USC Viterbi School of Engineering

AME 405 – Functional Approach to Computational Methods

Units: 3
Spring 2017—MW—10:00-11:50am

Location: VHE 210

Instructor: Dr. Inna Abramova
Office: RRB 220
Office Hours: TBD
Contact Info: abramova@usc.edu

Teaching Assistant:
Office: VHE 202
Office Hours: TBD
Contact Info: (213) 740-8253

IT Help: USC Information Technology Services
http://itservices.usc.edu/
Hours of Service: Monday-Thursday: 8:00 am-7:00 pm,
Friday: 8:00 am-5:00 pm,
Saturday-Sunday: 1:00 pm-5:00 pm
Contact Info: consult@usc.edu, (213) 740-555

Course Description
Introduction to computational methods in engineering: learning to develop and implement numerical algorithms for solving a range of mathematics, physics, and engineering problems. This course is suitable for undergraduate students in aerospace, bio, civil and mechanical engineering.

Learning Objectives

- Use Wolfram Mathematica modeling & simulation software to develop, implement, and debug numerical algorithms for a range of typical problems in math, physics, and engineering.
- Analyze posed problems, define the "best-fit" computational approach, and write well-designed, well-documented, modular code using an interpreter-based scripting language of Wolfram Mathematica.
- Utilize extensive Mathematica visualization/graphics capabilities to present the code output in an easily understandable, report/publication-style format.
- Get acquainted with Mathematica built-in suites of specialized functionality: Equations Solving (obtaining closed-form and numerical solutions for algebraic and differential equations and systems of equations), Matrices and Linear Algebra, Control Systems
- Get introduced to building of interactive applications with Mathematica

Prerequisite(s): None
Co-Requisite (s): None
Concurrent Enrollment: None
Recommended Preparation: 1 year of college-level calculus and physics

Course Notes
Class notes, assignments, handouts, and other class materials are downloadable from the Blackboard: https://blackboard.usc.edu/. Class announcements will also be posted on the Blackboard.

Required Readings and Supplementary Materials
Required Textbook:

Edward B. Magrab, "*An Engineer's Guide to Mathematica*", Wiley, May 5, 2014, 452 pp., ISBN: 978-1118821268

Supplementary Reading:

Paul Wellin, "*Programming with Mathematica: An Introduction*", Cambridge University Press, February 25, 2013, 728 pp., ISBN: 978-0521444859

Sal Mangano, "*Mathematica Cookbook*", O'Reilly Media, May12, 2010, 830 pp., ISBN: 978-0596520991

Roozbeh Hazrat, "*Mathematica: A Problem-Centered Approach (Springer Undergraduate Mathematics Series)*", Springer, July 12, 2010, 199 pp., ISBN: 978-1849962506

Jonathan Borwein, Matthew Skerritt, "*An Introduction to Modern Mathematical*

Computing: With Mathematica (Springer Undergraduate Texts in Mathematics and Technology)", Springer, August 4, 2012, 224 pp., ISBN: 978-1461442523

Wolfram Mathematica Tutorial Collection from Wolfram Research (a collection of free downloadable PDF files), http://www.wolfram.com/learningcenter/tutorialcollection/

Description and Assessment of Assignments

Two midterm exams, 10 sets of weekly homework problems, and the final exam will be assigned and graded during the semester. The accumulated points that a student earned from all the assignments will be used to assign a grade for the student. The grading is curved.

Grading Breakdown

Assignment	Points	% of Grade
10 Homeworks, 15 pts. Each	150	30
Midterm 1	75	15
Midterm 2	75	15
Final Exam	200	40
TOTAL	500	100

Assignment Submission Policy

Weekly homework assigned, and due the following week.

