# Training All Chemical Engineers in Computing and Data Science


**Prof. Alexander Dowling**

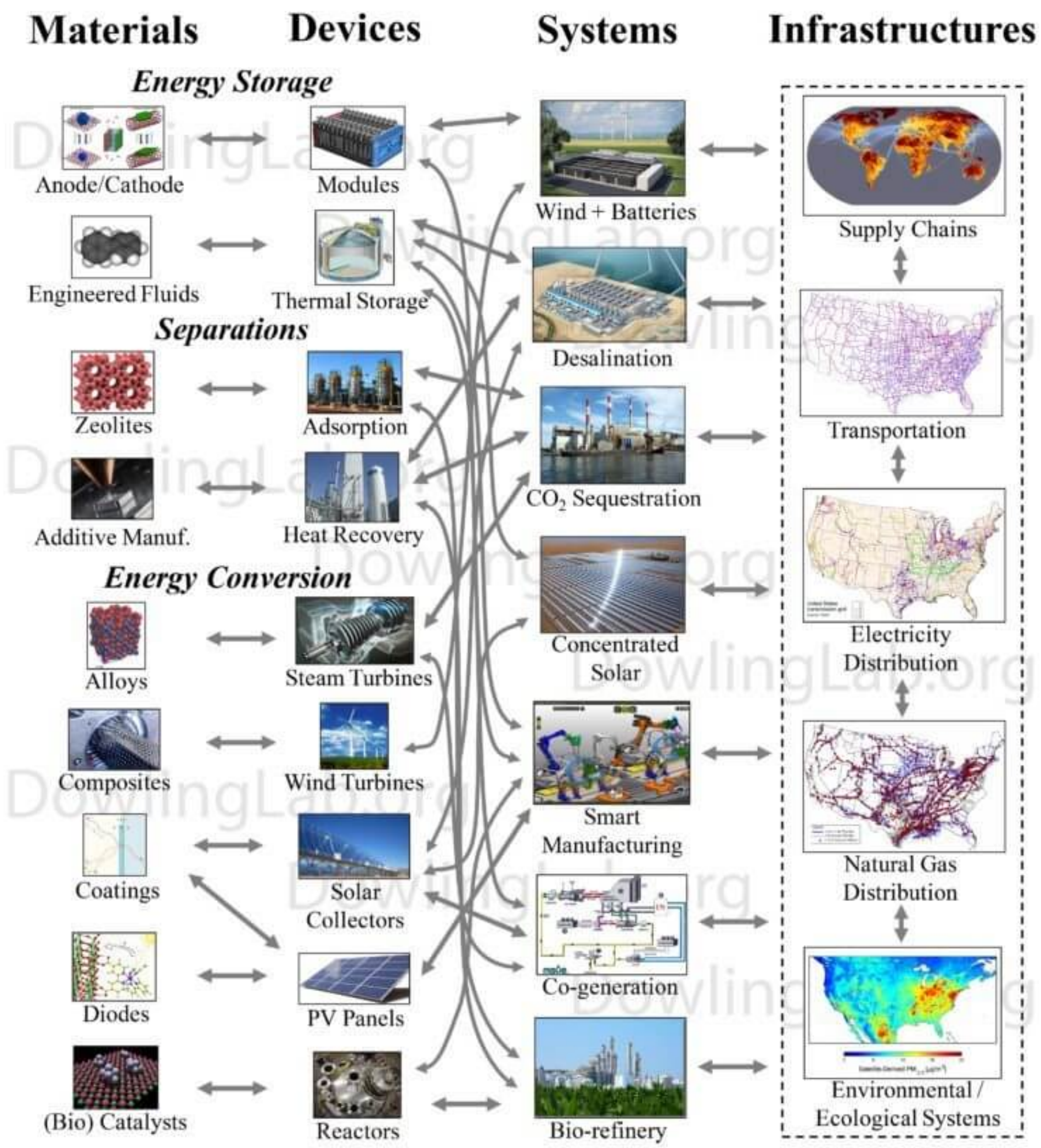
**adowling@nd.edu   dowlinglab.nd.edu**

Department of Chemical and Biomolecular Engineering

University of Notre Dame, Notre Dame, IN

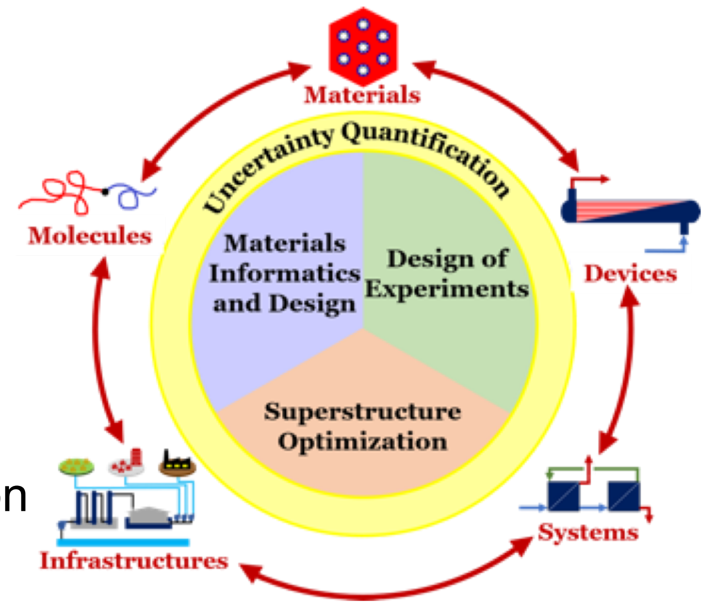

November 11, 2019   AIChE Annual Meeting   Orlando, FL

# Research: Process Systems Engineering

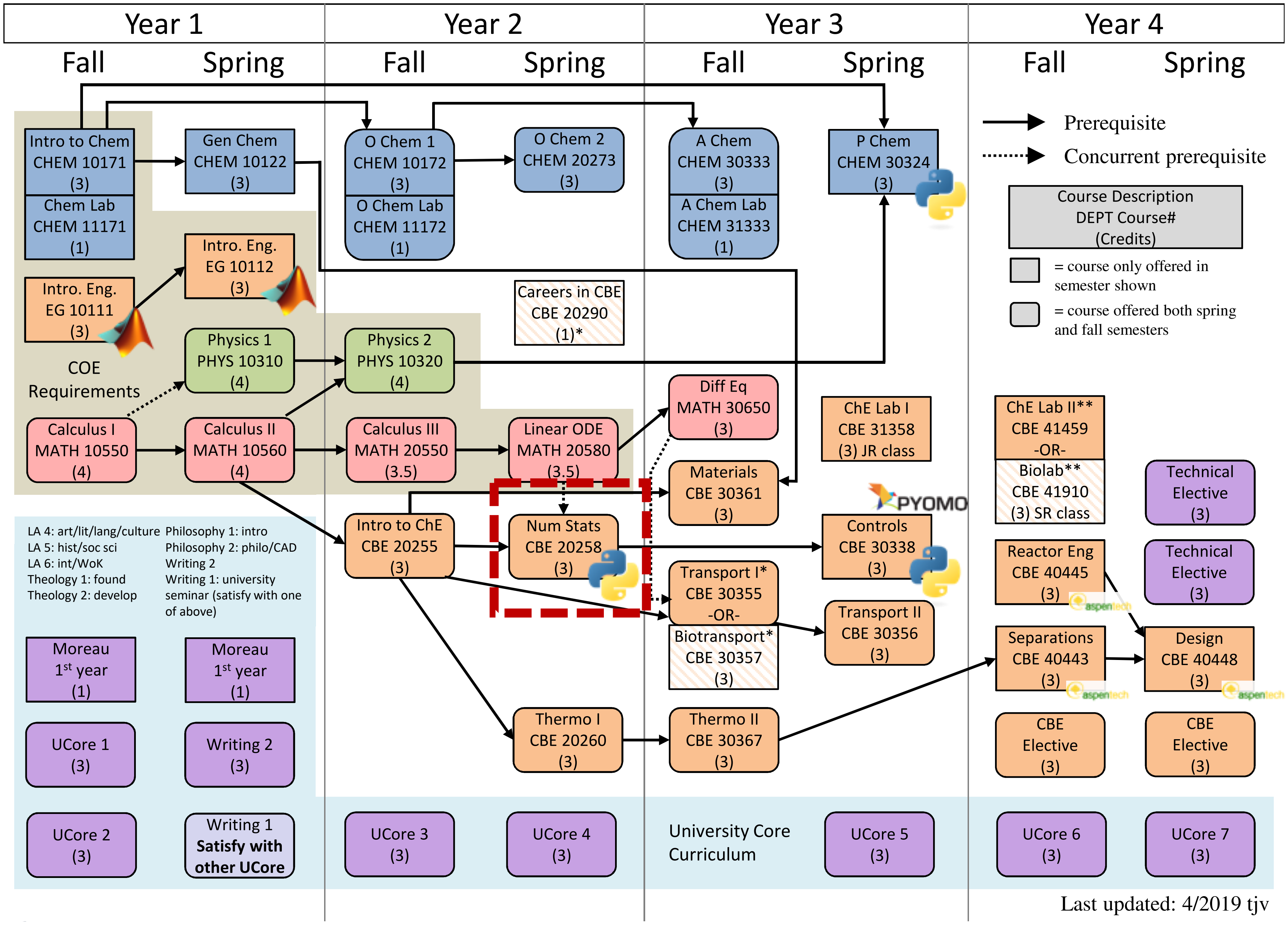



## Themes

- Mathematical Modeling
- Computational Optimization
- Applied Bayesian Statistics and Uncertainty Quantification
- Energy & Sustainability Applications




# Chemical Engineering Suggested 4 Year Curriculum University of Notre Dame entering FA18




## Current Practice: Computing & Statistics

MATLAB in freshman engineering sequence

Sophomore-required **Numerical & Statistical Analysis** (NSA)

