ENGR M10: PROGRAMMING AND PROBLEM-SOLVING IN MATLAB

Originator
srelle

College
Moorpark College

Discipline (CB01A)
ENGR - Engineering

Course Number (CB01B)
M10

Course Title (CB02)
Programming and Problem-Solving in MATLAB

Banner/Short Title
MATLAB Programming

Credit Type
Credit

Start Term
Spring 2020

Catalog Course Description
Utilizes the MATLAB environment to provide students with a working knowledge of computer-based problem solving methods relevant to science and engineering. Introduces the fundamentals of procedural and object-oriented programming, numerical analysis, and data structures. Uses examples and assignments in the course which are drawn from practical applications in engineering, physics, and mathematics.

Taxonomy of Programs (TOP) Code (CB03)
0924.00 - *Engineering Technology, General (requires Trigonometry)

Course Credit Status (CB04)
D (Credit - Degree Applicable)

Course Transfer Status (CB05) (select one only)
A (Transferable to both UC and CSU)

Course Basic Skills Status (CB08)
N - The Course is Not a Basic Skills Course

SAM Priority Code (CB09)
D - Possibly Occupational

Course Cooperative Work Experience Education Status (CB10)
N - Is Not Part of a Cooperative Work Experience Education Program

Course Classification Status (CB11)
Y - Credit Course

Educational Assistance Class Instruction (Approved Special Class) (CB13)
N - The Course is Not an Approved Special Class

Course Prior to Transfer Level (CB21)
Y - Not Applicable

Course Noncredit Category (CB22)
Y - Credit Course

Funding Agency Category (CB23)
Y - Not Applicable (Funding Not Used)

Course Program Status (CB24)
1 - Program Applicable

General Education Status (CB25)
Y - Not Applicable

Support Course Status (CB26)
N - Course is not a support course

Field trips
Will not be required

Grading method
Letter Graded

Alternate grading methods
Student Option- Letter/Pass
Pass/No Pass Grading

Does this course require an instructional materials fee?
No

Repeatable for Credit
No

Units and Hours

Carnegie Unit Override
No

In-Class

Lecture
Minimum Contact/In-Class Lecture Hours
35
Maximum Contact/In-Class Lecture Hours
35

Activity

Laboratory
Minimum Contact/In-Class Laboratory Hours
52.5
Maximum Contact/In-Class Laboratory Hours
52.5

Total in-Class

Total in-Class
Total Minimum Contact/In-Class Hours
87.5
Total Maximum Contact/In-Class Hours
87.5

Outside-of-Class

Internship/Cooperative Work Experience

Paid

Unpaid

Total Outside-of-Class

Total Outside-of-Class
Minimum Outside-of-Class Hours
70
Maximum Outside-of-Class Hours
70

Total Student Learning

Total Student Learning
Total Minimum Student Learning Hours
157.5
Total Maximum Student Learning Hours
157.5

Minimum Units (CB07)
3
Maximum Units (CB06)
3

Prerequisites
MATH M25A or MATH M25AH

Entrance Skills

Prerequisite Course Objectives
MATH M25A- determine analytically whether a limit fails to exist.
MATH M25A- determine whether a function is continuous or discontinuous at a point.
MATH M25A- use the formal definition of the derivative to find the derivative of an algebraic function.
MATH M25A- apply the basic rules of differentiation to find the derivative of a function including the constant, power, sum, product, quotient, and Chain rules.
MATH M25A- find first-order and higher-order derivatives of algebraic and transcendental functions and their inverses.
MATH M25A- find the derivatives of functions and relations using implicit differentiation.
MATH M25A- solve applied problems using the derivative including rates of change, the tangent line problem, and related rates.
MATH M25A- apply the method of logarithmic differentiation for finding derivatives.
MATH M25A- demonstrate an understanding of the connection between differentiability and continuity of a function.
MATH M25A- apply analytic techniques to a function and its derivatives to solve curve sketching problems.
MATH M25A- use differentials with linear approximation problems.
MATH M25A- solve applied optimization problems.
MATH M25A- find an approximate solution to an equation using Newtonrsquo;s Method. (optional*)
MATH M25A- apply the basic rules of integration for finding anti-derivatives for algebraic and transcendental functions.
MATH M25A- use summation notation with Riemann sums and upper and lower sums.
MATH M25A- use the formal definition of the definite integral to evaluate the integral of an algebraic function over a closed interval.
MATH M25A- evaluate definite integrals using the properties of integrals and the Fundamental Theorem of Calculus.
MATH M25A- integrate indefinite and definite integrals using change of variable techniques.
MATH M25A- use integration and analysis techniques to find the area of a region between two curves.
MATH M25A- solve exponential growth and decay problems.
MATH M25AH-determine whether a function is continuous or discontinuous at a point.
MATH M25AH-solve exponential growth and decay problems.
MATH M25AH- Honors: apply the basic rules of differentiation to derive the rules of differentiation for algebraic and trigonometric functions.
MATH M25AH- Honors: apply analytic techniques to a function and its derivatives to solve curve sketching problems for algebraic and transcendental functions.
MATH M25AH- Honors: use differentials to perform error analysis, and apply to real life projects.
MATH M25AH- Honors: apply optimization techniques to real life problems and projects.
MATH M25AH- Honors: perform error analysis on an approximate solution to an equation found by Newton’s Method.
MATH M25AH- Honors: apply the basic rules of integration to physics and other application problems.

