Chemical Engineering 2216:
Modeling and Simulations in Chemical Engineering

Prerequisites: CS1110/1111/1112 or equivalent, CHE 2215, APMA 2120, APMA 2130

Instructor: Gaurav Giri (Assistant Professor of Chemical Engineering)
E-mail: gg3qd@virginia.edu
Office Hours: Thu. 10–12 (Office: Chem. Eng. Bldg. Rm. 122), and by appointment

Teaching Assistant: Steven Merz (Ph.D. candidate in Chemical Engineering)
E-mail: snm8xf@virginia.edu
Office Hours: Mon. 2–5, tentative location

Undergraduate Graders/Programming Lab Assistants:
Michelle Sullivan, Andrew Biedermann, Matthew Bradshaw, Suzanne Johnson, Nick Morrow, Nicholas Grabar (Any questions about grading should go directly to Prof. Giri, not to the graders!)

Goals for students this semester:
After ChE 2216, a student will be able to start with any reasonable set of engineering equations, input data, and desired output and:
1. Diagnose the type of mathematical problem involved.
2. Identify appropriate techniques to solve the problem.
3. Make an informed choice of one of these techniques.
4. If it should be solved analytical, find the analytical solution (often from another class).
5. If it should be solved numerically, write a well-structured computer program that solves the problem.

Class meetings and their purpose: We meet regularly for lecture hours at 10 a.m. Monday, Wednesday, and Friday in ChE 005. I will lecture on the course material as well as include a number of active learning questions asked throughout the hour and short group activities to practice the ideas learned in class. Some lecture modules will be delivered before class in the form of videos to be watched before class, and some days will be primarily group work and working examples using the posted lecture material. There will be optional programming sessions on Monday and Thursday from 7-10 p.m at Stacks. The course TA and graders will be present during the sessions, and Prof. Giri will be present during office hours on Thursday.

Materials:
Textbook: *Applied Numerical Methods with MATLAB for Engineers and Scientists*, Steven C. Chapra, Third Edition, McGraw-Hill, 2011. Felder & Rousseau, *Elementary Principles of Chemical Processes* (ChE 2215 textbook, just Chapter 11). Additional required handouts are posted on the web page. The schedule of readings for the course are posted on the wiki to allow for unforeseen changes. Other editions of the book may be almost as useful, but will almost certainly have differently numbered homework problems, so be careful.

Recommended Reference: *Numerical Methods for Engineers*, Chapra and Canale, 6th edition, McGraw-Hill, 2009. A similar textbook that goes more in-depth on the mathematics covered in the class.

i>Clicker: You should pick up an i>Clicker at the bookstore (approx. $35). We will use this for multiple choice participation questions in the classroom. There will be 2-4 clicker questions per session.

Matlab: The class will focus on using Matlab to implement the numerical methods and programming discussed in the class. Matlab license keys must be purchased from the bookstore for $50, and will

allow you to use MatLab throughout your time at U.Va., using the process described at http://www.uvabookstores.com/uvatext/its_software.asp. In this class, we will use the version 'R2015b'. The Research Data Services site at http://data.library.virginia.edu/research-software/matlab/ is the best source of information for MATLAB.. Additional questions or concerns about installing or running these software packages should be directed to ITC help (http://www.its.virginia.edu/helpdesk/) by calling 434-244-4357 or emailing 4help@virginia.edu. Matlab can also be run over the Hive: see http://its.virginia.edu/hive/ for more information, though licenses may sometimes not be available because of demand.

Assessment:

You will be assessed on your learning in the class in a number of different ways. The total points for each of the assessment categories are as follows:

Homework	320	(8 x 40 points each)
Projects	180	(3 x 60 points each)
Quizzes	150	(50 points each, higher score doubled)
Class Participation	70	(See below)
Final Exam	280	
Total	1000	

Homework: Problems requiring numerical solutions are usually solved by writing a computer program. The weekly assignments are designed to help you get hands-on practice on relatively simple problems. Homework will be due on Tuesday by 5 p.m. Note that the help sessions are intentionally early in the week to help get a start on the homework. Late homework will not be accepted except by prior arrangement. Code should be submitted over Collab, not on paper. Problems that are non-coding problems can be handed in either electronically or on paper in the class box in the main office (note that the main office closes at 5 p.m.) and picked up from the TA after grading.

Homework can be discussed in groups, but the final work submitted must be your own, except in the case of pair programming (see below). A simple guideline to use: talk all you want about any problem with anyone in the class, but if you are looking at each other's code or paper, that is group work, not individual work.

