disks ranging from 170 MB to 670 MB. EMAC. 1-800-821-0806

PLI SuperFloppy 2.8 An external SCSI 3.5-inch floppy disk drive capable of reading and writing to 720 KB and 1.44 MB MS-DOS or UNIX formatted disks, as well as the new 2.88 MB disk standard. Peripheral Land, Inc. 1-800-288-8754

Rewritable dual-sided optical disks 512 MB NeXT-compatible optical disks. MicroNet Technology, Inc. (714) 837-6033

Scan-X Color A 24-bit true color scanner that supports resolution up to 300 dpi for 24-bit color images and up to 2400 dpi for line art. HSD Microcomputer. (415) 964-1400

Mathematica Resources

Exploring Mathematics with Mathematica A guide to how *Mathematica* can apply to real-world problems, including basic and advanced concepts and techniques. Each copy includes the book on a self-running CD-ROM disk. Theodore Gray and Jerry Glynn, Addison-Wesley. 1-800-447-2226

***Mathematica: A System for Doing Mathematics by Computer,* Second Edition** The premier *Mathematica* reference book. Stephen Wolfram, Addison-Wesley. 1-800-447-2226

Mathematica for the Sciences *Mathematica* examples from biology, chemistry, engineering, mathematics, and physics. Intended for students, teachers, researchers, and others wishing to explore science with *Mathematica*. Richard Crandall, Addison-Wesley. 1-800-447-2226

The Mathematica Graphics Guidebook A reference manual and idea book for *Mathematica* graphics programming. It features explanations of *Mathematica*'s graphics and the underlying PostScript level. Cameron Smith, Addison-Wesley.
1-800-447-2226

The Mathematica Journal A quarterly publication featuring scholarly articles, applied contributions, product updates, news features, and *Mathematica*-related events. Addison-Wesley. (415) 594-4423

Mathematica Quick Reference Guide A guide to *Mathematica* commands and functions. Variable Symbols, Inc. (415) 843-8701

***Programming in Mathematica,* Second Edition** A guide on using *Mathematica*'s built-in programming language that covers the new features in *Mathematica* 2.0. Roman Maeder, Addison-Wesley.
1-800-447-2226

NeXT User Groups

To find out more about a particular NeXT user group, please contact the group directly. If your user group is not listed here, or if you would like to start a new group, please send e-mail to user_groups@next.com for information on how to start a NeXT user group.

Australia

OzNeXT, Queensland Chapter
Bond University
(075)951-423
goose@hal.nmg.bu.oz.au

OzNeXT, Melbourne Chapter
University of Melbourne
(03) 344-5397
paul@terrapin.iaesr.unimelb.edu.au

Canada

Alberta

Club NeXT, University of Calgary
dibb@cpsc.ucalgary.ca

Edmonton NeXT Users
University of Alberta
(403) 492-2462
george_carmichael@mts.ucs.ualberta.ca

British Columbia

University of British Columbia NUG
(604) 228-3429
bob_bajwa@mtsg.ubc.ca

Vancouver NeXT Group
Simon Fraser University
(604) 291-4702
lionel_tolan@cc.sfu.ca

Ontario

TANG
Toronto Area NeXT Group
(416) 482-NeXT
david_lavallee@next.com

Ottawa NeXT User Group
University of Ottawa
(613) 594-3352
danielc%nx1%dciem@uunet.uu.net

University of Guelph NeXT User Group
(519) 824-4120, x4521
next@snowhite.cis.uoguelph.ca

University of Western Ontario
NeXT Group
(519) 663-0817
frank@hybrid.ccs.uwo.ca

Quebec

Montreal NeXT Section of Club
Macintosh
McGill University
(514) 939-0382
paulhus@quiche.cs.mcgill.ca

Denmark

NeXTiDK
bmayoh@daimi.aau.dk

France

FaNG
French area NeXT Group
73020.2442@compuserve.com

French NeXT User Group
nicolas@bora.inria.fr

Germany

Hannover NeXT User Group
(49) 511 421278
martin@cube.han.de

Japan

NeXus
Japan NeXT User Society
NeXus-office@etl.go.jp
Fax: +81-03-351-0880

Mexico

MeXNeXT
ITESM
(83) 58-20-00, x4071
dtrevino@mtecv2.mty.itesm.mx

The Netherlands

Netherlands NeXT User Group
urquijo@cst.prl.philips.nl

Portugal

Lisbon NeXT User Group
Instituto Superior Tecnico
351-1-802045, ext. 1619
amaro@ptifm.ifm.rccn.pt

Sweden

SnAG
Svenska NeXT-AnvandarGruppen
+46 8 752 15 45
snaginfo@sics.se

United Kingdom

UK NeXT User Group
+44 (0)81 458 0434
nextug@epsoft.uucp

United States

Alabama

University of Alabama, Birmingham
NeXT Group
(205) 731-9660
hsf!next17995!mark@uunet.uu.net