Ad-hoc computing & statistics in upper-level classes:

*"You learned this as sophomores… just figure it out" – Prof. Anonymous*

## Vision

*Vertically integrate computing and statistics throughout the undergraduate curriculum*

# Vertical Integration: Opportunities and Challenges

**Numerical Methods**

Equation solving (thermo., separations)

Optimization (controls, design)

Numeric integration (transport, reactions)

**Statistics and Data Analysis**

Probability (physical chemistry)

Visualization

Regression

Error Analysis

Uncertainty Propagation

labs

**Challenges**

Difficult to learn advanced topics sophomore year (e.g., PDEs, BVPs)

"Brian drain" without repeated exposure

Common software tools?

How to avoid burdening all faculty?

# Modernizing Numerical and Statistical Analysis

## Backward Course Design

## Set Clear Learning Objectives

At the end of the semester, you should be able to…

1. **Create mathematical models** and **apply computational methods** to analyze systems using basic principles of chemical engineering (e.g., mass and energy balances, thermodynamic equilibrium, etc.)

2. **Analyze data** and **quantify uncertainty** using standard statistical techniques and mathematical models grounded in engineering fundamentals

3. Independently plan, implement, and debug short (100 to 300 lines) **Python computer programs** to analyze data, solve engineering mathematical models, and visualize results

## Major Changes

**Reorganized class topics**

- Removed advanced topics (QR factorization, compression with SVD, trust regions, BVPs, PDEs)
- Emphasized fundamentals, especially probability & statistics
- Added mass and energy balance examples

