The Computational Science Revolution: Technology, Methodology, and Sociology

Larry Smarr
National Center for Supercomputing Applications
University of Illinois at Urbana-Champaign
Champaign, Illinois 61820

We are living in a period during which a new mode of science, **computational science**, is emerging. Its development parallels the development 300 hundred years ago of the other two modes of science: theoretical and experimental/observational. Just as Newton and Galileo stand as the historical geniuses who stated the rules of the game for those two methodologies, John von Neumann does so for computational science. In addition to new methodologies, new **technologies** or mathematical tools spurred on the previous revolutions. For example, Newton's invention of the calculus allowed him to codify mathematically the laws of nature and to develop the beginnings of analytic methods for solving simple cases. Similarly, the development of the von Neumann computer architecture allowed von Neumann to state how the digital computer could be used to solve the discretized laws of nature for general and complex cases.

Because of its ability to find solutions of **realistic complexity**, computational science relies heavily on **visualization** to represent these solutions. Images are the "closed form" for representing numerical solutions, similar to how functions, such as $\sin(x)$, play this role for analytic solutions. Image processing will take the place of analytic manipulations for computational science in the coming years.

Rather than replace theory or experiment, computation brings these two methodologies closer together. Typically, a theoretical analysis precedes any computation. Part of the validation procedure for a complex code is to run situations in which experimental or observational data are available for comparison. In many areas of research, the ability to make realistic simulations is pulling together theorists, experimentalists, and computationalists into a team effort to understand a complex problem.

This article explores in broad terms the three categories of the computational environment that I think are going to be rapidly developing in the next five years. The obvious focus is the **technology**, but I think a more fundamental change is occurring in the intellectual environment. This change is seen in how the computational science **methodology** itself is emerging and how the human enterprise or **sociology** of doing science is evolving.

Technology

The invention of the personal computer (PC) ten years ago was a key to the rapid technological developments in the user's computational environment. There are now more than ten million installed PCs in the United States. If you count the speed of a PC as 1/1000 of the speed of a CRAY-1 supercomputer, then there are more than 10,000 CRAY-1 equivalents on desktops, compared to roughly 100 installed

supercomputers. Gradually these two great technologies, the supercomputer and the personal computer, are becoming integrated into a national, distributed network system of computing.

The mass-market base of PCs has radically increased the rate of development of new tools for the user. One critical effect has been to lower exponentially the cost of memory, so that each user can have a full-color imaging station and can visualize vast amounts of numerical data. New software tools are rapidly appearing on PCs and workstations which, when coupled into the supercomputer, create a qualitatively new interface for the large-scale computational user.

The National Center for Supercomputing Applications (NCSA) is one of five supercomputer centers established by the National Science Foundation. NCSA is dedicated to building a comprehensive computing environment at the University of Illinois based on these emerging technologies. I will be drawing from our experience with a few specific examples of these new technologies.

Distributed Computing

Color Plate 1 has become our symbol for distributed computing. The system has three components. The large red pieces constitute our supercomputer: the CRAY X-MP and the Solid-state Storage Device (SSD). The little box is an Apple Macintosh personal computer. The human is a scientist, John Kogut, who is one of the contributors to this volume. He is doing most of his interactive work on the Mac, with some networking (a serial line) linking it to the CRAY that is running a large code. At the time this photo was taken, three weeks after the CRAY "hit the floor" on August 28, 1985, what is shown was the complete NCSA system. The Macintosh was the front end, mass storage, and central output facility. This photo emphasizes that all users are remote, even if only three feet from the CRAY, but because of the personal computer revolution, they now have a complete computing center at their fingertips.

The NCSA system has grown rapidly since August 1985, with the installation of a number of critical intermediate pieces (see figure 1). I will describe briefly these key parts of a distributed computational system, not giving any details but rather emphasizing the general trends that I see in each part.

Supercomputer. I will spend little time discussing the supercomputer because Davidson et al. have covered that topic in the preceding article. The computational environment in which the supercomputer operates is, in my opinion, as significant as the specifications of the supercomputer itself. Sometimes I think about this problem with an analogy to a stereo system. When designing such a system, you can put too much emphasis on the the total wattage of the power amplifier. But you are buying the power amplifier to be part of a balanced system for listening to music, which means that you must pay just as much attention to the capabilities of the preamp, tuners, disk or cassette players, speakers, wiring, and room acoustics.

In the same vein, one must create a balanced computational system. Thus, in this article, I'll assume that I already have a Cedar, a CRAY X-MP, a CRAY-2, a NEC SX-1, or some other power amplifier, with an operating system and software libraries. The question that I will address is how to design the rest of the supercomputer environment so that we can turn that raw computing power into advances in human knowledge.

Before leaving the supercomputer, I will reiterate what the authors have shown in the preceding article, namely that over the next five years parallelism will drive an exponentiation in the capacity of the power amplifier. Color Plates 2 and 3 show the expected growth path in computational power and memory at NCSA. This growth is caused by the upgrading of our CRAY X-MP every two years to the new state-of-the-art machine, which is built into our agreement with NSF. By 1990, assuming normal industry projections will materialize, NCSA's capacity will have grown to at least 50 CRAY-1 equivalents, with at least one billion words of fast memory. That is just the hardware. During the same time period, algorithms will also continue to be developed, so that the net speed-up for a user could be much greater than what is shown in Color Plates 2 and 3.

This rapid growth in capacity will drive an

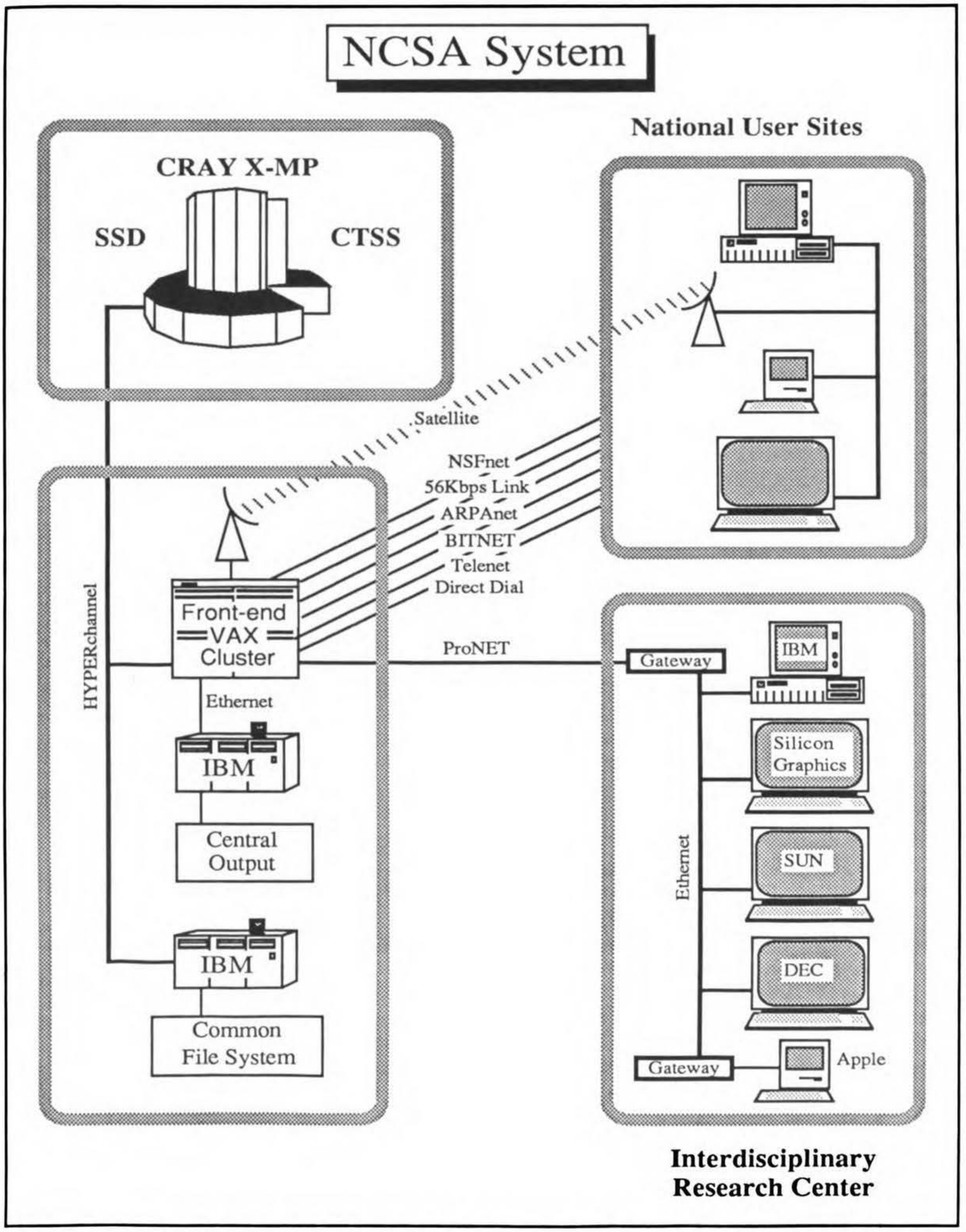


Figure 1. The parts of the distributed NCSA computing environment. Graphic by Scott Lathrop, Manager of User Services.

unprecedented explosion in application development. Contrast the next five years' projected growth rate to that of the last ten years when there has been at best a factor of two or three improvement in the speed of a single-processor supercomputer. I believe that this stagnant period has lulled to sleep this nation's university computational researchers. With a guaranteed growth in the power of the CPU, researchers should start planning now for attacks on heretofore intractable computational problems.

Mass Storage. I think that mass storage will be the greatest bottleneck in the computational environment during the next five years. The easiest way to see this is to recall that a ten gigaflop supercomputer (the standard by 1990) can multiply 10^{19} bits a year. NCSA has as sophisticated a mass storage system as any national NSF center (see figure 1). This system consists of three layers of hardware storage devices (magnetic disk, cartridges, and tape) and runs the Los Alamos Common File System (CFS). It will grow to an on-line capacity (no human tape mounts) of more than a trillion bits in the next few years. However, this means that NCSA will be able to store on-line only one of every ten million bits that are actually multiplied by the supercomputer each year.

What this calculation tells us is that, over the next five years, supercomputer capacity is going to increase more rapidly than mass storage capacity can match. Almost all users today work in a mode that assumes intermediate storage of their computed results. Therefore, central mass storage shortfall will drive a change in computational methodology. We can expect a shift toward local storage of images on laser disks and other exotic storage technology. Visualization in real-time of the supercomputer simulations may be required to avoid prohibitively high costs of central mass storage.

Central Output. In the past, the only way to get hard copy of your calculations was to dump to a central output or local (RJE) facility. This is still necessary for large volume output of paper, film, or video. The automation of the central output job queue and the control of quality on film and video are challenging problems. Constant human adjustment of equipment is still required. However, much of the output that previously would have gone to a central facility will now be routed to the user's workstation. There, the user's ability to scan rapidly data that has been dumped to a local hard disk from the supercomputer as the calculation progresses may reduce the blind dumping of data to central output. As video technology advances rapidly, hard copy of dynamic images rather than numeric data fields should show a great rise in user preference.

