

Using Matlab to Teach the Introductory Computer-Programming Course for Engineers

Asad Azemi
Department of Engineering
Pennsylvania State University
Delaware County Campus
Media, PA 19063
E-mail: azemi@psu.edu

Laura Pauley
Department of Mechanical and
Nuclear Engineering
Pennsylvania State University
University Park, PA 16802
E-mail: LPauley@psu.edu

Abstract

The introductory computer-programming course for engineers is usually taught using the C++ programming language. This work describes our current effort to introduce a pilot project, which can be used in an evaluation process by those departments that would like to substitute Matlab for C++. Those who would like to continue the current practice, but are looking for more challenging problems/projects involving Matlab can also use the project outcome. The project, at this time, is not calling for elimination of C++, but merely a reversal of our on-going practice at Penn State; namely, twelve weeks of Matlab and two weeks of C++. A discussion of the advantages and disadvantages of conducting a computer-programming course in this format are included.

I. Introduction

Computer programming has been part of the engineering curriculum since the dawn of the computer age. The course is typically taught during the freshman or first semester of the sophomore year to ensure that students have sufficient programming background for solving problems in engineering courses. Although the assignments usually require some mathematical and/or basic physics/engineering background, the course is focused around programming concepts. Most universities use a “teach-a-language” approach in teaching this course, which means students work with a general-purpose programming language (e.g. C++, Java, or FORTRAN) that is sufficiently flexible to build anything that needs to be built. This approach provides training in programming, but is so time consuming that there is little opportunity left to learn about computation.

II. Current practice

The introductory computer-programming course for engineers, which was once dominated by FORTRAN for several decades, was gradually changed to C and later to C++ during the late 1980’s and early 1990’s. Currently most universities require only one computer-programming

course for their engineering students (excluding computer engineering and computer science). This course is usually taught in C++ and in some cases in Java and FORTRAN. Currently at Penn State, we offer two versions of this course (C++ and FORTRAN), where the C++ version is required by the majority of the departments. The course covers fundamental concepts of programming (using C++ or FORTRAN), including introduction to computers and programming; data types, declaration and displays; assignment and interactive input; selection; repetition; functions; arrays; strings; and input/output methods, with emphasis on engineering and advanced mathematics problems, and a brief introduction to Matlab for two weeks. The Matlab portion includes how to use the Matlab editor/debugger; array and matrix operations; files, functions and data structures; selection; repetition and plotting. It should be mentioned that production of scientific graphics, using a general-purpose programming language, such as C++, is not an easy task and is beyond the scope of a normal introductory programming course. Matlab can be used to overcome this shortfall. Next, the justifications/benefits of incorporating a scientific simulation software will be discussed.

III. The role and benefits of simulation software packages

The benefits of using simulation software packages, such as Matlab, in various engineering courses have long been realized by many educators, e.g. [1]-[8]. One of the main advantages of using these tools is the reinforcement of student understanding of theoretical principles by means of enhanced graphical aids. Simulation results can be used effectively in the classroom to emphasize the characteristics of devices and show the similarities and differences that exist between actual and theoretical characteristics. The simple graphical nature of the simulation outputs tends to help students understand the operation of mathematically complex system behaviors. Another equally important advantage is the preparation for analysis and the design of more complex systems than those that can be treated with pencil and paper, much like the ones students will see in industry. With these tools, instructors can assign complex design problems that otherwise would be unrealistic without the help of such software. This is a key advantage that helps students apply the theoretical principles learned in the classroom to the real world problems associated with following a design cycle through completion. The benefits of using these packages in a university setting are confirmed by the number of new textbooks, and revisions of previously printed textbooks incorporating new exercises and problems based on these packages [9]-[19], among others.

IV. Why change?

As it was mentioned, the introductory computer-programming course for engineers at Penn State consists of teaching programming skills using C++ or FORTRAN with a limited exposure to Matlab. It has been recognized by many engineering faculty at Penn State, in various departments, that the current courses teaching programming skills using C++ or FORTRAN are not fully utilized in later required courses in the curriculum. Instead, increasingly in undergraduate courses in various engineering disciplines, they are using Matlab for problem

solving. Moreover, Matlab and its toolboxes, due to ease of use and functionality, have become important tools for simulation based engineering research and even in some experimental setups. Therefore, we have considered a pilot project to teach the programming course mainly using Matlab, with a limited exposure to C++ or FORTRAN. The limited exposure to C++ or FORTRAN is to satisfy those departments that feel their students need to have a general understanding of these programming languages. It should be mentioned there are some universities that either have converted their computer-programming course to Matlab [20] or offer different sections of the course with Matlab and C++/Java options [21].

