

Experiments with MATLAB

By Cleve Moler

Experiments with MATLAB is a free, online book for educators and high school students looking for material that goes beyond their standard courses. College students early in their careers will also find value in the materials and exercises.

Now a full-fledged technical computing language, MATLAB® started in the late 1970s as a simple “Matrix Laboratory.” *Experiments with MATLAB* builds on this laboratory tradition by describing experiments involving applied mathematics and technical computing. MATLAB programming is introduced with code snippets and small programs, many of which use interactive graphics.

To make the most of the book’s experiments, readers need a high-school-level knowledge of geometry, algebra, and trigonometry. *Experiments with MATLAB* introduces topics involving calculus, matrices, and differential equations, but it does not assume that students have completed courses in these subjects.

In this *Cleve’s Corner*, I’ve included edited excerpts from *Experiments with MATLAB* to give you some idea of the level and tone of the book. I encourage you not only to read about the experiments but also to modify and improve them.

Iteration

The first chapter asks you to pick a number —any number. Enter it into MATLAB with

```
x = your number
```

Now enter the statement

```
x = sqrt(1+x)
```

Use the up-arrow and enter keys to repeatedly execute the statement until the displayed values stop changing.

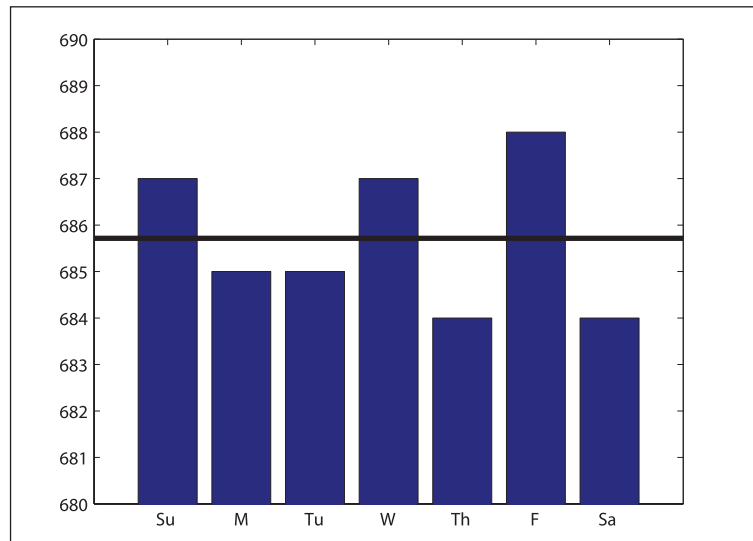


FIGURE 1. MATLAB plot showing that the 13th day of a month is more likely to be a Friday than any other day of the week.

Here are the first and last few lines that you will see if you choose $x = 3$ as your starting value.

```
3.0000000000000000
2.0000000000000000
1.732050807568877
1.652891650281070
1.628769980777233
. . .
1.618033988749915
1.618033988749901
1.618033988749897
1.618033988749895
1.618033988749895
```

No matter what value you start with, you always converge to the same final

value, 1.6180339... . Do you recognize this number?

This experiment motivates a discussion of the two meanings of the equals sign: as the assignment operator in programming languages and as the symbol for equality in equations.

How would you solve the following equation?

$$x = \sqrt{1+x}$$

Don’t use a computer! Do it in your head or on paper. Square both sides and move everything to one side of the equals sign to get a quadratic equation.

$$x^2 - x - 1 = 0$$

Then use the quadratic formula to find the positive root,

$$\varphi = (1 + \sqrt{5})/2 = 1.6180339 \dots$$

This is our old friend, the golden ratio.

