MAE 316 & MAE 316 (Hons.)

Analysis of Engineering Systems

Fall 2018

TR 3:30-4:45, MRB 113

Instructor: David S. Mebane, ESB 731, dsmebane@mix.wvu.edu

Office Hours: TR 2:15-3:30 (directly before class) or by appointment

TA: Olivia Santee and Geoffroy Gauneau

TA Office Hours: TBA

Course Text (not required): Gilbert Strang, *Computational Science and Engineering* Wellesley-Cambridge, 2007.

Course Website: eCampus

Objectives

The course is an introduction to computational modeling in mechanical, aerospace and materials engineering. Modeling is the application of scientific, mathematical and computational principles to replicate and predict the behavior of a physical phenomenon or device. It is a critical part of the engineering profession. An important auxiliary goal is the familiarization with numerical methods commonly used in science and engineering and their computational implementation.

Specific Learning Outcomes

ABET

1. Students will demonstrate the formulation, solution and computational implementation of models addressing engineering problems.

1

1. Students will demonstrate understanding of numerical methods for solution of differential equations, linear algebra, and statistics.

1

1. Students will demonstrate the ability to solve problems with MATLAB.

 1

Important Information

* **Homework** is a big deal in this course, since this is where the modeling action is. There will be MATLAB mini-projects along with written exercises on the methods we’ll be learning. One larger project will be assigned and due toward the end of the course. Homeworks and projects are 60% of the grade; the breakdown being 20% for written homework (2 assignments), 20% for mini-projects (2) and 20% for the final project.
* **Tests** are a necessary evil, which make up the remaining 40% of the grade. There will be two of them: one in the middle and one at the end of the course. (Neither are comprehensive.) Modeling work will not be required on the exams; these will merely test knowledge of the numerical methods presented in class. Exam questions will be drawn from topics covered in the written homework and projects. Everyone will be allowed to bring graded homework exercises and returned projects into the exam.
* The **course text** is *Computational Science and Engineering* by Gilbert Strang (Wellesley-Cambridge, 2007). Lecture material and homework assignments will largely – though not entirely – be drawn from that book. It’s a good book to have, it’s not too expensive and I recommend buying it if modeling and math is your thing. It is not required, however, and indeed its style is less of a textbook than that of a reference book: it is a compendium of class notes written for a numerical methods course taught at MIT.
* We will make extensive (and exclusive) use of **MATLAB** (MathWorks) as a framework for the implementation of models. Although everyone should have had some exposure to MATLAB already, I understand that some things will be new and others may have to be re-learned. Everyone should first try the extensive documentation available in the MATLAB help browser as well as online resources at [www.mathworks.com](http://www.mathworks.com) (go to Support->Answers) and sites like Stack Exchange for MATLAB-specific questions. There will be a lecture devoted to MATLAB review, example codes will be made available on eCampus, and MATLAB topics will come up in lecture frequently, especially around the time that projects are assigned. When these resources have been tried and questions remain, then the TA and myself are available to answer MATLAB-related questions, either in class or outside of class. But students *must* put in a significant amount of work honing their coding.
* It’s a big class, but I strongly encourage **discussion**. If you have a question about something, it’s likely that others have the same question. This is *especially* true when there is a misunderstanding: students’ questions have often completely changed the way a lecture has been presented, making it much more useful for everyone. Please stop me and ask: I don’t mind the interruption in the least, and we are going to get something positive out of every question that is asked in good faith. **Your questions make me a better teacher.**
* All **grades** assigned will be letter grades: A for work which shows both understanding and successful execution, B for understanding with mistakes in execution, C for misunderstanding of key concepts but attempted in good faith, and D or F for lack of effort. (Filling the page with material irrelevant to the problem will constitute “lack of effort.”) I will not be a tough grader, but I won’t put up with haggling over grades, either. I do not curve. Composite grades are calculated using a standard GPA system, with 4 for an A, 3 for a B, and so on. Composite scores of 3.5 and up are considered an A average, 2.5 to 3.5 is B, 1.5 to 2.5 is C, and so on. **Every student must score a C or better on two out of three projects in order to pass the course. If this requirement is not met, the instructor reserves the right to assign a D or an F as the final grade irrespective of a student’s score on exams or homeworks.**