The homework problems will require writing programs in Mathematica scripting language. A student must submit hardcopy in the form of a PDF file containing the developed code and the generated output, and softcopy in the form of the developed Mathematica notebook(s) (zipped archive is acceptable). Homework shall be submitted via the Blackboard prior to the indicated in every assignment due time. Homework problems will be graded according to the following criteria:

Points	***Criteria***
0	Problem not attempted, OR no hardcopy submitted OR no softcopy submitted
20% of available points	Hard and soft copy submitted, but notebook evaluation terminates with an error
21 to 30% of available points	Hard and soft copy submitted; notebook evaluation generates results that don't match the hardcopy.
31 to 70% of available points	Hard and soft copy submitted; notebook evaluation generates incorrect results
71 to 100% of available points	Hard and soft copy submitted; notebook evaluation generates correct results

In all cases better-written code earns more points.

Additional Policies

Late homework receives **NO** credits.

Course Outline

1. Introduction: Overview of main principles of Mathematica programming (Chapter 1, handout)
   - Starting up Mathematica and getting familiar with the notebook interface
   - Overview of elementary operations, entering input and evaluating it, displaying the output
   - Basic syntax: expressions, variables, comments, error messaging
   - Using Mathematica help: the Documentation Center, function information
2. Mathematica Language (Chapter 1, handout)
   - Expressions as the main element of Mathematica language: types, structure, evaluation
   - Defining variables and functions, assigning values to variables (immediate and delayed assignments), symbolic and numerical computations
   - Predicates and Boolean Operators
3. Numerics in Mathematica Language (Chapter 1, handout)
   - Representation of numbers in Mathematica: types of numbers, exact vs. approximate numbers, random numbers
   - Precision and accuracy of numerical computations
4. Data Structures: lists, arrays, tables (Chapter 2, handout)
   - Lists: structure, syntax, constructing and displaying
   - Operation on lists: measuring, rearranging, accessing elements, assigning components
   - Structure, syntax, and operations on arrays
   - Structure, syntax, and operations on tables
   - Multidimensional data structures: lists of lists
5. Procedural Programming (Chapter 3, handout)
   - Conditional functions (*If*, *Which*, *Switch*)
   - Loops and iteration (*Do*, *For*, *While*, *Break*, *Continue*)
   - Recursion
6. Strings (Chapter 2, handout)
   - Structure and syntax
   - Operations on strings, accessing string elements, finding substrings, partitioning strings
7. Graphics and Visualization (Chapter 6, handout)
   - Graphics primitives, their construction, display, manipulation
   - Function visualization (*Plot, Plot3D, ParametricPlot, ParametricPlot3D, PolarPlot, RegionPlot*)
   - Data visualization (*ListPlot, ListPlot3D, ListLinePlot, contour and density plots, visualization of tabular data*)
   - Vector Visualization, Charting & Information Visualization
   - Graphics/Plot styling
   - Combining and modifying graphics
8. Functions and Modules (Chapter 3, handout)
   - Patterns: structure, methods of matching

- Creating and applying rules, delayed vs. immediate application, rule substitution, evaluation process
- Functions: definition, syntax, evaluation process, parameter passing, checking for argument validity, function with variable number of arguments, function with defaults, function attributes
- Module as means to localize names of the variables and avoid the name conflicts between the global and local names

9. Packages (handout)

10. Specialized built-in Mathematica functionality suites (Chapters 4, 5, and 10; handouts and Wolfram online tutorials)

- Equations solving (*Solve, NSolve, RSolve, FindRoots, FindInstance, manipulating equations*)
- Differential Equations solving (*DSolve, NDSolve*)
- Matrices and Linear Algebra (*vector and matrix operations, LinearSolve, Eigensystem, LeastSquares minimization*)
- Control Systems (*TransferFunctionModel, StateSpaceModel, model connections and manipulations, StateResponse, OutputResponse*)

11. Building of interactive applications with Mathematica (Chapter 7, handout and Wolfram online tutorials) – optional, time-permitting

- Interactively manipulating variables, symbolic expressions, plots
- Creating expressions for which value dynamically updates and controlling the updates of dynamic values
- Creating, placing, and styling control objects
- Attaching dynamic expressions to control objects
- Animating expressions
- Creating a custom interface

