Engineering Educators Bringing the World Together
2025 ASEE Annual Conference & Exposition
Palais des congrès de Montréal, Montréal, QC • June 22–25, 2025
Paper ID #45523

Bridging Theory and Practice: Active Learning and Real-World Applications in Mathematical Analysis Course

Dr. Djedjiga Belfadel, Fairfield University

Dr. Djedjiga Belfadel is an Associate Professor and the Graduate Program Director for Electrical and Biomedical Engineering at Fairfield University, Connecticut. She earned her Ph.D. in Electrical and Computer Engineering from the University of Connecticut in 2015. Dr. Belfadel is deeply committed to teaching, with a focus on innovative, hands-on learning approaches in STEM education. She has successfully integrated active learning strategies into her courses, particularly in first-year engineering programs, to enhance student engagement and retention. Her teaching emphasizes providing students with practical, real-world experiences that foster critical thinking, innovation, and problem-solving skills. Through her teaching and research, Dr. Belfadel is dedicated to inspiring and preparing the next generation of STEM professionals, with a particular emphasis on increasing the representation of women and underrepresented groups in the field. Her contributions to both undergraduate and graduate education continue to have a significant impact on her students and the broader academic community.

Danushka Bandara, Fairfield University

DANUSHKA BANDARA received the bachelor's degree in Electrical Engineering from the University of Moratuwa, Sri Lanka, in 2009. He received his master's and Ph.D. degrees in Computer Engineering and Electrical and Computer Engineering from Syracuse University, Syracuse, NY, USA, in 2013 and 2018, respectively. From 2019 to 2020, he worked as a Data Scientist at Corning Incorporated, Corning, NY, USA. Currently, he is an Assistant Professor of Computer Science and Engineering at Fairfield University, Fairfield, CT, USA. His Current research interests include Applied machine learning, Bioinformatics, Human-computer interaction, and Computational social science.

Bridging Theory and Practice: Active Learning and Real-World Applications in Mathematical Analysis Course

Djedjiga Belfadel, and Danushka Bandara
Fairfield University

Abstract:

This Evidence-Based Practice Paper outlines the redesign of the sophomore-level *Mathematical Analysis* course at Fairfield University, focusing on enhancing the practical application of mathematical concepts for engineering students. The course has been redesigned from a predominantly theoretical framework into a hands-on, project-based approach. Building upon the successful integration of MATLAB, the redesign emphasizes active learning techniques and interactive programming tasks to enhance student engagement. The redesign also incorporates computer science education concepts such as algorithmic thinking, software engineering practices, and computational thinking. These methods provide students with practical experience in addressing real-world engineering challenges.

A cornerstone of the redesign is a practical project in which students gather and analyze real-time data from a flying drone equipped with an Inertial Measurement Unit (IMU). Through MATLAB, students process acceleration data, applying numerical integration to compute both velocity and position. This project reinforces essential concepts such as integration, data analysis, and mathematical modeling, while also highlighting the relevance of these techniques in real-world engineering scenarios.

The course structure strikes a balance between fundamental theory and practical applications. In the first half of the semester, students acquire core MATLAB skills, including loops, conditional statements, and plotting, alongside key mathematical principles. The second half delves into more advanced topics, such as numerical solutions for nonlinear equations, matrix analysis, and statistical modeling. This phase culminates in applied projects like the IMU data analysis, which integrates these advanced mathematical methods with practical, real-world problem-solving.

Flipped classroom techniques are employed to support this active learning environment. Students engage with online modules before class, freeing up lecture time for interactive programming exercises and collaborative problem-solving. Peer Learning Group (PLG) sessions also provide extra opportunities for practice and peer-assisted learning.

Preliminary feedback and assessment data suggest that this project-based approach significantly enhances students' understanding of mathematical and computational concepts and their ability to apply them in engineering contexts. By integrating MATLAB programming with real-world applications, the course prepares students with both the theoretical foundation and practical expertise required for advanced coursework and professional engineering challenges.

1. Introduction:

The growing complexity of engineering problems requires students to master computational tools early in their education. Many engineering programs address this need by incorporating programming courses, often using languages like Python, Java or C. However, these languages can pose challenges for students who are still developing their mathematical and problem-solving skills [1,6]. To better support students, the sophomore-level Mathematical Analysis course at Fairfield University has been redesigned to integrate MATLAB and its companion tool, Simulink, providing an accessible and practical introduction to computational problem-solving tailored to engineering contexts.

