Numerical Problem Solving across the Curriculum with Python and MATLAB Using Interactive Coding Templates

Presenters:

Ashlee N. Ford Versypt, University at Buffalo

Matthew D. Stuber, University of Connecticut

Robert P. Hesketh, Rowan University

ashleefv@buffalo.edu

matthew.stuber@uconn.edu

hesketh@rowan.edu

ASEE/AIChE Summer School for Engineering Faculty

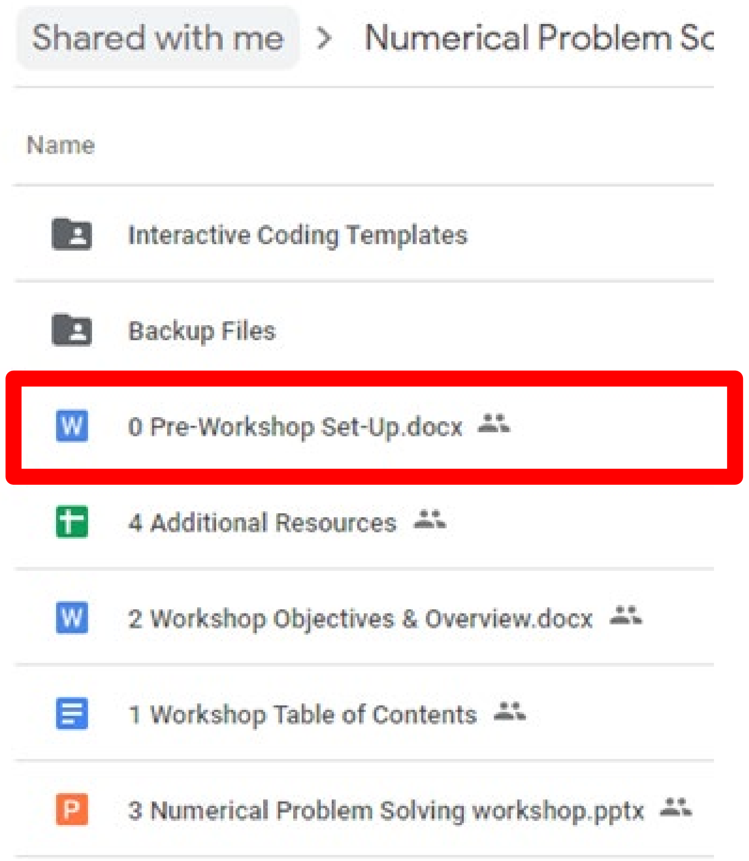
Assisted by Austin Johns, anjohns@buffalo.edu

Pre-Workshop Set-Up

- Pre-Workshop Set-Up Link
- Access workshop Google Drive folder
- Open interactive coding templates
  - Obtain MATLAB license (.mlx)

  OR

  - Google Colab (.ipynb)

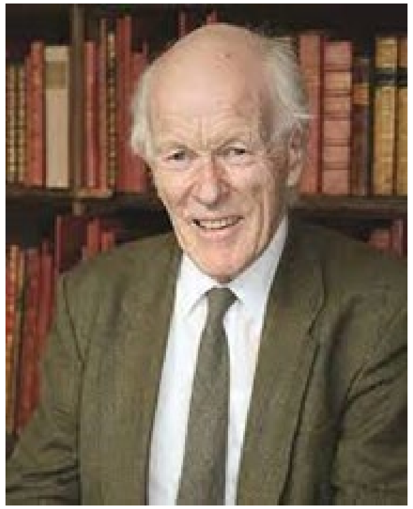


Shared with me > Numerical Problem Sc

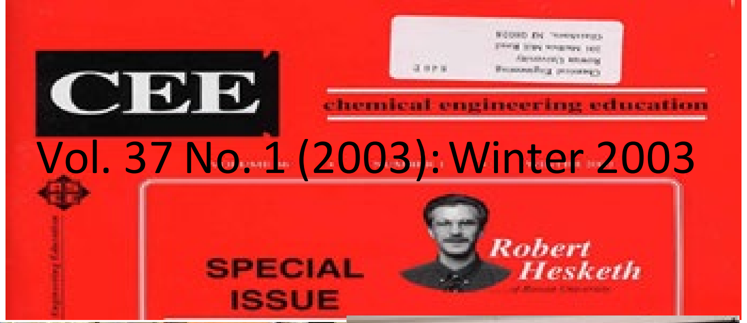
Name
Interactive Coding Templates
Backup Files
0 Pre-Workshop Set-Up.docx
4 Additional Resources
2 Workshop Objectives & Overview.docx
1 Workshop Table of Contents
3 Numerical Problem Solving workshop.pptx

Robert Hesketh

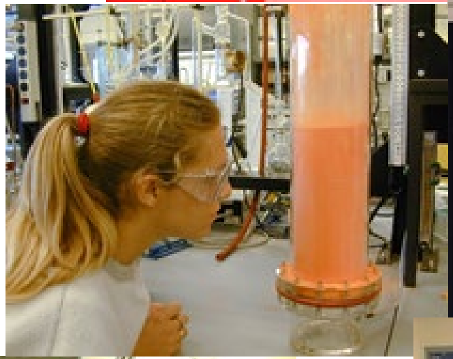
- B.S. University of Illinois, Champaign-Urbana, May 1982
- Ph.D. University of Delaware, May 1987
- Postdoc University of Cambridge, 1987
- University of Tulsa, 1990
- Rowan University, 1996

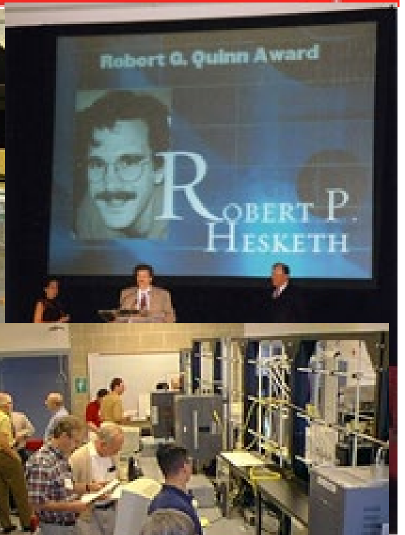


- CACHE Trustee
- ASEE Chester F. Carlson 2006
- ASEE Robert G. Quinn 2002
- AIChE Gary Leach 2007
- ASEE Chester F. Carlson 2006
- Featured Educator, Chemical Engineering Education, 37(1) 2003
- ASEE Robert G. Quinn 2002 Award
- ASEE Ray W. Fahien Award 1999
- ASEE Joseph J. Martin Awards
- Dow Outstanding New Faculty Award ASEE



