10-27-1993

A *Mathematica* tutorial.

This is a brief introduction to *Mathematica*, the symbolic mathematics program. This tutorial is generic, in the sense that you can use it no matter what kind of computer you have. However, this means that it doesn't discuss features which are specific to your version of *Mathematica*. Hence, if you can find a tutorial designed for your system, you might find it more useful.

This tutorial assumes that you know how to perform basic operations with your computer. If not, you should consult the documentation that came with your machine.

I'll start with basic arithmetic. Here is the *Mathematica* command to compute 3^{100}:

```
3^100
```

Type it in, then execute it by pressing **Shift-Enter**.

(If you are using the Macintosh version or the *Microsoft Windows* version, move the mouse until your cursor turns into a horizontal "I-beam". Click the mouse; a horizontal line appears. When you start typing, your input should replace the horizontal line. When you are finished typing, press **Shift-Enter** to execute the command.)

The long number that appeared is 3^{100}. Notice that *Mathematica* put "`In[1]:=`" next to your input and "`Out[1]=`" next to your output. *Mathematica* labels all of its input and output in this way so that you have an easy way to refer to things you've done earlier.

Notice that I'm using a different font and typeface for *Mathematica* commands. A *Mathematica* command will always look like this:

```
(* Here is a Mathematica comment.  *)
```

so that you can tell it apart from normal text.

As you go through the rest of the tutorial, my explanations will be interspersed with *Mathematica* commands for you to try. You should execute the commands as you read to see how they work.

Here are some things to pay particular attention to:

- Spaces (or the absence of spaces) can be important! For example, "`xy`" and "`x y`" are not the same! The first is the 2-letter name of a variable, i.e. a single word. The second is "`x` *times* `y`".
- Certain letters must be capitalized. For example, most of *Mathematica*'s built-in functions start with a capital letter, and some have capital letters "in the middle". "`Plot3D`" is correct; "`plot3d`" is not.
- Pay attention to punctuation. For example, "`(`", "`[`", and "`{`" mean different things.

1. More arithmetic.

Here is how to use*Mathematica* to perform addition, subtraction multiplication, and division. Execute the commands below; for practice, make up a few yourself.

```
100 + 200
117 - 43
77 * 99
77 99
63 / 9
```

2. Numerical results.

Of course, you can use *Mathematica* to compute transcendental functions. Execute the next line:

```
Sin[30]
```

The result isn't very enlightening. To express a result in approximate (decimal) form, use the `N[]` function:

```
N[Sin[30]]
```

Mathematica (like most mathematicians) reads angles in radians unless you tell it otherwise. So "`Sin[30]`" means the sine of 30 radians.

The names of *Mathematica*'s built-in functions, like `Sin`, always start with capitals. Some even have capitals in the middle of their names (like "`ParametricPlot`"). You must have exactly the right letters capitalized, or *Mathematica* won't know what you mean.

Notice that when you want to apply a function to something, you must use square brackets: "`[`" and "`]`", not parentheses. That is, instead of writing "$y = f(x)$" the way you do in calculus, you would need to write "`f[x]`" in *Mathematica*. So you write "`Sin[30]`", not "`Sin(30)`".

Suppose you want the sine of 30 to 50 places. Execute the next command:

```
N[%, 50]
```

The "`50`" tells `N[]` to approximate the result to 50 places.

What about the "`%`"? This is shorthand for the previous output. It is useful when you want to avoid retyping a long formula, or when you want to use the last result in your next computation.

Execute the following commands to compute some other transcendental functions:

```
N[ArcTan[7]]
```

Notice that the function is "`ArcTan`", not "`Arctan`"!

```
N[E^2]
```

Mathematica represents the constant $e = 2.71828\ldots$ with a capital E: "`E`".

```
N[Log[4]]
```

3. Symbolic algebra.

Mathematica really shines as a tool for symbolic manipulation. I'll discuss a few of its capabilities below. Execute the commands as you go.

Here is a binomial expansion:

```
Expand[(x + y)^10]
```

The "`Expand`" function is good for "multiplying stuff out".

You can also use *Mathematica* to put things together. For example, here is how to combine fractions over a common denominator:

```
Together[1/(x + 1) - 5/(x^2 + 2)]
```

You can also take a fraction apart (i.e. obtain its partial fractions decomposition). This will be familiar to people who have seen it used as an integration technique.

```
Apart[%]
```

Mathematica can also simplify expressions:

```
Simplify[x^2/(x - 1) - 1/(x - 1)]
```

Mathematica combined the fractions over a common denominator, then noticed that the numerator was divisible by the denominator and performed the cancellation.

Mathematica makes reasonable assumptions about how you'd like a simplification performed. However, the program can't anticipate every human intention, and sometimes you will have to "play around" to get answers in the form that you want. Sometimes, it's even necessary to "teach" *Mathematica* simplification rules that it doesn't know!

If you're doing serious work with *Mathematica*, you should refer to the manual for other functions and techniques that are useful in performing algebraic manipulations.