Engineering students often approach programming with a focus on solving engineering problems rather than mastering computational theory. Computer science pedagogy, encompassing methods such as active learning, project-based learning, and sociocultural approaches, offers structured frameworks to facilitate learning. These methods emphasize engagement, practical application, and collaboration, which are

particularly beneficial for students new to programming. The challenge lies in adapting these approaches to teach MATLAB, a matrix-based language widely used in engineering, in a way that is accessible and relevant to sophomore students who are the target group for this course.
The redesigned course introduces several key updates to create a more dynamic and applied learning experience. Simulink has been added to help students model and simulate dynamic systems, complementing MATLAB's computational capabilities. Additionally, an online MathWorks training module now enhances students' learning, offering certifications that validate their growing technical skills. A new team project further enriches the course by encouraging students to connect mathematical concepts with their own engineering disciplines. By selecting a topic covered in class and applying it to a real-world problem, students not only deepen their understanding but also develop teamwork and communication skills through classroom presentations.

This course stands out due to its inclusion of weekly 75-minute PLG sessions. These workshops, led by a teaching assistant, offer hands-on programming practice beyond lectures, reinforcing core concepts. The PLG is a non-credit corequisite, taught by a proficient former student, with all materials provided by the faculty. There is no direct grade assigned to the PLG because students are completing their Programming assignments during the PLG. The focus is to give students confidence to start writing code from scratch and let them develop their own programming style.

Through these updates, the course transitions from a predominantly theoretical focus to an applied, hands-on learning experience. By blending computational tools, real-world applications, and collaborative problem-solving, students are better prepared to tackle advanced coursework and engineering challenges in their future careers.

The primary motivation for redesigning the Mathematical Analysis course was to provide engineering students with an accessible introduction to programming, equipping them with the skills to use MATLAB for numerical calculations and data analysis. Prior to the redesign, many students entered the course with little or no programming experience, which limited their ability to apply mathematical concepts in practical engineering contexts. By integrating MATLAB and project-based learning, the course aims to bridge this gap, allowing students to develop both the computational skills and mathematical understanding needed for advanced coursework and real-world problem-solving.

2. Literature Review

A substantial body of research supports the use of MATLAB in introductory programming courses for engineering students due to its discipline-specific relevance and accessibility for non-computer science learners [10-13]. Active learning has emerged as a highly effective pedagogical strategy in STEM education, with a meta-analysis of 225 studies demonstrating that it improves examination performance by approximately half a letter grade and reduces failure rates by 55% compared to traditional lecturing [8]. Flipped classrooms in particular have seen success in computational modeling courses [16]. Furthermore, incorporating real-world data into mathematical analysis courses has been shown to significantly enhance student engagement [14-15]. These findings underscore the value of integrating active learning and real-world applications, along with MATLAB-based projects, to optimize educational outcomes in engineering curricula.

3. Considerations for Course Integration:

Integrating MATLAB and Simulink into the Mathematical Analysis course introduces a valuable opportunity for students to connect computational tools with mathematical concepts. However, as with any curriculum enhancement, several challenges must be carefully managed to ensure the course remains effective and accessible:

- **Steep Learning Curve:** For students with limited programming experience or weaker mathematical foundations, learning MATLAB and Simulink while tackling advanced mathematical topics can be overwhelming. The dual focus on computational tools and theory might discourage students if adequate support is not provided, potentially affecting their confidence and academic performance [4].

- **Balancing Course Content:** The inclusion of Simulink modules, online MathWorks training, and team projects enriches the curriculum but also adds to its intensity. This additional workload can limit the time available for covering theoretical concepts in depth. Striking a balance between practical applications and mathematical rigor is critical to ensuring students develop a comprehensive understanding of both.

- **Complexity in Assessment:** The expanded course structure introduces multiple layers of evaluation, such as programming tasks, mathematical problem-solving, and collaborative team projects. This can make it difficult for instructors to pinpoint the root causes of a student's struggles—whether they stem from programming, mathematical concepts, or the integration of both. Designing effective assessments and providing targeted feedback require additional planning and resources.

- Limited field specific knowledge of the students: Students' high school backgrounds offer little common ground due to the national and international diversity of standards. Even at the first year in college, some students excel at their college courses, and others requiring much additional study [9].

4. Course Overview

The "Mathematical Analysis" course at Fairfield University integrates mathematical concepts with programming to prepare engineering students for solving complex, real-world problems. With a small class size of about 15 to 20 students, the course provides a highly interactive and personalized learning experience, encouraging close collaboration between instructor, teaching assistant, and students.