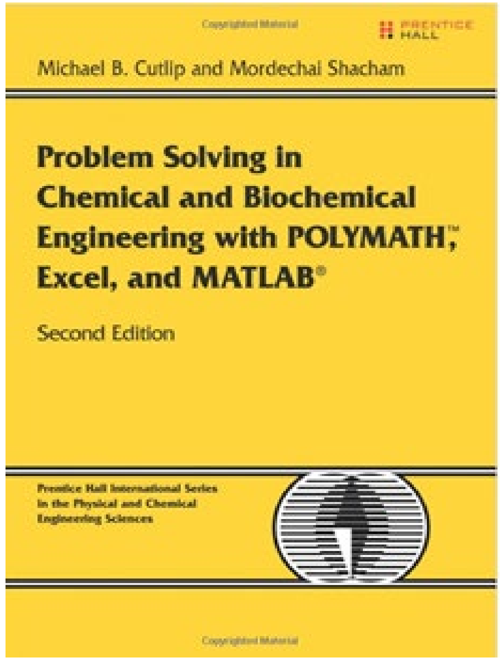
Vol. 37 No. 1 (2003): Winter 2003





Robert’s Use of Python in the Classroom

- Mass & Energy Balance (Felder, Rousseau, Bullard)
- ChE Fluid Mechanics (fluids 1)
- Process Fluid Transport (fluids 2)
- Separations I
- Chemical Reaction Engineering
- Transport Phenomena
- Grad Classes



Copyrighted Material

PRENTICE HALL

Michael B. Cutlip and Mordechai Shacham

Problem Solving in Chemical and Biochemical Engineering with POLYMATH™, Excel, and MATLAB®

Second Edition

Prentice Hall International Series in the Physical and Chemical Engineering Sciences

Copyrighted Material

Matthew D. Stuber

Process Systems and Operations Research Laboratory

Academics

- PhD MIT, 2013
- BChE UMN-TC, 2007

Professional

- Asst. Prof., University of Connecticut, 2016
- Entrepreneurial, 2012

Teaching

- Optimization
- Numerical Analysis
- Robust Design

- MIT Teaching Certificate
- UConn CETL Online Course Design
- UConn CETL Mini-Grant Competition
- CACHE Grant Award

Ashlee N. Ford Versypt (Pronounced: Ford Vurr-SIPPED)
Pronouns: she/her/hers
Associate Professor
Dept. of Chemical & Biological Engineering
University at Buffalo
Starting 9th year as faculty, 15th semester of teaching
PhD UIUC 2012 (ChE + Computational Science & Engineering)
BS OU 2005, postdoc MIT 2012-2014, Asst Prof OK State 2014-2021

Why I am teaching this course

- Have taught kinetics, fluids, heat transfer, process control, mass and energy balances, grad math (analytical & numerical methods), elective "applied numerical computing" featuring MATLAB & Python
- I'm very passionate about engineering education and computing
- I use math and computing applied to ChE principles for my research
- CACHE Trustee, AIChE CAST & ASEE CHED Exec Comm member
- AIChE CAST Himmelblau Award for Innovations in Computer-Based Chemical Engineering Education, UIUC Teaching Certificate, ASEE Fahien Award, NSF CAREER, DOE CSGF, CACHE minigrant

CACHE Development of Computational-Based Tools & Modules for Chemical Engineering Education RFP

Project Goal

This program seeks to fund 12-month-long projects (maximum budget of $5000 per project) that focus on the development of novel computational-based educational modules or tools.

Proposals are due by 5 pm EDST, August 31, 2022 and awards will be made by October 1, 2022. Details related to proposal preparation and submission can be found at https://cache.org/computational-tools-development

Examples of Past Projects

Modernizing the ChE Curriculum via Interactive Visual Aids for Traditional Courses

Development of Open Access Version of Applied Numerical Computing Course (Ford Versypt)

Coding Concepts and Chemical Engineering Analysis With Interactive Jupyter Notebooks (Stuber)

Learning objectives for the workshop

By the end of this workshop, participants will be able to

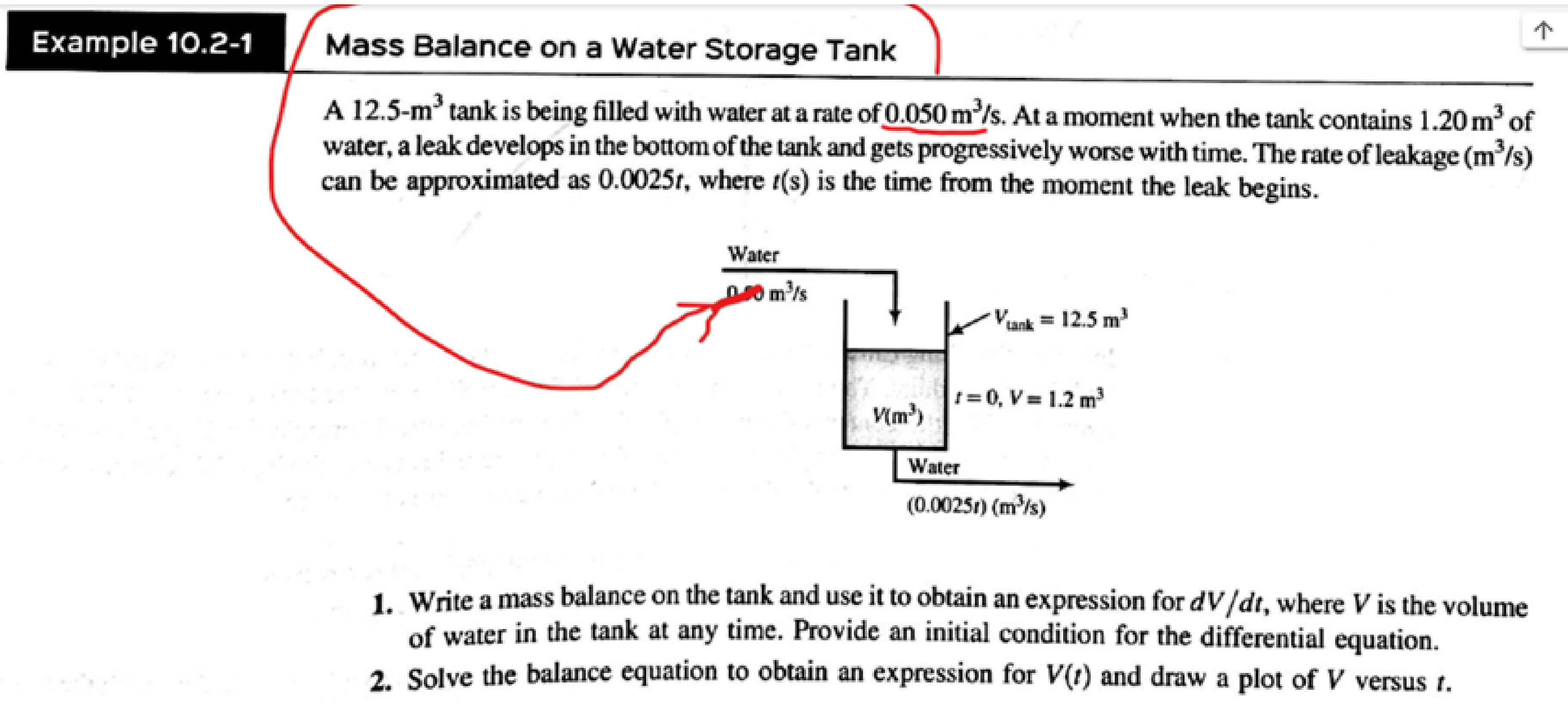
- Create interactive coding templates (MATLAB Live Scripts or Jupyter Notebooks using Python) for teaching chemical engineering concepts and problem-solving
- Select, run, and interact with a MATLAB Live Script or Jupyter Notebook template applied to a chemical engineering topic of their choice