V. Disadvantages of using Matlab as the main programming language

The main disadvantage of using Matlab as the main programming language is that it has not been put together to be a “programming language.” Its main purpose has always been to simplify the technical calculations/simulations. With this in mind, some basic programming rules such as variable declarations have been relaxed. Therefore, if one looks at Matlab from a purely computer science point of view, it would not be the choice to teach programming courses. Therefore, one needs to be more careful that it does not become a “teach-a-package” course. Another disadvantage is the limited available resources in using Matlab as a programming language, versus the wide range of books that are published for use with general-purpose programming languages such as C++ or FORTRAN. Moreover, Matlab may not be entirely suitable for cases that one needs to conform to commercial standards in industrial settings. Although, with the help of the Matlab compiler one can generate C files from m-files, which can be used for some settings [22].

VI. Advantages of using Matlab as the main programming language

By taking a “teach-a-language” approach using Matlab, we can teach general-purpose language skills and concepts and take advantage of its computational/graphical capabilities. This will give us the ability to discuss more advanced engineering/mathematical problems in just a short period of time, which cannot be accomplished with general-purpose programming languages. We should keep in mind that the main reason behind offering a programming course for engineering students is to help them with their future scientific computational tasks. We believe that Matlab can be used to serve this purpose. The fact that the majority of engineering jobs (excluding computer engineering which is not part of our discussion) do not require programming, but may require problem solving, should also justify the use of Matlab in engineering programming courses. Moreover, since Matlab is an integrated part of many advanced engineering courses and textbooks, an early exposure to this software is beneficial.

VII. Our proposed plan

Based on the presented discussion, and the fact that currently there are only limited resources available to teach Matlab as a general-purpose language [23]-[25], and there may be some departments that would like their students to be exposed to C++ or FORTRAN, we propose to teach the course, as a pilot project, by reversing the time spent on C++ and Matlab: namely,

twelve weeks on Matlab and three weeks on C++ or FORTRAN. The Matlab portion will cover an introduction to computers and programming; Matlab editor/debugger; data types, declaration and displays; assignment and interactive input; collections, indexing, and selection; repetition; functions; plotting and model building; input/output methods; numbers and precision; sounds and signals; and image processing. The C++ portion of the course will cover the syntax specific part of C++, as well as “good” programming practices that students would learn when taking a general computer-programming course.

The pilot project is a collaborative work between the Department of Engineering at Penn State Delaware County and the Department of Mechanical and Nuclear Engineering at Penn State University Park. Matlab assignments/examples will come from both EE courses (e.g. electric circuits, control systems, communications) and ME courses (e.g. dynamics, vibration, and systems control). We also propose that students taking the proposed engineering programming course have taken or are currently taking the MATH 220 (matrices) course.

Conclusion

In this paper, we have presented our pilot project to use Matlab as the major part of a general-purpose introductory programming course for engineers. Advantages of such a move would be the ability to incorporate more advanced engineering/mathematical assignments as well as helping students with their future courses that use Matlab. The main disadvantage of such a move is the extra work that is required for putting together the lectures, due to limited resources that focus on using Matlab as a general-purpose programming language. The proposed modified course will also include several lectures in C++ and “good” programming practices.

