

ENGR 11: PROGRAMMING & PROBLEM-SOLVING IN MATLAB

Foothill College Course Outline of Record

Heading	Value
Units:	5
Hours:	4 lecture, 3 laboratory per week (84 total per quarter)
Prerequisite:	MATH 1B or 1BH.
Advisory:	Demonstrated proficiency in English by placement via multiple measures OR through an equivalent placement process OR completion of ESLL 125 & ESLL 249.
Degree & Credit Status:	Degree-Applicable Credit Course
Foothill GE:	Non-GE
Transferable:	CSU/UC
Grade Type:	Letter Grade (Request for Pass/No Pass)
Repeatability:	Not Repeatable

Student Learning Outcomes

- Students who successfully complete this course should be able to write programs using Matlab to implement problem solving algorithms.
- Students who successfully complete this course should be able to demonstrate knowledge and skills about numerical methods.

Description

This course utilizes the MATLAB environment to provide students with a working knowledge of computer-based problem-solving methods relevant to science and engineering. It introduces the fundamentals of procedural and object-oriented programming, numerical analysis, and data structures. Examples and assignments in the course are drawn from practical applications in engineering, physics, and mathematics.

Course Objectives

The student will be able to:

- recall and apply basic computer environment and basic algorithm flows.
- describe types of error caused by computations.
- use basic numerical methods to find roots of equations.
- do basic manipulations on matrices and solve systems of equations using matrices.
- sort a list of data.
- fit selected models to dataset.
- numerically differentiate and integrate.
- solve basic ordinary differential equations numerically.
- apply different methods to solve simple optimization problems.
- simulate random variables.
- apply a top-down design methodology to develop computer algorithms.

L. create, test, and debug sequential MATLAB programs, as well as programs that use object-oriented techniques, in order to achieve computational objectives.

Course Content

- Basics
 - Overview of computer systems and the MATLAB environment
 - Floating point formats
 - Variables, expressions, and order of operation
 - Elementary functions
 - Pseudo-code, flowcharts, and documentation
 - Formatted input and output
 - Selection programming structures (IF, ELSE)
 - Repetition programming structures (FOR, WHILE)
 - User-defined functions
- Modeling, Computers, and Error Analysis
 - Mathematical modeling and engineering problem solving
 - Programming and software
 - Precision and accuracy
 - Approximations and round-off errors
 - Absolute and relative error
 - Truncation errors
 - Significant digits
- Roots of Equations
 - Bisection methods
 - Bracketing a root
 - Accuracy and speed
 - Fixed-point theorem
 - Fixed point of a function
 - Geometry of a fixed point iteration
 - Linear convergence and stopping criteria
 - Newton's Method
 - Linear and quadratic convergence
- Plotting
- Linear Algebraic Equations
 - Matrix algebra
 - Add, subtract, multiply, inverse
 - Gaussian elimination
 - Operation count
 - LU decomposition and matrix inversion
 - Matrix form of Gaussian elimination
 - Back substitution
 - Complexity
- Sorting
 - Bubble sort
 - Straight Insertion and Shell's Method
 - Quicksort and heapsort
- Curve Fitting
 - Least-squares regression
 - Inconsistent system
 - Fitting models to data
 - Conditioning of least square
 - Cubic spline interpolation
 - Properties of splines
 - End point conditions
 - Taylor polynomials
 - Error analysis
 - Fast Fourier transform
 - Orthogonality and interpolation
 - Least square fitting with trigonometric functions
 - Error analysis

- 6. Plotting
- G. Numerical Differentiation and Integration
 - 1. Differentiation
 - a. Forward divided-difference formulae
 - b. Centered divided-difference formulae
 - c. Backward divided-difference formulae
 - 2. Integration
 - a. Newton-Cotes integration formulas
 - b. Trapezoidal rule
 - c. Simpson's rule
 - 3. Error analysis
- H. Ordinary Differential Equations
 - 1. Theory of initial-value problems
 - a. Euler's Method
 - b. Runge-Kutta Methods
 - c. Stability region
 - 2. Theory of boundary-value problems
 - a. Linear Shooting Methods
 - b. Finite Difference Method
 - 3. Higher Order ODE
 - a. Solve as system
 - 4. Error analysis
- I. Optimization (Optional)
 - 1. Linear programming
 - 2. Simplex Method
 - 3. Duality Method
- J. Random Numbers and Applications
 - 1. Random digits and random numbers
 - 2. Pseudorandom numbers
 - 3. Monte Carlos Method
 - 4. Brownian motion
 - 5. Applications of random walks

Lab Content

- A. Finding Roots of Equations, Linear Programming (Optimization)
 - 1. Designing an algorithm to find roots
 - 2. Implementing the plan using MATLAB
 - 3. Including comments in the program
 - 4. Checking for reasonableness
 - 5. Debugging the program
- B. Solving Linear Algebraic Equations
 - 1. Designing an algorithm to solve linear algebraic equations
 - 2. Implementing the plan using MATLAB
 - 3. Including comments in the program
 - 4. Checking for reasonableness
 - 5. Debugging the program
- C. Curve Fitting
 - 1. Designing an algorithm for curve fitting
 - 2. Implementing the plan using MATLAB
 - 3. Including comments in the program
 - 4. Checking for reasonableness
 - 5. Debugging the program
- D. Numerical Integration and Differentiation
 - 1. Designing an algorithm for integration and differentiation
 - 2. Implementing the plan using MATLAB
 - 3. Including comments in the program
 - 4. Checking for reasonableness
 - 5. Debugging the program
- E. Solving Ordinary Differential Equations
 - 1. Designing an algorithm for solving ordinary differential equations
 - 2. Implementing the plan using MATLAB

- 3. Including comments in the program
- 4. Checking for reasonableness
- 5. Debugging the program

Special Facilities and/or Equipment

- A. A graphing calculator (e.g., TI-83 or TI-84) is recommended, but not required.
- B. A lecture room with computers.
- C. Projector equipment and instructor computer.
- D. MATLAB software.

Method(s) of Evaluation

Methods of Evaluation may include but are not limited to the following:

- A. Midterm(s)
- B. Quizzes
- C. Project/oral presentation and written report
- D. Homework assignments
- E. Lab assignments
- F. One comprehensive final exam

Method(s) of Instruction

Methods of Instruction may include but are not limited to the following:

- A. Lecture
- B. Laboratory
- C. Demonstration

Representative Text(s) and Other Materials

- Attaway, S. [MATLAB: A Practical Introduction to Programming and Problem Solving](#). 5th ed. Waltham, MA: Butterworth-Heinemann, 2018.
- Chapman, S. [MATLAB Programming for Engineers](#). 6th ed. Cengage, 2019.
- Gilat, A. [MATLAB: An Introduction with Applications](#). 6th ed. Wiley, 2016.
- Hahn B., and D. Valentine. [Essential MATLAB for Engineers and Scientists](#). 7th ed. Academic Press, 2019.

Types and/or Examples of Required Reading, Writing, and Outside of Class Assignments

- A. Required reading from sections of the textbook
- B. Written homework from the textbook
- C. Written reports on laboratory assignments
- D. Written and oral report on project

Discipline(s)

Engineering, Mathematics