# Incorporation of Fourth Generation Computing Environment into a Freshman Engineering Program: An Historical Perspective

**Christopher Rowe, Richard Shiavi, Jim Tung**
**Vanderbilt University/The MathWorks**


## Abstract

Historically, the incorporation of computing into Vanderbilt University School of Engineering's (VUSE) curricula began in the mid-1960s. Mainframe computers were purchased, and students initially learned computing using the language FORTRAN (formula translation). The early 1980s saw two significant advancements: the introduction of personal computers made computing hardware much more accessible to more people, while the introduction of fourth generation languages such as MATLAB® made it possible to solve real-world problems much more rapidly. This combination of advancements challenged engineering professors to provide students with the most up-to-date tools, techniques, and information.

In 1989, it was decided to introduce VUSE's first-year engineering students to techniques in engineering computing. The school required a technical computing environment that offered an interactive format that was relatively easy to learn, and applied frequently in varying engineering disciplines in industry. MATLAB, invented by Cleve Moler, co-founder of The MathWorks and longtime professor, had been introduced on the market several years earlier. MATLAB provided an interpretive capability along with an intuitive computing environment for expressing and solving problems. VUSE selected MATLAB as the school's principal computing environment.

This document will explore how VUSE's experience with MATLAB has evolved over the last 14 years with the environment's increasing capabilities. The 14-year history can be divided into three segments: 1989-1995, 1995-2002, and 2002 to present, based on the constantly changing educational demands of an introductory course. Offering integrated computation, data analysis, visualization, and programming in one package, MATLAB's ease of implementation and widespread use as a technical computing environment has made it appropriate to be used today in an introductory course for teaching computing principles, as well as a unifying environment that supports the development of a coordinated multi-year engineering curriculum.


## Introduction

A common challenge for most engineering schools is designing an introductory course that properly prepares freshmen for upper-level courses while giving them an initial impression of the profession of engineering. Designing a course that provides for a smooth transition to engineering school while effectively using students' already-obtained knowledge is proving to be increasingly difficult for college educators. A component of this situation is the fact that students have increasingly diverse amounts of advanced placement credit. Many engineering schools are placing the focus of introductory engineering courses on computing tools to perform analytical problem solving functions. Because of this migration it is even more critical to the success of a freshman engineering program to find a software platform that is versatile, easy to use, and utilized after students complete their formal engineering training. In 1989, VUSE decided to adopt MATLAB into its new and innovative introductory course, Engineering Science (ES) 130, focusing on computer modeling and simulation techniques for solving engineering and math problems. Since technology was just becoming user-friendly, MATLAB was a good choice because of its uncommon combination of ease of use and breadth of functionality.

MATLAB originated in the late 1970's when Cleve Moler wanted to provide interactive access to the FORTRAN linear algebra software packages EISPACK and LINPACK, motivated by his belief that a person should not have to learn FORTRAN in order to learn numerical computation. MATLAB was initially focused on constructing and manipulating matrices, and applying algorithms for eigenanalysis and linear algebra ("MATLAB" stands for "matrix laboratory", reflecting these origins). In 1984, Cleve Moler and Jack Little founded The MathWorks and released the first commercial version of MATLAB for MS-DOS personal computers. By then, MATLAB had become a full-featured programming language, whose noteworthy features included:

- Interpretive yet fast execution, which encouraged interactive exploration
- Numeric computation and graphics combined in a single environment
- Hundreds of built-in functions for technical computing
- A high-level matrix-based language (so that operations can be performed on an entire set of data without requiring *for* loops) with seamless complex arithmetic support (key for engineering), and built-in memory management (so users could focus on the algorithms and problem-solving rather than programming tasks such as declaring variables, datatyping, etc.)
- Extensibility, which enabled users to easily create their own functions and share them with others
- Multi-platform availability, ranging from PCs to engineering workstations to minicomputers to supercomputers

The extensibility of the MATLAB language led to the creation of "toolboxes", add-on packages of MATLAB functions for a specific application or technique, such as signal processing, control system design and analysis, or optimization. The relatively simple MATLAB syntax made it easy to learn, even for users without a programming background. Its high level computing methods resulted in efficient programs. It is possible to carry out computations in a line or two of MATLAB code that would, otherwise, require dozens or hundreds of lines in FORTRAN or C/C++. Its interpretive, interactive mode allows users to explore ideas without having to write a script or program.