Front End. Traditionally, the front end was the machine on which you prepared your job for the batch queue on the supercomputer. However, if you run the Cray TimeSharing System (CTSS) as we do at NCSA, you run your job directly on the CRAY. So the front end machine really should be thought of as a terminal concentrator or a traffic cop for the remote users on the system. The NCSA uses a DEC VAX cluster for the front end (see figure 1) because all the software needed for the networks runs under DEC's VMS. As TCP/IP becomes a universal protocol, point-to-point contacts between machines on the distributed network will make this criterion less important. Therefore, I believe that special-purpose hardware devices with much higher internal bandwidths than the present VAX will be used more and more for network concentrators, so that the bottleneck here in the communication path will be reduced greatly during the next five years.

Network. Thousands of remote users will access the supercomputer centers over the **national network** (NSFnet) that the National Science Foundation is establishing. For more details, see *Science* (28 Feb 1986, 231: 943). The topology of the network will be an overlapping set of regional or consortia networks all hooked together by a high-speed backbone connecting the NSF centers (see figure 2). By the end of 1986, some 100 universities should have access to the centers at 56 kbaud over NSFnet. Until then, many users are using dial-up modems or preexisting networks, such as BITNET, for file transfer. The goal is to operate NSFnet with a uniform national protocol of TCP/IP.

With the national network comes not only access to supercomputers, but also an unprecedented interconnectivity between users that does not exist today. Scientists who have previously

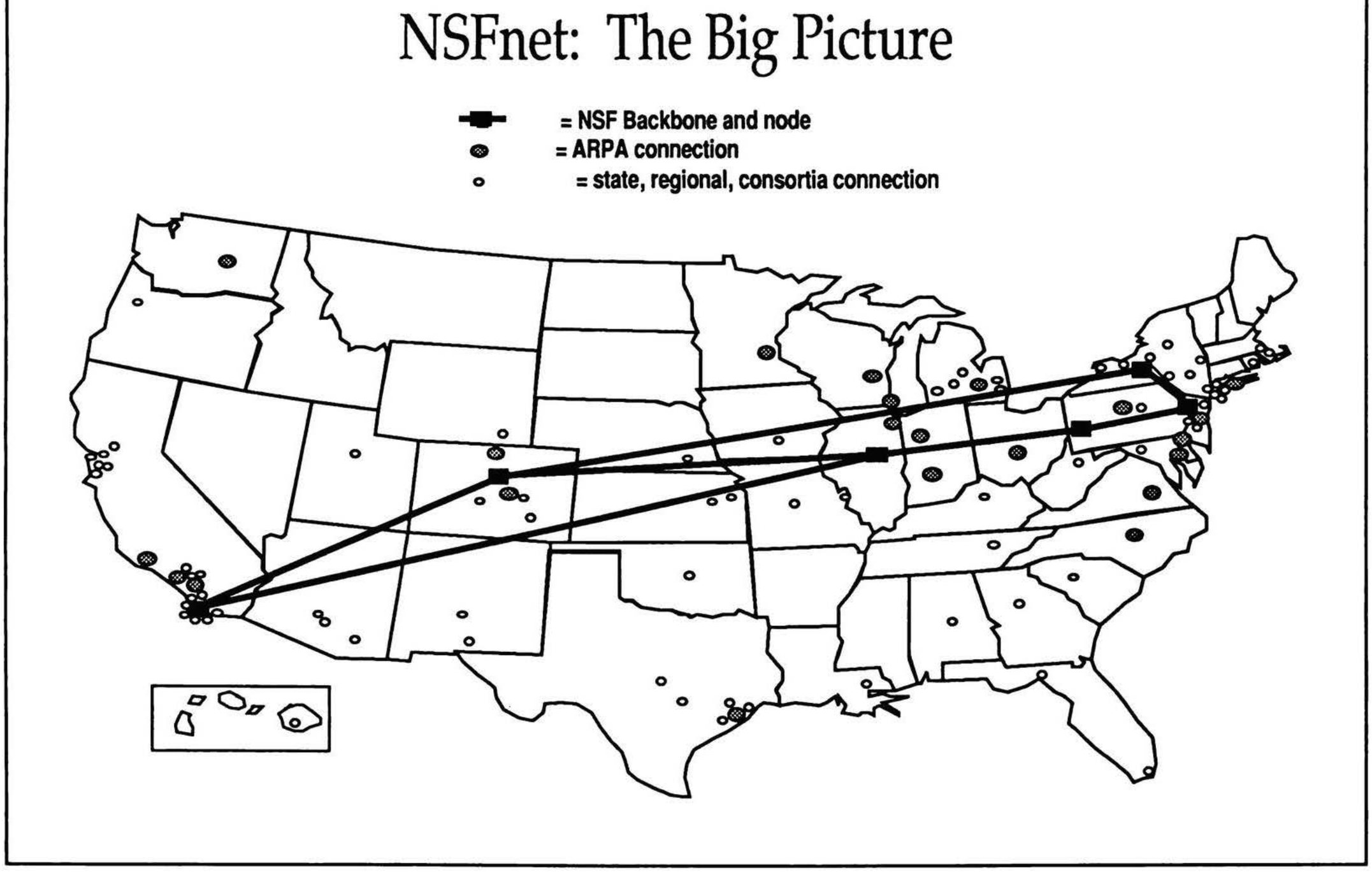


Figure 2: The planned NSFnet. The heavy dark line represents the NSFnet backbone that connects the five NSF centers and NCAR (located at Princeton, Ithaca, Pittsburgh, Urbana-Champaign, Boulder, San Diego) at 56kbps. ARPAnet, consortium, and regional nets attach to the backbone. Graphic by Charlie Catlett.

communicated annually at national meetings can now participate in real-time, multicolleague electronic mail discussions. I think this development is going to cause a major shift in the sociology of computational science.

Workstations. As I mentioned above, the real revolution in distributed computing is the replacement of dumb terminals by personal computers and workstations. This makes possible locally many of the functions that heretofore have been responsibilities of the central facility. Editing of files, compiling and testing of programs, and post-processing of data sets are now done conveniently off-line from the supercomputer. In addition, the software universe the user confronts is vastly richer on the workstation than it was on the mainframe. This suggests the possibility of "hiding the supercomputer" behind the software of the PC.

To hasten this transition, we at NCSA have established an Interdisciplinary Research Center (IRC) in which each desk has a workstation (see figure 1). Our goal is similar to Xerox's PARC experiment in the early 70s, namely to see how users will change their research methodology and sociology when a large amount of computing power and capability is placed on their desks. This environment of networked personal computers and workstations is what all users will have remotely in a few years as prices fall and capability increases. Our experiment will give us advance knowledge of the trends we should monitor as the landscape shifts.

Networking Issues

Because the distributed computing environment works only as well as the network that holds it together, it is important to discuss some key issues that exist in networking.

1) The gap that currently exists between the bandwidth of the remote users, many of whom are using 1200 baud modems, and the bandwidth that can be tapped locally coming directly out of the CRAY, which is 800 megabaud, is close to a factor of one million. Several projects nationally (Digital Productions, Los Alamos, Apple) are allowing users to work with images at a bandwidth near the top end of this range. What can be accomplished at a gigabaud appears to be a miracle compared to a kilobaud. The good news is that this spread allows for a wide range in capability on the network. The differential in user bandwidth, however, leads to real difficulties in deciding which technology receives the most priority.

2) I think narrowing this gap is very important. If we can get most remote users coming in at megabaud speeds over the NSF network, the gap will be narrowed down to a factor of a thousand. That is still enough of a gap to drive technology development. Technology development would take place in a more coherent environment, however, because the software developed would operate throughout the bandwidth range.

3) Most networked computer systems today are based on the model of a dumb terminal hooked to a mainframe where all of the CPU power is in the supercomputer. That is rapidly changing to a situation where every user has at least the computational power and software of a current superminicomputer on his or her desktop. For instance, the new Sun Microsystem SUN-3 with a floating-point accelerator is as fast as a DEC VAX 11/780. This distributes the CPU power and mass storage, so that the network must be completely rebalanced. For example, if you are sitting at a dumb terminal, you don't demand more than 9600 baud from the central CPU because the words scroll off the top of the screen too fast. But if you have a 400 megabyte disk sitting next to your SUN workstation, a megabit a second is not enough bandwidth to feed you. Thus, the widespread use of workstations requires system administrators to reevaluate where to put money into the networked system.

4) Much of the national network planning still labels legs of the network by the leg's bandwidth. That is very deceptive since the real question is, "What is the delivered throughput to the individual user?" The simplest reason this varies is that the number of simultaneous users varies with time of day and from leg to leg at a given moment. But even if these variations did not exist, the effective bandwidth over a single user leg can vary a great deal if the existence of a workstation means that you can compress and decompress at both ends. Putting these effects

together, orders of magnitude variation in the user-delivered throughput across two legs of the network labeled with the same bandwidth is possible.

These network issues all focus on the difficult problem of optimizing and balancing a whole computational system to deliver to the user those tools, capacities, and capabilities that most enable the user to work as efficiently as possible.

Imaging

Now, let me move on to the technology of imaging. Color Plate 4 illustrates the type of image used for years by radio astronomers. It shows the radio emission from a star-forming region in the constellation of Orion. The radio map data were taken with a Very Large Array (VLA) of radio telescopes in Socorro, New Mexico. The data were turned into a map on NCSA's CRAY X-MP by Richard Crutcher of the Astronomy Department of the University of Illinois, and the image was displayed on an IBM PC using a PGA graphics adapter and monitor. For many years radio astronomers have used color or gray scales to represent the spatial distribution of radio emission. That is, at each pixel of the image, the numerical value of radio intensity is mapped to one of 256 color or gray levels. Many different color palettes can be used to bring out different features of the radio map. The simplest is to divide the color bar in spectral order into 256 equal intervals as is done here, with red corresponding to the most intense radio emission. In some cases, the data can require many more color levels than 256 per pixel.

Several years ago Karl-Heinz Winkler transferred this idea of imaging from use in observational astronomy to use in solving a theoretical physics problem. He, Michael Norman, and I were working on understanding the nature of extragalactic supersonic gaseous jets. Using finite differencing techniques on a CRAY-1 supercomputer in Munich, we had simulated a variety of flows that satisfied the Euler equations of gas dynamics. Initially much of our understanding of the properties of the flow came from using standard 2D contour line plots. However, as we used finer and finer grids, the number of local maxima and minima in our flow field became so great that determining the global structure of the flow became very difficult and time consuming. By switching to color images we were able to solve this problem, because blue was low and red was high and the eye can immediately grasp the structure of the flow. The finite difference grid is mapped to the pixel space of the graphics screen, with the value of the appropriate physical variable (for example, density, pressure, and vorticity) at each pixel being mapped onto the color table. Winkler's simple idea is now widely used by many other workers in astrophysics and gas dynamics. It can be used in any problem that simulates a two-dimensional field quantity.

