Syllabus for Modeling Process Dynamics

Winter 2020

1 Course details

Instructor: Prof. Stevan Dubljevic

- email:stevan.dubljevic@ualberta.ca, DICE 13-265
- Office hours Mon-Wed 1.00pm-2.00pm or by appointment

Teaching assistants:

Guihlerme Ozorio Cassol ozorioca@ualberta.ca

Class will be on Tuesday and Thursday from 9:30-10:50am in NRE 2 020, Natural Resources Engineering Facility (NRE).

We will not be using a textbook. The course notes will be provided for you, and supplementary materials as required.

1.1 Computing Requirements

You may need to install some software for this course. If you are an incoming CPC student, this software is somethng that we require you to install. If you are not using your own laptop (desktop), you can use the open-source Anaconda Distribution which is the easiest way to use Python on Linux, Windows, and Mac OS X.

1. Anaconda Python 3.7.
   (a) Go to `https://www.anaconda.com/download/` and install the 3.7 version for your operating system. or
   (b) acces the University of Alberta jupyter server `https://ualberta.syzygy.ca/` Please see more information on `https://ist.ualberta.ca/research-computing/jupyter`. It provides an insight in the jupyter notbook at University of Alberta.
2. Git
   (a) I recommend you get it from `https://git-scm.com/downloads`.

1.2 Course description

Modeling approaches to describe relevant problems in chemical engineering practice and to gain insight in modeling realization by considering chemical engineering applications and software developments. In this course, the modeling mathematical topics relevant to chemical engineering will be used to model relevant complex chemical problems. Topics include systems based fundamental and empirical modelling, first principle models, continuous and discrete modeling, model representation, model properites, data driven models (fitting & regression analysis). The course will provide the insight in basic cooding (python) and will illustrate all sections with appropriate numerical problems.

Beartracks: Mechanistic and empirical modeling of process dynamics; continuous- and discrete-time models; model fitting and regression analysis. Corequisites: CH E 314, 318 and 345. Credit cannot be obtained in this course if previous credit has been obtained for CH E 572.

1.3 Course objectives

After completing this course you will be equiped with the modeling knowledge and you will able to use basic Python to address and/or solve problems involving the following:

- Install python and use Jupyter notebooks to create and document scientific programming solutions
  - Jupyter notebooks are one approach to integrating narrative text with code to solve scientific problems.
- Mathematical modeling of chemical engineering systems
  - You will be able to generate equations that model a chemical engineering process using principles of conservation laws, and other physical principles.
- Mathematical modelling and computational solutions to ordinary differential equations
  - You will be able to generate computational solutions to ordinary differential equations, and evaluate them for engineering problems.
- Mathematical modelling of continuous and discrete state space dynamics
  - You will be able to generate andd realize continuous and discrete model state space representations.
- Properties of the dynamical representation stability, controllability, observability

  - You will be able to characterize stability, observability and controllability of linear system

- Data based modeling
  - You will be able to use linear regression to fit data from models arising in chemical engineering

- Use scientific data to build interpolation or regression models
  - You will be ably to apply basic least square estimation technique

1.4 Course schedule

The course schedule is listed here: `https://github.com/stevand/W20-CHE472` and in the eclass

1.5 Grading

Each assignment will be in one of these categories. Your final grade will be determined by the weighted sum of all your grades.

Category	weight
homework	0.20
quiz	0.10
lab	0.20
exam-1	0.15
exam-2	0.15
final-exam	0.2
Total	1.0

1.6 Grading criteria

You are at the final years of your undergraduate education and in transitioning process to a young professional. Hence, the work need to be technically correct and in addition we will look for the professional completion of a assignments and tasks.

The following criteria will be used:

- "**A**" work has the following features. The correct approach is used and the problem solution is set up correctly. The work is not over-simplified and it is easy to follow and to confirm that it is done correctly. If any assumptions were made they must be stated and justified. The appropriate units are used and figures are appropriately labeled with units and axes. Everything is done as expected and you should be happy and proud of your work.

- "**B**" work is deficient in one or more aspects of "A" work. The essential details are missing and the presentation is sloppy. There will be no praise for this kind of work.
- "**C**" quality work is deficient in more than two of the properties of "A" work. This is the bare minimum of expected performance.
- "**D**" work is not considered as acceptable performance.
- "**F**" work is totally unacceptable performance. You will be fired.

At the end of the semester fraction of the possible points you have earned is calculated, and your grade will be based on class distribution.

1.7 Academic conduct

All work is expected to be completed by you (student in the class) and to be your original work. It is permitted to work with the class members to solve the homework problems, but you must turn in your own solutions since the HW are personalized for each student (two students who turn in the same work will receive zeros and warning). It is considered cheating to turn in someone else's work as your own. If you use code from the internet or the course notes, you should note this in your solution. Repeat offenses will be reported as academic dishonesty.

Here are some examples of acceptable collaboration:

- Clarifying ambiguities or vague points in class handouts or lectures.
- Discussing or explaining the general class material.
- Providing assistance with Python, in using Jupyter notebooks, or with editing, debugging, and Python tools.
- Discussing the code that we give out on the assignment.
- Discussing the assignments to better understand them.
- Getting help from anyone concerning programming issues which are clearly more general than the specific assignment (e.g., what does a particular error message mean?)

1.8 Accommodations

If you need to have an accommodations, please bring the letter from the Disability Resources office. Please discuss your accommodations and needs with me as early as possible. We will work with you to ensure that accommodations are provided as appropriate.