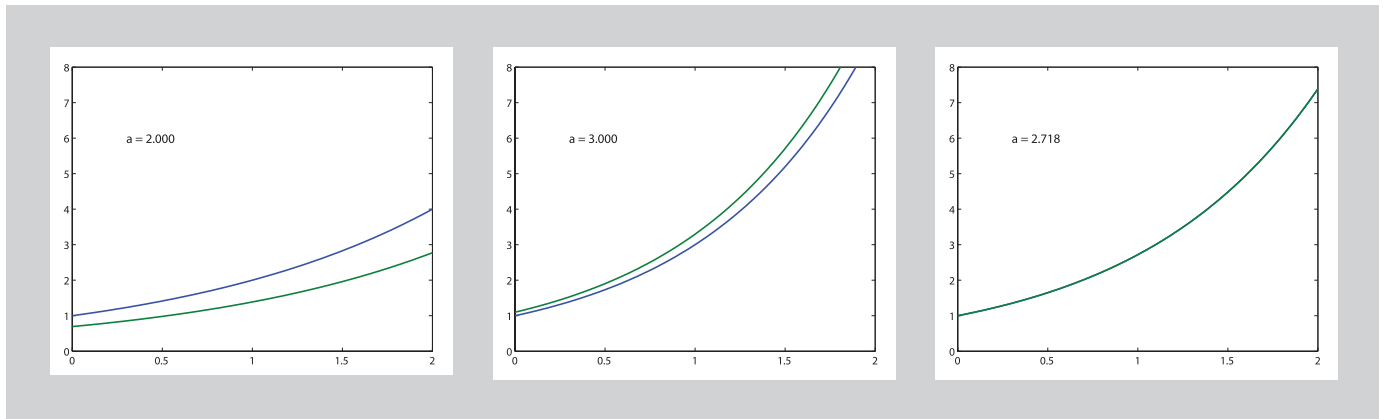


FIGURE 2. Varying the base in a plot of a^x and its approximate derivative to discover e .

Calendars and Clocks

Friday the 13th is sometimes considered unlucky, but is it unlikely? What is the probability that the 13th day of any month will fall on a Friday? The quick answer is $1/7$, but that is not quite right. The rules for leap years ensure that our Gregorian calendar repeats itself every 400 years, or 4800 months. A simple experiment using the MATLAB `datenum` and `weekday` functions can count how often the 13th day of each month falls on a Friday, producing the graph in Figure 1. It turns out that the 13th is more likely to fall on a Friday than on any other day of the week. The probability is $688/4800 = .143333$, which is slightly larger than $1/7 = .142857$.

The Exponential Function

Too many students who have recently taken a beginning calculus course think the derivative of e^x is xe^{x-1} . How can we better understand the number e and the function e^x ? MATLAB can compute powers, $y = a^x$, for a scalar a and a vector x .

```
a = 2
x = 0:0.01:2
y = a.^x
```

We can also compute an approximate slope or approximate derivative, without any formal rules of differentiation, using

```
h = 0.0001
yp = (a.^(x+h) - a.^x)/h
```

Then the statement

```
plot(x, [y; yp])
```

produces the first plot shown in Figure 2. The blue curve is the graph of 2^x . The green curve is its approximate derivative. The graph of the approximate derivative has the same shape as the graph of the original function but is below it. In fact, the ratio $yp./y$ is a constant independent of x .

The M-file `expgui` lets you move the blue line with the mouse and slowly vary the base of a^x . The second shows the graph of 3^x and its approximate derivative. The green curve is now above the blue curve. As you move from $a = 2$ to $a = 3$, you pass the situation shown in the third plot, where the green curve is on top of the blue curve. For this value of a , the function is its own derivative. The value displayed for a is e and the function is e^x , one of the most important functions in mathematics.

We've skipped an important but subtle mathematical and computational question:

Experiments with MATLAB

Table of Contents

- Iteration
- Fibonacci Numbers
- Calendars and Clocks
- T Puzzle
- Matrices
- Fractal Fern
- Magic Squares
- Tic Tac Toe Magic
- Game of Life
- Mandelbrot Set
- Linear Equations
- Google PageRank
- Ordinary Differential Equations
- Exponential Function
- Predators and Prey
- Shallow Water Equations

Download *Experiments with MATLAB* at www.mathworks.com/nn8/moler

How does MATLAB compute a^x without knowing about e ? Well, actually, it can't, so our whole argument is circular. But if anybody asks that question, we have a terrific topic for discussion.