* A word about **coming to talk to me**: my door is always open, and I encourage you to come in when you have questions about concepts. However, please be aware of a couple of things: One, there are many students in this class, and therefore I can’t afford to spend time going through entire lectures for students individually. When students come to see me, I want it to be clear that they have studied the material and tried to make sense of it, so that we can zoom in on the tricky concepts. If you come to me and say, “I just don’t get any of it,” I may direct your attention to a couple of things and ask you to go study (or watch lecture videos) and come back. *This does not mean that I don’t like you.* (I like everyone. Seriously. I am always glad to see you when you show up at my door, even if I occasionally may *seem* annoyed.) Two, I will not simply provide answers for homework problems and project issues – this is not fair grade-wise to other students, and undermines the goal of having everyone learn the material for themselves. I will instead guide students through the problems they are having. This may seem frustrating, but not only is it fair it is the best way for you to learn.
* The **TA** will be available for help with homework and projects. **Please see the TA first for help on assignments, if possible.** If it is hard to get an appointment with the TA or if the TA is unable to help for any reason, please come and see me.
* **Lecture videos** are available for live streaming: <https://vimeo.com/user31186215/videos>. The password is MAE316. Students have found this to be a useful resource for those who are confused on concepts from the lectures or if class must be missed.
* However, the videos are not a perfect substitute for **attending class**. If you choose to watch the videos instead of coming to class, your grade may suffer. Attendance is mandatory, and attendance will be taken at random times; students with unexcused absences will face up to a 10% penalty on their final grade.
* **Academic dishonesty** has got to be one of my least favorite topics. The College of Engineering is so awash in cheating that we are at a crisis point, and there are a bunch of new, harsher procedures now for handling it. Cheating in 316 is rife – I caught 16 people cheating (16% of the class) on the final project in Fall 2016. Cheating usually means copying code, either from a fellow student or from a previous student or from the web. Assignments are similar from one term to the next, and I know that old assignments (lots of them) are out there. It’s for this reason that I always change the assignments in subtle ways, and a good number of people who get busted have turned in solutions to old assignments without realizing it. Others simply share code with each other. If you are just starting out with coding and you copy from a classmate, it is almost certain that you will not be able to hide the fact that your code and your classmate’s have come from the same source. *In other words: if you are shaky enough with coding that you have to cheat, you probably can’t hide it from me.* The person who shared and the person who copied are equally guilty. **Take care that your code does not fall into the hands of someone who wants to copy it – if you lose a printed or electronic version or have one stolen, you must report that to me immediately to avoid possible sanctions.**

I used to handle most cheating myself, but now the policy is that we report cheaters to the Department and the system takes over. You do not want to get caught up in this system. Some people get away with it, sure – however, a lot of people get busted, and those people never thought they would get caught. I have had to make heartbreaking decisions, essentially ending peoples’ college careers over cheating. Don’t put me in that position. *If you are honest and work with me in good faith, I will help you make it through. If you cheat, I will be sad to do it but I will make sure you face the consequences.*

Ultimately, if you cheat and get away with it, you are cheating your honest classmates, who comprise the majority of students. **Bottom line: make my life, yours, and your classmates’ better by staying on the honest path.**

* Unquestionably related to the above, a word about **stress**: This is a course that creates a lot of it in students. This is often because of the coding aspect of the course, which many students feel uncomfortable with. However, each coding assignment has two parts, a written math part and a code part. Working code templates are provided for each assignment. If you do well on the written math part, and use the template and put in a good effort on the code, you can score as high as a B on the overall assignment *even if your code doesn’t work*.

