ENGR A220: PROGRAMMING AND PROBLEM-SOLVING IN MATLAB

Item	Value
Curriculum Committee Approval Date	12/08/2021
Top Code	090100 - Engineering, General (requires Calculus) (Transfer)
Units	4 Total Units
Hours	108 Total Hours (Lecture Hours 54; Lab Hours 54)
Total Outside of Class Hours	0
Course Credit Status	Credit: Degree Applicable (D)
Material Fee	No
Basic Skills	Not Basic Skills (N)
Repeatable	No
Grading Policy	Standard Letter (S)

Course 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. PREREQUISITE: MATH A180 or MATH A180H. Transfer Credit: CSU; UC. C-ID: ENGR 220.**C-ID:** ENGR 220.

Course Level Student Learning Outcome(s)

1. Students will create, execute, and test MATLAB applications using a development environment.
2. Students will write programs that correctly use MATLAB data types and functions to solve various kinds of problems.
3. Students will write programs that correctly use functions and control structures, including selection and iteration to solve different kinds of programming problems.

Course Objectives

- 1. Introduce the MATLAB software environment.
- 2. Reinforce a structured, top-down approach to formulate and solve problems.
- 3. Introduce common approaches, structures, and conventions for creating and evaluating computer programs.
- 4. Formulate algorithms and flowcharts to solve a problem and implement them in MATLAB.
- 5. Use MATLAB coding to perform basic computations, including routines, logic operations, and plotting.
- 6. Manipulate vectors and matrices and perform vector and matrix operations.
- 7. Write basic programs in MATLAB to analyze data with an input/output approach.
- 8. Utilize MATLAB routines to perform linear and non-linear interpolation of engineering data.
- 9. Learn a systematic approach to organize and debug MATLAB programs.
- 10. Deliver solving methods and results, including graphical representations, that meet engineering standards.
- 11. Apply a variety of common numeric techniques to solve and visualize engineering-related computational problems.
- 12. Be able to simulate classic engineering processes with MATLAB/SIMULINK.

Lecture Content

Introduction to Programming Basic Computer Architecture Computer Software Programming Languages Language Generations Compiled vs. Interpreted Program Overview of MATLAB environment MATLAB Script and Live Script Built-in Functions, Arrays, Polynomials, and Plots Programming Style System, Directory, and File Commands Data Types and Format Input/Output Commands MATLAB Help Problem-Solving methodology Numeric, Cell, and Structure Arrays Vectors: Creation (“:”, linspace, logspace) and Manipulation (transpose, appending) Vectors: Magnitude, Length, Absolute Value Matrices: Creation and Manipulation Arrays: Creation and Addressing Array Functions Multidimensional Arrays Array Operations: Addition, Subtraction, Matrix-Matrix Multiplication, Right and Left Division, and Exponentiation Array (element-by-element) Operations Special Matrices (eye, zeros, ones) Polynomial Functions: poly, roots, conv, deconv, polyval Cell Arrays Functions Structures Functions and Files Built-in Functions Operations with Complex Numbers Use of Functions with Arrays Function Composition Trigonometric Functions (incl. atan2) User-Defined Functions: function files, input-out functions Local and Global Variables Functions with Multiple Inputs and/or Outputs Functions with No Inputs and No Outputs Zeros of a Function Minimum of a Function of 1 and Multiple Variables Function Handles Anonymous Functions Importing Spreadsheet Files Prog ramming with MATLAB Algorithms and Control Structure Structured Programming Flowcharts Pseudocode Debugging: Syntax and Runtime Errors Large Programs Relational Operators Logical Class Accessing Arrays with Logical Arrays Logical Operators Logical Functions and the “find” Function The “if” Statement The “else” Statement The “elseif” Statement The “if-elseif-else” Structure Strings and Conditional Statements “for” Loops Loop Variable Expression Using a Logical Array as a Mask Vectorization Nested Loops “break” and “continue” command Using an Array as a Loop Index “while” Loops “switch” Structure Plotting Nomenclature Saving and Exporting Figures Requirements for a Correct Plot “fplot” Function Subplots Data Markers and Line Types Labeling Curves and Data Plotting Complex Functions Logarithmic Plots Specialized Plot Commands (bar, plotyy, polar, strairs, stem, errorbar) Plotting Implicit Functions 3-D Plotting Parametric Line Plots Mesh and Surface Plots Contour Plots Surface Plots of Implicit Functions Specialized Plot Commands (meshc, surfc, waterfall) Regressions Linear, Power, and Exponential Functions to Describe Data Function Discovery “polyfit” Function The Least-Squares Criterion Regression with 1st- to 4th- order Polynomials Quality of a Curve Fit The Sum of Squares of the Residuals The Sum of Squares of the Deviation Coefficient of Determination Analysis of the Residuals Scaling the Data Avoiding High-Degree Polynomials Multiple Linear Regression Probability, Statistics, and Interpolations Basic Statics Concepts (Mean, Mode, Median) Absolute, Relative, and Scaled Frequency Histograms and Histogram Functions Probability Calculations with Cumulative Sum Normal or Gaussian Function Mean, Variance, and Standard Deviation Effect of Variance on Gaussian Curve Probability of .±. and .±2. Range Probability Calculations with Error Function Sums and Differences of Random