The course is designed for sophomore engineering students who have completed Calculus I and II and an Introduction to Engineering fundamentals course. While students typically have a strong mathematical background, many enter with little or no prior programming experience, particularly with MATLAB or similar computational tools. To address this, the course introduces MATLAB programming skills progressively, ensuring that students can engage with numerical methods and computational problem-solving regardless of their starting point.

The course is organized into two distinct phases. The first focuses on foundational programming skills, introducing students to MATLAB for numerical computing. This includes topics such as vector and matrix operations, control structures, advanced functions, symbolic calculations, and plotting techniques. Simulink has been incorporated to teach system modeling and simulation, expanding students' computational toolkit. Starting right after the midterm, the course shifts into its second phase, delving into more complex topics in mathematical analysis, including Root Finding & Optimization, Linear Algebra, Statistics, Curve Fitting, Interpolation, and Numerical Integration & Differentiation.

To optimize hands-on learning, the course adopts a flipped classroom model. Students engage with pre-class material short instructional videos and preparatory worksheets before attending lectures. Class time is then devoted to practical applications, problem-solving, and interactive activities. Each session follows a structured learning process, transitioning from instructor demonstrations to group problem-solving and individual practice. Collaborative discussions and solution reviews at the end of each session ensure key concepts are reinforced.

5. Course Organization

Phase 1: Building Programming Foundations
The initial part of the course introduces essential MATLAB skills and emphasizes their application in solving engineering problems.

Key topics include:
- MATLAB Interface: Navigation and command basics.
- Vectors and Matrices: Core operations and their applications.
- Scripts & Functions: Writing efficient and reusable code.
- Control Flow: Conditional statements and loops for logical operations.
- Advanced Functions: Tackling complex problems with specialized tools.
- Symbolic Calculations & Data Visualization: Using symbolic math and creating informative plots.
- App Design: Building simple user interfaces with MATLAB.
- Simulink: Simulating dynamic systems to complement numerical programming.

Phase 2: Advanced Applications of Mathematical Analysis
The second phase of the course transitions from programming fundamentals to advanced mathematical techniques, focusing on real-world applications. Students first explore the theory behind each concept through manual problem-solving before implementing algorithms in MATLAB. Topics include:
- Error Measurement: Understanding true and approximate errors, as well as absolute and relative measures.
- Root Finding & Optimization: Solving equations with techniques like bisection, Newton-Raphson, and secant methods.
- Linear Algebra: Addressing systems of equations, matrices, and transformations.
- Numerical Integration & Differentiation: Applying methods like trapezoidal and Simpson's rules, and finite difference techniques.
- Data Analysis: Statistical tools for interpreting data and drawing insights.
- Interpolation: Techniques such as Lagrange, Newton, and spline methods for estimating values.
- Regression Analysis: Developing models to analyze relationships between variables and evaluate data trends.

Each week, students are assigned 2 HomeWorks: an initial weekly assignment - a preparatory worksheet to be completed before the lecture, graded mainly for completion, using pre-class videos and readings provided; and a second main assignment - programming assignments aligned with lectures and PLG sessions, due at the end of the second lecture and PLG session of that week and graded by teaching assistants. Additionally, the curriculum integrates online self-paced training auto-graded modules from the MathWorks platform.

An essential component of the course is the team project, introduced during the second phase. Students select a course topic and apply it to a problem relevant to their engineering discipline. This project encourages creativity, teamwork, and critical thinking while emphasizing real-world applications. Each team presents their findings to the class, honing their professional communication skills and showcasing the practical relevance of their learning.

6. **Course Outcome:**

[ ] course outcome link to the Blooms Taxonomy levels goal ( ) link to ABET student outcomes

1. Show proficiency in MATLAB including the understanding of the workspace, using m-files, graphics and plotting, and vector manipulation. [I] (1)
2. Demonstrate mastery of mathematical, numerical, and statistical engineering topics such as matrix algebra, data analysis and statistics, data interpolation, curve fitting, integration, and differentiation [II] (2)
3. Identify how programming and mathematical content applies to the field of engineering and understand the impact of engineering solutions in a global economic, environmental, and societal context. [I,II] (4) knowledge

This course supports ABET Student Outcomes: (1, 2, 4)

- ABET 1 an ability to identify, formulate, and solve complex engineering problems by applying principles of engineering, science, and mathematics.
- ABET 2 an ability to apply engineering design to produce solutions that meet specified needs with consideration of public health, safety, and welfare, as well as global, cultural, social, environmental, and economic factors.
- ABET 4 an ability to recognize ethical and professional responsibilities in engineering situations and make informed judgments, which must consider the impact of engineering solutions in global, economic, environmental, and societal contexts.