Example interactive coding templates

- Ex. 10.2-1 from MEB book by Felder, Rousseau, & Bullard
- Jupyter Notebook (Python) version: J6_TankDrainage.ipynb
- MATLAB version: M6_TankDrainage.mlx

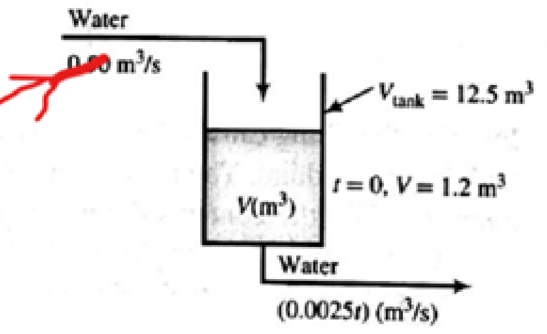
ODEfun(t, y_i's)

$$\frac{dV}{dt} = \dot{v}_{in} - \dot{v}_{out}$$



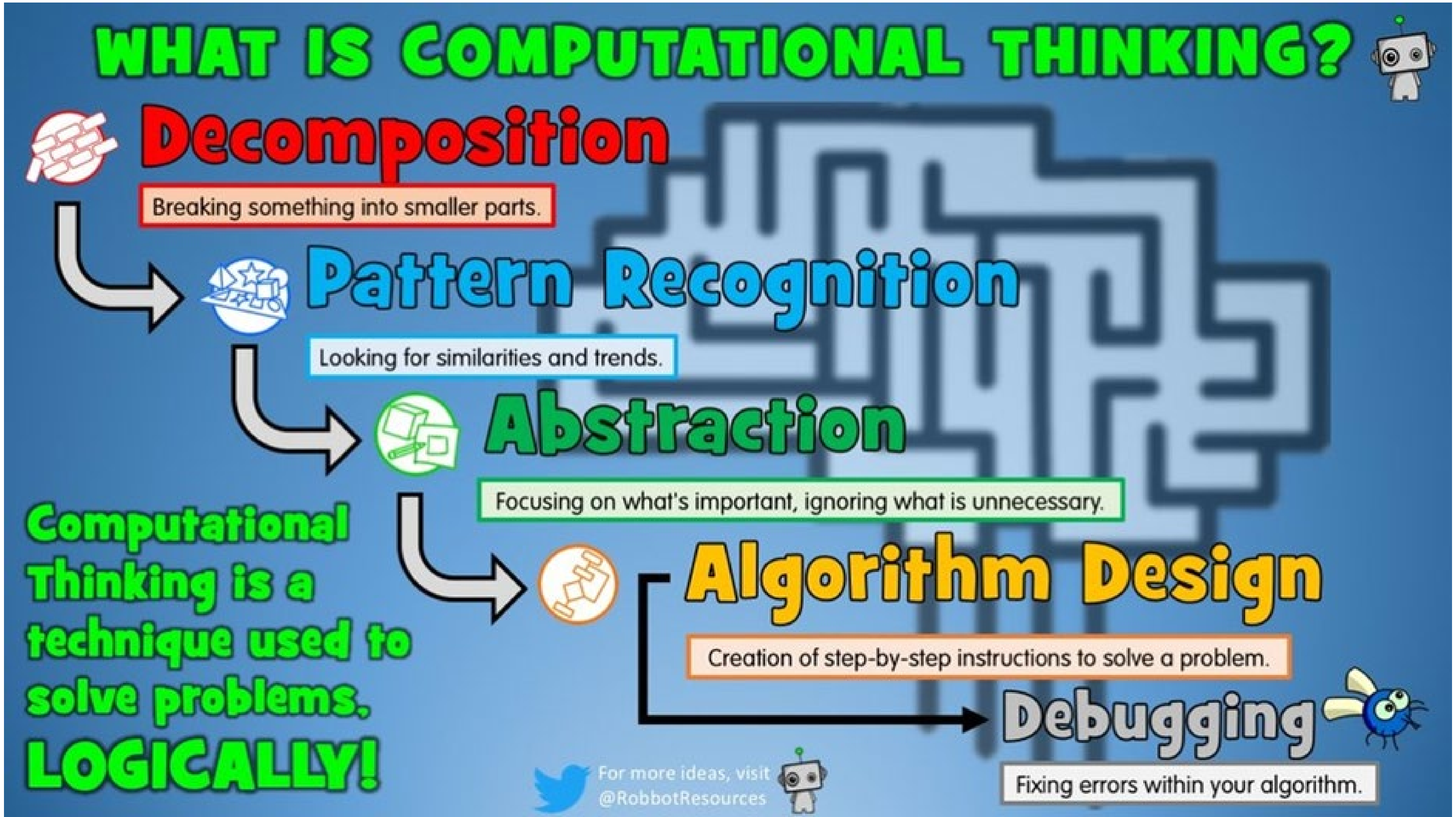
Example 10.2-1 Mass Balance on a Water Storage Tank

A 12.5-m^3 tank is being filled with water at a rate of 0.050 m^3/s. At a moment when the tank contains 1.20 m^3 of water, a leak develops in the bottom of the tank and gets progressively worse with time. The rate of leakage (m^3/s) can be approximated as $0.0025t$, where t(s) is the time from the moment the leak begins.



