AME 404 – Computational Solutions to Engineering Problems, Fall 2016

Lecture: MW 11:00-12:20 pm (class # 28755R) OHE 132
MW 11:00-12:20 pm (class # 29005D) DEN
Instructor: Takahiro Sakai, Ph.D.
Email: tsakai@usc.edu, **Office:** RRB213, **Phone:** (213)740-5367
Office Hours: TBA
Textbook: Not required.
Supplemental book: Gilat and Subramaniam, *Numerial Methods for Engineers and Scientists*, Wiley
Teaching Assistant / TA Office hours:
TBA (email): Office hour @Office# (office phone#)

Catalogue Data: Mathematical aspects of the solutions to typical advanced mechanical engineering problems. Modeling, simulation, computational aspects, computer solutions, and computational tools.

Course Objective: This course introduces numerical methods and effective numerical tools in simulating and analyzing typical problems that arise in aerospace and mechanical engineering. Mathematical modeling and analytical solutions to ideal problems and numerical error analysis will be also provided. At the end of semester students will have skills to: formulate a simple problem, implement numerical methods to a computer program, validate the program, and simulate the problem and evaluate the results. **Recommended Preparation:** Math245, ITP168 (MATLAB)

Selected Topics (some topics may change):

1. Numerical Simulations of Dynamical Systems
   - Modeling of the 1st order, the 2nd or higher order ODEs and the 1st order systems
   - Integration methods of Ordinary Differential Equations (ODEs)
   - Fourier spectral analysis, Frequency Power Spectrum
2. Numerical Solutions to Two-point Boundary Value Problems (BVP)
   - Iterative method (the shooting method)
   - Modal solutions (Eigenvalues and Eigenfunctions)
3. Numerical Solutions to Partial Differential Equations (PDEs)
   - Finite differences
   - Solution methods of linear system equations
   - 1D heat conduction problems (steady and unsteady)
   - 2D steady state heat conduction problems

Grading Plan: Homework 10% and three projects, 30% each.

Computer Programming Language: MATLAB programming language is required for this course. Students are assumed to have a fundamental, beginner-level MATLAB programing skill (e.g., ITP168 or equivalent), which includes array manipulations, if-constructs, for-loop, functions and basic 1-D plotting. MATLAB is freely available from USC ITS at https://itservices.usc.edu/software/. You can use any version of MATLAB in this course. Many of you took MATLAB for a while ago, and your programming skill might have become a bit rusty to date. If you are concerned about current standing of your programming skill, there is NO worry. You will learn programming *again* through many practical example codes and programing assignments throughout the semester. Of course, for several of you it may not be easy at the start of the semester, but you will catch up and get acclimatized in programming as you work through, and at the end of semester you will become a proficient MATLAB programmer. For those of you who *still* want some more MATLAB tutorials, Attaway, *Matlab: Practical Introduction to Programming and Problem Solving*, Butterworth-Heinemann, is recommended for its low price with ample contents.

Homework Policy (IMPORTANT! PLEASE READ CAREFULLY!):

- In general students are expected to work on homework independently.   Discussion with peer students is not discouraged. However, students must write computer programs and the other part of the work independently. Your work will be carefully monitored for academic integrity throughout the semester. If graders determined that academic plagiarism is highly likely, then the work will receive a score of zero. If plagiarism persists, then such a case will be filed to Student Judicial Affairs & Community Standards (http://www.usc.edu/student-affairs/SJACS) with a recommended course grade of F. This is the worst scenario and, should be avoided at all times. If you worked on a problem in a group, it would be likely that your work looks similar to others. In order to avoid your work miscounted as a plagiarism, therefore, please list all the group members' names on your work. This does not harm your score.
- Your questions on grading will be accepted for one-week counting from the day the homework is returned, and the homework grade will be frozen thereafter and no appeal or excuse will be accepted.
- Due date will never be compromised. Late homework is never accepted for any reason.
- In general, submitted work must include mathematical formulation (as needed), computer program and corresponding results other than that required by the assignment. It is strongly recommended to type and edit all your work (e.g., use Office, MathType, LaTeX, etc.) and paste results and MATLAB codes in the same document file, and burn the document into a pdf-format and turn it in.
- Grading is based on accuracy of the results and quality of presentation. Computational work would be valuable, only if correct results were produced timely and presented in clear way. Students are responsible for every line of their code, and must be very *careful* because even one incorrect line will produce entirely incorrect results which can cost substantial amount of the score. (If this sounds harsh or scares you, I am really sorry... But this is a *reality* of computational work, different from other courses. So I really appreciate your understanding!) If the presentation of your results is unclear, you will not be able to receive full credit even you obtained correct computational results. But *some* partial credit will be given, if the program does not output correct results, as long as your code is presented.
- **All assignments must be submitted electronically to Dropbox set up in DEN D2L.** Submitted document must be in a single **pdf file** in which all required work (e.g., results and codes) are included. Please do not turn in any hardcopy of your work and do not email your work to us, for whatever the reason is. All informal submissions will be rejected.
- Please do not email your codes to the instructor or TAs and ask for debugging. The instructor team will never debug your code online. Students are rather encouraged to consult us during our office hour.
- We will strictly and consistently follow these policies throughout the semester.

