
Veterans Day
HONORING ALL WHO SERVED

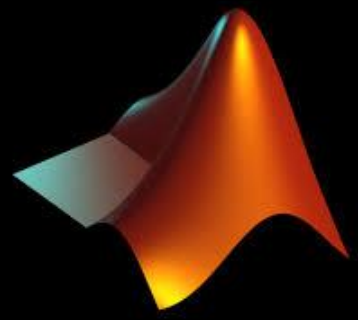
Steal This Course!
Scientific Computing for ChEs in Python

AIChE Annual Meeting 2019
Orlando, FL

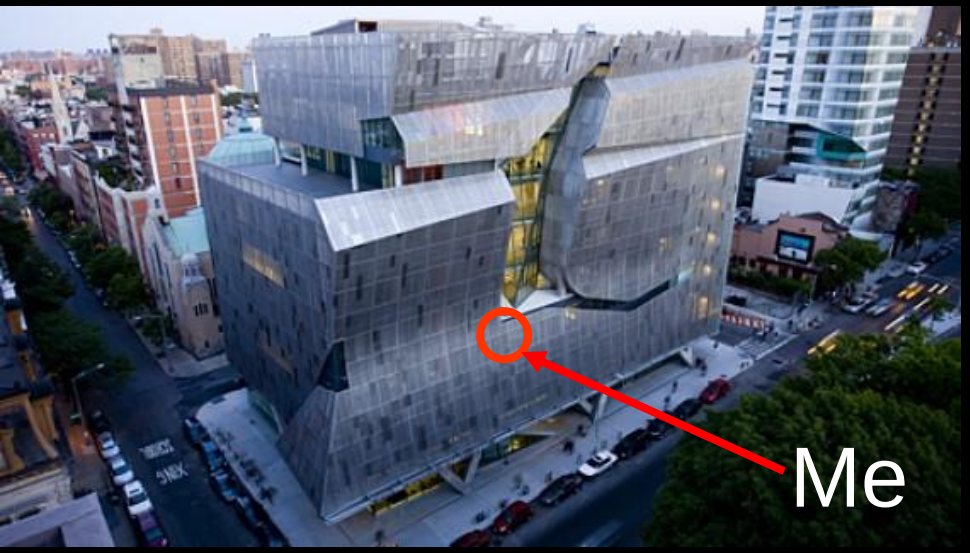
Professor Benjamin Davis

My Three Most Important Points

1. Learn to code `print("Hello, world!")`
2. Scientific computing teaches coding and ChE $f(x) = 0$
3. utility(Python) > utility(Matlab)



The Cooper Union for the Advancement of Science and Art



- We’re tiny! ~1000 students! Only 2 buildings!
- Full-tuition scholarships again in 2028?
- Urban, non-residential, primarily undergraduate
- 3 degree-granting schools: Art / Arch. / Eng.
- ChE undergrad class typically 20-30 students

Quick Finger Poll:

How many Credits / Student Course Hours of Coding are required at your college/university?

My Students aren't Comfortable w/ Coding

- Handful each year have coded in Matlab / Python / "high level" programming lang. before

Juniors = Can use ALL OF THE MATH!

- ChE352: Junior level REQUIRED(!) course, offered in Spring
- 6th semester ChEs, some BSEs
- Have Calc I/II/III, Lin. Alg., and Diff. Eqn.
- Students had 2 credits in C programming language 1st semester freshman year – is that enough?
- Switched from Matlab to Python in 2019

You Learn to Code by Writing your own Code

First half:

- Math Review and Error Analysis
- Root finding: **f(x) = 0, x in R**
- Interpolation/Extrapolation/Polynomial Approximation
- Numerical Differentiation & Quadrature
- Initial Value Problems (IVPs) and IVP Systems: **u' = f(t,u)**

Starting Simple on Homework 1

5. (20 points) Download / open Python 3.7. The IDLE IDE is available on the machines in the 8th floor computer lab, but you can also download Spyder from https://www.anaconda.com/download/. Click on the “New” icon to create a new *.py file (a “program”). Save it as “FirstnameLastname.py”, but use your first name instead of “Firstname” and your last name instead of “Lastname”. Type this in, one line at a time:

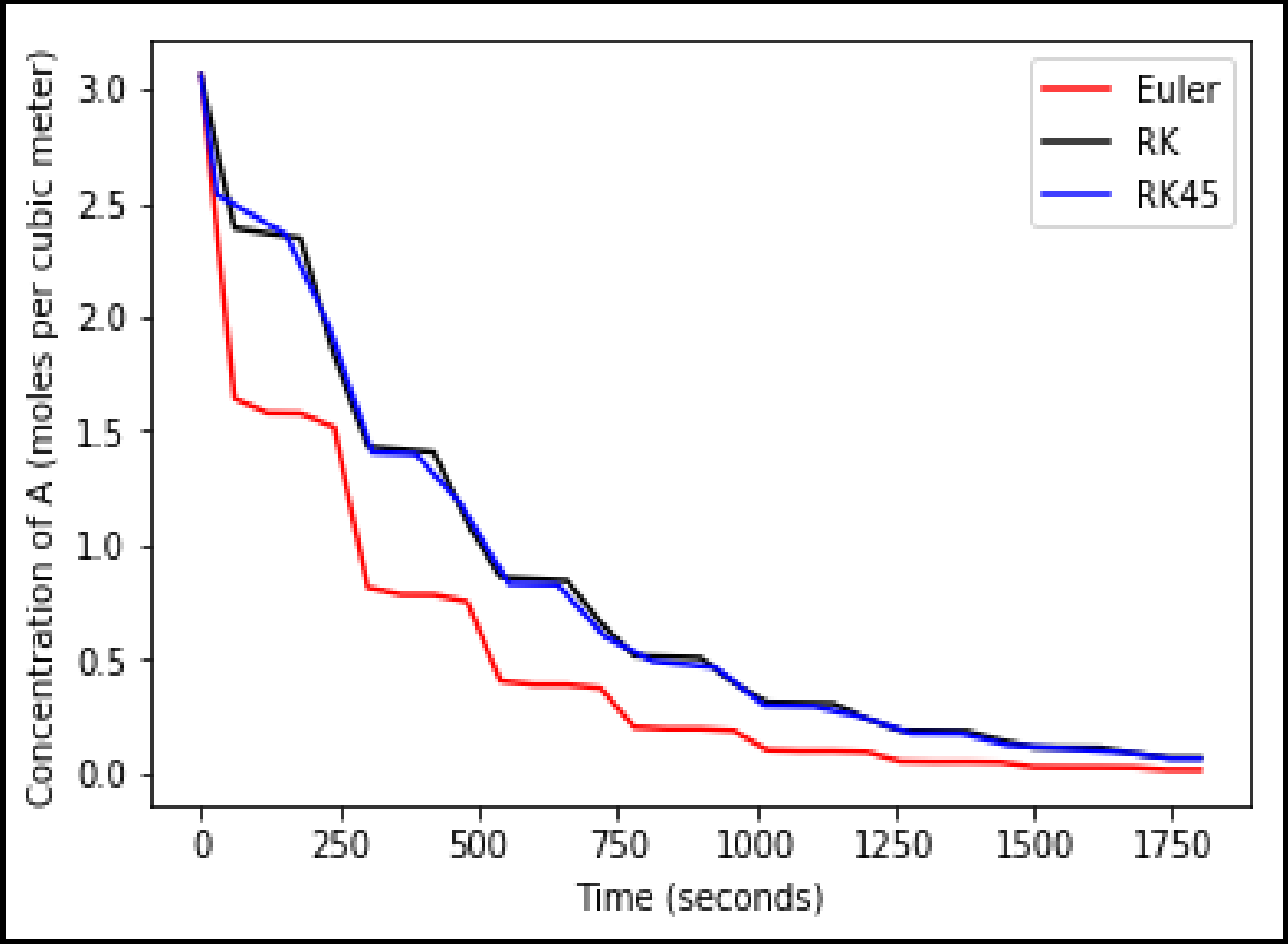
```
# You make comments in Python using the pound sign; it turns a different color
# I promise I typed this out with my own fingers and didn’t copy and paste it
# NumPy is the fundamental package for scientific computing with Python.
# -SciPy.org (Google "about numpy" to find the documentation)
# You need to import a module to use it; we're importing numpy and calling it
# "np", so when we use numpy commands we use "np.command(blah blah)"
import numpy as np

w = 'Hello, world!'  # This is how you make a “string” in Python
# The object w now represents a string of text, but you can’t do math
# with strings. . .
```

Build up to ChE Prob. from Other Courses

You're working as a process engineer for a pharmaceutical company and you're in charge of a batch reactor. The reactor, which is supposed to operate isothermally at 320 K, makes a certain isomer of a drug from a racemic mixture.