References

- [1] T.W. Martin, A. Azemi, D. Hewett, and C.P. Schneider, “PSpice in Electrical Engineering Laboratories,” *Proceedings of the 1992 ASEE Annual Conference*, pp. 1307-1308.
- [2] D. Andrews, A. Azemi, S. Charlton, and E. Yaz, “Computer Simulation in Electrical Engineering Education,” *Proceedings of the 1994 ASEE Gulf-Southwest Section Meeting*, pp. 77-82.
- [3] A. Azemi and E. Yaz, “PSpice and MATLAB in Undergraduate and Graduate Electrical Engineering Courses,” *Proceedings of the 24th Frontiers in Education Conference*, pp. 456-459, 1994.
- [4] E. Yaz and A. Azemi, “Utilizing MATLAB in two Graduate Electrical Engineering Courses,” *Proceedings of the 25th Frontiers in Education Conference*, pp. 2c6.1-2c6.4, 1995.
- [5] A. Azemi and C. Stook, “Utilizing MATLAB in Undergraduate Electric Circuits Courses,” *Proc. of the 26th Frontier in Education Conference*, Salt Lake, UT, vol. 1, pp. 599-603, 1996.
- [6] A. Azemi and E. Yaz, “Utilizing SIMULINK and MATLAB in a Graduate Nonlinear Systems Analysis Course,” *Proc. of the 26th Frontier in Education Conference*, Salt Lake, UT, vol. 1, pp. 595-599, 1996.
- [7] A. Azemi, and E. Yaz, “Using MATLAB in a Graduate Electrical Engineering Optimal Control Course,” *Proceedings of the 27th Frontier in Education Conference*, pp. 13-17, 1997.
- [8] A. Azemi, and E. Yaz, “Using Graphical User Interface Capabilities of MATLAB in Advanced Electrical

- Engineering Courses,” *Proceedings of the IEEE Control and Decision Conference*, pp. 4549-4554, 1996.
- [9] R. Dorf, and R. Bishop. *Modern Control Systems*, ninth edition. Addison-Wesley Publishing Company, 2001.
 - [10] G. Franklin, J. Powell, and Emami-Naeini. *Feedback Control of Dynamic Systems*. fourth edition. Prentice Hall, 2002.
 - [11] M. Roden. *Analog and Digital Communication Systems*. Discovery Press, 2003.
 - [12] S. Haykin. *Communication Systems*, fourth edition. John Wiley, 2001.
 - [13] J.G. Proakis, and M. Salehi. *Contemporary Communication Systems Using MATLAB*. Brooks/Cole Publishing Company, 2000.
 - [14] L.H. Turcotte, and H.B. Wilson. *Computer Application in Mechanics of Materials Using MATLAB*. Prentice Hall, 1998.
 - [15] D.M. Auslander, J.R. Ridgely, and J.D. Ringgenberg. *Control Software for Mechanical Systems: Object-Oriented Design in a Real-Time World*. Prentice Hall, 2002.
 - [16] P.I. Kattan. *MATLAB Guide to Finite Elements*. Springer-Verlag, 2003.
 - [17] B.D. Harper. *Solving Statics Problems in MATLAB*. John Wiley, 2002.
 - [18] R.C. Gonzalez, E. Woods and S.Eddins. *Digital Image Processing Using MATLAB*, Prentice Hall, 2004.
 - [19] D.T. Kaplan. *Introduction to Scientific Computation and Processing*. Brooks/Cole, 2004.
 - [20] M.E. Herniter, and D.S. Scott, “Teaching Programming Skills with MATLAB,” *Proceedings of the 2001 ASEE Annual Conference*, session 1520.
 - [21] P.E. Devnes, “MATLAB and Freshman Engineering,” *Proceedings of the 1999 ASEE Annual Conference*, session 3353.
 - [22] R. Bachnak, and R. Lee, “The MATLAB Compiler Suite: M-Files to C/C++ Executable Programs,” *Proceedings of the 2002 ASEE Annual Conference*, session 1520.
 - [23] W.J. Palm III. *Introduction to MATLAB 6 For Engineers*. McGraw-Hill, 2001.
 - [24] A. Gilat. *MATLAB: An Introduction with Application*. John Wiley, 2004.
 - [25] F. Gustafsson and N. Bergman. *MATLAB for Engineers Explained*. Springer-Verlag, 2003.

ASAD AZEMI

Asad Azemi is an Associate Professor of Engineering at Penn State University. He has received his B.S. degree from UCLA in 1982, M.S. degree from Loyola Marymount University in 1985, and Ph.D. degree from University of Arkansas in 1991. His professional interests are in nonlinear stochastic systems, signal estimation, neural networks, bioinformatics, and use of computers in undergraduate and graduate education.

LAURA L. PAULEY

Laura Pauley is a Professor of Mechanical Engineering at Penn State University, University Park. Since 2000, she has also served as the Professor-in-Charge of Undergraduate Programs in Mechanical and Nuclear Engineering. Dr. Pauley teaches courses in the thermal sciences and conducts research in computational fluid mechanics. She received her Ph.D. from Stanford University in 1988.