## The Introductory Curriculum & Computing (1989-1995)

For the first two years the ES130 syllabus was discrete time simulation and computer graphics. It was observed that it was too difficult for many students to grasp these topics. The syllabus was changed and the focus of this course became studying an array of topics in engineering in the standard analytical fashion. The innovative component of this course was that for the first time it implemented a computer laboratory session once per week. In addition to working standard introductory engineering problems out of a customized textbook, simple problems were presented and solved in the computer laboratory using MATLAB. These problems were largely consisting of simple data sets and graphical plotting. Since technology was not nearly as widely available, prior student computing experience was very low or nonexistent. The laboratory time was spent learning the software features and how to perform basic calculations using the matrix-based platform. This period of time was also before sophisticated graphical user interfaces had become widely popular and most students did not have a computer of their own. They had to rely on using the School of Engineering's computer laboratories almost exclusively.

**Introductory Problem Solving using Computers (1995-2002)**

In the mid-1990's, VUSE's incoming students started to possess an increasing range of experiences and skills in using computers; however, the vast majority were only slightly more computer proficient than in the recent past. The challenge of delivering an effective introductory course shifted from simply exposing students to the available technology to beginning to incorporate the computing as a component of their problem solving skills. One of the most significant decisions that had to be made was whether or not the modality of the course should be changed. Previously, the format of course consisted of two lecture sessions per week with a once per week computer laboratory session. The question arose whether instruction would be more effective if the course was conducted exclusively in a computer laboratory. A three-year study of this issue was conducted (Shiavi et al., 2000; Shiavi and Brodersen, 2002). Based on the results the format of the course was changed from a traditional lecture/laboratory modality to an all-laboratory modality. As the years passed, more and more students were bringing their desktop and laptop computers to the university or purchasing them shortly after enrolling in first-year courses.

During this time, a Windows-based version of MATLAB was introduced on the market. With the improvement in user interface came innovations in the capabilities of the software. MATLAB went from a user interface consisting of a single command-line on the MS-DOS screen to several user-friendly interfaces, including the GUI-based MATLAB Desktop, the MATLAB Notebook and user-created GUI applications. Updated versions of MATLAB provided an opportunity for students to gain additional computing experience, further developing their problem-solving skills, and start to explore how computing tools can be applied to problem-solving approaches.

Unlike general-purpose programming languages, MATLAB requires at least fundamental knowledge of mathematical concepts in the usage of its functions. The use of MATLAB allows concepts to be reinforced from classroom learning directly to computer application. When the instructional modality changed to an exclusively laboratory format, it further ensured the effectiveness of teaching computing as a vehicle for developing engineering problem solving skills. For example, the addition of the MATLAB Notebook feature enabled the integration of the powerful MATLAB environment with the widely popular word processing environment MS Word. The advantage of this integration was the ability for students to take an analytical problem, solve it, and present the results in an easy to understand, well-organized report format. Students also gained initial experience in software integration possibilities. The use of the MATLAB Notebook feature also allowed the introduction of more technical writing concepts, further increasing the value of the introductory course to the overall engineering curriculum.

Throughout the many changes in student skill and experience, software tools changed equally fast: Quattro Pro was replaced with Microsoft Excel, Visual Basic was replaced with markup languages HTML/VRML, and CADKEY was relocated in the curriculum as a result of programmatic changes in subsequent semesters. Regardless of these changes, MATLAB remained an integral component of the curriculum as a result of its own continuous innovations and expansions of capabilities.

**Introduction to Engineering Problem Solving & Computing (2002-present)**

Approximately two years ago, it was becoming more evident that students were entering the School of Engineering with significantly more computing skills and abilities. The availability of word processing and spreadsheet programs had reached new heights as a result of most high schools incorporating the software packages into their programs. In 2001-2002, VUSE launched its laptop initiative with the goal of equipping each engineering student with his or her own wireless laptop computer. The proliferation of Windows-based computing packages made this program introduction possible with nearly all entering students having at least basic abilities in their usage. This program is currently in its second full year of implementation and has been very successful. With all freshmen having identical laptops, the need for the introductory course to be taught in a traditional computer laboratory was eliminated. Now any classroom within the engineering building complex can instantaneously be converted to a wireless computer classroom while maintaining the instant transition from presentation to practice of course concepts.

The course, in its current format, was being stretched to its limits with computing material, but had not significantly changed its problem solving approach. Students had gained basic skill in using computing devices by this time, thus the need arose for a radical innovation in course content and format.

