Thoughts on Mathematica

by Brian Hayes

Sometime in the 1850's William Shanks, a British schoolkeeper and amateur mathematician, began calculating the digits of the decimal expansion of π. Twenty years later, when he published his results in the *Proceedings of the Royal Society,* he had gotten as far as 707 digits. Today, with the aid of the computer program Mathematica, performing the same calculation requires no more effort than typing the expression:

```
N[Pi,707].
```

In response to this command, the labor that occupied Shanks for two decades is completed in less than eight seconds. Furthermore, Mathematica produces the *correct* 707 digits. Shanks made an error in the 528th decimal place that spoiled all the rest of his work.

Mathematica does many other impressive tricks. Perhaps you have some urgent need to know the value of the hundred-millionth prime number. Evaluating the expression

```
Prime[100000000]
```

generates the answer—which happens to be 2,038,074,743—in well under a second. The program shows equal facility in identifying *non*-primes. According to legend, when Leonhard Euler heard of Pierre de Fermat's conjecture that all numbers of the form 2^2+1 are prime, he pondered a moment and then snapped back: "No, 2^2+1 is equal to 4,294,967,297, which has the prime factors 6,700,417 and 641." Equipped with Mathematica, you could go Euler one better. The calculation invoked by the expression

```
FactorInteger[2^2^6 +1]
```

reveals that 2^2+1, which is equal to 18,446,744,073,709,551,617, has the prime factors 274,177 and 67,280,421,310,721.

And Mathematica's skills are not limited to numerical computations. The program can factor a polynomial as readily as it can an integer. When you type in the command

```
Factor[x^4 - 10 x^2 + 9],
```

what comes back is a list of four binomial factors: $(x+3)$, $(x-3)$, $(x+1)$ and $(x-1)$. This ability to manipulate symbolic expressions as well as numeric ones is central to the design and operation of the program.

Mathematica is by no means the first or the only computer software to solve problems like these. Twenty years ago Joel Moses and his colleagues at M.I.T. undertook to create a program that would automate various aspects of algebra and calculus and also perform exact numerical calculations; the result, called MACSYMA, is still in use today and has recently been made available for microcomputers. A number of other programs with similar aims have been developed. For example, REDUCE was started by Anthony C. Hearn when he was at Stanford University in the early 1970's, SCRATCHPAD is an IBM product, and Maple is the creation of a group at the University of Waterloo in Canada. Still another system, called SMP, was written a decade ago by Stephen Wolfram, when he was at Caltech. For more on Wolfram, see below.

Out of all these programs, however, it is Mathematica that has been getting most of the attention lately. It is the subject of numerous magazine articles (including this one). A Mathematica Conference is scheduled to be held January 11-13. Soon there will even be a *Mathematica Journal.* Why has the program attracted so much interest? Part of the answer is vigorous and effective promotion. For example, issuing the user manual as a hard-cover book (published by Addison-Wesley) has helped to elevate Mathematica out of the category of mere commercial product. And the shrewdest maneuver was agreeing to supply Mathematica bundled with the NeXT computer, an arrangement in which both the hardware and the software have the benefit of reflected glamour.

An added measure of luster emanates from the presence of Stephen Wolfram, the principal architect of Mathematica, who is full of honors at an early age. With both SMP and Mathematica to his credit, Wolfram qualifies as a major software developer, and yet writing software is only a sideline for him; in real life he is a physicist and mathematician. He earned a Ph.D. in physics at age 20; he has held appointments at Caltech and at the Institute for Advanced Study; he is now professor of physics, mathematics and computer science at the University of Illinois as well as director of the Center for Complex Systems Research. In 1981 he was awarded a MacArthur Prize Fellowship. He has just turned 30.

But apart from all these factors external to the program itself, there are also some deeper reasons for Mathematica's sudden prominence. The program is simply a marvel—a vast and impressive piece of work. It is clearly a useful tool for the professional in mathematics or the quantitative sciences, as well as being great fun for the dilettante (which is my own status). And, in the competition with other computer-mathematics systems, it has one special attraction: It produces the most engaging pictures. Solving a system of differential equations is quite a feat for a computer program, but it takes a degree of mathematical sophistication to appreciate it. A picture of the solution, on the other hand, is accessible to everyone. (Figure 1, for example, shows part of a solution of the Schrödinger wave equation; the surface outlines a component of an orbital of the hydrogen atom.) This discussion of Mathematica will focus on the program's graphics facilities.