Project Policy (IMPORTANT! PLEASE READ CAREFULLY!):

- Project policy follows the homework policy except that **students are required to work on projects all independently. No collaborating work is permitted.** This means that all the students must submit their original work. This is because the projects are considered as take-home "exams". TAs and instructor will assist you only in clarifying the problems but not in solving. Due date will never be compromised. Late submittal is never accepted for whatever the reason is.

DEN Desire2Learn: All course documents (lecture notes, homework, project, etc.) will be posted to D2L (https://courses.uscden.net/d2l) and available for all the students. However, online lectures and video archives are available for DEN students only (section 29005). Submit your work only electronically into Dropbox that is set up for every assignment in D2L. All grades will be posted on D2L grade center.

Piazza: This course adopts piazza online discussion system (https://piazza.com). In the beginning of the semester students will be notified to set up your piazza account for this course. Please note that when you post a comment, disclose your name and **DO NOT make your name Anonymous** so that everybody is *responsible* for one's comment, and this avoids irrelevant or nonconstructive sort of posts. For the purpose of this online *discussion* system, do not make your post private.

Academic Integrity: The Department of Aerospace and Mechanical Engineering adheres to the University's policies concerning Academic Integrity as described in SCampus. All faculty, staff and students share the responsibility for maintaining an environment of integrity. Students are expected to be aware of, and to observe, the academic integrity standards set forth in SCampus. We will collectively follow these standards in this section of AME 404.

Disclaimer: This syllabus is tentative and subject to change as needed during the semester. Any changes will be announced in class in advance.

Tentative Schedule and Assignments

week	Day	Covered Topics	Assignment
1	22-Aug	Intro to the 1st order models, Numerical solution of the 1st order ODE, Euler's method	
	24-Aug	Numerical accuracy, Runge-Kutta method, ode45 built-in function, integration error control, Stiff ODEs and Matlab family of ODE solvers (ode113, ode23, ode15s, ode23s, etc.)	
2	29-Aug	Numerical simulation of the 2nd-order model (spring-mass-damper system)	
	31-Aug	Simple pendulum model; Intro to higher order model (multi-degree of freedom problem, application of Lagrange's equation of motion)	Hw 1
3	5-Sep	**Labor day (Holiday)**	
	7-Sep	Numerical simulation of high order model - Heat exchanger system	Hw 2
4	12-Sep	Simulation of Airplane longitudinal dynamics; system matrix, eigenvalue/vectors, stability	
	14-Sep	Fourier series - Review: periodic functions, Fourier modes, convergence, Gibbs phenomenon	Hw 3
5	19-Sep	Fourier spectral analysis - complex Fourier series, Discrete Fourier Transform (DFT), Inverse DFT, Frequency Power spectrum	Project 1
	21-Sep	Fast Fourier Transform (FFT), numerical treatment of non-uniformly sampled signals & non-periodic signals, aliasing error	Hw 4
6	26-Sep	2pt Boundary Value Problem (BVP) - the shooting method, numerical solution to the laminar boundary layer (Blasius) equation	
	28-Sep	Modal solutions of 2pt BVP - Torsional vibration of a shaft: modal shapes, modal frequencies	Hw 5
7	3-Oct	Torsional vibration of a shaft (cont.) - Implementation of shooting method; Project 1 - review	
	5-Oct	Intro to PDE: classification, derivation of the heat conduction equation, boundary conditions	Hw 6
8	10-Oct	1D steady heat equation: Dirichlet problem - Finite differences, tridiagonal system	
	12-Oct	1D steady heat equation: Neumann problem, consistency, numerical accuracy, convergence	Project 1due
9	17-Oct	Steady heat conduction in a circular rod submerged in water, numerical treatment of the pole	Project 2
	19-Oct	Steady heat conduction in a gas turbine blade, variable shape, convection boundary condition	Hw 7
10	24-Oct	Revisit modal solutions of 2pt BVP - direct solution by the finite difference method	
	26-Oct	2D steady heat conduction: Fourier series representation of solution, scalar/vector field plotting	Hw 8
11	31-Oct	2D steady heat conduction: Finite difference formulation, array index mapping via pointer	
	2-Nov	2D steady heat conduction: numerical implementation of direct method	
12	7-Nov	2D steady heat conduction: iterative solution approach - Jacobi, Gauss-Seidel, SOR methods	
	9-Nov	1D unsteady heat conduction: Fourier series representation of solutions, making computer animation of time-dependent solutions by using Matlab	Hw 9
13	14-Nov	1D unsteady heat conduction: Explicit scheme (FTCS scheme), Numerical stability	Project 2due
	16-Nov	1D unsteady heat conduction: Implicit scheme I (Fully-Implicit method), Neumann BC	Hw 10
14	21-Nov	No class	Project 3
	23-Nov	**Thanks giving holiday (no class)**	
15	28-Nov	1D unsteady heat conduction: Implicit scheme II (Crank-Nicolson method), Convection BC	
	30-Nov	Project 3 - review, Course wrap up	
	7-Dec		Project 3 due