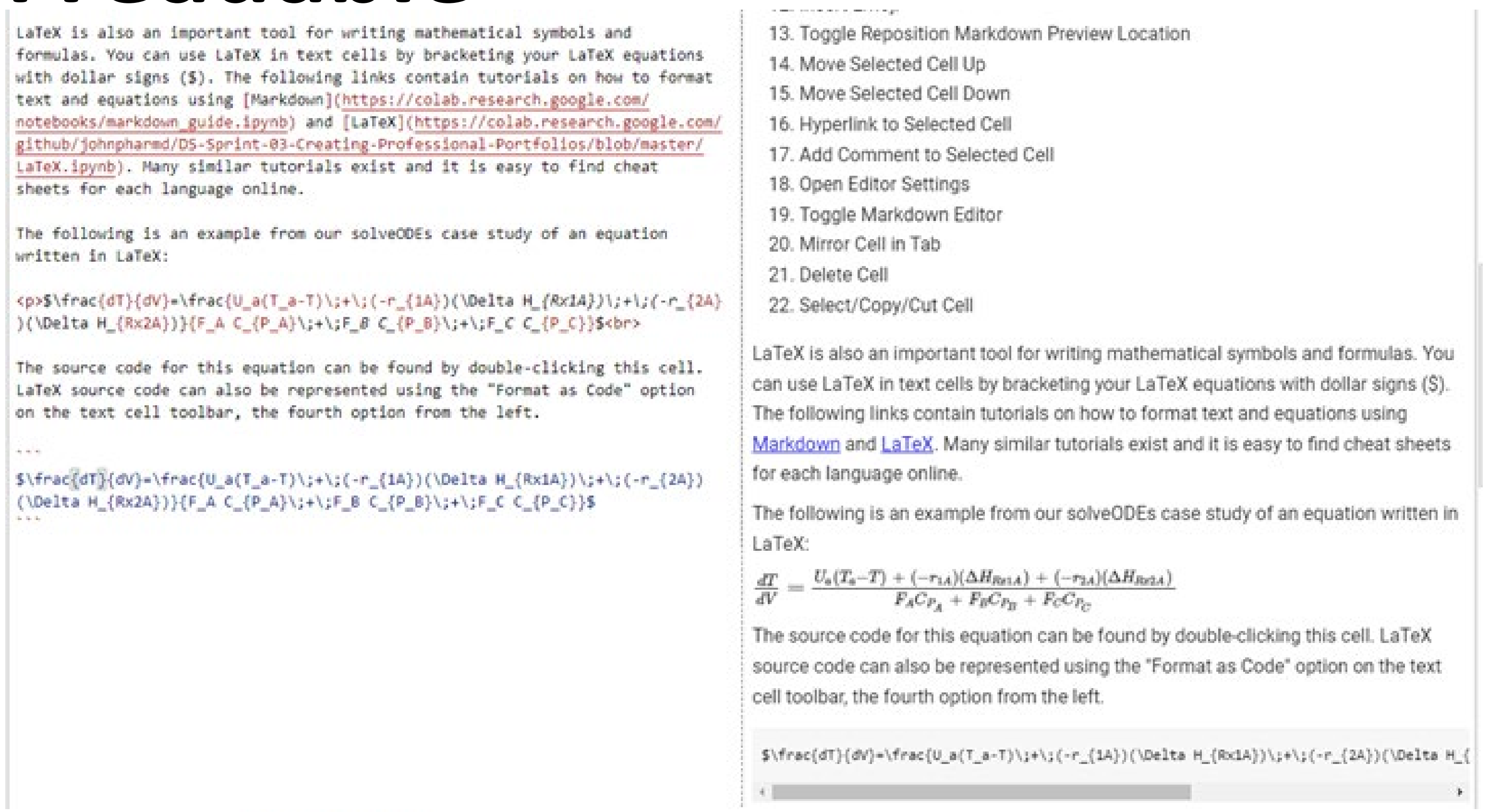
1. Write a mass balance on the tank and use it to obtain an expression for dV/dt, where V is the volume of water in the tank at any time. Provide an initial condition for the differential equation.
2. Solve the balance equation to obtain an expression for $V(t)$ and draw a plot of V versus t.

Educational goal: computational thinking for problem solving


WHAT IS COMPUTATIONAL THINKING?
Decomposition
Breaking something into smaller parts.
Pattern Recognition
Looking for similarities and trends.
Abstraction
Focusing on what's important, ignoring what is unnecessary.
Algorithm Design
Creation of step-by-step instructions to solve a problem.
Debugging
Fixing errors within your algorithm.
Computational Thinking is a technique used to solve problems. LOGICALLY!
For more ideas, visit @RobbotResources

Literate programming making code human readable

- Focus on natural language
- Text and code blocks
- Formatting for clear documentation
  - Alternative to extensive commenting
- Educational applications
- Markup language vs programming language
- [Literate Programming - Wikipedia]()



```
LaTeX is also an important tool for writing mathematical symbols and
formulas. You can use LaTeX in text cells by bracketing your LaTeX equations
with dollar signs ($). The following links contain tutorials on how to format
text and equations using [Markdown](https://colab.research.google.com/
notebooks/markdown_guide.ipynb) and [LaTeX](https://colab.research.google.com/
github/johnpharmd/DS-Sprint-03-Creating-Professional-Portfolios/blob/master/
LaTeX.ipynb). Many similar tutorials exist and it is easy to find cheat
sheets for each language online.

The following is an example from our solveODEs case study of an equation
written in LaTeX:

<p>$\frac{dT}{dV}=\frac{U_a(T_a-T)\;+\;(-r_{1A})(\Delta H_{Rx1A})\;+\;(-r_{2A}
)(\Delta H_{Rx2A})}{F_A C_{P_A}\;+\;F_B C_{P_B}\;+\;F_C C_{P_C}}$<br>

The source code for this equation can be found by double-clicking this cell.
LaTeX source code can also be represented using the "Format as Code" option
on the text cell toolbar, the fourth option from the left.

```
$\frac{dT}{dV}=\frac{U_a(T_a-T)\;+\;(-r_{1A})(\Delta H_{Rx1A})\;+\;(-r_{2A})
(\Delta H_{Rx2A})}{F_A C_{P_A}\;+\;F_B C_{P_B}\;+\;F_C C_{P_C}}$
```
```

13. Toggle Reposition Markdown Preview Location
14. Move Selected Cell Up
15. Move Selected Cell Down
16. Hyperlink to Selected Cell
17. Add Comment to Selected Cell
18. Open Editor Settings
19. Toggle Markdown Editor
20. Mirror Cell in Tab
21. Delete Cell
22. Select/Copy/Cut Cell

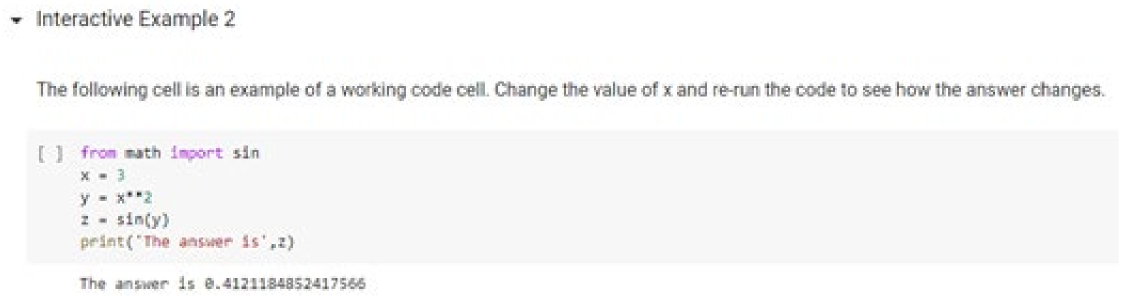
LaTeX is also an important tool for writing mathematical symbols and formulas. You can use LaTeX in text cells by bracketing your LaTeX equations with dollar signs ($). The following links contain tutorials on how to format text and equations using Markdown and LaTeX. Many similar tutorials exist and it is easy to find cheat sheets for each language online.