Color Plates 5 and 6 are an example of the use of this technique on a recent calculation that Michael Norman did at NCSA, in collaboration with Dean Sumi and me. The computation is a planar symmetric (2D) supersonic jet of gas flowing into a two-phase atmosphere. The figure represents one physical variable, density, at two moments in the evolution. The Euler equations are finite differenced (time explicit with monotonic advection) on a grid of 250,000 square or rectangular zones. This computation can just be contained in the memory of a four million word CRAY X-MP/24. The photo is from a SUN-2 color graphics workstation with a resolution of 640 x 480 x 8 bit display. The number of pixels on the screen is about the same as the number of grid zones in the calculation. This illustrates why as you use more powerful supercomputers you will need higher resolution graphics screens to see the results. The 8 bits of color depth yield a possible 256 colors at each pixel. Blue is the lowest density gas and red is the highest, in spectral order.

The simulation starts with the two atmospheres at rest and jet material starting to flow in from the left. With time the jet bores its way across the grid to the position seen in Color Plate 5. From this one image you can see that this jet develops a nonlinear instability as it passes from the higher density atmosphere (yellow-green) into the lower density atmosphere (blue). The jet itself is red, which means it is considerably denser than the surrounding atmosphere (the jet is 10 times denser than the yellow-green atmosphere and 100 times denser than the blue). You

can read out quantitative information at each pixel by use of the mouse and screen cursor. To understand the time development of the instability, you must see the movie in which each frame is an image, like Color Plate 6. At each moment in the evolution you need to look at several images, each of which represents a different physical variable (for example, density, pressure, velocity). With this approach, you can read out the physics in the solution which is otherwise represented by a quarter of a million numbers per variable per time step.

A key point to realize is that, as you scan this image, your eye-brain system is doing the pattern recognition that is identifying the jet, its structure, the interaction with the atmosphere, and other factors. What originally comes out of the supercomputer is a serial stream of numbers representing the values of density in the two-dimensional array. The graphics workstation converts those numbers to a sequential series of colors: green, green, green, aquamarine, blue, blue, blue. . . . It is only when the image enters your eye-brain system, as it is doing right now as you look at Color Plates 5 and 6, that the data are converted into the structures you pick out and analyze. This is why imaging is so critical to understanding the results of supercomputer runs. Imaging allows for a direct coupling of the supercomputer output into the eye-brain system, which brings our heuristic, intuitive, and associative memory capabilities into play.

Single images, such as those in Color Plates 5 and 6, illustrate the **spatial complexity** that is routinely generated by today's supercomputer simulations. However, for continuum field simulations, such as gas, fluid, or solid dynamics, magnetohydrodynamics, plasma physics, and radiation, the **dynamics or time complexity** is just as critical to the understanding of the science. The partial differential equations that encode the laws of physics state that the time derivatives locally are driven by spatial derivatives and combinations of field variables locally. The **rate of change of field variables** will vary widely in space and time for spatially complex problems.

Color Plates 7, 8, 9, and 10 show an example of the density field generated by the head-on collision of two neutron stars, recently computed at NCSA by Charles Evans, a postdoctoral fellow at the University of Illinois (now at CalTech). The four frames from the movie show 1) the initial state with the two stars at rest, 2) the end of the period of coherent free fall, 3) the initial fusion of the two stars, and 4) the late time in which the inner core is phase decoupled from the outer atmosphere. Once again you see a transition from an ordered initial state to a spatially complex final state, such as happened with the supersonic jet (see Color Plates 5 and 6). Although you can infer from a series of still images, such as Color Plates 7-10, that the rates of change vary with time, it is only by watching the dynamics as a movie that the human can couple his or her keen senses of time variation, causality, and intuition into the computation.

At NCSA we are pursuing several technological dynamics. The traditional methods are to generate video tape or film movies of certain physical variables. Although it is very useful, there are a number of drawbacks to this approach. First, it prevents the user from seeing the images quickly, because he or she must await the outcome of the production facility and then possess the specialized playback equipment. Second, the production process is usually so costly and time consuming that movies are made of only one or two key variables rather than all variables. Third, NTSC video or 16 mm film, the standard options available, are low resolution output media compared to the resolution obtained today by algorithms such as PPM (see the articles by Woodward and by Woodward and Porter in this volume). Although HDTV and 35 mm film can be used in principle, very few users have playback facilities.

Therefore, several groups nationally are developing "digital movie" capabilities. The idea is for the user to create color or gray scale images of the variables and dump them to storage disks on a graphics workstation. The user then selects the variable he or she wishes to see the dynamics of and loads the frame sequence into either fast disks or a large memory bank. The images are then sent to the screen at many frames a second. Two current implementations of this idea are on SUN-3 workstations at NCSA and on Karl-Heinz Winkler's Ultra Speed Graphics project at Los Alamos. The SUN-3

with 16 megabytes of memory can hold up to 50 images (512 x 512 x 8 bit) and show them at about seven frames a second for a total movie of 7 seconds. The Los Alamos Ultra Speed Graphics project is built around Gould equipment. The real-time disks hold some 3200 images (512 x 512 x 8 bit) that can be shot to the screen at 30 frames per second for a 100-second movie. The reality of the dynamics is stunning on Winkler's machine. Our plan at NCSA is for remote users to rough out the dynamics on their workstations and then come to NCSA for intensive high resolution sessions on a system (described below and shown in figure 3) functionally similar to the Los Alamos system. In addition, both the Los Alamos and NCSA implementations are planned to include image-processing capabilities so that a user will be able to zoom on details, combine images, and do FFTs on the fly. This is a very new technology that I believe has the promise of radically altering our ability to gain understanding from supercomputer simulations.

The exponential growth rate in supercomputer capacity and memory over the next few years will lead inevitably to full three-spatial-dimensional dynamics as commonplace as two-spatial-dimensional runs are today (Color Plates 2 and 3). Therefore, technology for **visualizing four-dimensional space times** (time + 3-space) must be developed today. Again there is a spectrum of tools that can be used which I will briefly describe.

Adding specialized graphics processors to workstations allows the user to fly around remarkably complex three-dimensional (3D) surfaces or wireframes. The Silicon Graphics Iris workstation developed by James Clark, who attended our meeting, is probably the best known of these devices. Color Plate 11 is a still image taken off the Silicon Graphics Iris workstation at NCSA. It shows a myoglobin molecule on which Peter Wolynes and Atsuo Kuki, University of Illinois chemistry researchers, have been calculating quantum electron transfers on the NCSA CRAY X-MP. They have used Monte Carlo simulations of the path integral formulation to calculate the quantum tunneling of electrons from one side of the molecule to the other. The CRAY X-MP calculated more than 800,000 such paths. A few representative paths are shown in yellow, orange, and green. Most of the time the electrons hover around one side of the molecule or the other. Occasionally they tunnel between the two sides. The 3D atomic bonds of the molecule (blue and magenta) and the location of the paths are downloaded into a standard chemistry community molecular software (Midas) on the Silicon Graphics. The user then can rotate and zoom the computed results in real-time with a joy stick, completely disconnected from the supercomputer.

This technique works well with 3D lines or surfaces, but many simulations produce 3D volume information that the user needs to "look into." To investigate this problem, NCSA has set up a joint study with IBM to use their IBM 7350 imaging workstation. Color Plates 12 and 13 show 3D cloud simulations computed by atmospheric scientist Robert Wilhelmson and colleagues on the NCSA CRAY X-MP. The 3D volume data are then downloaded to the IBM 7350 through an IBM mainframe. Each cube of the finite differenced model can be colored according to the physical variable represented. Thus, one has a "solid" with internal colors. With this workstation you can use a menu and a mouse to split open this "solid" to see what is inside. Or you can cut out a chunk of it to look inside. This is a "volume" approach for looking at three-dimensional data as opposed to a "surface" approach, as used on the Silicon Graphics. The specific problem and the user's desires determine which approach is more useful.

Finally, there are approaches so sophisticated that no workstation today is powerful enough to do the imaging. Instead, the CRAY or a mini-supercomputer is needed. Color Plate 14 shows a large cube of the universe at two moments, the initial and final states, in a simulation of the evolution of the density of dark matter in the universe. This simulation was carried out by Joan Centrella, a cosmologist at Drexel University. She did this work on a CRAY X-MP at Digital Productions; the imaging was created using DP software by Craig Upson, now a research scientist at NCSA, and his team. The colored surfaces represent isodensity contours in three dimensions. Three nested contour levels are shown with blue, green, and red representing increasing levels of density. The CRAY took

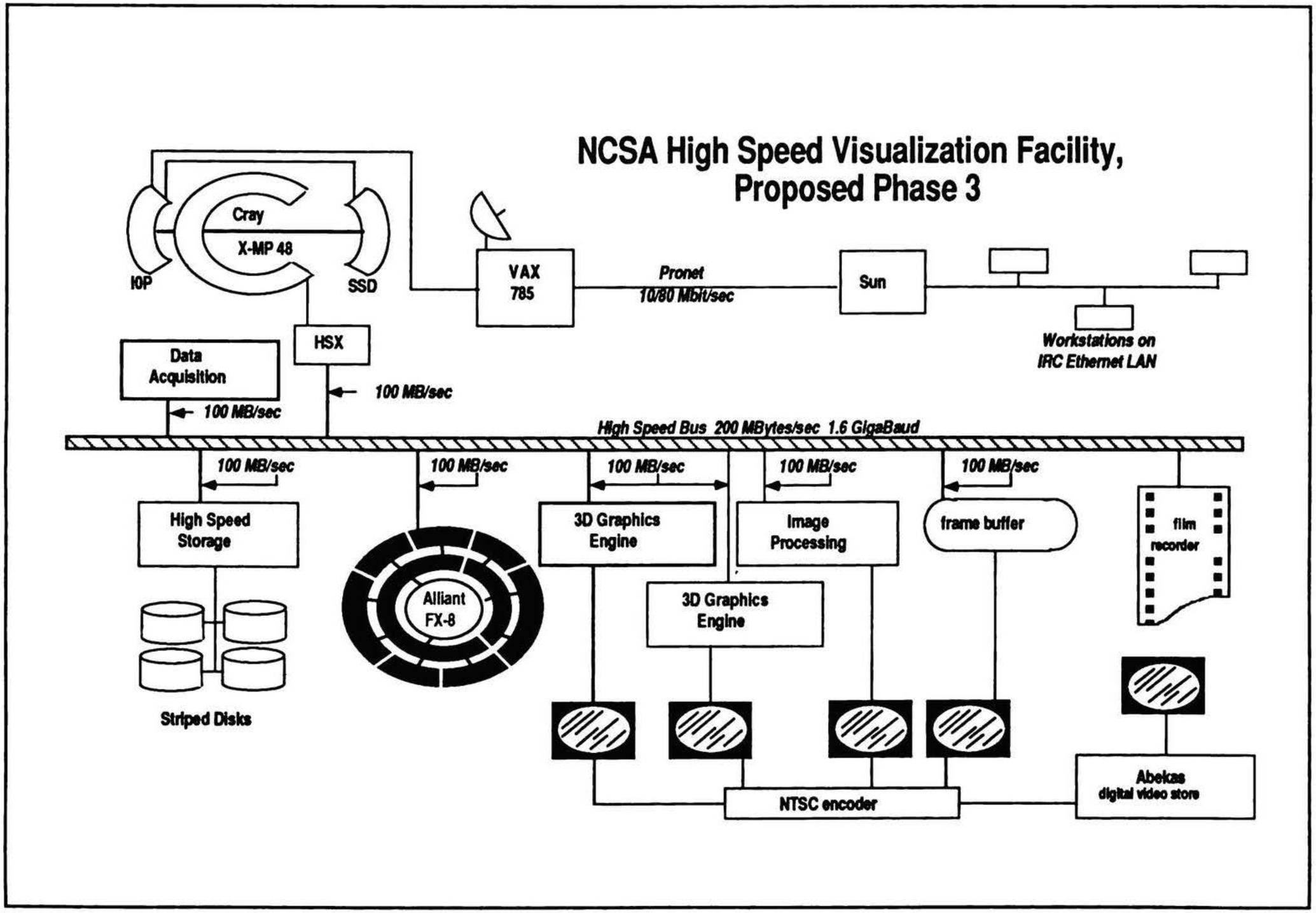


Figure 3: This represents a schematic diagram of Phase III of the NCSA Visualization Program led by Nancy St. John, Craig Upson, and Bob Haber. Graphic by Colleen Bannon, staff artist at NCSA.