Historically, most people in this course (since I’ve been teaching it) have earned a B or better – probably about 2/3. A’s are hard to come by, but it is quite possible to get one – they go to those who really understand the material and who put the work in. (I have found that A’s are more highly correlated with effort than with raw ability.) Those who don’t understand everything but do what they must to get by should expect B’s and C’s. D’s and F’s are reserved for those whose effort is particularly lacking. Try your best on every assignment, come to class, ask questions of me or the TA when you don’t understand something (not just when you are stuck on an assignment), and I can guarantee you’ll pass the course, probably with a B or better, regardless of how good or not-so-good you think you are at math or programming. **If you are feeling overwhelmed, because of something in the class or events outside it, please come and talk with me about it. I am interested in helping you.**

* **Honors students** will have a bit of extra lecture material (delivered when the other students are let out early) and a bit more sophistication in the project assignments.

Lecture schedule

(subject to change)

8/16 Introduction / Motivation

8/21 Matrix Notation

8/23 Equilibrium Problems I (NP due)

8/28 Vector Calculus

8/30 Finite Differences I

9/4 Finite Differences II

9/6 MATLAB Crash Course (MP I Written Part Due)

9/11 Gaussian Elimination I

9/13 Gaussian Elimination II (MP I Due)

9/18 Equilibrium Problems II

9/20 Eigendecomposition I

9/25 Eigendecomposition II

9/27 Oscillations I (HW I Due)

10/2 Oscillations II

10/4 Review

10/9 Exam I

10/11 Finite Volume I

10/16 Finite Volume II (MP II Written Part Due)

10/18 Stationary Iteration

10/23 Positive Definite Matrices (MP II Due)

10/25 Conjugate Gradient I

10/30 Conjugate Gradient II

11/1 Project Review

11/6 Newton’s Method I

11/8 Newton’s Method II (Project Written Part Due)

11/13 Linear Least Squares

11/15 Probability Distributions (Project Due)

11/20 Fall Break

11/22 Fall Break

11/27 Maximum Likelihood I

11/29 Maximum Likelihood II

12/4 Review (HW II Due)

Exam II will be given at the final exam time.

Statement of Academic Integrity

The integrity of the classes offered by any academic institution solidifies the foundation of its mission and cannot be sacrificed to expediency, ignorance, or blatant fraud. Therefore, I will enforce rigorous standards of academic integrity in all aspects and assignments of this course. For the detailed policy of West Virginia University regarding the definitions of acts considered to fall under academic dishonesty and possible ensuing sanctions, please see the [Student Conduct Code](http://studentlife.wvu.edu/office_of_student_conduct/student_conduct_code). Should you have any questions about possibly improper research citations or references, or any other activity that may be interpreted as an attempt at academic dishonesty, please see me *before* the assignment is due to discuss the matter.

Statement of Social Justice

West Virginia University is committed to social justice. I concur with that commitment and expect to maintain a positive learning environment based upon open communication, mutual respect, and non-discrimination. Our University does not discriminate on the basis of race, sex, age, disability, veteran status, religion, sexual orientation, color or national origin. Any suggestions as to how to further such a positive and open environment in this class will be appreciated and given serious consideration. If you are a person with a disability and anticipate needing any type of accommodation in order to participate in this class, please advise me and make appropriate arrangements with the Office of Disability Services (304 293-6700).

Additional Course References

Celik, Ismail, *Introductory Numerical Methods for Engineering Applications*, 2nd Ed., Ararat, 2008.

Kreyzig, Erwin, *Advanced Engineering Mathematics*, 8th Ed., Wiley, 1999.

LeVeque, Randall J., *Numerical Methods for Conservation Laws*, 2nd. Ed., Berkhauser, 1992.

Strang, Gilbert, *Introduction to Applied Mathematics*, Wellesley-Cambridge, 1986.

Strang, Gilbert, *Introduction to Linear Algebra*, 3rd Ed., Wellesley-Cambridge, 2003.

Thomas, J.W., *Numerical Partial Differential Equations: Finite Difference Methods*, Springer, 1995.