Wolfram Research, Inc., has released versions of Mathematica for more than a dozen computers. Most of these machines are Unix workstations, such as those made by Apollo, DEC, Hewlett-Packard, IBM, MIPS, Sony, Silicon Graphics and Sun, as well as the NeXT computer. There are also versions for the DEC VAX, for Cray supercomputers, for the Apple Macintosh and for MS-DOS computers based on the Intel 80386 chip. The experiments described here were done with a Macintosh SE/30.

The Mathematica system has two main parts, called the kernel and the front end. The kernel, which is where all the mathematics is done, is essentially the same in all versions of the software. The front end, which handles interactions with the user, is adapted to the peculiarities of each machine. In principle, any front end will work with any kernel, so that you might well use a Macintosh front end to communicate with a kernel running on a larger computer.

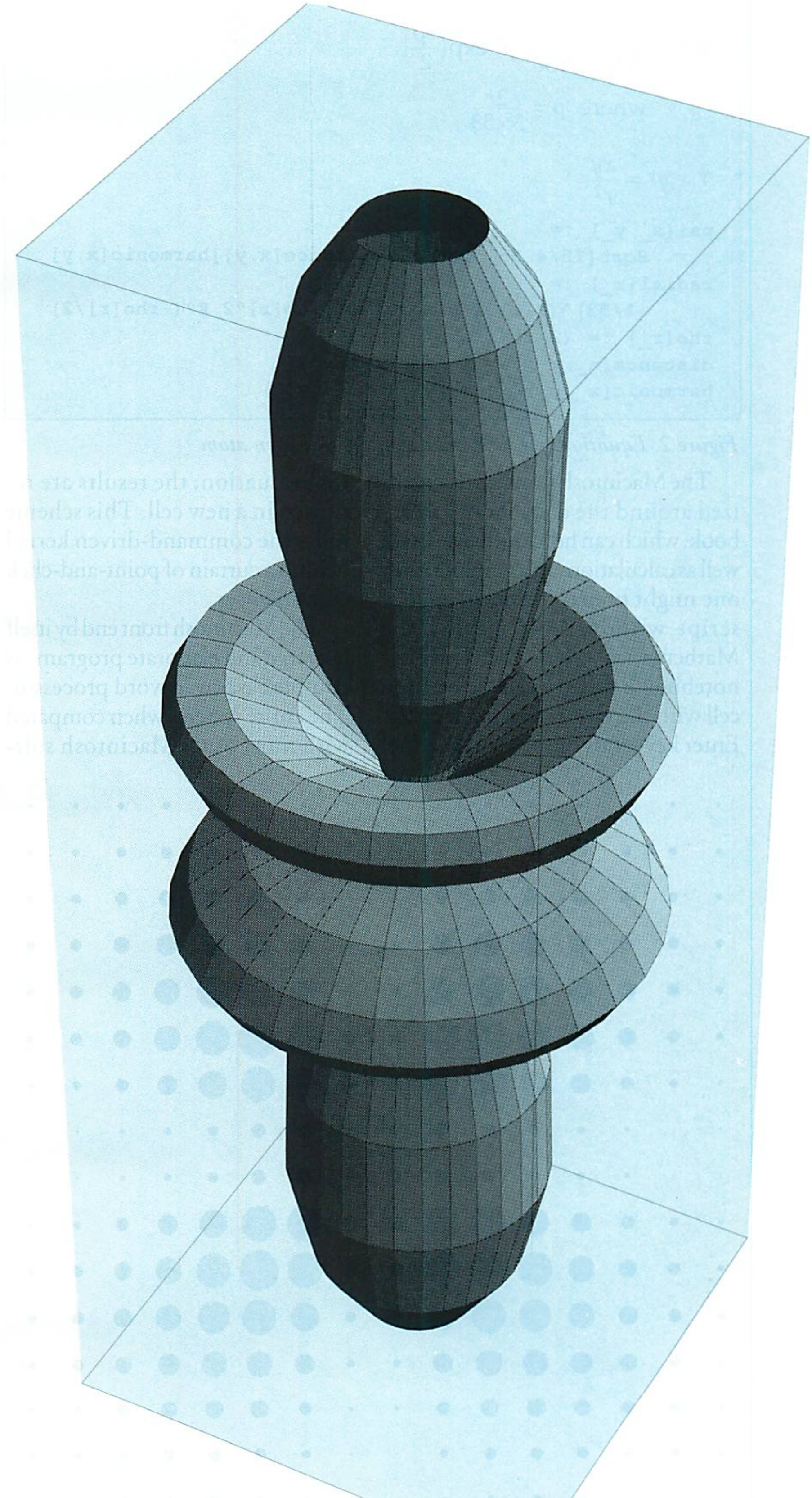


Figure 1. Parametric plot of a spherical harmonic

$$\psi(\mathbf{r}) = \left(\frac{15}{4\pi}\right)^{\frac{1}{2}} R(r)\, Y(x,y)$$

$$R(r) = \left(\frac{1}{53}\right)^{\frac{3}{2}} \left(\frac{1}{9\sqrt{30}}\right) \rho^2 \exp\left(\frac{-\rho}{2}\right)$$

$$\text{where } \rho = \frac{2r}{3\times 53}$$

$$Y(x,y) = \frac{xy}{r^2}$$

```
psi[x_,y_] :=
     Sqrt[15/4 Pi] radial[distance[x,y]]harmonic[x,y]
radial[r_] :=