ten minutes to find the three surfaces, polygonize them, and calculate the image using transparent level surfaces and lighting sources. This technique reveals that the dark matter fragments into an intricate structure of filaments, leaving large voids in the volume. Several thousand 3D images were computed, in which the viewer flies around the volume as the evolution proceeds. The imaging project required about 100 hours of CRAY time, while the evolution simulation itself took less than one hour. This example may be representative of what we may come to expect in four-dimensional simulations in the future—the computer time to explore the solution visually may greatly exceed the time to compute the solution.

These kinds of visualization technologies will be inadequate to deal with the output of the early 1990's supercomputers, which will have 500 times the fast memory and 50 to 100 times the speed of a CRAY-1. So at NCSA we are developing a visualization task force, led by Nancy St. John, which will be exploring the latest technologies that give the user new resources to deal with the output of his or her supercomputer solutions.

Figure 3 is a schematic diagram of an NCSA-proposed visualization system, which is related to Winkler's Los Alamos System. Robert Haber is the primary architect of the NCSA system. One way to think of it is that this is what a graphics workstation should look like in ten years. It takes seriously that we are moving from a number (bytes) as a basic unit of data to an image (megabytes) as the basic unit. This shift in the data unit implies a shift in computational system architecture toward a system such as that shown in figure 3. Such a system will require a bus throughput of billions of bits per second. There will be striped disks or solid state disks with capacities of gigabytes (thousands of megabyte images). The attached minisupercomputer will run sophisticated 3D animation software. Frame buffers with 2048 x 2048 x 24 bits will be used to show the animation in real time or to hold the images for high-resolution video or film recorders. Special-purpose hardware will render up to 100,000 polygons per second, much faster than a supercomputer. Powerful stand-alone image processing devices will go far beyond the volume-imaging capabilities of today's devices.

A single user will have access to this imaging power for computational science, much as a single user in the prime-focus cage of the 200" Mt. Palomar telescope has access to an awesome amount of imaging power in astronomy. By the end of 1987 such a device could be built out of commercially available components, so that it would be easily replicable. These powerful visualization tools will alter the nature of computational science.

Software Tools

The last area of technology I will discuss is that of software tools. NCSA has initiated several software projects to help users become more productive. Many other researchers in other institutions are promoting other approaches. I will use ours to illustrate the kinds of user difficulties that software toolscan solve.

1) As discussed previously, the enormous growth in the installed base of personal computers has driven a rapid development in software for these computers. The user interface for such computers is now much more advanced than it is for supercomputers. Market forces are likely to make this software gap grow with time. Therefore, a rational strategy is to integrate the supercomputer into the personal computer software so that the user has the speed advantage of the supercomputer and the ease of use of the PC.

Figure 4 illustrates this strategy with the Apple Macintosh. The Macintosh is a special personal computer because Apple imposed strict internal standards on third-party software authors. This allows you to easily "cut and paste" from one application to another. Using the Mac Switcher, a commercial software product, you can open up simultaneously four types of tools: a word processor (e.g., MacWrite), a drafting package (e.g., MacDraw), a spreadsheet and graphing package (e.g., Cricket Graph by Cricket Software, Inc.), and a terminal emulator (e.g., VersaTerm-PRO).

Figures 5-8 illustrate this approach with a series of screen dumps from the Apple Macintosh. Suppose you are calibrating an ODE

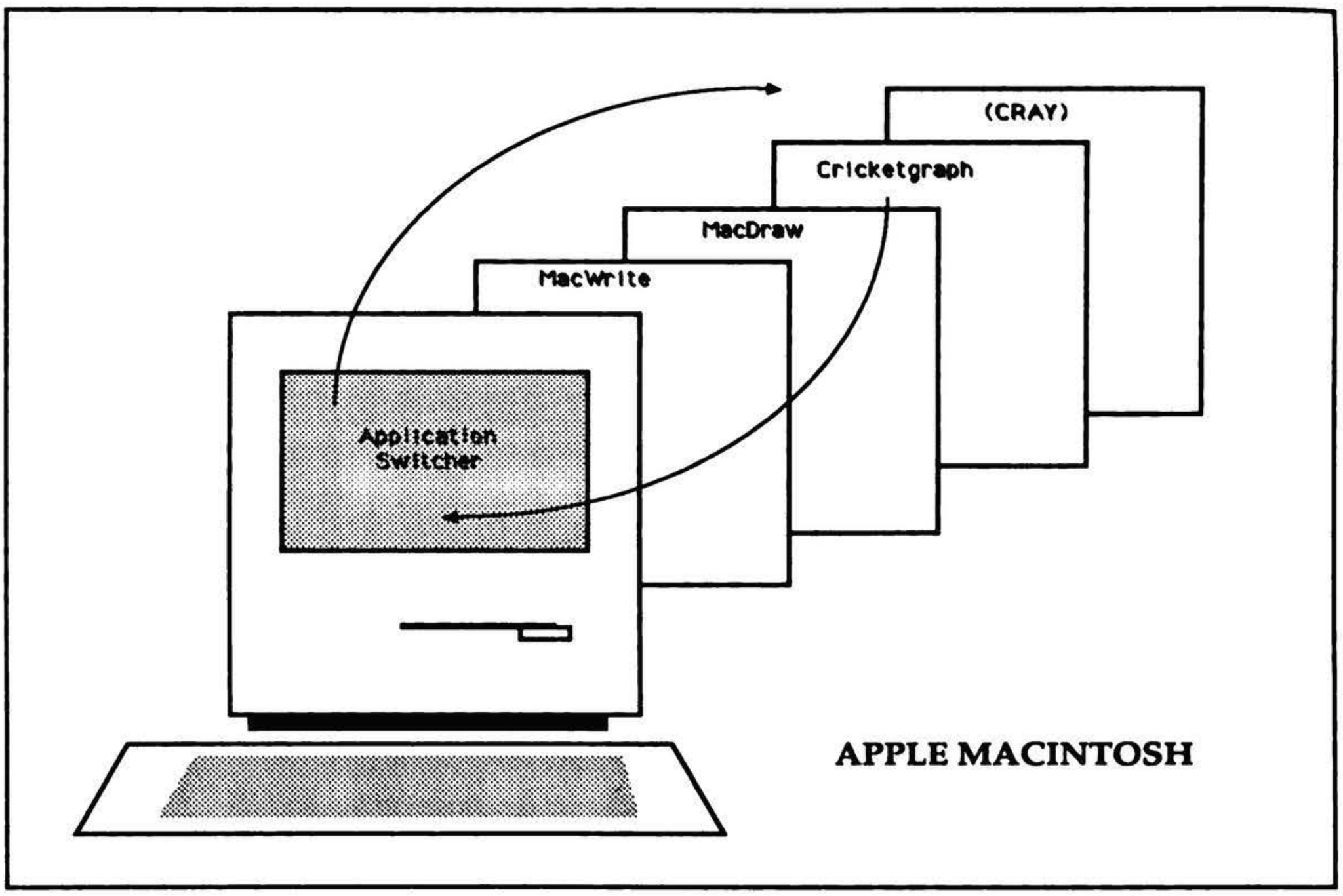
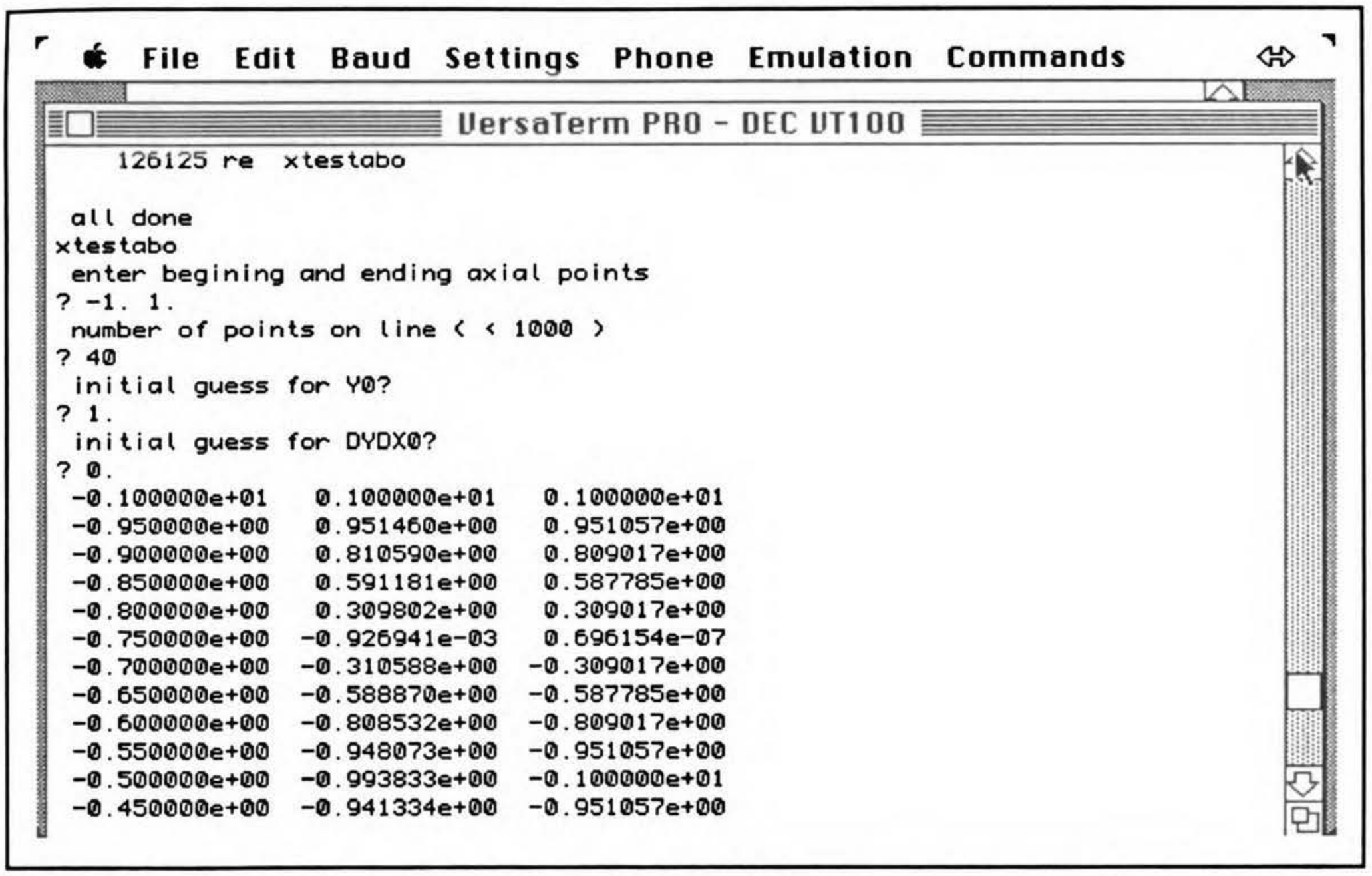