Requisite Justification

Requisite Type

Prerequisite

Requisite

MATH M25A or MATH M25AH

Requisite Description

Course not in a sequence

Level of Scrutiny/Justification

Required by 4 year institution

Student Learning Outcomes (CSLOs)

	Upon satisfactory completion of the course, students will be able to:
1	apply numerical methods in order to solve problems in science and engineering.
2	use the MATLAB environment to implement moderately complicated algorithms in a coherent and structured manner to solve problems in science and engineering.

Course Objectives

	Upon satisfactory completion of the course, students will be able to:
1	apply a top-down design methodology to develop computer algorithms.
2	create, test and debug sequential MATLAB programs, as well as programs that use object-oriented techniques, in order to achieve computational objectives.
3	apply numeric techniques and computer simulations to analyze and solve engineering-related problems.
4	use MATLAB effectively to analyze and visualize data.
5	demonstrate understanding and use of standard data structures.

Course Content

Lecture/Course Content

- **10% - Pseudocode, flowcharts, and documentation:**
  - learn to write pseudocode (a text-based algorithm) and draw flowcharts before writing MATLAB codes to implement numerical methods for problem solving
  - generate documentation to aid the understanding of the MATLAB codes
- **15% - Numerical analysis techniques (embedded within topics above):**
  - solving systems of linear equations
  - vector analysis
  - data interpolation
  - least-squares regression and linearization
  - numerical differentiation and integration
  - solving ordinary differential equations
  - series approximation and error
  - solving equations of one variable
  - optimization (optional)
  - stochastic simulation (optional)
- **15% - Computational problem-solving methodology:**
  - mathematical models as a functional relationship between dependent variable, independent variables, parameters, and forcing functions
  - conservation laws, steady-state and dynamic solutions
  - differences between analytical or exact and numerical or approximate solutions
  - different types of numerical solutions
- **15% - Object-oriented programming:**
  - learn when to use object-oriented programming techniques to simplify complicated programming tasks in MATLAB Data structures:
  - numeric, character, and cell arrays

  - tables and data type conversion MATLAB functions and user-defined functions:
  - create and use MATLAB functions with both single and multiple inputs and outputs
  - create and use anonymous functions
  - create and use function handles
  - create and use subfunctions and nested subfunctions
  - store and access functions in toolboxes
  - understand the computational limits of MATLAB
  - recognize and be able to use the special values and functions built in MATLAB
- **10% - Overview of computer systems and the MATLAB environment:**
  - start the MATLAB program and solve simple problems in the command window
  - identify and use the various MATLAB windows
  - define and use simple matrices
  - understand MATLAB's use of matrices
- **10% - Repetition programming structures and recursion:**
  - write and use loops and while loops
  - create midpoint break structures
  - measure the time required to execute program components and understand how to improve program execution times
- **10% - Selecting programming structures, logical functions, sorting and searching data:**
  - understand how MATLAB interprets relational and logical operators
  - understand the appropriate uses of the if/else family of commands
  - understand the switch/case structure
  - use the find and search function
- **5% - Plotting:**
  - create and label two- and three-dimensional plots
  - adjust the appearance of plots, divide the plotting window into subplots, use the interactive MATLAB plotting tools
- **10% - Data files:**
  - import, export, save, and read data formatted input and output:
  - prompt the user for input to an M-file program
  - create output with the "disp" command
  - create formatted output by using the “fprintf” command
  - create formatted output for use in other functions with the “sprint” functions
  - use graphical techniques to provide program input
  - use the cell mode to modify and run M-file programs