One day you notice there's a slow periodic disturbance in the temperature in the reactor. You check your controller data and the reactor temperature (T) seems to be following a regular pattern. You plot the data and you were able to fit the curve with a periodic function; the equation $T = \lambda_1 \cos(\lambda_2 t) + 320$ seemed to fit it okay.

A. Write your own algorithm in Python which implements Euler's method to solve for $C_A(t)$ numerically.
B. Modify your Euler code from part A. so that it solves the problem for $C_A(t)$ using the Runge-Kutta order 4 method also.
C. Note in your submission the primary differences between RK4 and Euler.
D. Use RK45 in Python to solve this problem.


Euler
RK
RK45
Concentration of A (moles per cubic meter)
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
250
500
750
1000
1250
1500
1750
Time (seconds)

Add Math Complexity in 2nd Half

Second half:

- Linear Systems: **Ax = b**
- Iterative Methods, Eigenvalues/Eigenvectors, Optimization: $x^{k+1} = x^k + \alpha_k d^k$
- Systems of Nonlinear Equations (SNLE):

  f(x) = 0, x in R^n
- Boundary-Value Problems (BVPs) for ODEs
- Num. Soln.s to Partial Differential Equations (PDEs) – mostly 2nd order parabolic / diffusion equation with finite differences

Final Project is Writing and Coding

ChE352: Process Simulation and Mathematical Techniques for ChEs
Prof. Davis

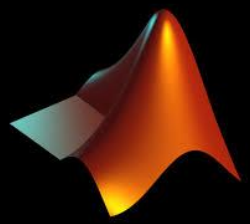
Spring 2019 Final Project: 100 points total
Due Thursday, May 9th (submit to Moodle by 11:59 PM)

The goal of this project is to have you demonstrate your skill at (in no particular order): estimating error in numerical approximations, writing algorithms, solving systems of equations, fitting data by polynomial approximation or regression, finding roots, solving PDEs, and communicating the results of your work effectively. Each student will be assigned a different set of four species to use for their project in order to differentiate between submissions. Your particular species are up on Moodle.

You have just started work as a research and development engineer for AspenTech (100 Fake Road, Newtown, PA 18940). Your boss (Mr. Sean F. Askebom) hired you to help write the code for the newest version of their process simulation and design software. Your first assignment is to write a "flash tank" unit operation (or just a

- Create simple code, use it in a more complicated algorithm for SNLE, solve a PDE
- Write a memo and 3 algorithm flow charts explaining what you did and how it works