Figure 4. Current scientific working environment on an Apple Macintosh at NCSA. The application Switcher allows you to move easily (switch) between operating on the CRAY and using commercial applications on the Mac. Graphic by NCSA staff Virginia Worthington and Ronnie Johnston.

solver on the supercomputer while using commercial software on a Macintosh. Typically, you would use a DEC VT100 emulator on the Mac, such as VersaTerm-PRO, connected through a modem to the supercomputer. Figure 5 shows a dump of numbers from the CRAY to the screen; the first column is the value of x, the second is the integrated value for y, and the third column is the exact value. These numbers are "dead" if you are at a traditional dumb terminal. If you want a graph, you have to write a separate program to do that and wait until the central output device creates the hard copy graph. Then you have to walk over to the central output location to get it. The hard copy can take hours or days to generate.

With the Macintosh, you can simply highlight the columns of numbers with the mouse, switch over to Cricket Graph, and electronically copy the figures into the spreadsheet (figure 6). To get a graph, simply pull down the graph menu, select the style of graph, and instantly you have a graph of the two y values vs x (figure 7). Relabeling the graph is quick and easy. Because the numbers are "live" on the Mac, you can perform calculations with them on the local PC, (i.e., taking the difference of the two y values to see how good the solver is) and graph that result (figure 8), all without having to go back to the remote supercomputer. In addition, you can write up a report for your electronic notebook with MacWrite, combine the two figures by using MacDraw, and then transfer the combined figures via the Mac Clipboard into the MacWrite text. At the click of a mouse, you can have camera-ready hard copy from the Apple LaserWriter. Thus, for supercomputing a great advantage of personal computers over dumb terminals is that the entire scientific production cycle is optimized, not just the computation



Figures 5-8 provide an example of the Mac's actual working environment. The application is an ordinary differential equation solver running on the CRAY by a user sitting at an Apple Macintosh. This demonstration was provided by Mark Johnson, an NCSA User Consultant.

Figure 5. Interacting on the CRAY using the software package Versa Term-PRO. The Mac captures the session in memory.

itself. Furthermore, any user can afford to purchase this set of tools.

2) Software tools can hide much of the redundancy in the multiple computers confronting the user in a comprehensive computing system. If you are sitting at an IBM personal computer and using the NCSA supercomputer, you are using three operating systems: PC-DOS on the IBM PC/AT, VMS on the DEC VAX front end, CTSS on the CRAY X-MP. You are also using the Common File System software on the IBM mainframe (which itself is running MVS). Each operating system has different commands for such operations as listing file directories, copying files, and sending file contents to the screen.

At a minimum, the computer used most directly, the personal computer, should be able to do all these common functions on the system computers, using only the commands on the PC. Figure 9 shows an example of a screen on the IBM PC when using CTALK. This is an experimental environment being developed at NCSA by the Workstation Development Team, led by Larry White, and UIUC computer science students Swami Natarajan and Aaron Contorer. Each computer is activated by typing a disk command ["c:" for the local PC, "k:" for the CRAY, "v:" for the VAX, " m:" for the mass store]. Thus, different computers appear as different IBM PC disk drives. The command "k:" automatically issues a series of commands that log you on remotely to the CRAY X-MP. From then on, the normal PC-DOS commands send out their corresponding commands to the CRAY. For instance, to list files on *any* computer, you enter "DIR," the IBM PC-DOS command for directory. This command sends out "files" under CTSS on the CRAY or "list" under CFS. To copy a file from one machine to another, enter the DOS command "copy." With this

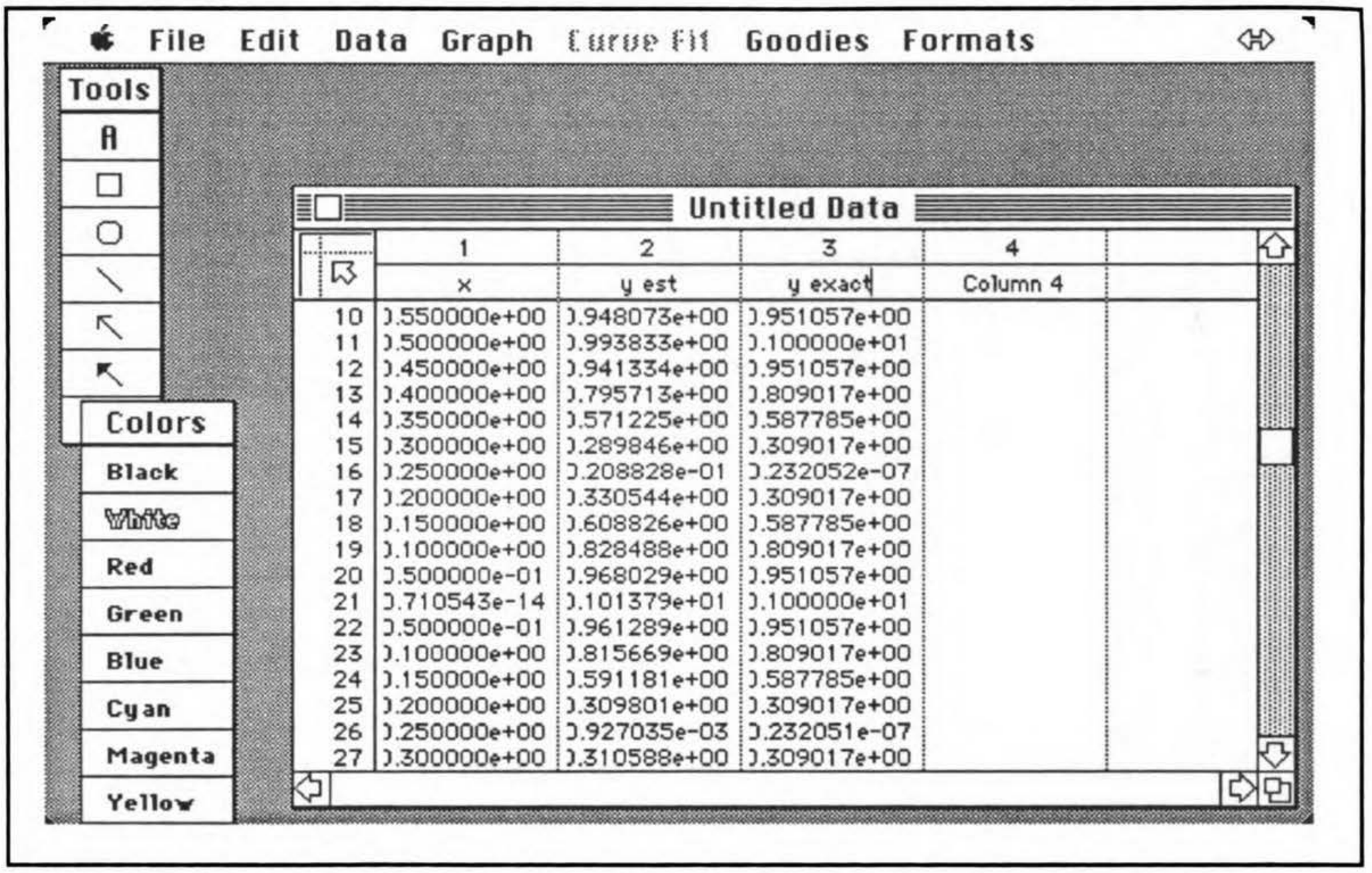


Figure 6. The columns of numbers dumped from the CRAY to Versa Term-PRO have been transferred to a spreadsheet in Cricket Graph.

software, you only need to know the operating system of the computer on your desk. Life is too short to have to learn more than one operating system, if software tools can hide the others.

3) The previous two ideas, in principle, can be integrated into what we call the *Macintosh CRAY Finder*. This proposed software interface, which NCSA hopes to develop, would "iconize" the entire computing network shown in figure 1 and provide a uniform user interface for commands mentioned previously. Because this software has not yet been written, we can only describe the functions that we hope to include in it. However, the Mac itself can be used to simulate what the screen on a future Mac would look like if the software were available today.

In the proposed Mac Finder system, each computer would be represented by an icon (figures 10 and 11). In fact, the network itself would be an icon. Double clicking on an icon, such as the DEC VAX in Figure 10, would automatically display the files on that computer. To move a file from the VAX to the CRAY, for instance, with the mouse you would highlight the file in the VMS window and then drag it over to the CRAY. When the CRAY icon turns black, the file transfer is complete. This is the iconized version of CTALK for the IBM PC. Here, as in CTALK, the machines are symbolized as disk drives on the Mac. Double clicking on an icon actually logs you on to that machine.

Figure 11 shows that in the case of CTSS on the CRAY you can start up multiple sessions. CTSS allows for up to five simultaneous sessions. Here three of them are symbolized by *A*, *B*, and *C* and their respective icons in the CTSS window. Double clicking on those icons can open up to five windows on the Mac, each running a separate CTSS session. Thus the Mac would be acting as a multitasking device, even though only one of its sessions is "live" at any one moment. Again, the basic idea behind such a software tool is to minimize the amount of time that a user spends on getting around in the computational network and to maximize productivity for doing science.

