

# Using MATLAB to Teach Programming to First-year Engineering Students at Boston University

By Dr. Stormy Attaway, Boston University

In the 20 years that I've been teaching *Introduction to Engineering Computation*, we've used several different languages, including Pascal, Fortran, C, and C++, but the course remained fundamentally unchanged until recently, when an engineering faculty task force met to rethink the course objectives.

Faculty from the electrical and computer engineering department initiated the change, telling the committee that they would like all students to learn the basic programming concepts using MATLAB® so that students could then effectively use MATLAB in their upper-level courses and beyond.

Some faculty initially balked, believing that it was a choice between MATLAB and introductory programming—not realizing that MATLAB has all the requisite programming constructs.

It may seem at first that a biomedical engineering student would only need to know how to use MATLAB efficiently and that a computer systems engineer would only need to know programming concepts, but in engineering today, both skill sets are vital. Another consideration was that students who learn a low-level language such as C++ or Java before learning MATLAB tend to use MATLAB inefficiently, at least at first. Instead of using code vectorization techniques and the many powerful functions built into MATLAB, they rely

on **for**-loops and other low-level constructs. Not only does this approach take longer, the resulting solutions are usually not optimized for performance.

For these reasons, in *ENG EK 127 Introduction to Engineering Computation* we decided to take the unique approach of teaching programming concepts and the built-in functions side-by-side, using MATLAB as the vehicle for both. Each semester, about 200 engineering students in all engineering disciplines learn how to work with variables, data structures, loops, conditionals, functions, input and output, and other programming concepts within MATLAB and how to use MATLAB efficiently to solve engineering problems.

## Jumping Right In

One of the biggest hurdles that I faced when I taught the course using C++ was simply getting the students started. I had to give them a two-page, single-spaced handout on how to open the appropriate code editor and start typing, and it took an entire lecture to go through the basic setup steps.

## Product Used

- MATLAB®

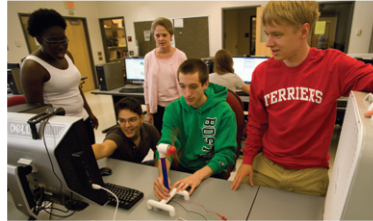
With MATLAB, students can jump right in; they type commands in the Command Window and immediately see the results. There's no intimidation whatsoever. For a student new to programming, this is much easier—and more motivating—than the edit-compile-execute-debug cycle required by other languages.

Instead of spending the first lecture talking about how to use the development environment, I can now cover assignment statements, variables, expressions, operators, precedence, and vectors.

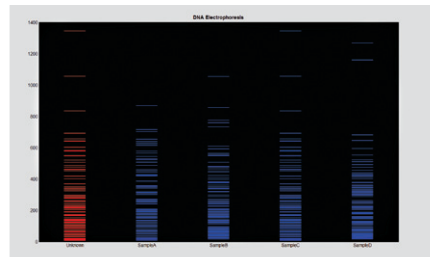
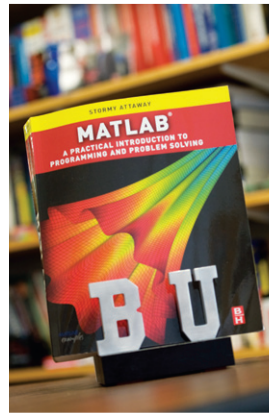
Students are also motivated when they can visualize results in new ways. I cover MATLAB plots early in the semester because using a **for**-loop to create plots and charts is much more interesting to the students than repeatedly printing a string.

## Course Structure

The course is based on my book *MATLAB: A Practical Introduction to Programming and Problem Solving*. It consists of two one-hour lectures per week, each followed by an



*Left to right, top to bottom: Dr. Attaway lecturing on for-loops, students demonstrating the wind turbine data collection project, a student problem-solving in MATLAB at the chalkboard, the course text book, students working with Dr. Attaway in the lab, an engineering building at BU.*



*MATLAB plot created by students to compare DNA samples.*

hour and a half in the computer lab. A one-hour informal discussion session at the end of the week offers more opportunities for students to ask questions.

In the lectures, I repeat the process of talking for five to ten minutes, showing an example using MATLAB, and then giving the students a problem to work on. Lectures used to be held in a computer lab, with students working on the problems directly in MATLAB. I've found, however, that they learn better when they think the problem through using pen and paper instead of immediately going to code. (For

the same reason, tests are given in a written question-and-answer format rather than within MATLAB.) Following each lecture they work with MATLAB in the lab to solve more advanced problems, which gives them an opportunity to immediately apply the abstract concepts they just learned.