Variables Random Number Generation Uniformly Distributed Random Numbers Normally Distributed Random Numbers Random Integers 1-D Random Walk Linear Interpolation 2-D Interpolation Cubic-spline Interpolation Linear Algebraic Equations Analysis of Solutions Matrix Rank Augmented Matrix Existence and Uniqueness of Solutions Homogeneous Case Matrix Inverse Method Left-Division Method Pseudoinverse Method Reduced Row Echelon Method Underdetermined Systems Overdetermined Systems Exact vs. Least-Squares Solutions Numerical Methods Numerical Integration: Trapezoidal Integration Integration of Functions Single, Double, and Triple Inte grals Polynomial Integration Numerical Differentiation Backward, Central, and Forward Difference Polynomial Differentiation Gradients 1st-Order Differential Equations Euler Method Built-in Functions for ODEs Non-linear ODEs Higher-Order Differential Equations Linear and Non-linear ODEs Small-angle Approximation State Variable Form Matrix Form Input/Output Models Introduction to Simulink (Optional) Simulation Diagrams Block Diagrams Model Window Common Simulink Blocks Variable Import and Export with the Workspace Linear State-Variable Models Piecewise-Linear Models Model of a Relay-Controlled DC Motor

Lab Content

Introduction to MATLAB Interactive Sessions Operations with Arrays, Matrices and Structures Default and User-Defined Functions, including INPUT/OUTPUT files Creating and Working with M-files File INPUT/OUTPUT Implementing different types of Operators, Selection Structures and Repetition Structures Visualization and Plotting in 2D and 3D Interpolation and Curve fitting Solving Systems of Linear Equations with Matrix Methods Numerical Analysis Techniques, including Numerical Differentiation/Integration, and Solving Differential Equations Numerically

Method(s) of Instruction

- Lecture (02)
- DE Live Online Lecture (02S)
- DE Online Lecture (02X)
- Lab (04)
- DE Live Online Lab (04S)
- DE Online Lab (04X)

Instructional Techniques

Lecture, Assignments, Lab, Projects

Reading Assignments

Textbook reading (1.5 hours/week)

Writing Assignments

Project reports (0.25 hours/week)

Out-of-class Assignments

Programming exercises in MATLAB (4 hours/week) Practice Problems to reinforce course material (1 hour/week)

Demonstration of Critical Thinking

Solving programming problems is a demonstration of critical thinking.

Required Writing, Problem Solving, Skills Demonstration

Students will be required to solve mathematical and enginnering problems by coming up with programming solutions.

Eligible Disciplines

Engineering: Masters degree in any field of engineering OR bachelors degree in any of the above AND masters degree in mathematics, physics, computer science, chemistry, or geology OR the equivalent. (NOTE: A bachelors degree in any field of engineering with a professional engineers license is an alternative qualification for this discipline.) Masters degree required. Title 5, section 53410.1

Textbooks Resources

1. Required William Palm III. MATLAB for Engineering Applications, 4th ed. McGraw-Hill Education, 2019 Rationale: This is a technical subject with applications to Engineering and Mathematics, and requires a professional text. 2. Required Moore, Holly . MATLAB for Engineers, 5th ed. Pearson, 2018 3. Required Chapman, J., Stephen. MATLAB Programming for Engineers, 5th ed. Cengage Learning, 2016