Laboratory or Activity Content

- **25% - Project(s) to create a MATLAB program or programs which will incorporate:**
  - naming, using, and saving the value of variables in a MATLAB session
  - understanding the order of operations and the different kinds of data used in MATLAB
  - understanding the difference between scalar, array, and matrix calculations in MATLAB
  - expressing numbers in either floating-point or scientific notation
  - saving a series of commands in an M-file
  - using a variety of common mathematical functions in MATLAB (trigonometric, statistical, and data analysis functions, uniform and Gaussian random-number matrices)
  - using array definitions and operations (numeric, character, and multidimensional arrays) and accessing data in those array
  - creating and using cell and structure arrays
- **25% - Project(s) to create a MATLAB program or programs which will incorporate:**
  - using various analytical and numerical techniques to solve science and engineering related problems
- **25% - Project(s) to create a MATLAB program or programs which will incorporate:**
  - writing pseudocode, drawing flowcharts, and generating documentation to aid the understanding of the MATLAB codes
  - using various input and output commands such as user- prompted input, graphical input, and ‘disp’, ‘fprintf’, ‘sprint’ outputs
  - importing, exporting, saving, and reading data files
  - plotting
- **25% - Project(s) to create a MATLAB program or programs which will incorporate:**
  - selecting the appropriate logical functions and sorting and searching data
  - writing and using loops and while loops to improve program execution times
  - creating user defined functions, subfunctions, and MATLAB functions
  - using various data structures such as numeric arrays, character arrays, cell arrays, tables, and data type conversion using object-oriented programming techniques to simplify complicated programming tasks in MATLAB when appropriate

Methods of Evaluation

Which of these methods will students use to demonstrate proficiency in the subject matter of this course? (Check all that apply):

Problem solving exercises
Skills demonstrations

Methods of Evaluation may include, but are not limited to, the following typical classroom assessment techniques/required assignments (check as many as are deemed appropriate):

Classroom Discussion
Computational homework
Individual projects
Laboratory activities
Objective exams
Projects
Problem-solving exams
Participation
Quizzes
Reports/Papers/Journals
Skills demonstrations
Simulations

Instructional Methodology

Specify the methods of instruction that may be employed in this course

Computer-aided presentations
Collaborative group work
Class activities
Class discussions
Distance Education
Demonstrations
Group discussions
Guest speakers
Instructor-guided interpretation and analysis
Instructor-guided use of technology
Internet research
Laboratory activities
Lecture
Small group activities

Describe specific examples of the methods the instructor will use:

The instructor will explain and demonstrate how programming in the MATLAB environment can be used to solve science and engineering problems by the following teaching technique:

- introducing the programming concepts to the class either on the white board or by utilizing PowerPoint presentations
- reviewing the applicable mathematical analysis tools
- reviewing the applicable science or engineering concepts
- applying the MATLAB programming language
  - writing the pseducode
  - demonstrating the code in action in the MATLAB environment

Representative Course Assignments

Writing Assignments

- Writing a pseudocode, a detailed “text-based” description of a MATLAB program, for solving a science or engineering problem. An example would be: Write a pseudocode to predict the velocity of a bungee jumper as a function of time during the free fall portion of the jump.
- Documenting engineering design solutions. An example would be: Write a laboratory report documenting the process of verifying the calibration of a pressure sensor using MATLAB. Include the following sections in the report: Abstract, Introduction, Background, Data Acquisition, Data Analysis, Discussion of Results, Conclusion, References.
- Answering concept questions from lectures. An example would be: Name and explain the 4 major conservation laws used in civil, mechanical, chemical, and electrical engineering.

Critical Thinking Assignments

- Analyzing engineering or science problems, synthesizing solutions, and critically evaluating the results. An example would be: The growth of populations of organisms has many scientific and engineering applications. Assuming first an unlimited population

growth and then one that is limited by shortages of food and space, model the rate of change of population 'p' with respect to time 't' from 1950 to 2000 in 5-year increments. Then write a MATLAB program to solve the model numerically and display your simulation results on a plot. Compare your population growth model results to those published for the same time span from 1950 to 2000. Based on this comparison, would you change your model or your chosen numerical method? If yes, how? If no, why not?
- Applying knowledge of mathematics and coding to enhance MATLAB's built-in functions. An example would be: You need to determine the thermocline depth and temperature gradient for Platte Lake, Michigan. You need to use natural spline end conditions to perform this analysis. However, the built-in MATLAB spline function does not meet your needs as it does not return the 1st and 2nd derivatives which are required for this analysis. Write a MATLAB code to develop your own M-file to implement a natural spline and return the derivatives.