Switched to **Python**, with great student buy-in

Incorporated **active learning** into lectures

**Shortened assignments**

# Active Learning is Essential for Computing and Statistics

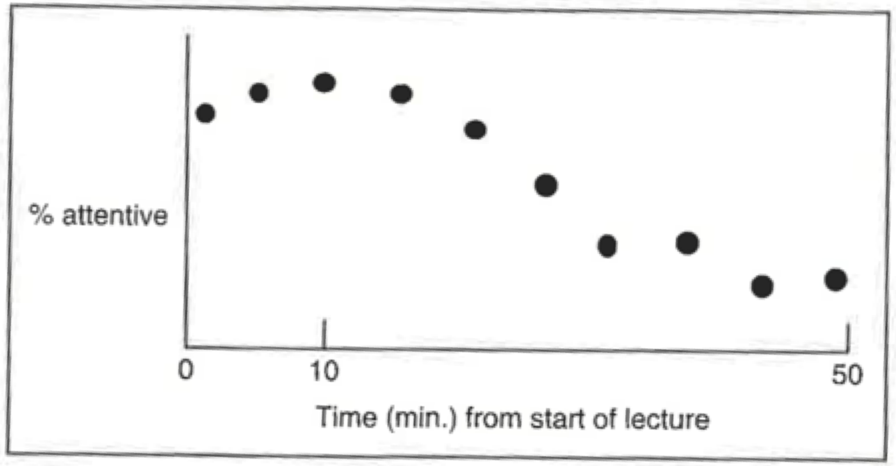



Figure 6.3–1: Attentiveness versus Time in Lecture—No Activities

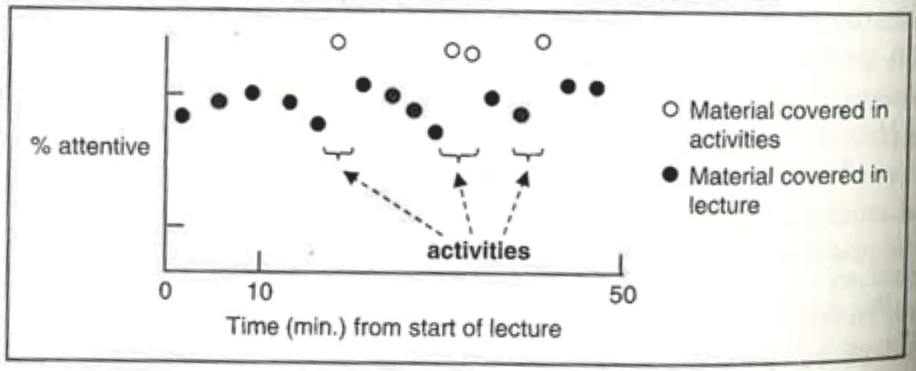



Figure 6.3–2: Attentiveness versus Time in Lecture—Activities Interspersed



Right: Felder and Brent (2015), *Teaching and Learning STEM: A Practical Guide*

# Spring 2019: Cloud-based Google Colaboratory (Jupyter Notebooks)

`colab.research.google.com`

**Benefits of Google Colaboratry:**

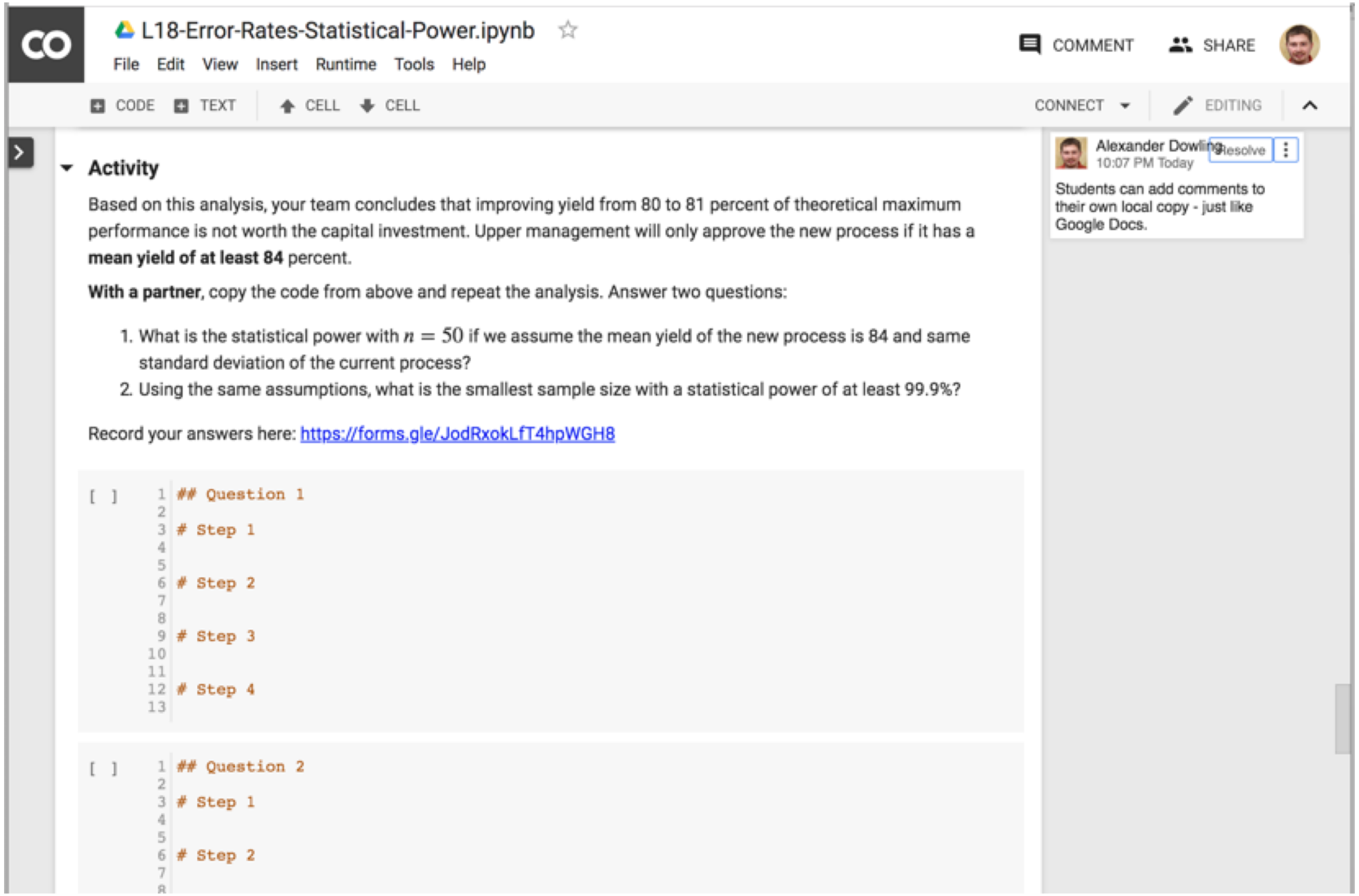
Like Google Docs, but for code

Integrated with **Google Drive**: automatic versioning, easy sharing

Removes barriers to access: students can complete assignments from **any internet connect computer** – no need to support 80+ local Python installations

Facilitates **active learning**

Free




UNIVERSITY OF NOTRE DAME

*I show this slide on day 1.*

# Making your time more effective

**Easy** — Complexity of Concepts and Applications — **Difficult**

### *Traditional Class (plan)*

$\underbrace{\textbf{Lecture } 3 \text{ hr / wk}}_{\textit{Initial exposure}} + \underbrace{\textbf{Tutorial } 1 \text{ hr / wk}}_{\textit{Practice}} + \underbrace{\textbf{Studying } 2 \text{ hr / wk}}_{\textit{But how?}} + \underbrace{\textbf{Homework } 3 \text{ hr / wk}}_{\textit{Builds on lectures}} = \textbf{Total } 9 \text{ hr / wk}$