The following is an example from our solveODEs case study of an equation written in LaTeX:

$$\frac{dT}{dV}=\frac{U_a(T_a-T)\;+\;(-r_{1A})(\Delta H_{Rx1A})\;+\;(-r_{2A})(\Delta H_{Rx2A})}{F_A C_{P_A}\;+\;F_B C_{P_B}\;+\;F_C C_{P_C}}$$

The source code for this equation can be found by double-clicking this cell. LaTeX source code can also be represented using the "Format as Code" option on the text cell toolbar, the fourth option from the left.

```
$\frac{dT}{dV}=\frac{U_a(T_a-T)\;+\;(-r_{1A})(\Delta H_{Rx1A})\;+\;(-r_{2A})(\Delta H_{
```



Interactive Example 2

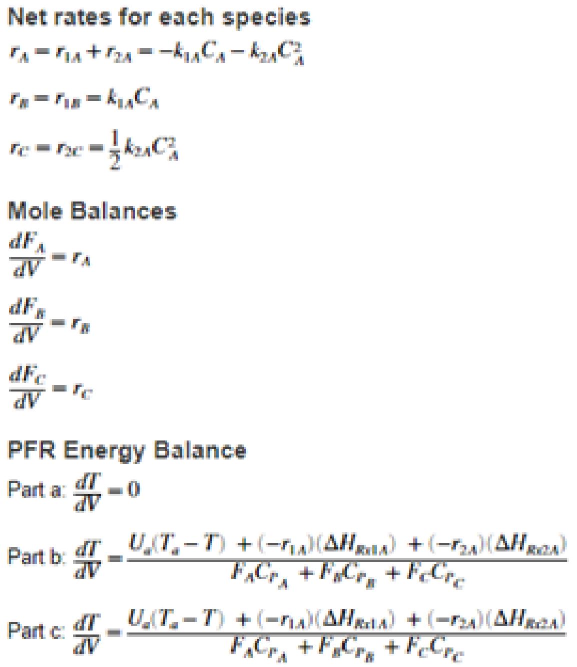
The following cell is an example of a working code cell. Change the value of x and re-run the code to see how the answer changes.

```
from math import sin
x = 3
y = x**2
z = sin(y)
print('The answer is',z)
```

```
The answer is 0.4121184852417566
```

MATLAB Live Script (.mlx)

- MathWorks literate programming
- Works using online and standalone client
- MATLAB code + Simplified Markdown
  - LaTeX available for equations
- Functions defined at end of document
- Interactive elements



Net rates for each species

$r_A = r_{1A} + r_{2A} = -k_{1A}C_A - k_{2A}C_A^2$

$r_B = r_{1B} = k_{1A}C_A$

$r_C = r_{2C} = \frac{1}{2}k_{2A}C_A^2$

Mole Balances

$\frac{dF_A}{dV} = r_A$

$\frac{dF_B}{dV} = r_B$

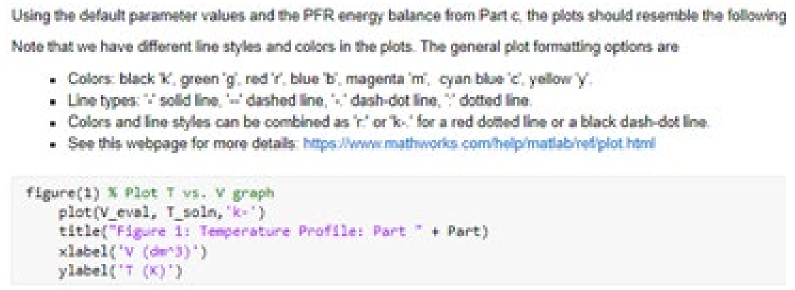
$\frac{dF_C}{dV} = r_C$

PFR Energy Balance

Part a: $\frac{dT}{dV} = 0$

Part b: $\frac{dT}{dV} = \frac{U_a(T_a - T) + (-r_{1A})(\Delta H_{Rx1A}) + (-r_{2A})(\Delta H_{Rx2A})}{F_A C_{P_A} + F_B C_{P_B} + F_C C_{P_C}}$

Part c: $\frac{dT}{dV} = \frac{U_a(T_a - T) + (-r_{1A})(\Delta H_{Rx1A}) + (-r_{2A})(\Delta H_{Rx2A})}{F_A C_{P_A} + F_B C_{P_B} + F_C C_{P_C}}$

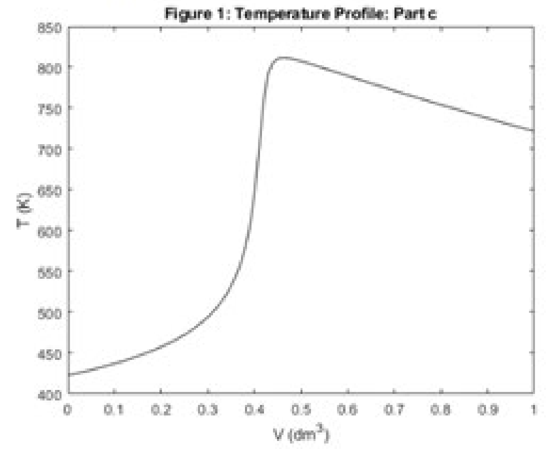


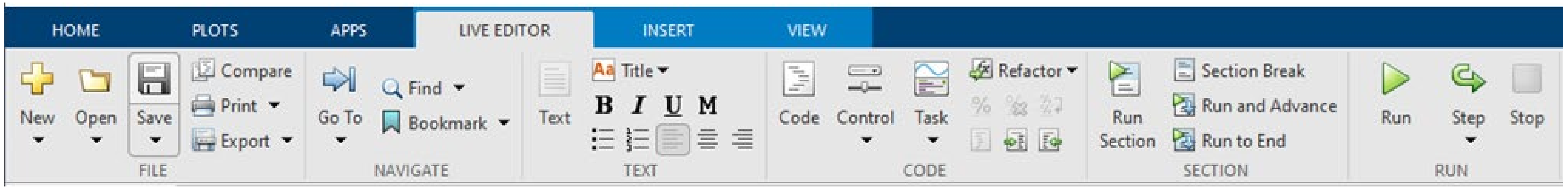
Using the default parameter values and the PFR energy balance from Part c, the plots should resemble the following