Reading Assignments

- Read a MATLAB program code that when executed produces an incorrect result, then analyze and debug the program code.
- Read scientific or engineering peer reviewed journal articles in which MATLAB programming was utilized to solve or analyze a problem, and then write a pseudocode to recreate that particular MATLAB programming code.

Skills Demonstrations

- Demonstrate the use of common mathematical functions such as matrices, and trigonometric, statistical, and data analysis functions in MATLAB programming.
- Demonstrate the ability to debug an erroneously written MATLAB code.

Outside Assignments

Representative Outside Assignments

- Writing MATLAB programs to solve engineering problems with greater than average difficulty which may require analysis tools beyond the textbook. An example would be: Using the fourth-order Runge-Kutta method solve a system of nonlinear ordinary differential equations developed by Edward Lorenz to relate the intensity of atmospheric fluid motion in the x-direction to variation of temperature in the y and z directions.
- Conducting library and/or Internet research to gather information and prepare for group design projects. An example would be: Design a computer simulation using a mass-spring model to gain insight into the dynamics of multi-story buildings under the influence of earthquakes.

Articulation

C-ID Descriptor Number

ENGR 220

Status

Approved

Equivalent Courses at 4 year institutions

University	Course ID	Course Title	Units
UC San Diego	MAE 8	MATLAB Programming for Engineering Analysis	4
UCLA	C&EE/MECH&AE M20	Introduction to Computer Programming with MATLAB	4
UC Berkeley	ENGIN 7	Introduction to Computer Programming for Scientists and Engineering (MATLAB)	4

Comparable Courses within the VCCCD

ENGR R148 - Programming in MATLAB
ENGR V14 - MATLAB

Equivalent Courses at other CCCs

College	Course ID	Course Title	Units
East Los Angeles College	ENG GEN 122	Programming and Problem-Solving in MATLAB	3

District General Education

A. Natural Sciences

B. Social and Behavioral Sciences

C. Humanities

D. Language and Rationality

E. Health and Physical Education/Kinesiology

F. Ethnic Studies/Gender Studies

Course is CSU transferable
Yes

CSU Baccalaureate List effective term:
Spring 2017

CSU GE-Breadth

Area A: English Language Communication and Critical Thinking

Area B: Scientific Inquiry and Quantitative Reasoning

Area C: Arts and Humanities

Area D: Social Sciences

Area E: Lifelong Learning and Self-Development

CSU Graduation Requirement in U.S. History, Constitution and American Ideals:

UC TCA

UC TCA
Approved

IGETC

Area 1: English Communication

Area 2A: Mathematical Concepts & Quantitative Reasoning

Area 3: Arts and Humanities

Area 4: Social and Behavioral Sciences

Area 5: Physical and Biological Sciences

Area 6: Languages Other than English (LOTE)

Textbooks and Lab Manuals

Resource Type
Textbook

Classic Textbook
No

Description
Chapman, S. J. (2019). *MATLAB programming for engineers*, (6th ed.). Cengage Learning.

Resource Type
Textbook

Classic Textbook
No

Description
Chapra, S.C. (2017). *Applied numerical methods with MATLAB for engineers and scientists*, (4th ed.).   McGraw-Hill.

Resource Type
Textbook

Classic Textbook
No

Description
Moore, H. (2017). *MATLAB for engineers,* (5th ed.). Pearson.

Resource Type
Textbook

Classic Textbook
No

Description
Attaway, S. (2019). *MATLAB: A practical introduction to programming and problem solving*, (5th ed.). Butterworth-Heinemann.

Library Resources

Assignments requiring library resources
Computer simulations

Sufficient Library Resources exist
Yes

Example of Assignments Requiring Library Resources
Research using the Library's print and online resources to gather information and prepare to design a computer simulation model in the MATLAB environment to solve engineering problems.

Distance Education Addendum

Definitions

Distance Education Modalities
Hybrid (51–99% online)
Hybrid (1–50% online)
100% Online

Faculty Certifications

Faculty assigned to teach Hybrid or Fully Online sections of this course will receive training in how to satisfy the Federal and state regulations governing regular effective/substantive contact for distance education. The training will include common elements in the district-supported learning management system (LMS), online teaching methods, regular effective/substantive contact, and best practices.
Yes

Faculty assigned to teach Hybrid or Fully Online sections of this course will meet with the EAC Alternate Media Specialist to ensure that the course content meets the required Federal and state accessibility standards for access by students with disabilities. Common areas for discussion include accessibility of PDF files, images, captioning of videos, Power Point presentations, math and scientific notation, and ensuring the use of style mark-up in Word documents.