### *Traditional Class (reality for many)*

$\underbrace{\textbf{Lecture } 3 \text{ hr / wk}}_{\textit{Difficult to focus 100\% of time}} + \underbrace{\textbf{Tutorial } 1 \text{ hr / wk}}_{\textit{Too busy to start homework before}} + \underbrace{\textbf{Studying } 1 \text{ hr / wk}}_{\textit{Either skipped or not effective}} + \underbrace{\textbf{Homework } 10+ \text{ hr / wk}}_{\textit{Really difficult if not comfortable with lecture examples}} = \textbf{Total } 15+ \text{ hr / wk}$

# Making your time more effective

*I show this slide on day 1.*

**Easy** Complexity of Concepts and Applications **Difficult**

**We'll start some homework problems during class.*

***This Semester***

**Class Preparation**
2 hr / wk

*Initial exposure at home*

\+

**Tutorial**
1 hr / wk

*Practice & jump-start homework*

\+

**Studying**
1 hr / wk

*I'll teach you how to do this & give extra practice problems*

\+

**Class***
3 hr / wk

*Problem solving together*

**Homework***
2 - 4 hr / wk

*Easy extensions of home and class activities*

=

**Total**
9 - 11 hr / wk

*This is 100% on task time… i.e., Facebook closed, not watching Netflix, not texting*