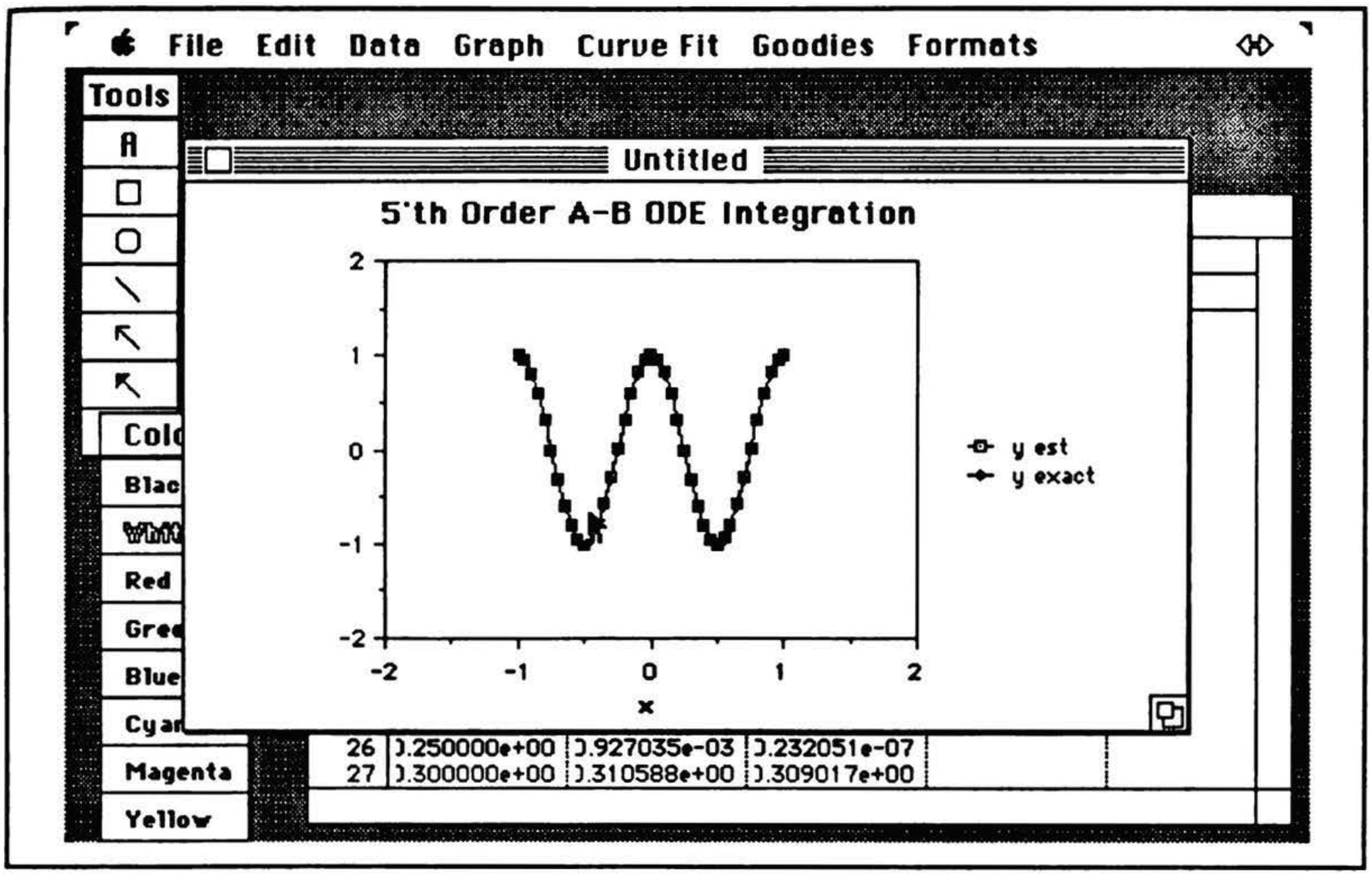


Figure 7. Cricket Graph can instantly plot the columns of numbers by selecting from the graph pull-down menu.

4) An example of a useful graphics software tool for a remote user is what we call *Ove windows*. When you are running a multidimensional simulation on a supercomputer, it is critically important to be able to spot-check the full solution, both as the calculation proceeds and afterward, when you initially peruse the space-time that has been computed. In the case of Einstein's equations of general relativity, for instance, there are more than ten variables of two or three space dimensions and time which can be inspected. The inspection process is not only important for understanding the physics, but also for improving the algorithm.

The screen of an IBM PC using a standard IBM EGA color graphics card and monitor, which was provided by IBM through its joint workstation study with NCSA, is shown in Color Plate 15. Each window represents one of the gravitational variables ("P") at a different time in the evolution. The simulation is that of a 3-torus universe in which two space dimensions have variation with time. The colors go through sixteen levels, from high to low values of the gravitational variable P. Roger Ove, a postdoctoral fellow with me, used assembly language on the IBM PC to develop this windowing environment and to make it fast enough to be practical. Even at the low resolution per window, the major features of the evolution of the gravity waves in the universe can be seen, with each window representing an entire universe at that moment. The click of a mouse changes the windows to represent all ten general relativity variables at the same moment or enlarges one window to full screen size. With color image compression and decompression, it becomes practical to do such color imaging remotely over NSFnet.

Our workstation development staff is considering implementing Ove windows as FORTRAN-callable subroutines so the user can directly download the state of his calculation from the supercomputer to his personal computer in real time. This single software tool increases enormously the ability of a remote user to run

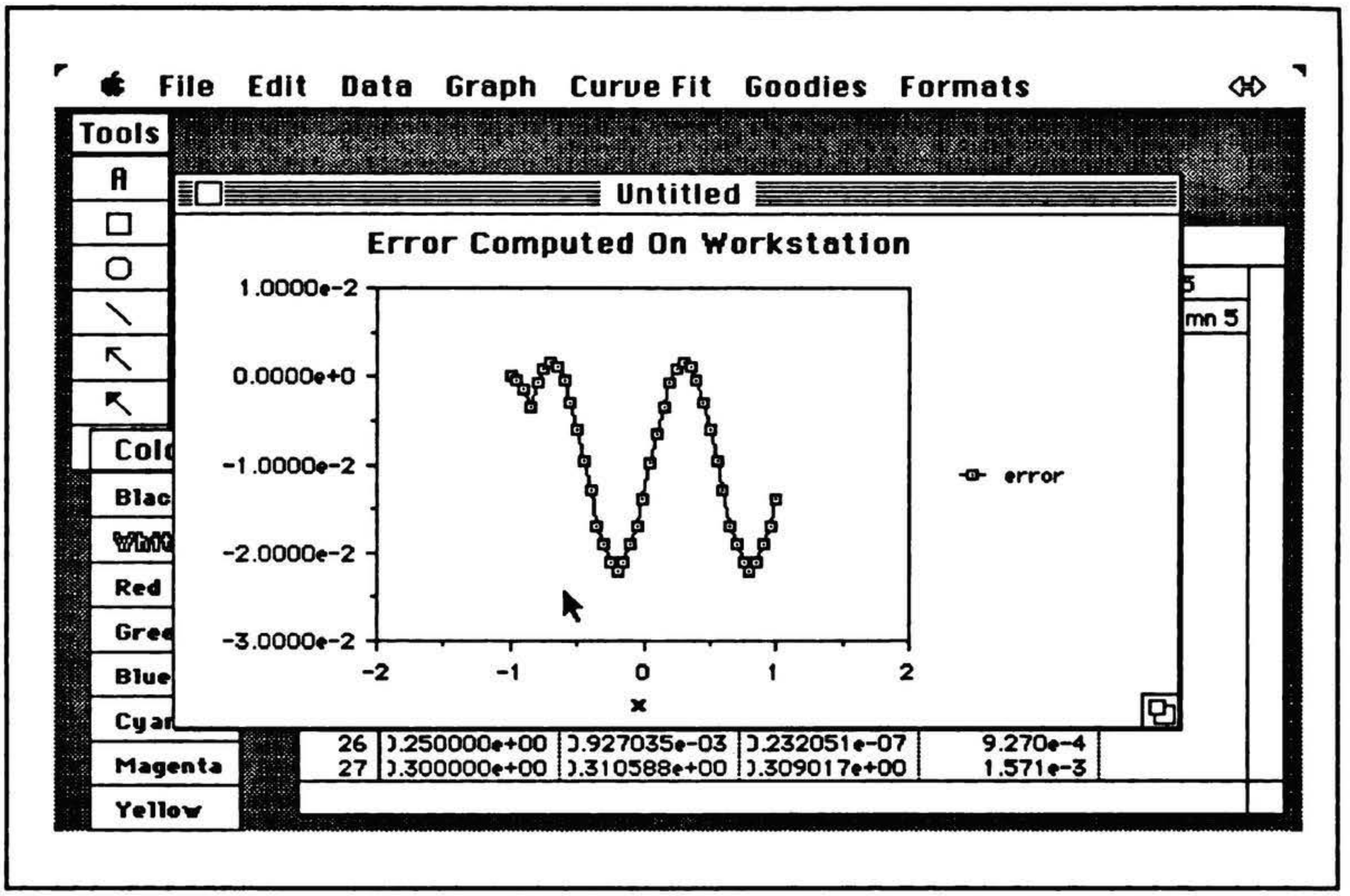


Figure 8. The Cricket Graph spreadsheet can take the difference of two columns of numbers and graph that result without having to go back to the CRAY.

complex simulations effectively on a supercomputer. It is important to note that this tool was developed by a user to solve his own real problem. A national center's role is to take a specific user's raw source code and turn it into a generic tool for many other users, regardless of their fields of science.

5) The previous four examples have been software tools for personal computers because, as of 1986, the installed base of Apple Macintosh and IBM personal computers is so large that it is likely that a remote user will have one of these devices as a port of entry into the computational network.

However, even more powerful windowed UNIX workstations such as SUN Microsystems' SUN-3 are beginning to become widely available. The screen of a SUN-3 workstation used by Steve Christensen, an NCSA research scientist in quantum gravity, is shown in Color Plate 16. The icons in the upper part of the screen, which Steve has created, represent the CRAY, the VAX, other SUNs. Each of these icons is a live session to that machine and simply clicking on the icon opens a window into that session [much as we envisioned for the Macintosh CRAY Finder in 3) above]. The SUN-3, in fact, supports up to sixteen windows, each simultaneously executing on the SUN-3. Different windows for various physical variables represented as color images can be run, such as those shown on the screen in Color Plate 16, or independent calculations can be performed on the workstation in C or FORTRAN.

Once the supercomputer itself has a UNIX operating system and the entire network has a uniform protocol (such as TCP/IP), then the unification of the workstation with the supercomputer will be straightforward. Software tools such as Remote Procedure Calls or the Network File System will allow us to come close to achieving our dream of hiding the entire computational network inside the desktop computer being used by the researcher.