Note that we have different line styles and colors in the plots. The general plot formatting options are

- Colors: black 'k', green 'g', red 'r', blue 'b', magenta 'm', cyan blue 'c', yellow 'y'.
- Line types: '-' solid line, '--' dashed line, '-.' dash-dot line, ':' dotted line.
- Colors and line styles can be combined as 'r:' or 'k-.' for a red dotted line or a black dash-dot line.
- See this webpage for more details: https://www.mathworks.com/help/matlab/ref/plot.html

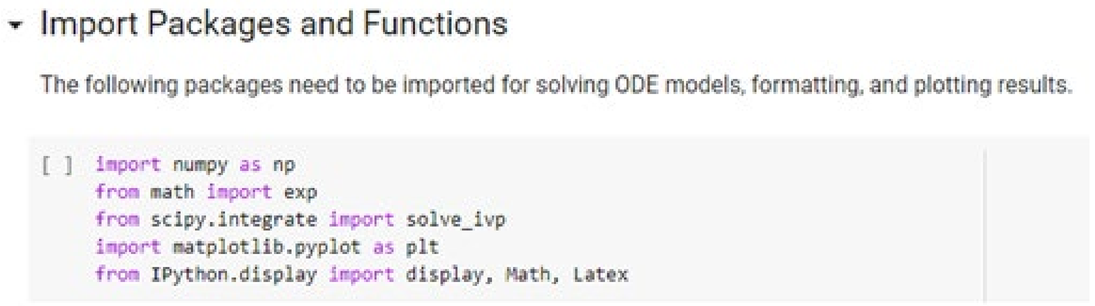
```
figure(1) % Plot T vs. V graph
    plot(V_eval, T_soln,'k-')
    title("Figure 1: Temperature Profile: Part " + Part)
    xlabel('V (dm^3)')
    ylabel('T (K)')
```





Jupyter Notebook (.ipynb)

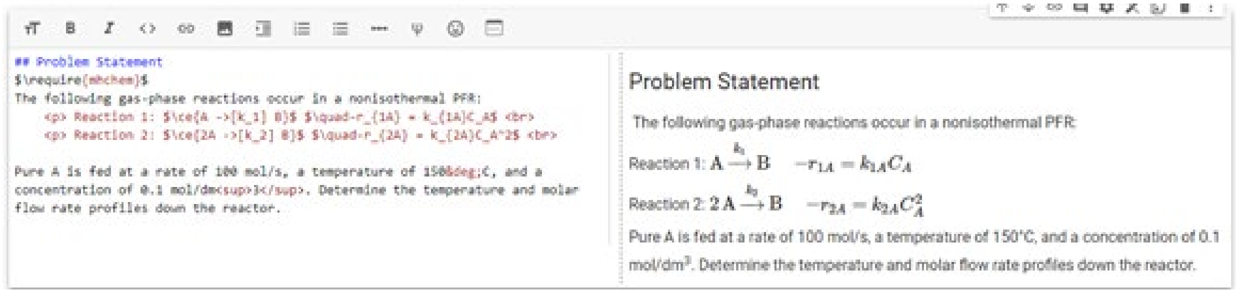
- Open-source literate programming option
- Works using online and standalone client
  - Google Colab opens directly from Google Drive
- Python code + Markdown default
  - >40 programming languages available
- Functions must run before called
- Notebooks render directly on GitHub
  - Non-interactive / Read-only



Import Packages and Functions

The following packages need to be imported for solving ODE models, formatting, and plotting results.

```
[ ] import numpy as np
    from math import exp
    from scipy.integrate import solve_ivp
    import matplotlib.pyplot as plt
    from IPython.display import display, Math, Latex
```



```
## Problem Statement
$\require{mhchem}$
The following gas-phase reactions occur in a nonisothermal PFR:
    <p> Reaction 1: $\ce{A ->[k_1] B}$ $\quad-r_{1A} = k_{1A}C_A$ <br>
    <p> Reaction 2: $\ce{2A ->[k_2] B}$ $\quad-r_{2A} = k_{2A}C_A^2$ <br>

Pure A is fed at a rate of 100 mol/s, a temperature of 150&deg;C, and a
concentration of 0.1 mol/dm<sup>3</sup>. Determine the temperature and molar
flow rate profiles down the reactor.
```

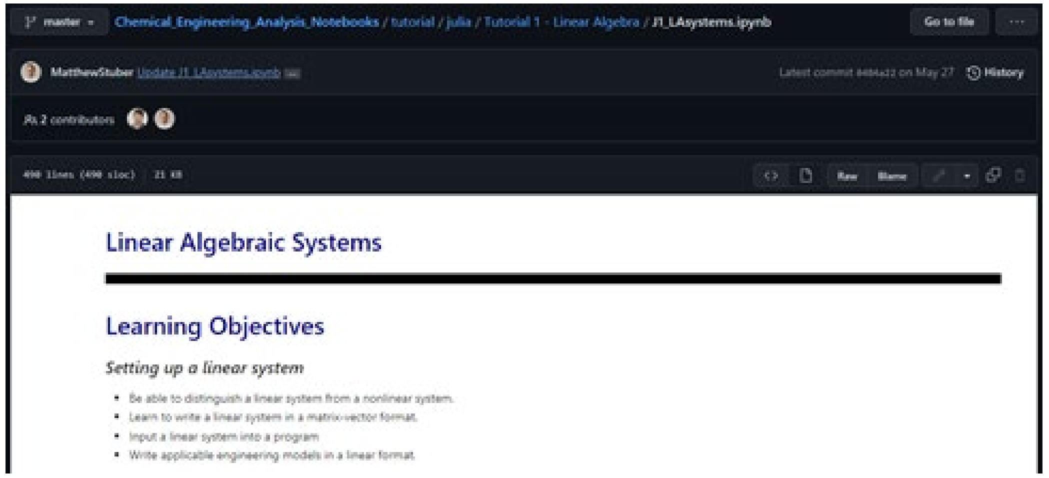
Problem Statement

The following gas-phase reactions occur in a nonisothermal PFR:

Reaction 1: $\mathrm{A} \xrightarrow{k_1} \mathrm{B} \quad -r_{1A} = k_{1A}C_A$

Reaction 2: $2\,\mathrm{A} \xrightarrow{k_2} \mathrm{B} \quad -r_{2A} = k_{2A}C_A^2$

Pure A is fed at a rate of 100 mol/s, a temperature of 150°C, and a concentration of 0.1 mol/dm^3. Determine the temperature and molar flow rate profiles down the reactor.



master
Chemical_Engineering_Analysis_Notebooks / tutorial / julia / Tutorial 1 - Linear Algebra / J1_LAsystems.ipynb
Go to file
MatthewStuber Update J1_LAsystems.ipynb
Latest commit [illegible] on May 27
History
2 contributors
490 lines (490 sloc) 21 KB
Raw
Blame

Linear Algebraic Systems

Learning Objectives

Setting up a linear system

- Be able to distinguish a linear system from a nonlinear system.
- Learn to write a linear system in a matrix-vector format.
- Input a linear system into a program.
- Write applicable engineering models in a linear format.