     (1/53)^(3/2) 1/9 Sqrt[30] rho[r]^2 E^(-rho[r]/2)
rho[r_] := (2 r)/(3 53)
distance[x_,y_] := Sqrt[x^2 + y^2]
harmonic[x_,y_] := (x y)/(x^2 + y^2)
```

Figure 2. Equations for a 3d orbital of the hydrogen atom

The Macintosh front end is organized around the concept of a notebook, which can hold ordinary text as well as calculations and graphics; thus one might prepare an entire manuscript without ever leaving the Mathematica system. Each entry in a notebook is called a cell. Selecting a cell with the mouse and pressing the Enter key sends the cell to the kernel for evaluation; the results are returned in a new cell. This scheme hides the command-driven kernel behind a curtain of point-and-click conveniences.

The Macintosh front end by itself is a large and elaborate program, as complex as, say, a word processor. The entire system, when compared with most other Macintosh software, is enormous. It comes on eight 800-kilobyte floppy disks, and even without the dozen or so sample notebooks supplied, it fills up three and a half megabytes of disk space. The minimum RAM requirement is four megabytes. Working with a five-megabyte machine, I have found it all too easy to exhaust available memory. (The response to this condition tends to be disagreeable. Sometimes only the kernel dies, so that you have a chance to save any open notebooks, but on other occasions the only recourse is a cold boot.)