Yes

Regular Effective/Substantive Contact

Hybrid (1%–50% online) Modality:

Method of Instruction	Document typical activities or assignments for each method of instruction
Asynchronous Dialog (e.g., discussion board)	Instructor will post a question or an erroneous code, students will respond to the question or attempt to analyze the code and suggest how to debug the code.
E-mail	Instructor will email students with announcements about the course or an upcoming event. Students in turn may email the instructor with their questions or concerns. Students will email their programs and projects to the instructor.
Face to Face (by student request; cannot be required)	Students will have the option to meet the instructor and work in the computer lab in the presence of the instructor to get one-on-one help from the instructor.
Other DE (e.g., recorded lectures)	Instructor may record the lectures and post them for students to view within a specified time frame to be ready for the accompanying programming assignments. Students will upload their programs and projects to the course webpage.
Synchronous Dialog (e.g., online chat)	Instructor may be available on a certain day or days of the week within a certain time frame to help students and answer their questions via an online chat.
Video Conferencing	Instructor may be available on a certain day or days of the week within a certain time frame to help students and answer their questions via live video conferencing.
Telephone	Instructor may provide a phone number for the students where they can leave a voicemail and expect a call back within 24 hours.

Hybrid (51%–99% online) Modality:

Method of Instruction	Document typical activities or assignments for each method of instruction
Asynchronous Dialog (e.g., discussion board)	Instructor will email students with announcements about the course or an upcoming event. Students in turn may email the instructor with their questions or concerns.
E-mail	Instructor will email students with announcements about the course or an upcoming event. Students in turn may email the instructor with their questions or concerns. Students will email their programs and projects to the instructor.
Face to Face (by student request; cannot be required)	Students will have the option to meet the instructor and work in the computer lab in the presence of the instructor to get one-on-one help from the instructor.
Other DE (e.g., recorded lectures)	Instructor may record the lectures and post them for students to view within a specified time frame to be ready for the accompanying programming assignments. Students will upload their programs and projects to the course webpage.
Synchronous Dialog (e.g., online chat)	Instructor may be available on a certain day or days of the week within a certain time frame to help students and answer their questions via an online chat.
Video Conferencing	Instructor may be available on a certain day or days of the week within a certain time frame to help students and answer their questions via live video conferencing.
Telephone	Instructor may provide a phone number for the students where they can leave a voicemail and expect a call back within 24 hours.

100% online Modality:

Method of Instruction	Document typical activities or assignments for each method of instruction
Asynchronous Dialog (e.g., discussion board)	Instructor will post a question or an erroneous code, students will respond to the question or attempt to analyze the code and suggest how to debug the code.
E-mail	Instructor will email students with announcements about the course or an upcoming event. Students in turn may email the instructor with their questions or concerns. Students will email their programs and projects to the instructor.
Face to Face (by student request; cannot be required)	Students will have the option to meet the instructor and work in the computer lab in the presence of the instructor to get one-on-one help from the instructor.
Other DE (e.g., recorded lectures)	Instructor may record the lectures and post them for students to view within a specified time frame to be ready for the accompanying programming assignments. Students will upload their programs and projects to the course webpage.
Synchronous Dialog (e.g., online chat)	Instructor may be available on a certain day or days of the week within a certain time frame to help students and answer their questions via an online chat.
Telephone	Instructor may provide a phone number for the students where they can leave a voicemail and expect a call back within 24 hours.
Video Conferencing	Instructor may be available on a certain day or days of the week within a certain time frame to help students and answer their questions via live video conferencing.

Examinations

Hybrid (1%–50% online) Modality

Online
On campus

Hybrid (51%–99% online) Modality

Online
On campus

Primary Minimum Qualification

ENGINEERING

Review and Approval Dates

Department Chair

09/26/2019

Dean

09/26/2019

Technical Review

10/03/2019

Curriculum Committee

10/15/2019

DTRW-I

MM/DD/YYYY

Curriculum Committee

MM/DD/YYYY

Board

MM/DD/YYYY

CCCCO
MM/DD/YYYY

Control Number
CCC000579717

DOE/accreditation approval date
MM/DD/YYYY