Course Schedule: A Weekly Breakdown

	Topics/Daily Activities	Readings and Homework	Deliverable/ Due Dates
Week 1 1/09-1/15	Introduction Mathematica language	Chapter 1, handout HW0 assigned	HW0 due 1/18
Week 2 1/16-1/22	Mathematica language	Chapter 1, handout HW1 assigned	
Week 3 1/23-1/29	Numerics and precision Data structures: lists	Chapter 1 and 2, handout HW2 assigned	HW1 due 1/25
Week 4 1/30-2/5	Data structures: lists (flat and nested), arrays, tables	Chapter 2, handout HW3 assigned	HW2 due 2/01
Week 5 2/6-2/12	Data structures: list operations Functional programming: patterns	Chapter 2 and 3, handout HW4 assigned	HW3 due 2/08
Week 6 2/13-2/19	Functional programming: rules, delayed vs. immediate application, evaluation process	Chapter 3, handout HW5 assigned	HW4 due 2/15
Week 7 2/20-2/26	Functional programming: functions Review for Midterm 1	Chapter 3, handout	HW5 due 2/22
Week 8 2/27-3/5	**Midterm 1,** 3/01 Procedural programming: loops and iteration	Chapter 3, handout HW6 assigned	
Week 9 3/6-3/12	Procedural programming: recursions Functional programming: custom & specialty functions, modules	Chapter 3, handout	HW6 due 3/08
Week 10 3/20-3/26	Strings	Chapter 2, handout HW7 assigned	
Week 11 3/27-4/2	Graphics and Visualization	Chapter 6, handout HW8 assigned	HW7 due 3/29
Week 12 4/3-4/9	Graphics and Visualization Review for Midterm 2 **Midterm 2,** 4/05	Chapter 6, handout	HW8 due 4/05
Week 13 4/10-4/16	Specialized functions: solving equations (algebraic & differential)	Chapters 4 and 5, handout HW9 assigned	
Week 14 4/17-4/23	Specialized functions: linear algebra, control systems Interactive applications	Chapters 4, 5, 7, and 10, handout HW10 assigned	HW9 due 4/19
Week 15 4/24-4/28	Interactive applications Packages Review for final exam	Chapter 7, handout	HW10 due 4/26
FINAL	**Final Exam**		**5/08** **8:00 – 10:00 am**

NOTE:

Shown in the above Course Schedule midterm dates and homework due dates are tentative and may be changed as logical progress of the course topics requires.

Statement on Academic Conduct and Support Systems

Academic Conduct

Plagiarism – presenting someone else's ideas as your own, either verbatim or recast in your own words – is a serious academic offense with serious consequences. Please familiarize yourself with the discussion of plagiarism in *SCampus* in Section 11, *Behavior Violating University Standards*https://scampus.usc.edu/1100-behavior-violating-university-standards-and-appropriate-sanctions/. Other forms of academic dishonesty are equally unacceptable. See additional information in *SCampus* and university policies on scientific misconduct, http://policy.usc.edu/scientific-misconduct/.

Discrimination, sexual assault, and harassment are not tolerated by the university. You are encouraged to report any incidents to the *Office of Equity and Diversity* http://equity.usc.edu/ or to the *Department of Public Safety* http://capsnet.usc.edu/department/department-public-safety/online-forms/contact-us. This is important for the safety whole USC community. Another member of the university community – such as a friend, classmate, advisor, or faculty member – can help initiate the report, or can initiate the report on behalf of another person. *The Center for Women and Men* http://www.usc.edu/student-affairs/cwm/ provides 24/7 confidential support, and the sexual assault resource center webpage sarc@usc.edu describes reporting options and other resources.

Support Systems

A number of USC's schools provide support for students who need help with scholarly writing. Check with your advisor or program staff to find out more. Students whose primary language is not English should check with the *American Language Institute* http://dornsife.usc.edu/ali, which sponsors courses and workshops specifically for international graduate students. *The Office of Disability Services and Programs* http://sait.usc.edu/academicsupport/centerprograms/dsp/home_index.htmlprovides certification for students with disabilities and helps arrange the relevant accommodations. If an officially declared emergency makes travel to campus infeasible, *USC Emergency Information http://emergency.usc.edu/*will provide safety and other updates, including ways in which instruction will be continued by means of blackboard, teleconferencing, and other technology.