Ways to use these interactive coding templates in ChE classrooms across the curriculum

- Reinforce computational thinking
- Introduce/enhance familiarity with MATLAB and/or Python
- Student use cases
  - Lecture notes with in-class activities
  - Pre-class readings with embedded activities
  - Worked example case studies
  - In-class problems
  - Homework or project problems

Modeling a student use case: reading with embedded interactive activities

- Choose either MATLAB Live Script or Jupyter Notebook (Google Colab)
- Open M0_HowToCreate.mlx or J0_HowToCreate.ipynb file on How to create a MATLAB Live Script/How to create a Jupyter Notebook
- Read through the file and complete the interactive activities
- We will circulate to answer your questions

Your turn to build your own interactive coding template

- Choose either MATLAB Live Script or Jupyter Notebook (Google Colab)
- Find a partner who is interested in using the same language (MATLAB or Python)
- Work together to build a notebook for students to solve an equation of your choice. The notebook must include code blocks for numerical computation (solving) and the following text blocks (in any order)
    - learning objectives
    - problem statement
    - mathematical equation(s)
    - an image

Going further: case study examples to explore in groups

- We have prepared a set of interactive coding templates that correspond to mathematical topics and chemical engineering topics across the curriculum.
- Pick a file from our set of templates. Play with the interface and reflect on how to adapt for your purposes: lecture/in-class activity, homework, group projects, other.

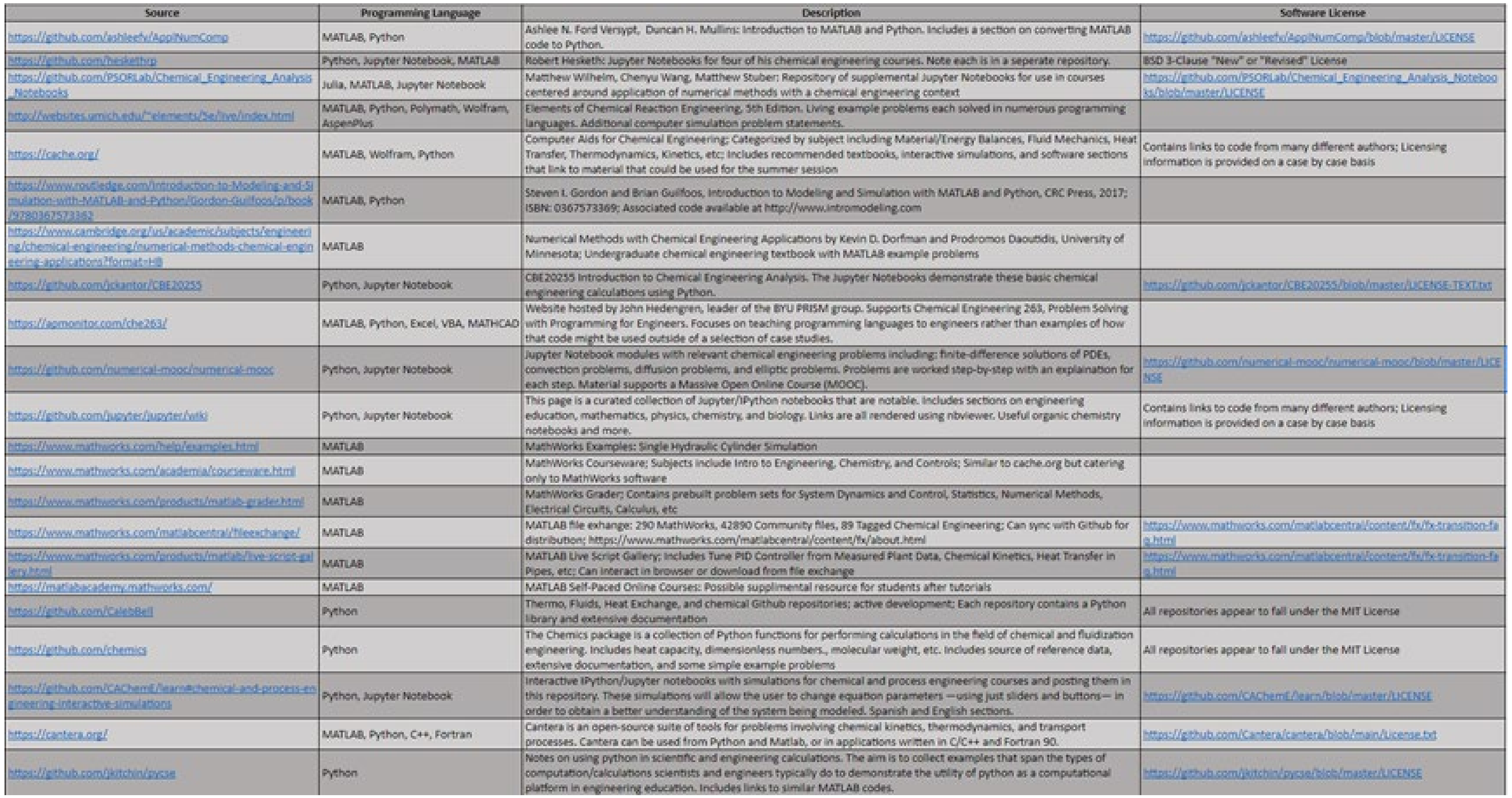
Workshop Table of Contents

More computational educational materials from us

- Video highlighting key features of MATLAB Live Scripts and Jupyter Notebooks (Austin Johns, Ford Versypt Lab): https://youtu.be/u5YkzFl6FbE
- Ford Versypt: https://github.com/ashleefv/ApplNumComp
- Hesketh: https://github.com/heskethrp
- Stuber: https://github.com/PSORLab/Chemical_Engineering_Analysis_Notebooks