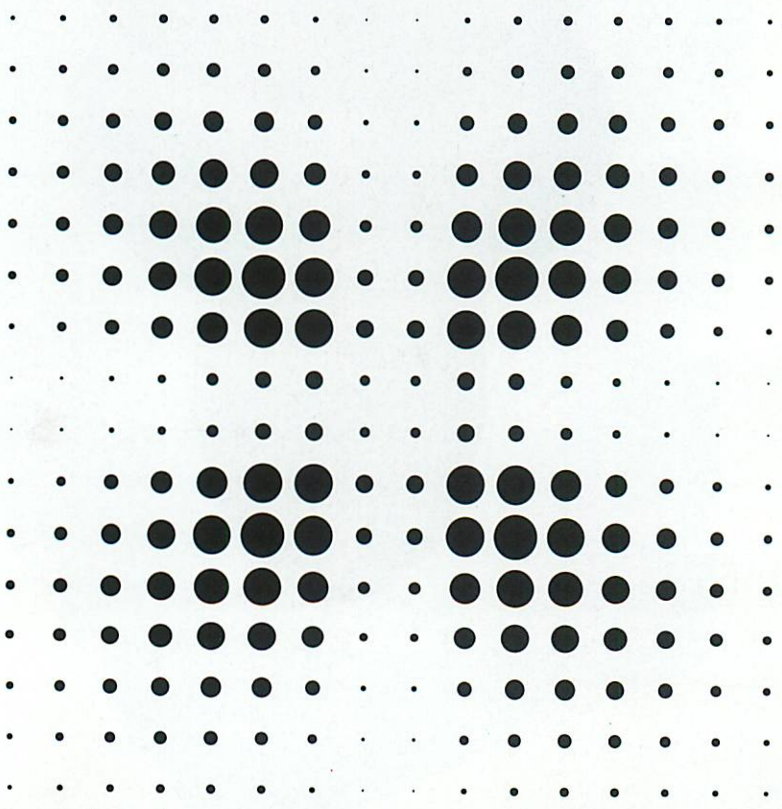


Figure 3. Wave-function amplitude encoded as disk diameter

What would William Shanks think of Mathematica? Nothing is known of his personality or character (except what can be guessed from his preoccupation), but it would take a man of extraordinary equanimity not to resent the invention that trivializes a life's work. I have a touch of ambivalence on this score myself. Although I have certainly not spent anything close to 20 years churning out numbers, I have occasionally invested a weekend in performing some routine but lengthy calculation. For example, I was once called upon to illustrate the physicist's concept of a field, meaning a quantity defined at every point throughout some region of space. This happened sometime between the age of the slide rule and the age of the computer, and so I worked out a few hundred values of an example field with a ten-key calculator, and had an artist plot the results by hand. Mathematica could now take over both my work and the artist's, and do it better as well as faster. I am relieved that I shall never again have to grind out those yards of adding-machine tape, but when I look back on my numerical labors, I also have a vague sense of chagrin, as if I had been caught in a foolish and self-indulgent waste of time.

My hand-drawn illustration necessarily showed a field with a very simple structure; with Mathematica we are free to choose a more intricate example. Accordingly, I have explored several ways of showing the field defined by the wave function of the electron in a hydrogen atom. The equations for the selected orbital are shown in Figure 2, both in standard mathematical notation and in standard Mathematica notation. The total wave function ψ is the product of a radial-distance function R and a spherical harmonic Y. The equations shown are those for the state of hydrogen with quantum numbers $n = 3$, $l = 2$ and $m = 0$, which is one of the $3d$ states. For convenience in constructing graphs, the wave function is defined in Cartesian coordinates x, y and z, rather than in the more usual polar coordinates r, θ and ϕ.

My manually prepared illustration employed the following graphic scheme. I defined a square grid and evaluated the magnitude of the field at each grid point. Then the artist drew a disk at each point, with the disk diameter proportional to the field value. What made this process so tedious was the need to repeat the entire exercise several times, until we found a suitable range of x and y coordinates and a suitable scaling factor relating field magnitude to disk diameter. The same iterative process is needed in Mathematica; the difference is that each iteration takes only a few seconds.

A graph encoding field values as disk diameters is not one of the standard graph types built into Mathematica, but it is easy to construct it. The program can be written as a single command, made up of four nested function calls:

```
Show[Graphics
    [Table
        [Disk[{x,y},
Abs[psi[x,y]] scale]
{x, xmin, xmax, dx},
{y, ymin, ymax, dy}]]
```

Disk is a function of two arguments, namely a list whose two elements give the coordinates of the disk center, and a number that defines the disk radius. In this case the expression supplying the latter argument is the product of a scale factor and the absolute value of the wave function. **Table** constructs an array of **Disk** objects, extending from **xmin** to **xmax** and from **ymin** to **ymax**, with disks distributed at intervals of **dx** and **dy**. **Graphics** interprets the array as an object to be rendered graphically, and **Show** displays the image on the screen. Figure 3 was created by substituting various scale factors and coordinates in this expression until the result "looked right." The pattern of disks shows that the orbital has four lobes symmetrically distributed around the origin.