Alaska

Arctic Circle NeXT User Group
University of Alaska, Fairbanks
(907) 479-2247
fsapm@alaska.bitnet

Arizona

Motorola NeXT User Group
(602) 952-3417
ray@doskocil.sps.mot.com

Phoenix NeXT User Group
(602) 869-0316
jsoft!ggf@uunet.uu.net

Tucson NeXT User Group
University of Arizona
(602) 621-2284
layhe@rcnext1.rc.arizona.edu

California

BANG
Bay Area NeXT User Group
(415) 549-2684, 243-9140
bang-request@bang.org

Berkeley Mathematics Software Group
(415) 843-8701
nb@cs.stanford.edu

Berkeley NeXT User Group
University of California, Berkeley
(415) 644-0139
vanroy@arpa.berkeley.edu

BMUG, NeXT SIG
(415) 549-2684, 243-9140
bfd@meta-x.stanford.edu

CaJUN
Caltech and JPL Users of NeXT
(818) 356-9258
ernest@pundit.cithep.caltech.edu

Cal Poly NeXT User Group
(805) 541-6879
mengel@data.acs.calpoly.edu

ESL NeXT User Group
dml@mozart.esl.com

FOGNUG
San Francisco State University
(415) 406-4873
preuss@sutro.sfsu.edu

NeXTUCLA
University of California, Los Angeles
(213) 824-1174
denison@euphemia.math.ucla.edu

Nuggets
California State University, Los Angeles
(213) 343-2400

Orange County NeXT Group
(714) 938-8850

Santa Barbara NeXT User Group
University of California, Santa Barbara
(805) 968-5584
erone%pumpkin@hub.ucsb.edu

Santa Cruz NeXT User Group
University of California, Santa Cruz
taff@spica.ucsc.edu

SCaN
Southern California NeXT Group
(213) 985-1550
mahoney@grafix.cse.csulb.edu

SNuG
San Diego NeXT User Group
(619) 481-7535, 565-9738
tfinn@next.com

UC Riverside NeXT User Group
(714) 787-3883
plowe@ucrac1.ucr.edu

UC, Irvine NeXTclub
(714) 856-5543
jchoi@next.acs.uci.edu

Colorado

rmNUG
(303) 530-2560
davehieb@boulder.colorado.edu

CSU NeXT Computer Users Club
(303) 491-1423, 484-4785
tn505981@longs.lance.colostate.edu

District of Columbia

Naval Research Labs NeXT User Group
(202) 767-2165
yiannis@prologos.nrl.navy.mil

WANSIG
Washington Area NeXT SIG
(703) 938-NeXT
joel@next.com

Florida

Miami NeXT User Group
(305) 854-8954
mgilula@miasun.miami.edu

Orlando NeXT User Group
(407) 649-3011, 872-0727

Georgia

BuzzNUG
Georgia Tech NeXT User Group
(404) 352-5551
erica%kong@gatech.edu

Hawaii

Hawaii NeXT Group
(808) 225-3830
74220.242@compuserve.com

Idaho

Boise State University NeXT User Group
grantham@tohobit.idbsu.edu

Illinois

Argonne NeXT User Group
(708) 972-5963
henderson@mcs.anl.gov

AT&T NeXT User Group
(708) 713-4374
coco@ihcoco.att.com

ChiNUG
Northwestern University
(708) 491-5368
bill_ parod@nwu.edu

Indiana

Notre Dame NeXT User Group
(219) 239-5600, x6421
mahesh@darwin.cc.nd.edu

THaNG
Rose-Hulman Institute of Technology
(812) 877-9030
thang-request@
dev1.sem.rose-hulman.edu

Iowa

NUGI
University of Iowa
(319) 338-4741
tdawson@shumun.weeg.uiowa.edu

Iowa State University NUG
cyliao@eng.umd.edu

Kentucky

Kentucky NeXT User Group
University of Kentucky, Lexington
(606) 258-8655
neil@graphlab.cc.uky.edu

Maryland

Maryland NeXT User Group
(301) 764-1878
73340.3167@compuserve.com

University of Maryland NUG
whodges@wam.umd.edu

Massachusetts

Boston Computer Society NeXT SIG
1-800-922-NeXT
dlavin@nextworld.com

NeXT Users of Harvard University
(617) 493-6264, 495-1260
montoya@husc9.harvard.edu

StrataNUG
Stratus NeXT User Group
(508) 460-2915
williams@cac.stratus.com

Michigan

Michigan State University NeXT
User Group
(517) 353-4973
erickson@frith.egr.msu.ed

University of Michigan NeXT User Group
(313) 763-9857
nug-admin@um.cc.umich.edu

Minnesota

Minnesota NeXT User Group
Carleton College
(507) 663-4067, (612) 645-9062
mtie@carleton.edu

Missouri

St. Louis NeXT User Group
(314) 342-4996
71511.125@compuserve.com

New Hampshire

DANUG
Dartmouth College
(603) 646-2085
next-contact@othello.dartmouth.edu

New Jersey

Central New Jersey NeXT Group
mmatlack@next.com

Princeton NeXT User Group
Princeton University
(609) 258-8683
jwjames@pucc.princeton.edu