Solving the Amazing T Puzzle

I first saw the wooden puzzle shown in Figure 3 at Puzzling World in Wanaka, New Zealand. Now their most popular puzzle, it was a well-known toy in the 1800s and an advertising tool in the early 1900s.

Figure 4 shows the MATLAB electronic version of the four pieces. They all have the same width, but different heights. One has an unshapely chunk cut out of it, resulting in an irregular pentagon. Using MATLAB and the GUI for the book's sample T Puzzle M-file, you could move the pieces around with the mouse. It turns out that the four pieces can be arranged to form the capital "T" shape shown in Figure 5. What happened to all those 45° angles—and to that chunk?

Our program for manipulating the T puzzle uses complex arithmetic. For example, the coordinates of the largest puzzle piece are given by the complex vector

$$z = [0 \ 1 \ 1+2i \ 3i]$$

where the horizontal and vertical dimensions are represented by the real and imaginary components, respectively.

Translating the piece in response to mouse motion is easily done by a complex vector subtraction

$$z = z - w$$

We rotate the piece about its center by an angle θ using the following statements:

```
mu = mean(z)
omega = exp(i*theta)
z = omega*(z - mu) + mu
```

Where do you learn that complex multiplication by $e^{i\theta}$ is a rotation?



FIGURE 3. This wooden puzzle is manipulated by hand. Image courtesy of Shop New Zealand www.shopnewzealand.co.nz

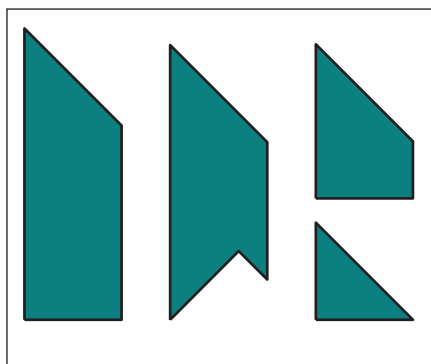


FIGURE 4. This electronic puzzle is manipulated by complex arithmetic.

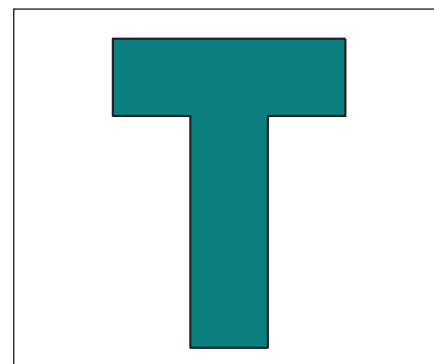


FIGURE 5. Make this T.

The Game of Life

John Horton Conway's "Game of Life" made its debut on the cover of *Scientific American* in October 1970 and has had a small but loyal following ever since. It is an example of the complexities that can be generated by apparently simple *cellular automata*.

In "Life", the universe is an infinite, two-dimensional rectangular grid. The population is a collection of grid cells marked as alive. The population evolves at discrete time steps known as generations. At each step, the fate of each cell is determined by the vitality of its eight nearest neighbors and by this

rule: A live cell with two live neighbors, or any cell with three live neighbors, is alive at the next step.

This deceptively simple rule leads to an incredible variety of patterns, puzzles, and unsolved mathematical problems.

The “Game of Life” MATLAB program discussed in *Experiments with MATLAB* is a beautiful example of the use of the *sparse matrix* data structure. The universe is a sparse matrix X with a finite number of 1s marking the live cells. The size of X grows to accommodate any expanding population.