In 2003, the course underwent a radical redesign where its focus was shifted from a skills-based approach with a combination of several software packages to a more intense analytical problem solving, challenge-based approach using only pencil/paper, Excel, and MATLAB. The goals of this innovation were to more thoroughly introduce students to and educate them on the various major courses of study within engineering, while giving them an intense and challenging introduction to engineering as a profession and the computing tools to be successful as an engineering student. In order to maximize student skill and engineering content, the decision was made to implement a module-based course beginning with basic engineering problem solving content and completing with discipline-specific modules emphasizing real-world examples based on current events/research within the professional field. Much of the decision on the exact new format was based on the accumulated findings of the recently established VaNTH ERC (Vanderbilt, Northwestern, Texas, Harvard-MIT Engineering Research Center in Bioengineering Education) at VUSE.

The first module spans the first half of the semester and consists of the analytical computing components that formed the foundation of the course for more than 10 years. This module focuses on using the engineering problem solving process to formally set up and solve contextual engineering problems using traditional paper and pencil, Excel, and MATLAB. Emphasis on teaching the features of the computing environments was reduced and left partially to student discovery, and efforts to reinforce the problem solving process were strengthened. As a result, more analytical methods were introduced including optimization, descriptive statistics, and mathematical modeling. These methods were taught in addition to basic methods such as mathematical functions, graphing, regression statistics, programming basics, symbolic math, and modeling and simulation. Similar methods were introduced in Excel and MATLAB in order to illustrate specific strengths and features of each tool.

The second half of the semester was divided into two discipline-specific modules in which students select two modules of interest. Individual departments were responsible for creating a challenge-based module consisting of a loose set of design criteria. The modules contained a significant analytical component building on the concepts presented in the general engineering problem solving module. MATLAB was widely used in many of these modules in more advanced contexts than in the first general module. For example, the biomedical engineering module focused on various brain imaging modalities. This module contained a thresholding assignment requiring students to use the image processing toolbox to determine approximate tumor volume using a series of digitized brain scans. Applications of MATLAB in other discipline-specific modules incorporated scripts, advanced matrix calculations, and various regression statistics as they related the module content in a real-world or research intensive context.

**Vertical Integration/Expanded Usage**

Through consistent software innovations over the past 14 years, MATLAB has played a large role in enabling the curriculum to move from a "how to use the tool" emphasis to a "how to solve engineering problems" emphasis. Over the time that MATLAB has been a significant component of the introductory engineering course, upper-level engineering courses have been redesigned to incorporate MATLAB not only as a computing tool but a learning environment. A new version of the introductory computer programming course was launched using MATLAB as the programming environment. MATLAB was chosen as the environment because of its ease of use, powerful programming applications, and prior student knowledge of the software functionality and programming language. Other upper-level courses include the use of add-ons to MATLAB, including the Image Processing Toolbox, the Signal Processing Toolbox, and Simulink®. The addition of MATLAB into traditional core engineering courses is a great example of the vertical integration of this versatile computing tool into engineering programs.

## Bibliography


Shiavi, R., Brodersen, A., Bourne, J., Pingree, A., Comparison of Instructional Modalities for a Course "Introduction to Computing in Engineering", *Frontiers in Education Conference*, Kansas City, Missouri, October, 2000.

Shiavi, R., Brodersen, A., "Study of Suitability of Studio Model for Introductory Computing", ASEE Annual Conference, Montreal, Canada June 16-19,2002.


## Biographical Information

CHRISTOPHER ROWE received his Bachelor of Engineering degree in Biomedical Engineering and Master of Engineering degree in Management of Technology from Vanderbilt University in 1996 and 1998, respectively. He joined the Engineering faculty in January 2003. His research interests include technical program and project management and serves as the Director of the Freshman Year for the Engineering Dean's Office.

RICHARD SHIAVI is a Professor of Biomedical Engineering, Electrical Engineering. His main professional interests are in applied signal processing and innovations in engineering education. He is serving presently as Associate Editor for the IEEE Transactions on Biomedical Engineering.

JIM TUNG is a MathWorks Fellow at The MathWorks, where he joined in 1988. His responsibilities include market and technology strategy and analysis, along with the company's book and Connections partner programs. He has over 20 years' experience in the technical software and data acquisition markets. He has a bachelor's degree from Harvard University.