The Python Pivot – Teaching Chemical Engineering Computing in the Wake of COVID-19

Dr. J. Patrick Abulencia
Department of Chemical Engineering

Abstract

The COVID-19 pandemic of 2020 caused a major disruption in the education system with the abrupt transition to remote learning. In addition to the quick adaptation of this unfamiliar mode by both students and professors, resources typically used for the course was no longer an option, and course outcomes and deliverables had to be re-imagined.

In this poster presentation, the challenges encountered during this transition and how they were addressed in a junior-level computer simulation course will be discussed. More specifically, the instructor had to rapidly pivot towards teaching Python, because the software used was not available to students remotely. This result in the creation of projects wherein students had to use Python programming to solve three styles of chemical engineering problems that they have been already been exposed to in the curriculum: 1) a multi-step solution, 2) an iterative solution, and 3) an integration. Class time evolved into working periods where groups could interact with their colleagues and instructor within Zoom breakout rooms, and assessments took the form of: 1) a short presentation where groups explained their problem and Python code, 2) a collaborative wiki where each group would describe the Python skills or structures that they needed to solve their problem, and 3) a working Python program that would compute their solution.

The Computer Simulation and Design Course

CHML 316 is Computer Simulation and Design; a junior-level course that primarily teaches the process simulator ASPEN-PLUS (approximately 70% of the course), as well as the use of numerical methods using VBA programming (approximately 30% of the course). There is also a short unit during the beginning of the course on using Microsoft Visio, as well as document generation using LaTeX. As a point of reference, VBA is taught during the freshman-level ENGS-116 Introduction to Engineering Computation course, which all engineering students are required to complete.

Issues Post-COVID

Classes at Manhattan College were suspended on March 10th, 2020 because of the escalation of COVID-19 in New York City. The next slide outlines the major issues encountered during the transition to remote learning:

Issues Post-COVID

Transition to Remote Learning

Remote learning was a new environment for both students and professor. The initial hurdles were logistical implementations; how to use the conferencing software, how to present slides/whiteboard, and in-class logistics. More specifically, class time before the shutdown included group work, where students would collectively work on an in-class problem. These logistics were resolved shortly after the transition after learning the Zoom software, and understanding its capabilities such as breakout rooms.

Access to Software

ASPEN-PLUS, which was the software that the class was working on before on-site instruction was suspended, is licensed on a per-seat basis and could only be used on a prescribed set of computers located on-campus. Because of the transition to remote, this was the primary problem for this course because students had no access to this software from their homes. Despite efforts by our institution's IT staff in finding a way to allow remote access to ASPEN-PLUS, a reasonable solution was not reached, warranting the need to change direction and teach another chemical engineering computing skill.

The Python Pivot

Why Python?

The primary issue for this course post-COVID was the lack of access to the process simulator ASPEN-PLUS. Thus, objectives of the course had to be re-directed towards computer programming because of student inability to perform their simulations. Students have already been exposed to VBA in their freshman class, so the decision was made to teach them Python because:

1) Python is free and accessible
2) There is a multitude of online resources for students to learn
3) Employers have increasingly been looking to hire candidates that possess the ability to program in Python
4) It is more straightforward compared to other computing languages, which allows the possibility of several project during the remaining five weeks of the course.
5) Dr. Ben Davis shared his class notes (hooray AIChE EdDiv!)

The Python Pivot

Implementation

Week 1 - Teaching Python basics

Weeks 2-5 - Achievement of three milestones of increasing difficulty

- Milestone 1 - Solve a series of equations using Python (e.g. calculating vapor pressure and using it to calculate bubble point and dew point pressures)
- Milestone 2 - Solve an iterative solution (e.g. flash calculation)
- Milestone 3 - Integrate under a curve (e.g. reactor sizing)

Deliverables

- An oral presentation/demonstration of your code
- Contribution to a Python Wiki
- The Python Program Itself

Student Reaction

Course and Teacher Evaluations responding to the question, "What aspects of this course were most beneficial to you?". There were ten responses, with seven directly addressing their opinion on learning Python (listed below):

S1: I enjoyed learning Python coding and feeling more confident in my coding abilities
S2: Learning the coding language of these programs is very beneficial to us when we go into industry
S3: I think learning Python and the review of VBA were helpful. I think Aspen was cut short but I think getting more comfortable with that program would be helpful
S4: Learning Aspen and Python were most beneficial
S5: I thought all aspects of the course were beneficial, especially the coding in Python
S6: Programming a new language
S7: Learning Aspen and Python

Additionally, students generally felt that Python should be taught instead of VBA (comments not shown).

Acknowledgement and References

1) Dr. Ben Davis - Cooper Union, New York, NY
2) Kitchin Research Group
3) EDUCBA: https://www.educba.com/category/software-development/software-development-tutorials/python-tutorial/ (accessed 24 March 2020)
4) Python Documents: https://docs.python.org/ (accessed 24 March 2020)
5) Software Carpentry: http://swcarpentry.github.io/python-novice-gapminder/index.htm (accessed 25 March 2020)
6)nbviewer:http://nbviewer.jupyter.org/github/jckantor/CBE20255/blob/master/notebooks/00.01-Getting-Started-with-Jupyter-Notebooks-and-Python.ipynb (accessed 25 March 2020)