New Mexico

Albuquerque NeXT User Group
Sandia National Laboratories and
University of New Mexico
(505) 846-2613
jnjortn@cs.sandia.gov

NeXT User Groups

(continued)

Los Alamos NeXT User Group
(505) 667-8870
dmb@lanl.gov

New York

Cornell NeXT User Group
(607) 253-0687
jiro@shaman.com

Columbia University NeXT User Group
mara@woof.columbia.edu

GUN
Gotham Users of NeXT
(718) 522-3776
nobugs!gun@uunet.uu.net

Rochester Area NeXT User Group
University of Rochester
(716) 473-0757
brown@ee.rochester.edu

RPI NeXT User Group
Rensselaer Polytechnic Institute
lasteve@rpi.edu

North Carolina

Triangle Area Users of the NeXT Group
(919) 677-8000
pranav@unx.sas.com

Columbus NeXT User Group
The Ohio State University
(614) 771-9187, 292-7211
tsg+@osu.edu

Oklahoma

University of Oklahoma NUG
(405) 325-5550
mlroider@uokmax.ecn.uoknor.edu

Oregon

Portland NeXT User Group
(503) 697-0676
baker@next.com

Oregon State University NUG
leach@satchmo.oce.orst.edu

Pennsylvania

Carnegie Mellon Computer Club, NeXT SIG
(412) 461-3598
jh4r@andrew.cmu.edu

Philadelphia NeXT User Group
mmatlack@next.com

Rhode Island

NeXT-BUG
Brown University
(401) 272-2262
agm@cs.brown.edu

Texas

Austin NeXT User Group
(512) 343-1111
pensoft!lorne@cs.utexas.edu

DaNG
Dallas Area NeXT Group
(214) 385-2991
gtenext1!hofbauer!DaNG@bunny.gte.com

hAng
Houston Area Next Group
University of Houston
(713) 749-1820
glover@uh.edu

TexNeXT
Texas A&M NeXT User Group
(409) 845-1308
daugher@cs.tamu.edu

Utah

Salt Lake City NeXT User Group
(810) 240-1017
tarbet@chemistry.chem.utah.edu

INFO...
Brigham Young University
(801) 371-3256
info@zapotec.math.byu.edu

Washington

Seattle Area NeXT Group
(206) 448-0845
0004454438@mcimail.com

University of Washington NUG
(206) 543-5611
corey@cac.washington.edu

Washington State University NeXT Group
(509) 334-9594
gerkman@bongo.csc.wsu.edu

Wisconsin

University of Wisconsin NeXT Group
(608) 262-0138
thomson@macc.wisc.edu

Special Interest E-mail Groups

Special interest e-mail lists provide a way for users to exchange information. Each listing provides two e-mail addresses: the first is to subscribe to the list, and the second is to exchange mail with subscribers.

Classroom and Courseware

To subscribe, send mail to: MAILSERV@gac.edu with the following text: "Subscribe next-classroom <your name>"
next-classroom@gac.edu

FrameMaker

framers-request@drd.com
framers@drd.com

Mathematica

mathgroup-request@yoda.ncsa.uiuc.edu
mathgroup@yoda.ncsa.uiuc.edu

Network and Security Management for Installed Labs

next-lab-request@cs.ubc.ca
next-lab@cs.ubc.ca

Managers (quick, technical answers)

next-managers-request@stolaf.edu
next-managers@stolaf.edu

Medical

NeXTMed-request@ulnar.biostr.washington.edu
nextmed@ulnar.biostr.washington.edu

Music

nextmusic-request@silvertone.princeton.edu
nextmusic@silvertone.princeton.edu

Programmers

next-prog-request@cpac.washington.edu
next-prog@cpac.washington.edu

NeXT Bulletin Boards

comp.sys.next
Accessible via Usenet newsreaders

CompuServe NeXT Forum
Accessible via CompuServe

To receive a subscription to *NeXT on Campus*™, please complete and return this card. Please print or type.

Full Name ....

Title ....

Age: ☐ 18-25 ☐ 26-40 ☐ 41-60 ☐ 61 or older

Institution Name ....

Department ....

Address ....

City .... State ....

Zip .... Country ....

Daytime Phone ....

E-mail Address ....

Check all that apply:

Faculty: ☐ Assistant Professor ☐ Associate Professor

☐ Full Professor ☐ Department Chairperson

Staff: ☐ Administrative ☐ Technical

Student: ☐ Undergraduate ☐ Graduate

☐ Post Graduate

Other: ☐ Developer ☐ Reseller

☐ Service Center ☐ Support Center

Do you currently use a NeXT™ computer?

☐ Yes ☐ No

Describe what you use your NeXT computer for and which applications you use. (Feel free to attach additional pages.)

Suggestions, contributions, and subscription requests can also be sent to us electronically at next_on_campus@next.com