4. Solving equations.

You can use *Mathematica* to solve equations, or systems of equations. For example:

```
Solve[{x + y == 2, x - y == 0}, {x, y}]
```

In some cases, *Mathematica* can't obtain an exact solution:

```
Solve[x^5 + 2x == 7, x]
```

You can use `N[]` to obtain a numerical approximation to the solution:

```
N[%]
```

5. Calculus.

Here's how do compute $\dfrac{d}{dx}\dfrac{x}{\sin x + \cos x}$:

```
D[x/(Sin[x] + Cos[x]), x]
```

You can also compute higher-order derivatives. This is the 3-rd derivative of $\dfrac{x}{\sin x + \cos x}$:

```
D[x/(Sin[x] + Cos[x]), {x, 3}]
```

You can also compute partial derivatives:

```
D[(x^2 + y^2) E^(-x^2 - 4y^2), x]
```

Mathematica has a sizeable repertoire of integration rules:

```
Integrate[Cos[x]^4, x]
```

It is also possible to teach *Mathematica* rules that it doesn't know.

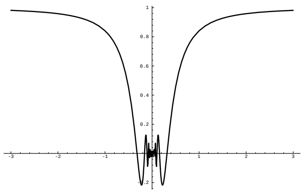
If *Mathematica* can't evaluate a definite integral in closed form, you can use `N[]` as usual to obtain a numerical approximation:

```
Integrate[E^(x^4), {x, 0, 1}]
N[%]
```

6. 2-dimensional Graphs.

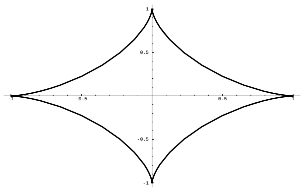
This is how to draw the graph of $y = x \sin \dfrac{1}{x}$ for $-3 \le x \le 3$:

```
Plot[x Sin[1/x], {x, -3, 3}]
```



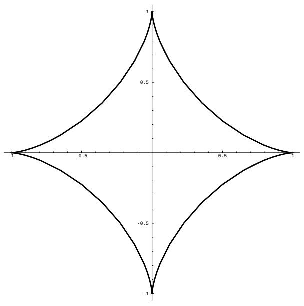
Parametric plots are also very simple:

```
ParametricPlot[{Cos[t]^3, Sin[t]^3},
    {t, -2 Pi, 2 Pi}]
```



The aspect ratio defaults to `1/GoldenRatio`; if you want better scaling, you can set the aspect ratio yourself, or have *Mathematica* do it automatically:

```
ParametricPlot[{Cos[t]^3, Sin[t]^3},
    {t, -2 Pi, 2 Pi}, AspectRatio->Automatic]
```



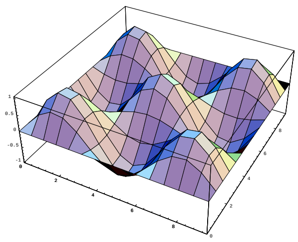
There are many other options for the graphics functions; see the *Mathematica* manual for details.

Mathematica produces graphics in **PostScript**, the page description language. This makes it possible to export *Mathematica* graphics for use in other programs.

7. 3-dimensional Graphs.

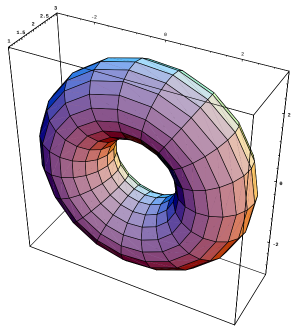
Here is the graph of $z = \sin x \sin y$, for $0 \leq x \leq 3\pi$ and $0 \leq y \leq 3\pi$:

```
Plot3D[Sin[x] Sin[y], {x, 0, 3 Pi}, {y, 0, 3 Pi}]
```



Many surfaces can't be represented as graphs of functions. *Mathematica*'s `ParametricPlot3D[]` function plots parametrized surfaces:

```
ParametricPlot3D[{(2 + Cos[u]) Cos[v], 2 + Sin[u], (2 + Cos[u]) Sin[v]},
    {u, 0, 2 Pi}, {v, 0, 2 Pi}]
```



8. More...

There are many features of *Mathematica* which I haven't discussed. For example, some versions of *Mathematica* support **Notebooks**: interactive textbooks containing text, graphics, and *Mathematica* commands. Some versions support animated graphics and sound. I also haven't said much about the *Mathematica* programming language.

To find out more about these things, and many others, you should consult the references below. Note that some features are specific to certain versions of *Mathematica*; for these features, you should consult the documentation for the version you're using.

9. Further reading.

Mathematica - A system for Doing Mathematics by Computer (2-nd edition) by Stephen Wolfram; Reading, MA: Addison-Wesley Publishing Co., 1991; ISBN 0-201-51502-4 (hard), 0-201-51507-5 (paper).

Programming in Mathematica by Roman Maeder; Reading, MA: Addison-Wesley Publishing Co., 1990; ISBN 0-201-51002-2.

Mathematica in Action by Stan Wagon; New York: W. H. Freeman and Company, 1991; ISBN 0-7167-2229-1 (hard), 0-7167-2202-X (paper).

Wolfram's book is the *Mathematica* reference amnual. It is over 700 pages long, well-written, but technical. It is essential if you want to do a lot of work with *Mathematica.* Maeder's book covers advanced topics in Mathematica programming. He explains many tricky points in more detail than the reference manual. Wagon's book explores various topics in mathematics using *Mathematica.*

The *Mathematica Journal* is published quarterly, and contains articles on programming *Mathematica* and applications. It costs $55 a year, and can be ordered from:

The Mathematica Journal
600 Harrison Street
San Francisco, California 94107-9602

Mathematica runs on a large number of platforms, including the Apple Macintosh, MS-DOS 80386 machines, and UNIX workstations. For information about other versions, you can contact Wolfram Research:

Wolfram Research
P.O. Box 6059
Champaign, Illinois 61826-6059
217-398-0700