7. **Grade Distribution:**

- Participation 5%
- Homework (WS) 10%
- Programing Assignments (PA) 20%
- Mathworks Certificates 10%
- Team Project 5%
- Quizzes 10%
- Midterm Exam 20%
- Final Exam 20%

8. **Integrating MATLAB Online Training into the Curriculum**

To enhance in-class learning and Peer Learning Group (PLG) sessions, students are required to participate in MATLAB's online training programs. These self-paced, auto-graded, interactive courses provide hands-on experience and are freely available to students. They cover a wide range of topics, from foundational skills to advanced applications, allowing students to progress at their own pace while earning certifications to demonstrate their expertise.
Key courses included in the curriculum are as follows:

- **MATLAB ONRAMP:** This beginner-friendly course introduces students to MATLAB's fundamental features. It covers topics such as basic commands, data importation, and data visualization. The course culminates in a final project where students analyze astronomical data, bridging the gap between theoretical understanding and practical application.

- **MATLAB Fundamentals:** Building on ONRAMP, this course provides a comprehensive exploration of MATLAB programming, with an emphasis on data analysis, visualization, and problem-solving. It is particularly suited for students aiming to deepen their programming skills beyond the basics.

- **App Design with MATLAB:** For students interested in creating graphical user interfaces (GUIs), this course introduces techniques for designing and building interactive applications. These skills are especially valuable for those who wish to apply programming concepts in a teaching or professional context.

- **Specialized MATLAB Courses:** Students are encouraged to further explore advanced modules on the MathWorks training portal, such as Simulink Onramp, Data Processing and Visualization, and MATLAB Programming Techniques. These courses provide specialized knowledge tailored to various engineering fields, offering tools to solve complex, real-world problems.

- **Certificates and Recognition:** Students who complete these courses earn certificates from MathWorks, validating their technical skills and providing tangible evidence of their learning progress. These certificates also serve as an added incentive for active participation in the training.

Topic Description of the ONRAMP [7]

Table I: Main topics of MATLAB Onramp training

Topic	**Description**
Commands	Enter commands in MATLAB to perform calculations and create variables
Vectors and Matrices	Create MATLAB variables that contain multiple elements
Importing Data	Bring data from external files into MATLAB
Indexing into and Modifying Arrays	Use indexing to extract and modify MATLAB arrays
Array Calculations	Perform calculations on entire arrays at once
Calling Functions	Call functions to obtain multiple outputs
Obtaining Help	Use the MATLAB documentation to discover information about MATLAB features
Plotting Data	Visualize variables using MATLAB's plotting functions
MATLAB Scripts	Write programs in script
Logical Arrays	Use logical expressions in MATLAB
Final Project	Bring together the introduced concepts with a project

9. Student Feedback

The feedback from students offered meaningful insights into the course's strengths and opportunities for improvement. Students largely appreciated the practical focus of the curriculum while providing constructive suggestions to further improve the learning experience.

What Students Valued
Students highlighted the programming assignments and Peer Learning Group (PLG) sessions as some of the most effective components of the course. These aspects were praised for making abstract concepts more tangible and fostering collaboration among peers. Many students also shared that learning MATLAB enhanced their interest in programming, emphasizing the relevance of integrating computational tools into engineering education.

Areas for Growth
Several students suggested incorporating more in-class demonstrations to help bridge the gap between theory and application. Adjustments to the distribution of coursework were also recommended, with some students seeking a better balance to ensure the workload remains engaging but manageable.

The MathWorks online training modules received mixed reactions. While some students appreciated the structured approach to learning advanced MATLAB skills, others felt the modules could be streamlined or better aligned with course objectives to maximize their value.

Guiding Future Improvements
This feedback, alongside faculty observations, will play a key role in shaping future iterations of the course. Planned enhancements include:

- Adding more guided, in-class examples to reinforce key concepts.
- Optimizing the workload to maintain student engagement while preventing overload.
- Refining the integration of online training resources to ensure better alignment with course goals and to meet student needs effectively.

In addition to qualitative feedback, future versions of the course will incorporate structured evaluation methods to assess the redesign's impact on student learning. Planned efforts include the use of pre- and post-course assessments to measure improvements in students' mathematical understanding and programming skills. There is also an intention to explore tracking students' performance in subsequent engineering courses to evaluate the long-term transfer of these skills.
These assessment strategies will help ensure that the course not only enhances student engagement but also supports sustained academic success and practical skill development.