# Fall 2019: Cloud-based Vocareum (Jupyter Notebooks)

www.vocareum.com

**Benefits of Vocareum:**

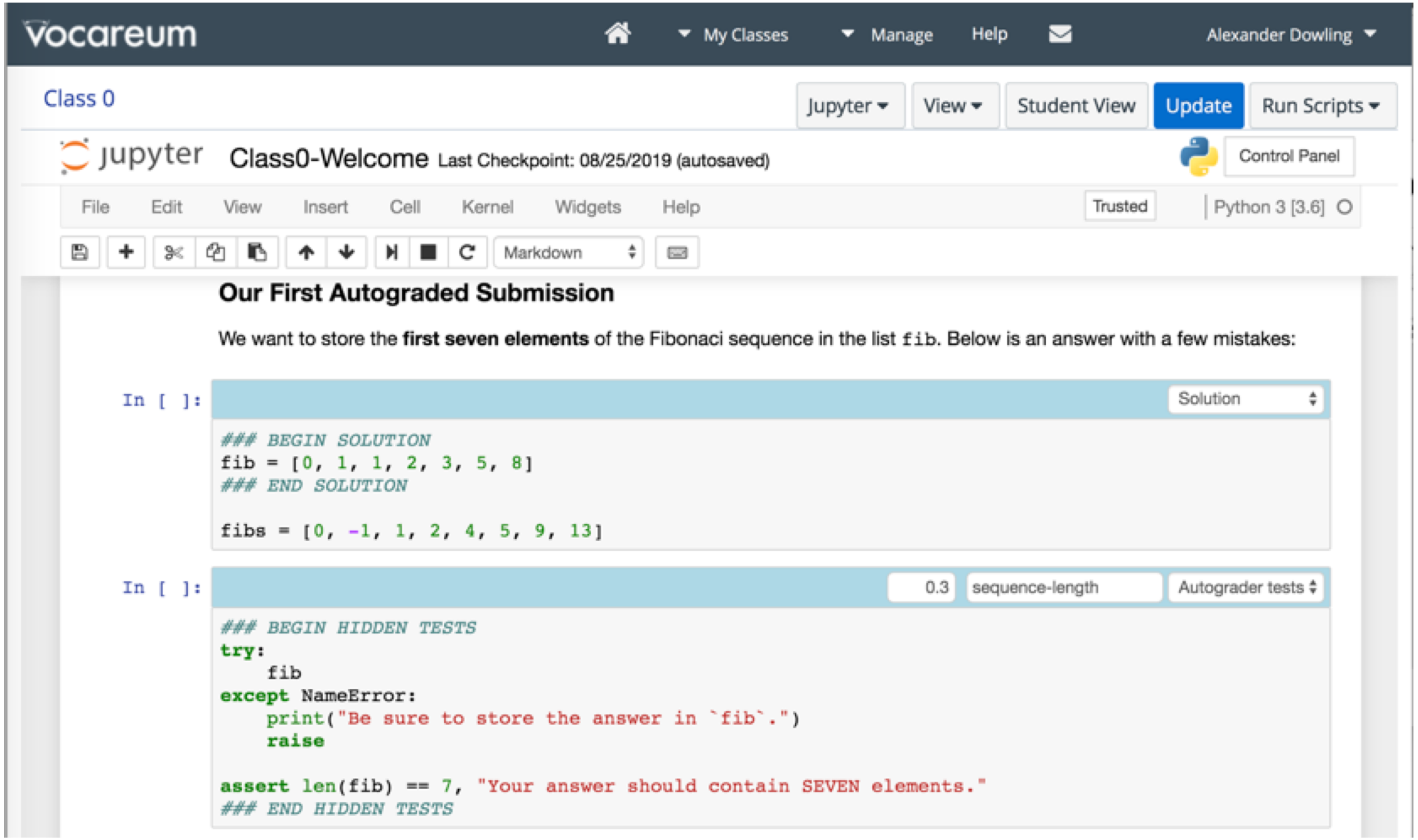
Many of the same cloud-based benefits as Colaboratory

Integrated with **Learning Management System** (e.g., Sakai) and gradebook

Supports **autograding** via nbgrader (with some enhancements)

Supports plagiarism detections (if you want it)