The statements

```
n = size(X,1);
p = [n 1:n-1];
q = [2:n 1];
Y = X(:,p)+X(:,q)+X(p,:) + X(q,:) + ...
    X(p,p)+X(q,q)+X(p,q)+X(q,p);
```

generate another sparse matrix Y with elements between 0 and 8 that count the number of live neighbors. The rule of “Life” is then implemented in a single MATLAB statement:

```
X = (X & (Y == 2)) | (Y == 3);
```

Because all populations follow this rule, the initial population configuration determines how the game of life plays out. In Bill Gosper’s initial population, known as the *glider gun* (Figure 6), the central portion of the gun oscillates, emitting an infinite stream of gliders that pass out of view and into the void. Gosper’s configuration was the first to create an unbounded population.

Further Experimentation

Reading about the exercises in *Experiments with MATLAB* is not nearly as fun—or as educational—as running them yourself. So if you find them interesting, or if you know students who would be interested, please take a look at *Experiments with MATLAB*. Then run, discuss, and improve the experiments together. Programs, especially MATLAB programs, are vehicles of discourse with people, and not simply ways to send commands to a machine. ■

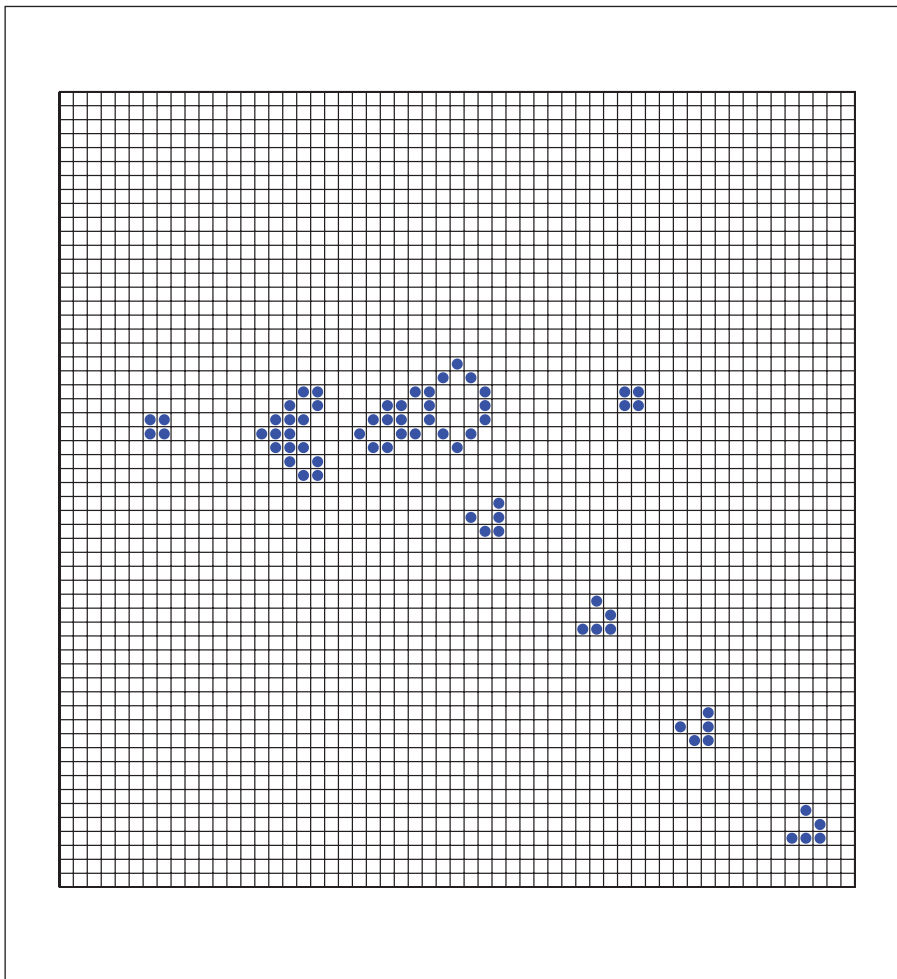


FIGURE 6. Bill Gosper’s glider gun.

Resources

EXPERIMENTS WITH MATLAB
www.mathworks.com/nn8/moler

CLEVE’S CORNER COLLECTION
www.mathworks.com/nn8/clevescorner