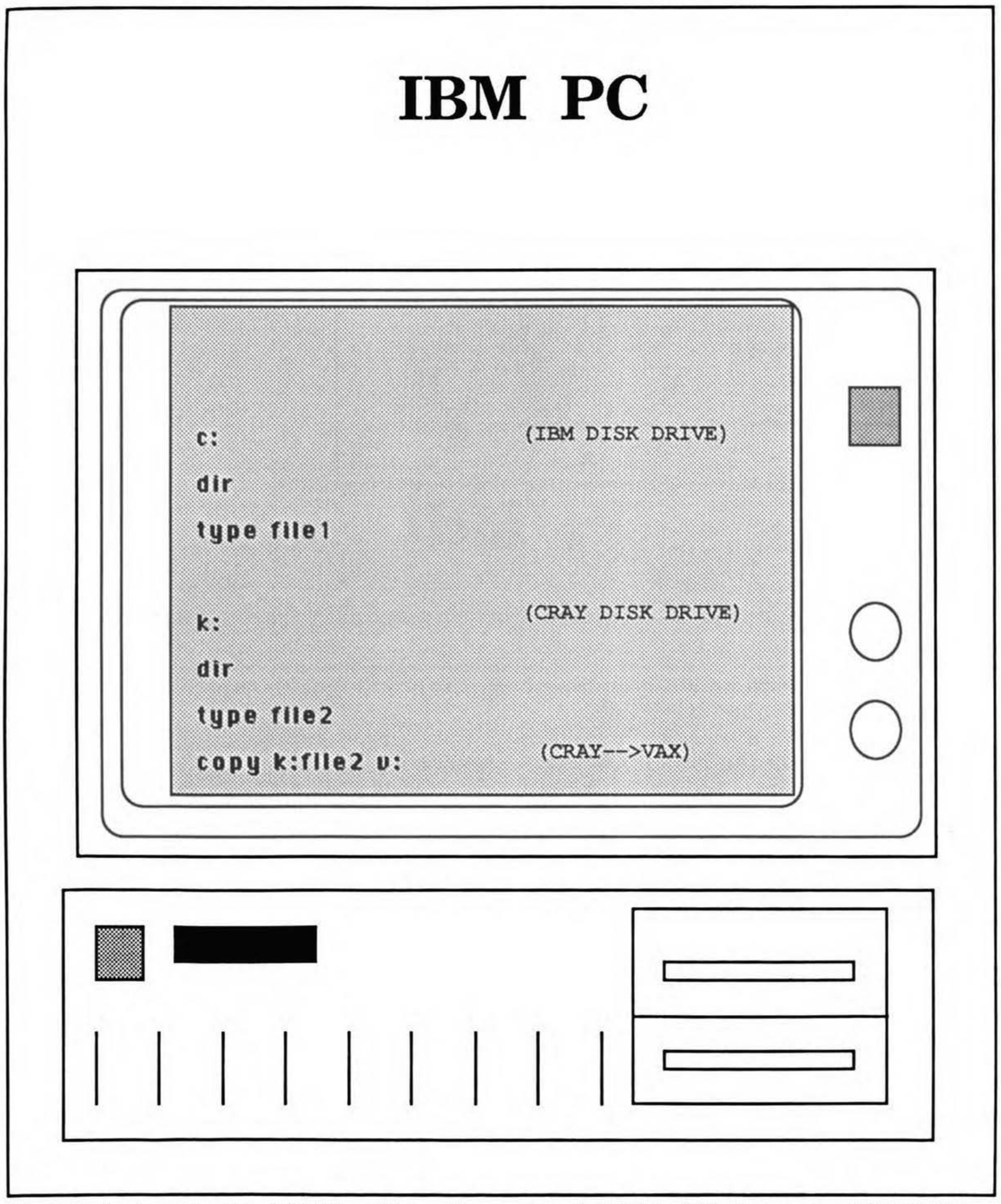
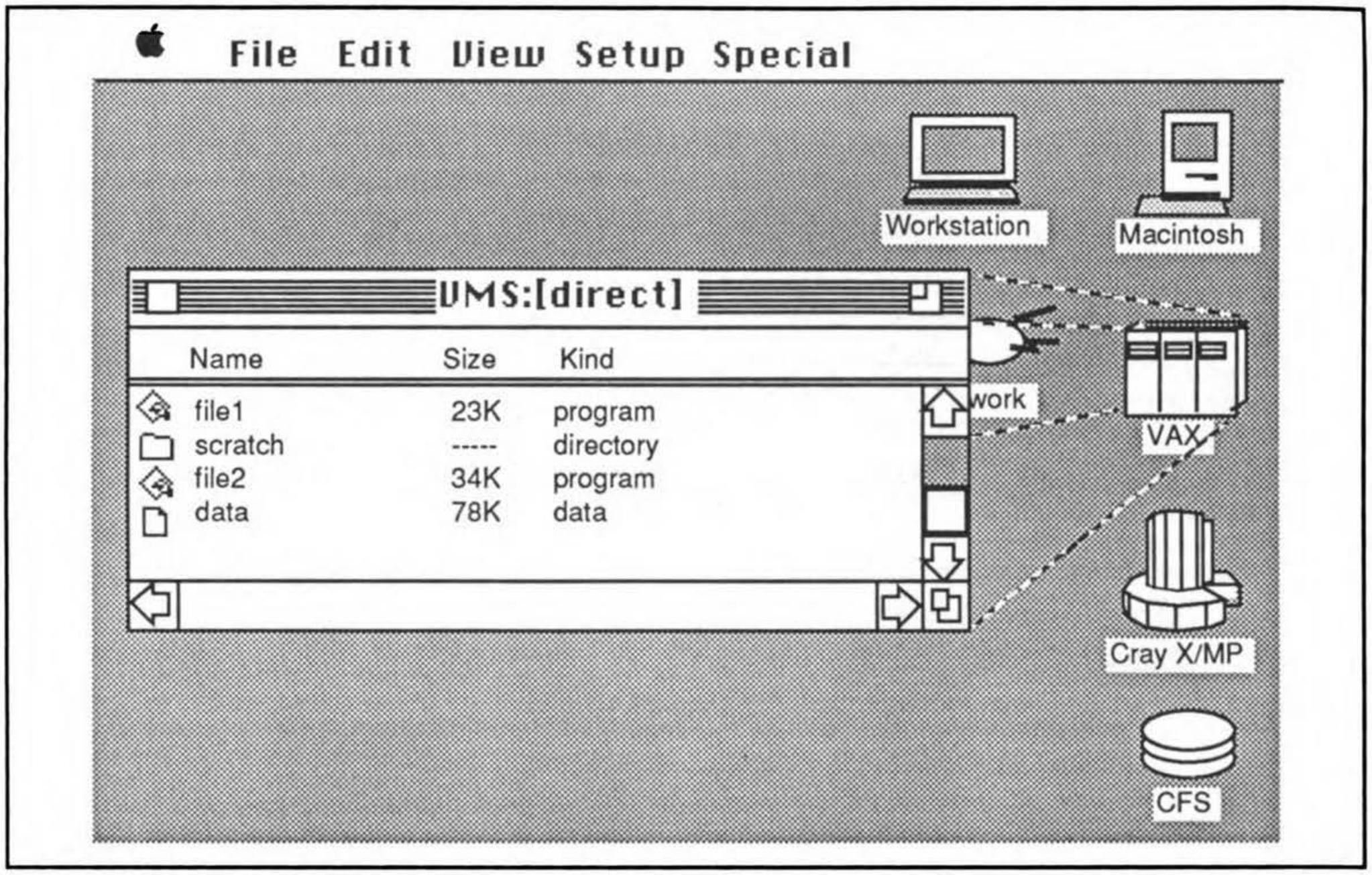


Figure 9. A schematic of the CTALK working environment on the IBM PC/AT. You can type PC-DOS like commands to different computers just by specifying which disk drive you are on (e.g., c: for the IBM PC, k: for the CRAY, v: for the front end VAX, and m: for the mass store). This software environment is being developed by Swami Natarajan and Aaron Contorer, computer science students working with the NCSA Workstation Development Group led by Larry White. Graphic by Virginia Worthington and Ronnie Johnston.



Figures 10 and 11 provide an Apple Macintosh simulation of a possible future interface to the CRAY which we call the CRAY Finder. The proposed NCSA project extends the Mac Finder desktop environment to the entire distributed computing environment. Graphics by Brand Fortner.
Figure 10. The CRAY Finder providing access to the front-end DEC VAX computer.

Methodology

The previous section on technology constitutes the bulk of this article because we are still at a very early stage in the development of computational science. In such an early stage, the tools themselves are the focus of a great deal of attention by researchers in the field. As the field matures and the tools become more invisible, I believe that there will be a greater emphasis on the way in which this new technology changes our methodology of science by allowing us to attack problems of a complexity heretofore unthinkable. The methodology of analytic science will be inappropriate to the numerically generated solutions of computational science. However, the two will overlap, because we are still dealing with mathematical solutions of the laws of nature.

This section will be fairly brief because I have touched on the change in methodology in more depth in an article in *Science* (26 April 1985, 228, 403). The fundamental problem of computational science is to understand the mathematical solutions of partial differential equations. I think there is a coherent way of approaching this subject, assuming you have a powerful graphics workstation.

I will illustrate the basic approach with some results of John Hawley, a postdoctoral fellow at the California Institute of Technology, and myself on gas spiraling into a black hole. This is meant to illustrate any complex solution of a system of partial differential equations. The basic setup of the problem is that there is an initial condition and, throughout the evolution, fixed boundary conditions. In our case, the initial condition is a spherical grid on which low pressure gas is placed that is freely falling toward the hole. This is matched at the outer boundary (a radius fifty times greater than the radius of the hole) with a continual influx of

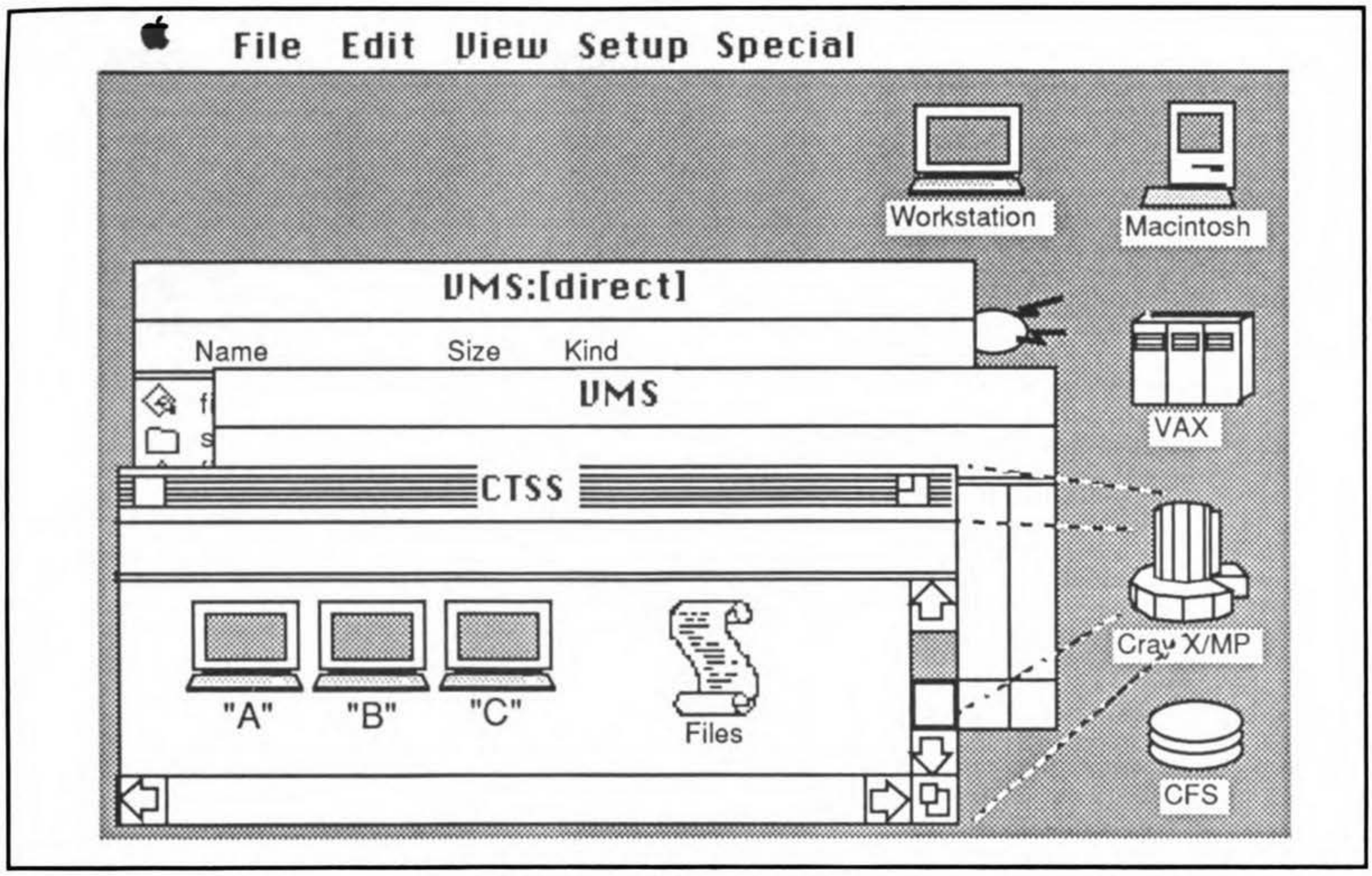


Figure 11. The Cray Finder providing access to the CRAY. Files will be moved from one computer to another by selecting the file with the mouse and dragging it to the icon of the other computer.

rotating matter with or without pressure (i.e., either hot or cold, relativistically speaking).