Although Mathematica has already made life a good deal easier, the system offers still better and easier ways to show the nature of the wave function. Figure 4 is a contour plot, analogous in form to a topographic map. Again the fourfold symmetry of the wave function is apparent. Writing a program that generates high-quality contour plots is a fairly tricky undertaking, because you need to know (or approximate) the inverse of the function being plotted. That is, having selected a value of $f(x,y)$, you need to find corresponding values of x and y. But the programming chore is unnecessary in Mathematica; the algorithm has already been implemented. The contour plot of Figure 4 is generated by substituting appropriate numerical values in the expression:

```
ContourPlot[psi[x,y],
{x, xmin, xmax},
{y, ymin, ymax}].
```

Something important gets lost in both the disk plot and the contour plot: the sign of the wave function. In

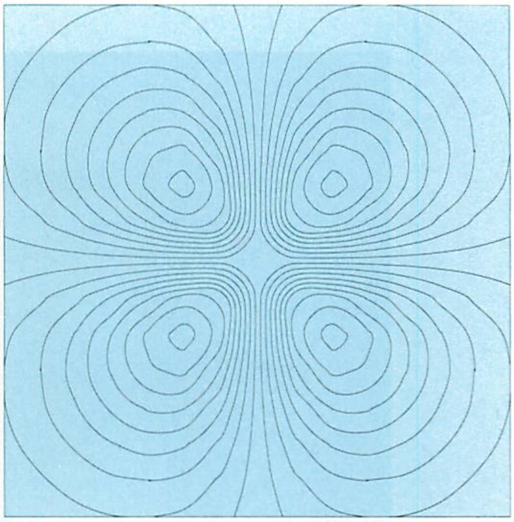


Figure 4. Contour plot of the orbital

the disk plot the sign was deliberately suppressed (by taking the absolute value of y) in order to avoid trying to draw a disk with a negative diameter. In the contour plot the sign information is present but invisible: two of the lobes in the graph represent hummocks and two are depressions, but because the graph is so symmetrical there are no clues to depth or orientation.

Another kind of plot presents the sign information to good effect. The idea is to encode the amplitude of the wave function as a gray level. In Figure 5 a field value of zero is represented by a medium gray; positive amplitudes are lighter, and negative amplitudes are darker. Now it becomes apparent that the northeast and southwest lobes of the wave function are positive, whereas the northwest and southeast lobes are negative.

Plots that use the gray-level encoding are also created by a built-in Mathematica command:

```
DensityPlot[psi[x,y],
        {x, xmin, xmax},
        {y, ymin, ymax}].
```

With suitable hardware, the graph than can be drawn in color rather shades of gray.