Projects: Projects are special homework assignments that allow you to get a deeper feel for all aspects of solving an engineering problem, from the first step of setting up the mathematics problem corresponding to the engineering problem, all the way to the final validation of the result. They will count for 50% more than normal homework assignments. They are pledged, and you are expected to work on them on your own. You will be given the projects two weeks ahead of time; plan on starting them well ahead of time. Projects cannot be completed by pair programming.

Class participation points: Maximum 70 points, but more than 70 are available. Ways to earn participation points:

a) *Evaluations* (up to 20 points): Students who participate in pair programming will have a chance to fill out a pair programming evaluation for 10 points. Students who fill out the end of class evaluation will receive 10 points.
b) *i>Clicker questions* (up to 70 points): There will be 1-4 multiple choice questions per lecture session to gauge student understanding and give you practice with the concepts. Half credit for answering,

full credit for a correct answer. The total score will be normalized to 70 points, such that if you get 70% of the questions correct, you will get the full 70 points.

c) *Catch the prof* (up to 15 points): Points will be given for the first to find mistakes in any class material (lecture slides, handouts, homework, sample code). Typo credit ranges from 0.5 to 2 points, depending on the difficulty of catching them, and how significantly they affect the meaning. Maximum 15 points.

Quizzes and Final Exams:

There will be two hour-long quizzes given during the semester outside of class time with the precise date and time to be determined later. The schedule of quizzes and final exam will be posted on the wiki in the Collab site. The final exam will be cumulative. Exams will not require you to write or correct code, though I will expect you to be able to describe basic coding principles and Matlab usage.

Tips for doing well in class:

Do the readings: Reading should be done ***before*** lecture. All of us must review any concept multiple times before it really sinks in, and since class time is the best place to get questions answered, it is vital to have seen the information at least once before class. There will also usually be a clicker question near the beginning of class that requires basic knowledge of the reading. Rather than lecturing on all the material, I want to focus in class on the material that you most need clarification on. *This will be a key to your success the class!*

Start homework early: Starting early allows you to think about the questions as the material is discussed in class. It also means that you will have your questions ready in time for lab hours and office hours.

Attend the evening lab hours: Evening lab hours bring in several TA's to help in one-on-one time.

Use pair programming: For each homework question, one may either work alone or use pair programming. Pair programming is a strategy that is gaining popularity in the workplace and in computer science departments, making up for the fact that programming is a difficult process for humans.

In pair programming, two programmers:

1. Are both present at all times when the program is being written.
2. Only one person (the "driver") types at a time; the other person (the "navigator") observes, catching errors and offering suggestions.
3. The driver and navigator switch places every 20 min, spending equal time in each role.

Please see the handout posted on Collab on pair programming for more information. If you elect to do a homework problem by pair programming, you must pledge to follow these principles. Note that if you do the equivalent of two full homework assignments with pair programming, you can fill out the pair programming questionnaire for participation points.

You may make the decision to use pair programming on a question by question basis for each assignment. You must indicate on each question if you used pair programming, and who your partner was. Both people should turn in their work individually. Switching from pair programming back to working alone on a question should be cleared with Prof. Giri.

Other notes

Class Schedule: The up-to-date class schedule with readings can be found on the Collab site, under the “Course Wiki” section. I will keep this schedule up to date throughout the semester.

Unforeseen Circumstances: In the event of very high absenteeism related to unforeseen circumstances, many of the class policies regarding attendance, homework submission, and the dates of midterm exams are subject to change.

E-mail Policy: E-mail is the best way to get in touch with Steve and myself for most class administrative matters. We try our best to answer e-mails within 24 hours. Email is not the place to have a discussion on course material, as it’s very easy to create confusion! Instead, please come to office hours, ask during the Stacks sessions, or set up a time to come by.

Academic Integrity: The School of Engineering and Applied Science relies upon and cherishes its community of trust. We firmly endorse, uphold, and embrace the University’s Honor principle that students will not lie, cheat, or steal, and we expect all students to take responsibility for the System and the privileges that it provides. We recognize that even one Honor infraction can destroy an exemplary reputation that has taken years to build. Acting in a manner consistent with the principles of Honor will benefit every member of the community both while enrolled in the Engineering School and in the future. If you have questions about your Honor System or would like to report suspicions of an Honor offense, please contact the SEAS undergraduate representatives.