Python is Open Source and Widely Used



Matlab

- 4M users
- $40-99
- Not widely used outside academia

Python

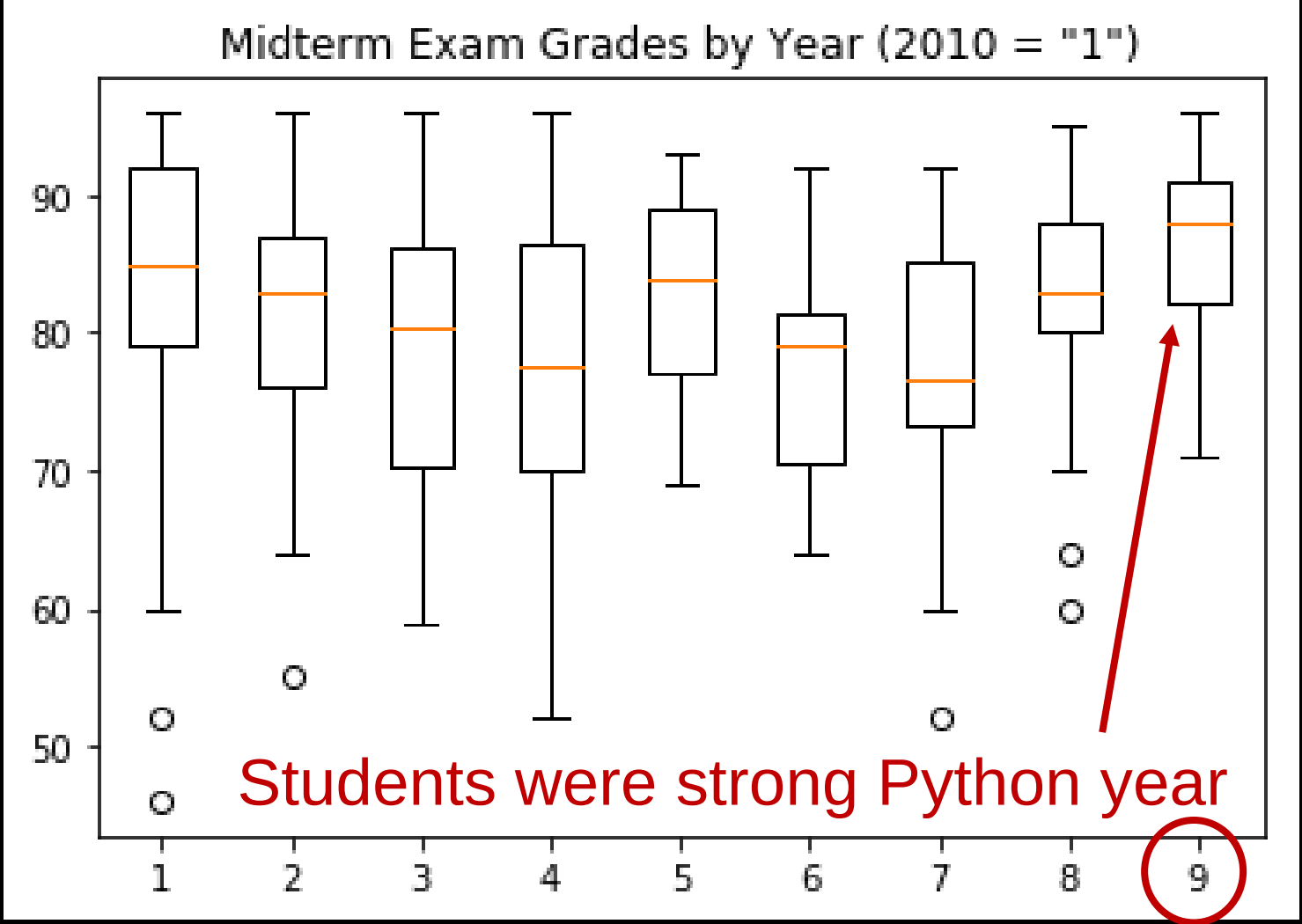
- 8M users
- Free
- Currently a high demand skill by employers

But all your Precious Code?