With these well-posed initial and boundary conditions, the computer then integrates step by step the Einstein equations for relativistic hydrodynamics. Color Plate 17 shows the gas density at the time in the evolution when the infalling rotating matter is just beginning to approach the hole. We are again using visualization techniques to represent a function (density) of two space variables (the vertical axis is an axis of rotational symmetry) at a moment in time. The value of the density at each point in space is denoted by a color that ranges from blue to red in spectral order as previously discussed in the imaging section. Color Plate 18 shows the same density function at a later moment. Comparing Color Plates 17 and 18 shows how the density distribution has changed in time with the creation of sharp shock waves forming a large figure eight, as well as inner shock waves near the hole.

To complete the space-time description of this computation, you need to give a similar time evolution visually for the pressure function and the three components of velocity. In their research project, the background metric or gravitational field was held fixed. If it were allowed to vary, then at least another dozen gravitational variables would need to be collected visually.

You are visually encoding the global structure of the solution, both spatially and as it changes in time. The pattern recognition capacity of the human mind is becoming an integral part of the mathematical representation of the solutions to the partial differential equations.

In summary, you mathematically represent the space-time by n movies of the n fundamental variables in the problem. These variables solve the coupled nonlinear partial differential equations by interacting with each other in space and in time. With the movie, the spatial and temporal variation in the time rate of change of each of the variables can be directly experienced. Therefore to study the mathematical structure of a given space-time, subject to fixed initial and boundary conditions, a large number of movies

running simultaneously must be viewed. This will be possible with the technologies of the multiple-windowed workstations described above. A researcher can then interrogate the space-time mathematically to discover quantitative as well as qualitative relationships among the variables.

However, simply computing one space-time solution is not enough. As the initial or boundary conditions are varied, a sequence of space-times is obtained. For instance, Color Plate 19 shows the same density function as Color Plate 18, but in a case in which the boundary conditions have been altered to give the gas a high pressure as it enters the grid rather than the low pressure boundary condition used in Color Plates 17 and 18. The result of the high pressure boundary condition is to slow the infall of the gas so that very few shock waves form and a much narrower evacuated funnel occurs along the axis. The density function has a quite different structure in this space-time in comparison to the density function of the space-time in Color Plates 17 and 18. This is so, even though the initial data are the same and only one parameter has been altered in the boundary condition. This systematic change in the global structure of the solutions is part of the answer that we seek in performing the computation.

To quantify mathematically this difference in solutions, imagine setting up a multidimensional solution space. The axes of this solution space are the independent parameters that specify the boundary and initial conditions. These simple parameters help us order an otherwise extremely complex flow field. Each point in the solution space represents an entire space-time solution of all the physical variables discussed in the paragraph above. As the parameters specifying the initial or boundary conditions are slowly varied, we move from point to point in the solution space. The space-times each point represents will slowly change as well. After enough movement through solution space, the space-time has become qualitatively different from the space-time we started with. For instance, Color Plates 17 and 18 would be part of a space-time in one regime of solution space which is characterized by shock-dominated solutions, whereas Color Plate 19 would be part of space-time in another regime of solution space characterized by smoother flows. By this method of concentrating on the qualitative changes of the space-time solutions as our simple parameters vary, we gain intuitive knowledge that generalizes the behavior of our particular computed solutions shown in Color Plates 17-19.

This new mathematical methodology for computational science is only beginning to emerge as supercomputers become fast enough to allow us to sample many space-time solutions and as graphics workstations become powerful enough to allow us to set up visually a taxonomy of the space-time solutions. However, the prospect of doing this routinely is very compelling and will, I think, lead to enormously powerful insights into nature which have escaped either experimental observational science or analytical theoretical science.

My last point, which has been emphasized a great deal at this workshop, has to do with the nature of algorithms. We must always remember that we are *not* solving partial differential equations themselves but only finite representations of those differential equations. Thus, changing a single line of code can produce a qualitative difference in the finite solution we obtain from the computer. Therefore, there exists a parallel and separate science of algorithm development which seeks the most accurate and efficient methods for solving partial differential equations or other mathematical equations. Historically, it has been so difficult to develop a code to solve a complicated problem that one did not bother detailing the nature of the code but merely published the results of the science. This cannot be adequate, however, since the so-called results depend critically on the nature of the algorithm used in the code.

Therefore, I see in a number of fields a new standard arising that requires the scientist to document his or her algorithm as well as the scientific results obtained from using that algorithm. I believe this is a very positive development because it will spur interaction among computer scientists, mathematicians, and scientific researchers to perfect algorithms for a wide variety of scientific fields. I believe that after a decade or so of intensive focus, algorithms will once again recede into the background because

there will become universally agreed-upon algorithms that are well documented and understood in a given subfield of science.

Again, I see the coupling of the immense speed of today's supercomputers with the software tools on the workstations as being critical for allowing this exploration of algorithm space. The focusing on algorithms is another sign of the relative youth of computational science. In the century after Newton invented the calculus, a great number of scientific paperswere published that developed and verified new analytical methodologies that we all take for granted today. The fact that we are in the early years of our field lends an excitement to the enterprise.

Sociology

The last topic is sociology, or how human beings do science. I believe that we are seeing a new model emerge for the fundamental human unit for pursuing research in computational science. Traditionally, analytic science has been done in the "single warrior combat mode." This is possible because analytic techniques have been refined for three centuries so that an individual worker is capable of utilizing an entire array of analytic tools on a given problem. In computational science, by contrast, there are so many new technologies in hardware and software that experts in the use of such tools need to work with the scientist as part of a team. In addition, the power of the supercomputer allows a researcher to tackle problems so complicated that they span several subdisciplines in the normal way of thinking about the work. This may bring a number of scientists together on the team. We have seen similar teams develop in experimental work and have taken that for granted. However, because many of the practitioners of computational science come from the analytic world rather than the experimental world, they are more disturbed by this change in sociology. I think the disturbance or the change is very healthy and will lead to more rapidly paced progress.

There is another level of interdisciplinary teamwork that transcends the particular field of investigation. Because of the universal character of the computer each of us, regardless of field of study, must edit files, transfer files, learn vectorization, and use workstations. Until now, each of us had to spend a large proportion of our time learning how to use this technology on our own. We had to investigate the literature to discover what products were available, then try them out and wire them up ourselves, wade through manuals to figure out how to use the software, and so forth. However, there seems to me to be a major opportunity for the National Science Foundation supercomputing centers to alter this process fundamentally. The national centers should serve as a reservoir of expertise on all these issues, and they should provide national training centers where a researcher can go to get brought up to date on the technological side of our field in a short period of time. The centers have already done a great deal of investigation to determine which commercial pieces of hardware and software are most effective, and the centers have learned how to integrate these into a national network. If the centers can provide this training function, an enormous amount of time for individual researchers can be saved.

But beyond these technological considerations, computational communities are emerging that differ from the traditional departmental structure of a university. For instance, any investigator who is solving a quantum mechanical problem will be led to inherently probabilistic algorithms such as Monte Carlo. These investigators may come from such departments as physics, chemistry, material sciences, or biochemistry. However, they have a common algorithmic interest which, if shared, would again increase their productivity. Right now a good forum for this shared interest does not exist because most journals are defined along content lines rather than algorithmic lines. Again, the national centers have a unique opportunity to bring together in workshops and conferences researchers from all over the country who share the same basic algorithmic (and therefore technological) approaches to a problem.

Finally, new kinds of interactions will emerge among disciplines that heretofore have been very far removed. At NCSA we have a major interdisciplinary group bridging computer art and various sciences. The computer artist is an expert in the kind of color transfer functions in visuali-

zation that we have seen becoming critical for scientific researchers. In addition, the artist as an artist gains new images of artificial computer-generated realities from which to create new art works. The scientist is able to dig deeper into his or her results by working as a colleague with the computer artist. Both gain something in their joint professional endeavor, while aiding each other in ways that would not happen without the partnership. I believe this convergence of art and science will be repeated across many other disciplines as computational science develops.

Barriers need to be broken down not only inside universities, but also between universities and other centers for research, such as industrial or government laboratories. NCSA, for instance, has created an Industrial Supercomputing Program that is bringing researchers from up to ten major corporations on site as part of its interdisciplinary center. These researchers join in the active intellectual discussion of technology, algorithms, and science. They bring a real-world focus that is often lacking in the university, but gain direct involvement with long-term basic research, which often does not occur in industry. Similarly, NCSA has been working closely with a number of the government laboratories to transfer technology between them and the university community.

One way to consider these new interactions is in the model of an increase in scientific communication. In that regard, the national network, NSFnet, will have a profound sociological impact. Currently, colleagues meet at annual meetings or jointly work on papers in which the turnaround time is set by the U.S. mail. With a rich electronic network environment, I believe we will see a drastic reduction in the average time for interaction of colleagues. Besides being able to work jointly on research in real time, there will very likely develop bulletin boards for broad-ranging discussions of topics among many colleagues throughout the country. This development is already well advanced in computer science.

The final barrier that needs radical revision is that of publishing. These proceedings illustrate the state of the art in publishing computational science, in that color images are included in the volume. However, it is clear from the discussion above that we need to go far beyond what can be captured on a printed page. Presumably the technologies of laser disks will enable us to document our visual representation of space-times so that we can share a solution with a colleague who can investigate that solution on his or her own computer. Until we get to the point that we can build on each other's work the way researchers have been able to in analytic science, the pace of our field will remain somewhat slow. Publishing has always been a crucial part of this building process, and it is currently one of the critical bottlenecks to progress in our field. Many of the technologies discussed here will help break open the prospects for future publishing as colleagues begin to depend more and more on the electronic network rather than on the printed page for doing science.