The most effective style of presen-

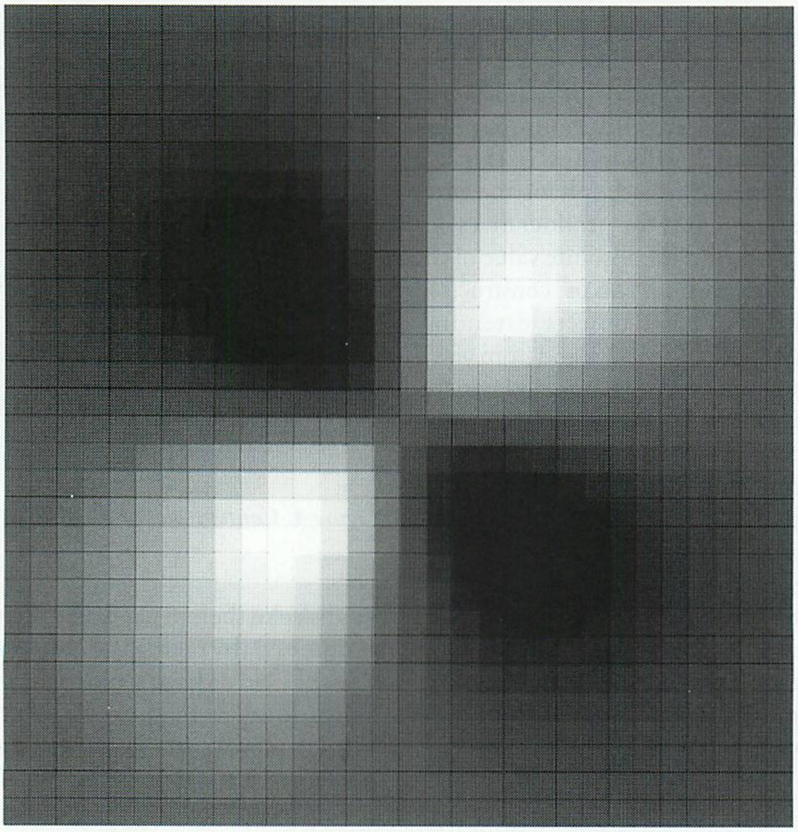


Figure 5. A gray-scale rendering shows the sign of the wave function

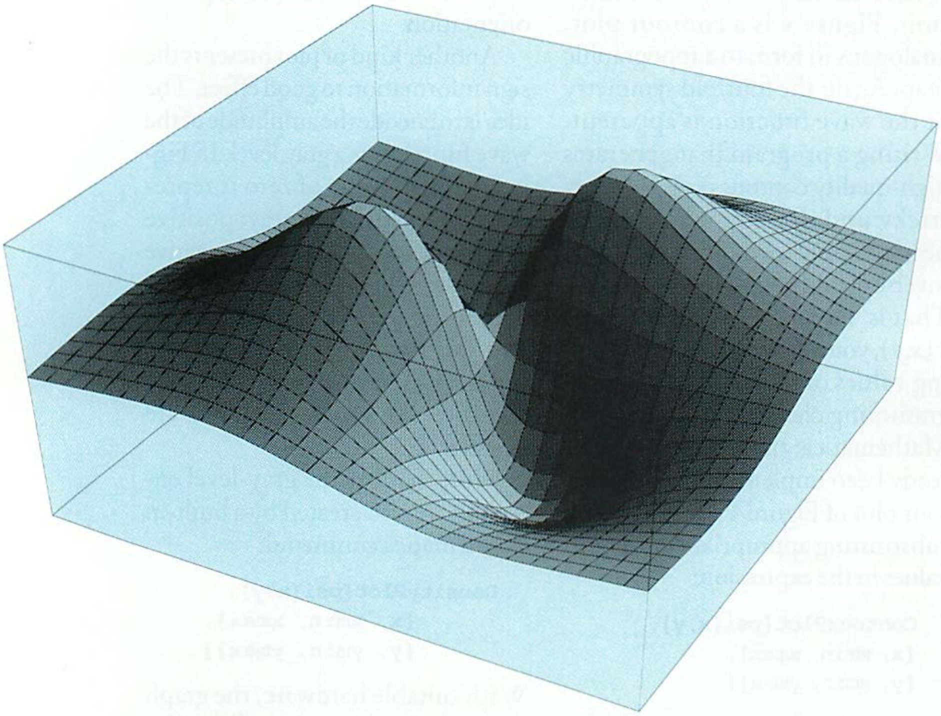


Figure 6. Amplitude represented by the undulations of a surface

tation, however, requires a leap into the third dimension. If the contour plot of Figure 4 can be regarded as a topographic map of a surface in three-dimensional space, then why not give a perspective drawing of that surface? Figure 6 is just such a drawing. It shows the amplitude of the wave function by means of surface elevation; the peaks are the points of greatest positive amplitude, and the pits are where the function is most negative. As with the contour plot and the density plot, this image was created merely by invoking a built-in Mathematica function:

```
Plot3D[psi[x,y],
    {x, xmin, xmax},
    {y, ymin, ymax}].
```

The graph in Figure 1, discussed above, employs yet another standard Mathematica drawing technique. It is a parametric plot: Instead of showing height along the *z* axis as a function of *x* and *y* coordinates, it records a locus of *x*, *y* and *z* points as a function of two independent parameters, θ and ϕ, which can be interpreted as angles swept out by planes rotating around the origin. The graph shows only the spherical-harmonic component of a wave function, without the contribution of the radial-distance function. It turns out that spherical harmonics are among the large collection of special functions built into Mathematica, and so creating the image was a simple matter of applying **`ParametricPlot3D`** to the function **`SphericalHarmonicY`**, and supplying a few additional arguments. Calculating and drawing the surfaces takes about 10 minutes.