Working in pairs or groups of three, students complete three projects during the course. More complex than the ones I give during the lab sessions, these projects are designed to give students an idea of how various engineering disciplines use MATLAB to solve real problems. I typically

include one project from each College of Engineering department every semester. Recent projects have included image processing for quality control on turbine blades, detecting differences in images, fingerprint recognition, finite differences for heat transfer applications, and elastic scattering spectroscopy for cancer detection.

### **Beyond Basic Programming Concepts**

MATLAB is so easy for the students to use and learn that I can cover all the basic programming concepts that we used to

cover using other platforms plus a great deal of mathematics and basic problem-solving, including introductions to linear algebra, curve fitting, and statistical analysis.

MATLAB has also enabled me to try new teaching approaches. For example, in a summer class the students used the Lego Mindstorms NXT system and MATLAB to build robots and program them to navigate an obstacle course. More recently, we set up a desktop wind turbine in a wind tunnel and used MATLAB to collect and analyze data, perform curve fitting, calculate the power generated, and run simulations.

### Positive Feedback

In open meetings with sophomores and juniors who took the MATLAB based course as freshmen, we heard over and over how pleased they were to have learned MATLAB in their first year at BU. I've heard nothing but positive remarks about the course from the engineering faculty as well. I'm working with several professors to further integrate MATLAB into the engineering curricula. We plan, for example, to use it more extensively in the linear algebra and engineering mechanics classes.

Among the only complaints I've heard were a few from upperclassmen who took the C++ version of the course and wished that they had had an opportunity to learn MATLAB (which was then provided for them in the form of wildly popular Saturday tutorials).

Teaching programming concepts with MATLAB has yielded several advantages that we envisioned and a few that we did not. We expected that the visual, interactive MATLAB environment would engage students and help them learn new concepts. It turns out that MATLAB also gives them a more thorough understanding of

those concepts. For example, many students typically have trouble grasping the notion of nested loops. When we get to matrix operations, I show how these are implemented in MATLAB by walking through the MATLAB files. Students see exactly how nested loops work and why they work. Many are surprised that a concept that was at first difficult is now so simple to them.

It is a common misconception among engineering faculty that teaching a computation course using MATLAB means forgoing the programming concepts. This is not true at all. We have shown at Boston University that you can impart programming skills, efficient coding, and basic problem solving, all within the user-friendly MATLAB environment. ■

### About the Author

Dr. Stormy Attaway is an Assistant Professor and Director of Curricular Assessment and Improvement in the College of Engineering at Boston University. Dr. Attaway holds a B.S. in geology from the University of South Carolina and an M.S. in computer science and an interdisciplinary Ph.D. in computer science, applied math, and engineering from Boston University. She is the author of the widely used textbook *MATLAB: A Practical Introduction to Programming and Problem Solving*.

## Resources

### VISIT

[www.mathworks.com/academia](http://www.mathworks.com/academia)

### TECHNICAL SUPPORT

[www.mathworks.com/support](http://www.mathworks.com/support)

### ONLINE USER COMMUNITY

[www.mathworks.com/matlabcentral](http://www.mathworks.com/matlabcentral)

### DEMOS

[www.mathworks.com/demos](http://www.mathworks.com/demos)

### TRAINING SERVICES

[www.mathworks.com/training](http://www.mathworks.com/training)

### THIRD-PARTY PRODUCTS AND SERVICES

[www.mathworks.com/connections](http://www.mathworks.com/connections)

### Worldwide CONTACTS

[www.mathworks.com/contact](http://www.mathworks.com/contact)

### E-MAIL

[info@mathworks.com](mailto:info@mathworks.com)

© 2010 The MathWorks, Inc. MATLAB and Simulink are registered trademarks of The MathWorks, Inc. See [www.mathworks.com/trademarks](http://www.mathworks.com/trademarks) for a list of additional trademarks. Other product or brand names may be trademarks or registered trademarks of their respective holders.

91877v00 08/10

## For More Information

- Stormy Attaway. *MATLAB: A Practical Introduction to Programming and Problem Solving*  
[mathworks.com/books/attaway](http://mathworks.com/books/attaway)
- Cleve Moler. *Experiments with MATLAB*  
[mathworks.com/moler/exm](http://mathworks.com/moler/exm)