Paid service, but responsive technical support



# Fall 2019: Cloud-based Vocareum (Jupyter Notebooks)

`www.vocareum.com`

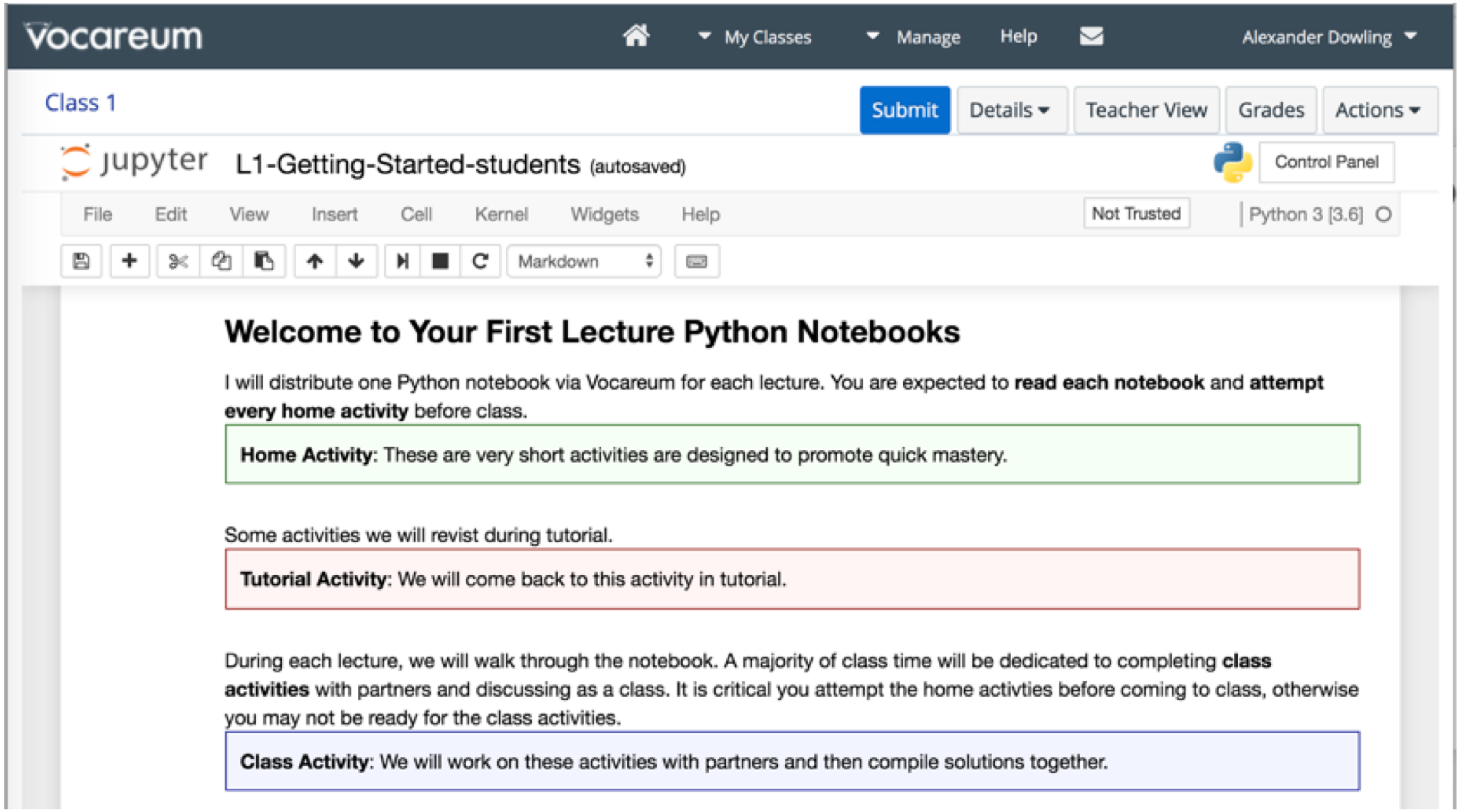
**Benefits of Vocareum:**

Many of the same cloud-based benefits as Colaboratory

Integrated with **Learning Management System** (e.g., Sakai) and gradebook

Supports **autograding** via nbgrader (with some enhancements)

Supports plagiarism detections (if you want it)




**Bottom Line:** Autograder (Vocareum) enables accountability for meaningful home activities before class, which translates to more engaging class sessions.

# Excited Students to Become Long-Term Learners

*Success Stories and Lessons Learned*

Show students how computing and statistics:
- Makes them competitive for jobs
- Helps them in future classes & career
- **Connects to chemical engineering & society**

Set them up for **early success**
- Require pseudocode
- Embrace the autograder

Show them how to **study & learn independently**

**Extension Assignments**

Watch TED talk, listen to podcast, etc.

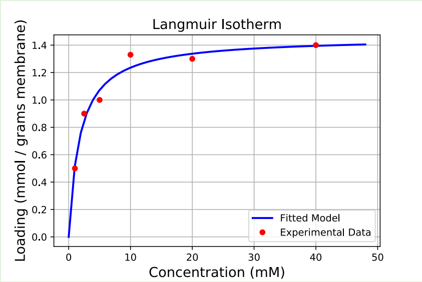
Answer brief reflection questions

Completion grade, counts towards dropped homework

Examples:
*Planet Money*, “What Causes What?”
*Hidden Brain*, “The Scientific Process”
*TED Talks*: Bias and Algorithms