Mathematica produces its graphic output not as a bitmap or as a vector drawing but as code in the Postscript page-description language. The Postscript code must then be interpreted for display. Each Mathematica front end has a Postscript interpreter for rendering graphics on

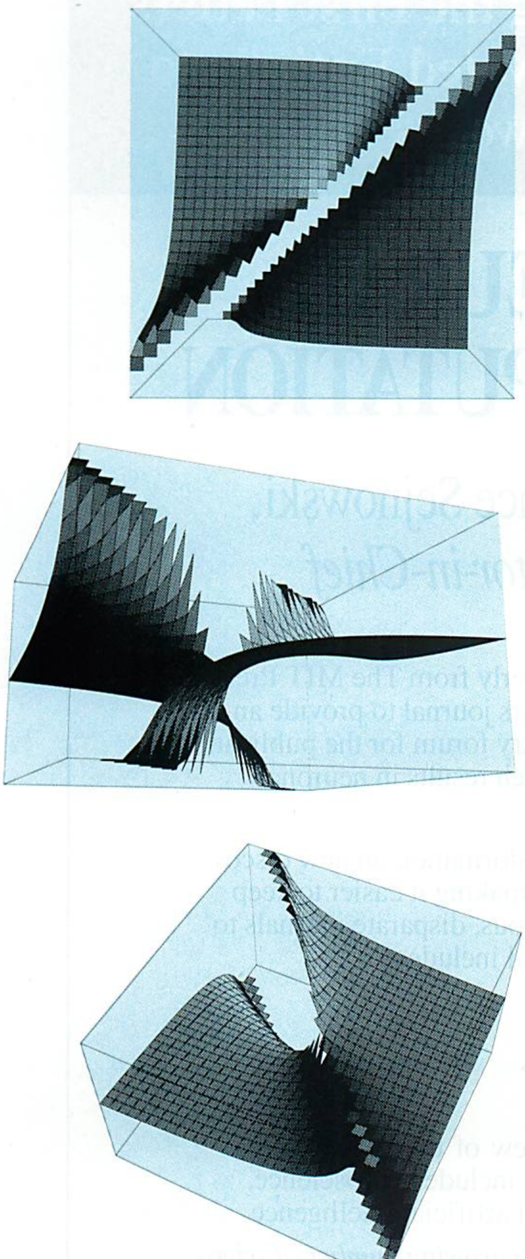


Figure 7. Three views of a torn sheet

the screen. The same code can also be processed by any other device or program that understands Postscript. For example, Mathematica images can be imported into Postscript drawing programs such as Adobe Illustrator.

The use of Postscript has two important advantages over most other rendering schemes. First, Postscript images are device-independent. The illustrations that accompany this article began as 72-dot-per-inch images on the Macintosh screen. Initial proofs were printed out at 300 dots per inch on a laser printer, and final copies were made at 1,693 dots per inch on a phototypesetter. The same Postscript code was used without alteration in all three cases.

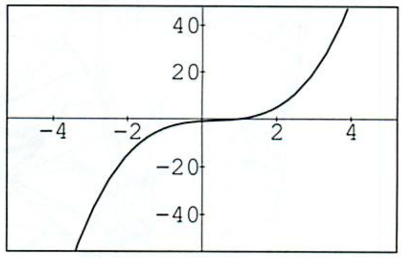
Second, Postscript images are also independent of scale and viewpoint. In Mathematica's Macintosh front end you can change the size or the proportions of an image merely by pulling on "handles" attached to the image frame. For three-dimensional graphics a viewpoint is selected by using the mouse to rotate a small wire-frame cube. Figure 7 shows three views of a three-dimensional graph, which together give a better sense of the shape of the surface than any one view could.