Shareable Handout

More related computational educational materials from others



Source	Programming Language	Description	Software License
https://github.com/ashleefv/ApplNumComp	MATLAB, Python	Ashlee N. Ford Versypt, Duncan H. Mullins: Introduction to MATLAB and Python. Includes a section on converting MATLAB code to Python.	https://github.com/ashleefv/ApplNumComp/blob/master/LICENSE
https://github.com/hesketrhp	Python, Jupyter Notebook, MATLAB	Robert Hesketh: Jupyter Notebooks for four of his chemical engineering courses. Note each is in a seperate repository.	BSD 3-Clause "New" or "Revised" License
https://github.com/PSORLab/Chemical_Engineering_Analysis_Notebooks	Julia, MATLAB, Jupyter Notebook	Matthew Wilhelm, Chenyu Wang, Matthew Stuber: Repository of supplemental Jupyter Notebooks for use in courses centered around application of numerical methods with a chemical engineering context	https://github.com/PSORLab/Chemical_Engineering_Analysis_Notebooks/blob/master/LICENSE
http://websites.umich.edu/~elements/5e/live/index.html	MATLAB, Python, Polymath, Wolfram, AspenPlus	Elements of Chemical Reaction Engineering, 5th Edition. Living example problems each solved in numerous programming languages. Additional computer simulation problem statements.	
https://cache.org/	MATLAB, Wolfram, Python	Computer Aids for Chemical Engineering; Categorized by subject including Material/Energy Balances, Fluid Mechanics, Heat Transfer, Thermodynamics, Kinetics, etc; Includes recommended textbooks, interactive simulations, and software sections that link to material that could be used for the summer session	Contains links to code from many different authors; Licensing information is provided on a case by case basis
https://www.routledge.com/Introduction-to-Modeling-and-Simulation-with-MATLAB-and-Python/Gordon-Guilfoos/p/book/9780367573362	MATLAB, Python	Steven I. Gordon and Brian Guilfoos, Introduction to Modeling and Simulation with MATLAB and Python, CRC Press, 2017; ISBN: 0367573369; Associated code available at http://www.intromodeling.com	
https://www.cambridge.org/us/academic/subjects/engineering/chemical-engineering/numerical-methods-chemical-engineering-applications?format=HB	MATLAB	Numerical Methods with Chemical Engineering Applications by Kevin D. Dorfman and Prodromos Daoutidis, University of Minnesota; Undergraduate chemical engineering textbook with MATLAB example problems	
https://github.com/jckantor/CBE20255	Python, Jupyter Notebook	CBE20255 Introduction to Chemical Engineering Analysis. The Jupyter Notebooks demonstrate these basic chemical engineering calculations using Python.	https://github.com/jckantor/CBE20255/blob/master/LICENSE-TEXT.txt
https://apmonitor.com/che263/	MATLAB, Python, Excel, VBA, MATHCAD	Website hosted by John Hedengren, leader of the BYU PRISM group. Supports Chemical Engineering 263, Problem Solving with Programming for Engineers. Focuses on teaching programming languages to engineers rather than examples of how that code might be used outside of a selection of case studies.	
https://github.com/numerical-mooc/numerical-mooc	Python, Jupyter Notebook	Jupyter Notebook modules with relevant chemical engineering problems including: finite-difference solutions of PDEs, convection problems, diffusion problems, and elliptic problems. Problems are worked step-by-step with an explaination for each step. Material supports a Massive Open Online Course (MOOC).	https://github.com/numerical-mooc/numerical-mooc/blob/master/LICENSE
https://github.com/jupyter/jupyter/wiki	Python, Jupyter Notebook	This page is a curated collection of Jupyter/IPython notebooks that are notable. Includes sections on engineering education, mathematics, physics, chemistry, and biology. Links are all rendered using nbviewer. Useful organic chemistry notebooks and more.	Contains links to code from many different authors; Licensing information is provided on a case by case basis
https://www.mathworks.com/help/examples.html	MATLAB	MathWorks Examples: Single Hydraulic Cylinder Simulation	
https://www.mathworks.com/academia/courseware.html	MATLAB	MathWorks Courseware; Subjects include Intro to Engineering, Chemistry, and Controls; Similar to cache.org but catering only to MathWorks software	
https://www.mathworks.com/products/matlab-grader.html	MATLAB	MathWorks Grader; Contains prebuilt problem sets for System Dynamics and Control, Statistics, Numerical Methods, Electrical Circuits, Calculus, etc	
https://www.mathworks.com/matlabcentral/fileexchange/	MATLAB	MATLAB file exhange: 290 MathWorks, 42890 Community files, 89 Tagged Chemical Engineering; Can sync with Github for distribution; https://www.mathworks.com/matlabcentral/content/fx/about.html	https://www.mathworks.com/matlabcentral/content/fx/fx-transition-faq.html
https://www.mathworks.com/products/matlab/live-script-gallery.html	MATLAB	MATLAB Live Script Gallery; Includes Tune PID Controller from Measured Plant Data, Chemical Kinetics, Heat Transfer in Pipes, etc; Can interact in browser or download from file exchange	https://www.mathworks.com/matlabcentral/content/fx/fx-transition-faq.html
https://matlabacademy.mathworks.com/	MATLAB	MATLAB Self-Paced Online Courses: Possible supplimental resource for students after tutorials	
https://github.com/CalebBell	Python	Thermo, Fluids, Heat Exchange, and chemical Github repositories; active development; Each repository contains a Python library and extensive documentation	All repositories appear to fall under the MIT License
https://github.com/chemics	Python	The Chemics package is a collection of Python functions for performing calculations in the field of chemical and fluidization engineering. Includes heat capacity, dimensionless numbers., molecular weight, etc. Includes source of reference data, extensive documentation, and some simple example problems	All repositories appear to fall under the MIT License
https://github.com/CAChemE/learn#chemical-and-process-engineering-interactive-simulations	Python, Jupyter Notebook	Interactive IPython/Jupyter notebooks with simulations for chemical and process engineering courses and posting them in this repository. These simulations will allow the user to change equation parameters —using just sliders and buttons— in order to obtain a better understanding of the system being modeled. Spanish and English sections.	https://github.com/CAChemE/learn/blob/master/LICENSE
https://cantera.org/	MATLAB, Python, C++, Fortran	Cantera is an open-source suite of tools for problems involving chemical kinetics, thermodynamics, and transport processes. Cantera can be used from Python and Matlab, or in applications written in C/C++ and Fortran 90.	https://github.com/Cantera/cantera/blob/main/License.txt
https://github.com/jkitchin/pycse	Python	Notes on using python in scientific and engineering calculations. The aim is to collect examples that span the types of computation/calculations scientists and engineers typically do to demonstrate the utility of python as a computational platform in engineering education. Includes links to similar MATLAB codes.	https://github.com/jkitchin/pycse/blob/master/LICENSE

Additional Resources