**Final Project: Heavy Metal Water Treatment**

*Fit Isotherm Model & Estimate Uncertainty*

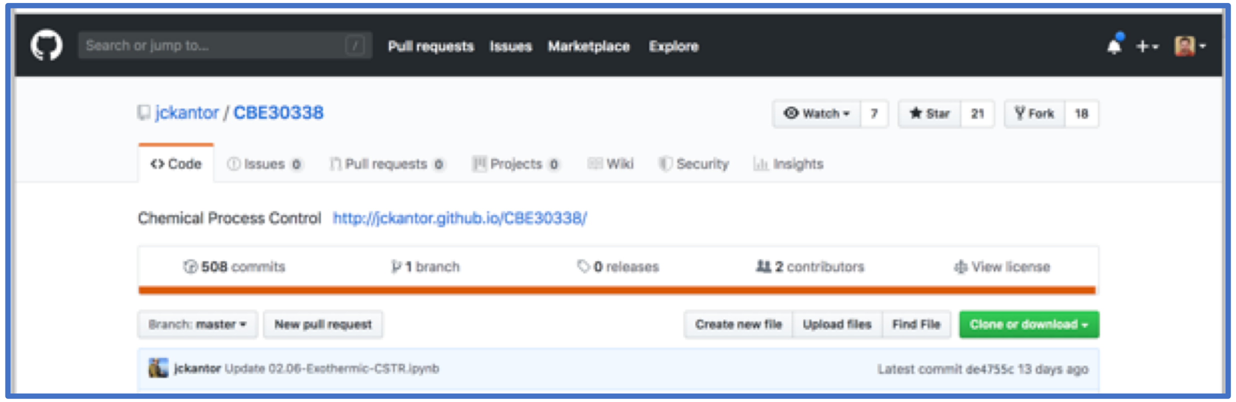



*Calculate Filter Lifespan*

$$\frac{d}{dt}q(t) = \frac{F}{m}(c_{in} - c(t))$$

*Draw Conclusions with Uncertainty*

# Special Thanks to Prof. Jeff Kantor



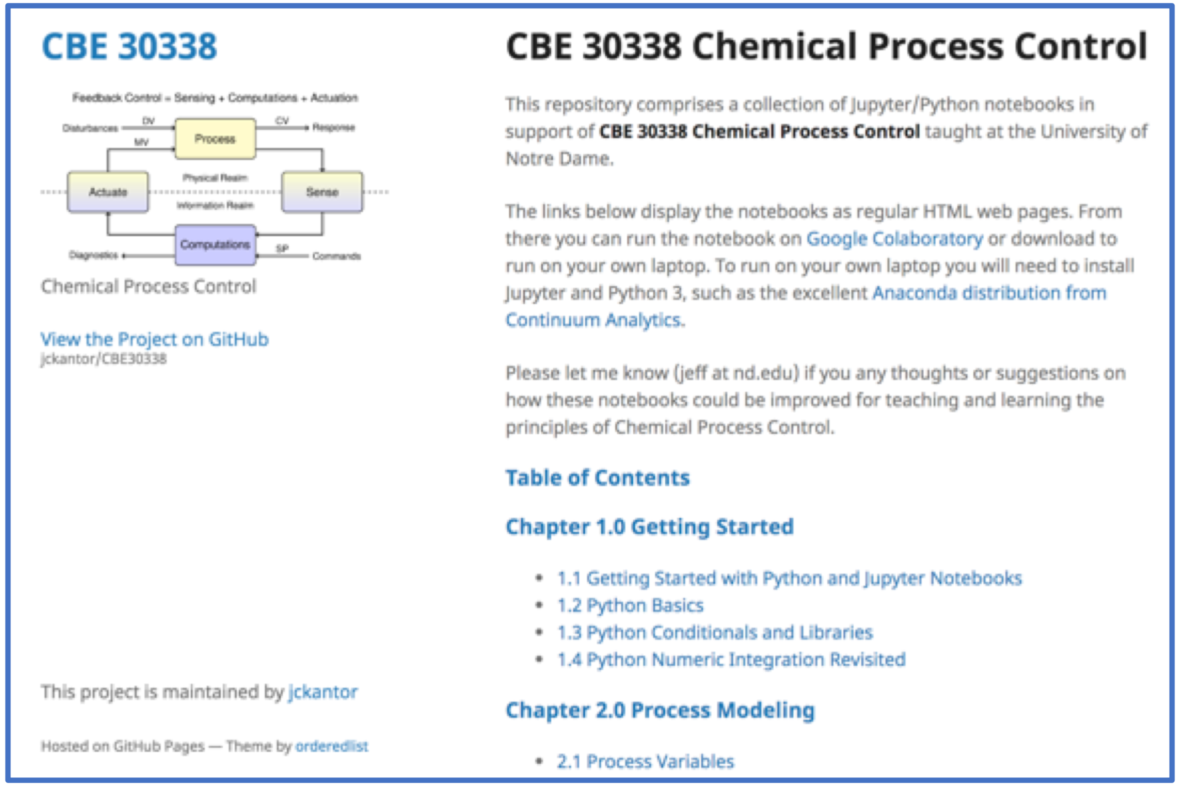

https://github.com/jckantor

Chemical Process Control

Introduction to Chemical Engineering Analysis

Introduction to Operations Research

Process Operations




# Training All Chemical Engineers in Computing and Data Science

**Prof. Alexander Dowling**

**adowling@nd.edu    dowlinglab.nd.edu**

Department of Chemical and Biomolecular Engineering

University of Notre Dame, Notre Dame, IN

November 11, 2019    AIChE Annual Meeting    Orlando, FL

colab

vocareum

python™

pandas

NumPy

UNIVERSITY OF NOTRE DAME