Figure 7 demonstrates another pleasant property of Mathematica's graphics routines. The three images are graphs of the function $z = (x + y)/(x - y)$, which has a line of singularities along the diagonal $x = y$. In preparing the graphs, I took no precautions to avoid points where the function is undefined. Mathematica issued a series of warning messages when it was asked to plot points with values such as 1/0, but it did not give up on drawing the graphs. Moreover, the program chose an appropriate scale for the z axis; it did not distort the graph by attempting to include the very large values of z found near the $x = y$ diagonal.

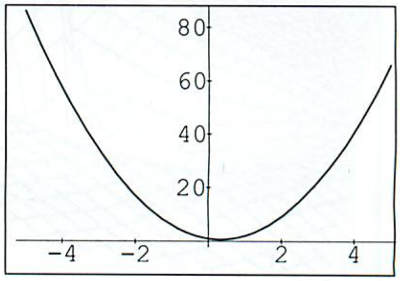
Still another example of "good judgment" is on exhibit in Figure 8, which is a printout of a small Mathematica notebook. Here the calculations begin with the definition of a cubic function; then, using the **D[]** operator, Mathematica calculates the first and second derivatives of the function. When the original function and its derivatives are plotted, Mathematica chooses a suitable vertical scale for each separate graph; when the three plots are superimposed, the curves are automatically rescaled.

```
fn0 := x^3 - x^2 + x - 1
fn1 = D[fn0,x]
1 - 2 x + 3 x^2
fn2 = D[fn1,x]
-2 + 6 x
```

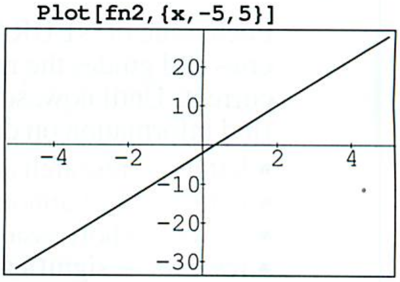
```
Plot[fn0,{x,-5,5}]
```




```
Plot[fn1,{x,-5,5}]
```




```
Plot[fn2,{x,-5,5}]
```




```
Plot[{fn0,fn1,fn2},{x,-5,5}]
```

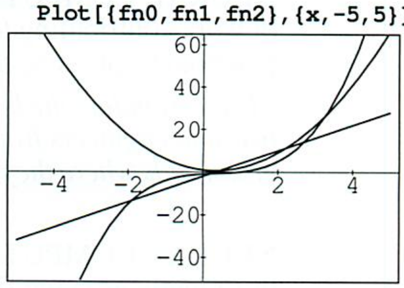


Figure 8. Graphs of derivatives

There is probably no need to point out that fooling around with derivatives in this way would be an excellent way to learn the rudiments of calculus. The graphs make plain the connection between the slope of a curve and the value of its derivative. Moreover, determining quantities such as intercepts, minima and maxima is almost effortless. Issuing the command

```
Solve[fn0 == 0]
```

quickly confirms what the graphs suggest: that the cubic equation has one real root, at $x = 1$. (Mathematica also finds the two imaginary roots, at $\pm i$.) Turning to the second derivative, the command

```
Solve[fn3 == 0]
```

yields the result $x = 1/3$, which must therefore be the position of the inflection point in the cubic curve. A numerical calculation of the minimum of the first derivative provides confirmation:

```
FindMinimum[fn1, {x, 0}]
{0.666667, {x ->0.333333}}.
```

In the two-element list returned by **FindMinimum**, 0.666667 is the minimum value of the **fn1** curve, and 0.333333 is the position where that minimum value is attained.

We tend to think of the computer as being naturally suited to doing mathematics. It is, after all, a machine whose most basic operations include + and −, × and ÷. The fact is, however, a computer right out of the box is not much use in mathematics. An elaborate layer of software is needed to introduce all the necessary concepts and apparatus—functions, equations, points and lines, vectors and matrices, series and limits, derivatives and integrals, not to mention graphs. Just how elaborate the layer of software needs to be is suggested by the size and complexity of Mathematica.

To put the same idea another way: It is no easier to erect the structure of modern mathematics when starting with the instruction set of a microprocessor than it is when starting with the axioms of set theory. And hence Mathematica is comparable in size and complexity to *Principia Mathematica.*