10. Conclusion

The integration of MATLAB programming into the *Mathematical Analysis* course at Fairfield University has proven to be an effective approach to bridging the gap between theoretical mathematics and practical applications. By combining interactive visualization with hands-on programming, the course enhances students' understanding of mathematical concepts while developing essential skills in problem-solving, computational thinking, and teamwork.

Student feedback and demonstrated improvements in programming proficiency highlight the value of this blended approach. The course prepares students for advanced studies and equips them to address real-world engineering challenges with confidence. The use of MATLAB, along with MathWorks online training modules, ensures that students gain both technical expertise and practical experience.

Although the current model is well-suited to smaller class sizes, expanding it to larger groups will require adjustments. A flipped classroom design, where students engage with foundational material before class, supports active programming practice and collaborative problem-solving during class time. Adding structured group work, peer feedback opportunities, and teaching assistant support will further enhance the learning experience and maintain its interactive nature.

Requiring MathWorks certification and using auto-graded programming modules help ensure consistent engagement and measurable learning outcomes. This approach provides a solid framework for fostering active learning and supports the development of both individual and team-based skills in engineering education.

References

[1] P. T. Goeser, W. Johnson, S. L. Bernadin, and D. A. Gajdosik-Nivens, "*Work-in-Progress: The Impact of MatLab Marina - A Virtual Learning Environment on Student Learning in a Computing for Engineers Course*", ASEE Annual Conference and Exposition, 2013.

[2] R. Talbert, "*Learning MATLAB in the Inverted Classroom*", ASEE Annual Conference and Exposition, 2012.

[3] K. Larsen, A. Hossain And M. Weiser, "*Teaching an Undergraduate Introductory MATLAB Course: Successful Implementation for Students Learning*". ASEE Annual Conference and Exposition, 2016.

[4] D. Belfadel, M. Zabinksi and I. Macwan. *Introduction to MATLAB Programming in a Fundamentals of Engineering Course,* ASEE: Annual Conference and Exposition, Long Beach, California, July 2021.

[5] D. Belfadel, M. Zabinksi and R. Munden. *Walking on Water Term Design Project in Fundamentals of Engineering*. ASEE: Annual Conference and Exposition, Montreal, Canada. June 2020.

[6] D. Belfadel, M. Arambulo, M. Zabinksi, R. Munden, and James Cavallo. *"Use of the Arduino Platform in Fundamentals of Engineering."* ASEE: Annual Conference and Exposition, Tampa, Florida. June 2019.

[7] D. Belfadel: *MATLAB Integration in Sophomore Mathematical Analysis Course,* ASEE: Annual Conference and Exposition, Portland, June 2024.

[8] Freeman, Scott, et al. "Active learning increases student performance in science, engineering, and mathematics." Proceedings of the national academy of sciences 111.23 (2014): 8410-8415.

[9] A. Haubold, "Matlab for first-year college engineers," 2007 37th Annual Frontiers In Education Conference - Global Engineering: Knowledge Without Borders, Opportunities Without Passports, Milwaukee, WI, USA, 2007, pp. F1H-7-F1H-12, doi: 10.1109/FIE.2007.4418108.

[10] González-Cortés JJ, Cantero D, Ramírez M. Project-Based Learning in Bioprocess Engineering: MATLAB Software as a Tool for Industrial-Scale Bioreactor Design. Computer Applications in Engineering Education. 2025 Jan;33(1):e22811.

[11] Herber DR, Deshmukh AP, Mitchell ME, Allison JT. Project-based curriculum for teaching analytical design to freshman engineering students via reconfigurable trebuchets. Education Sciences. 2016 Feb 25;6(1):7.

[12] Hensel R, Sun Y. Application of object scaffolding to develop a hands on, problem centered, and project based freshman Matlab® course. In2006 Annual Conference & Exposition 2006 Jun 18 (pp. 11-219).

[13] Hussain S, Jamwal PK, Munir MT. Computer-aided teaching using simmechanics and matlab for project-based learning in a robotics course. International Journal of Social Robotics. 2022 Jan;14(1):85-94.

[14] Sujatha S, Vinayakan K. Integrating Math and Real-World Applications: A Review of Practical Approaches to Teaching. International Journal of Computational Research and Development. 2023;8(2):55-60.

[15] Marley SA, Siani A, Sims S. Real-life research projects improve student engagement and provide reliable data for academics. Ecology and Evolution. 2022 Dec;12(12):e9593.

[16] Ortega-Alvarez JD, Vieira C, Guarín-Zapata N, Gómez J. Flipping a computational modeling class: Strategies to engage students and foster active learning. In2020 IEEE Frontiers in Education Conference (FIE) 2020 Oct 21 (pp. 1-4). IEEE.