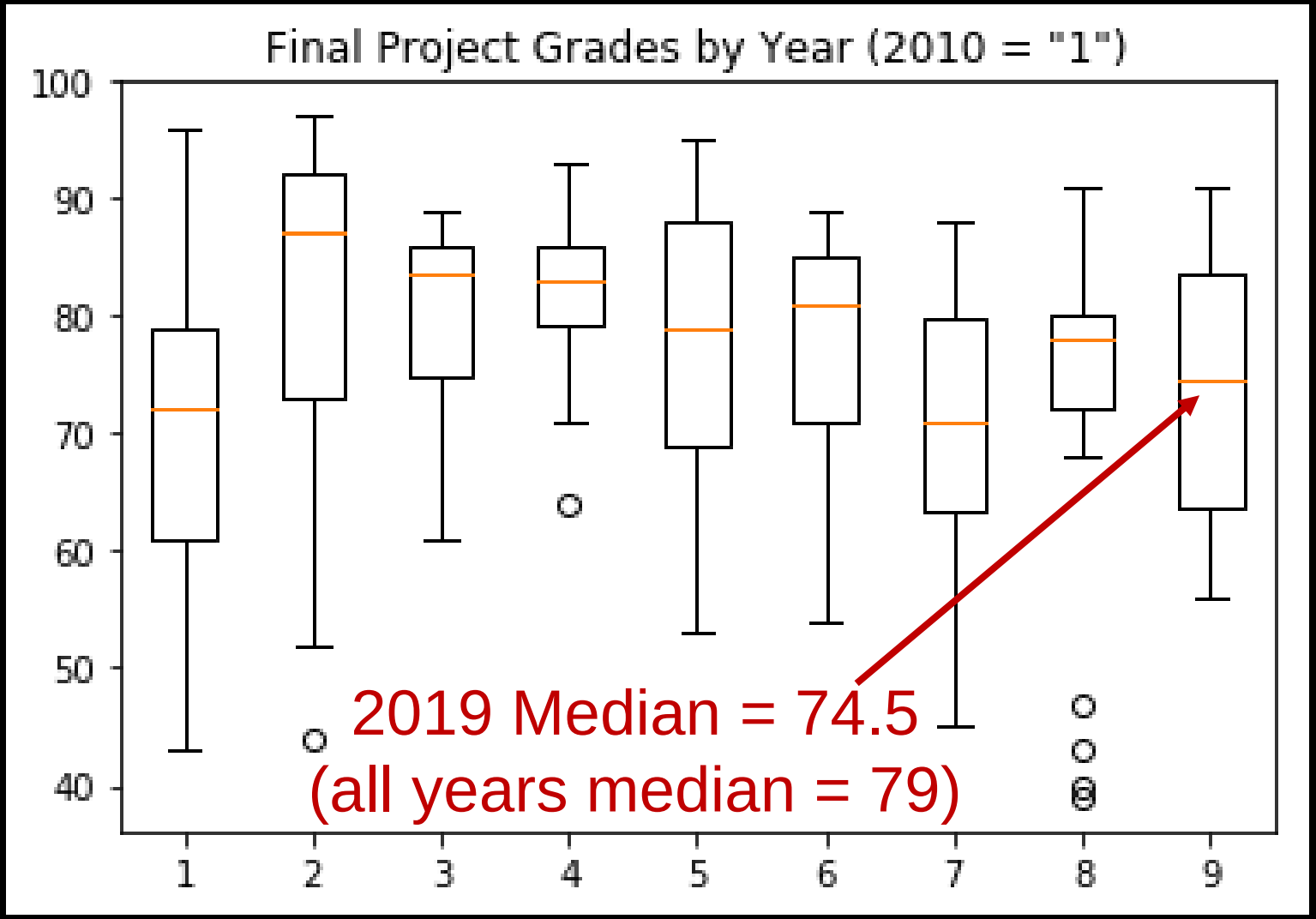
cts.vcf | output.txt | NewtRaph.py

```
# -*- coding: utf-8 -*-
"""
Created on Mon Feb 25 17:21:55 2019

@author: bdavis

This algorithm will use Newton-Raphson to find the root of g(x)
To change g(x), change the gp and gprime definitions on lines 13, 14, 34,
and 35.  Make sure the initial guess for p is "pretty close" (line 13)."""

Nmax = 20           # Maximum number of iterations; Newton-Raphson needs fewer
TOL = 1e-9          # Tolerance for stop condition - function value
p = 0.7             # Initial guess for root (p0)

c = [-0.045504, 0.1217664, 0.426226, -0.3139]  # Coefficient values for g(x)
d = [3*c[0], 2*c[1], c[2]]                     # Coefficients for g'(x)

gp = c[0]*p**3 + c[1]*p**2 + c[2]*p + c[3]     # First value of g(p)
gprime = d[0]*p**2 + d[1]*p + d[2]             # First value of g'(p)

# Check to see if the derivative is zero to avoid error / overflow
if abs(gprime) < 1e-16:
    p = 'Error: Derivative is zero.'  # Error message
    N = 0                             # Set number of iterations for output

else:

    for N in range(Nmax):         # Continue until Nmax is reached

        if abs(gp) <= TOL:        # True if function value < TOL

            break  # Break the for loop if the function value is small enough

        else:

            p = p - gp/gprime  # Calculate pN = pN-1 - g(pN-1)/g'(pN-1)
            gp = c[0]*p**3 + c[1]*p**2 + c[2]*p + c[3]  # Find new g(p) and g'(p)
            gprime = d[0]*p**2 + d[1]*p + d[2]

if N == Nmax:
    print('Error: maximum iterations reached.')

print('The answer is', p, '.')              # Print the answer to the console
print('\nIt took', N, 'iterations.\n')      # Print the number of iterations
```

Python Switch didn't Affect Grades



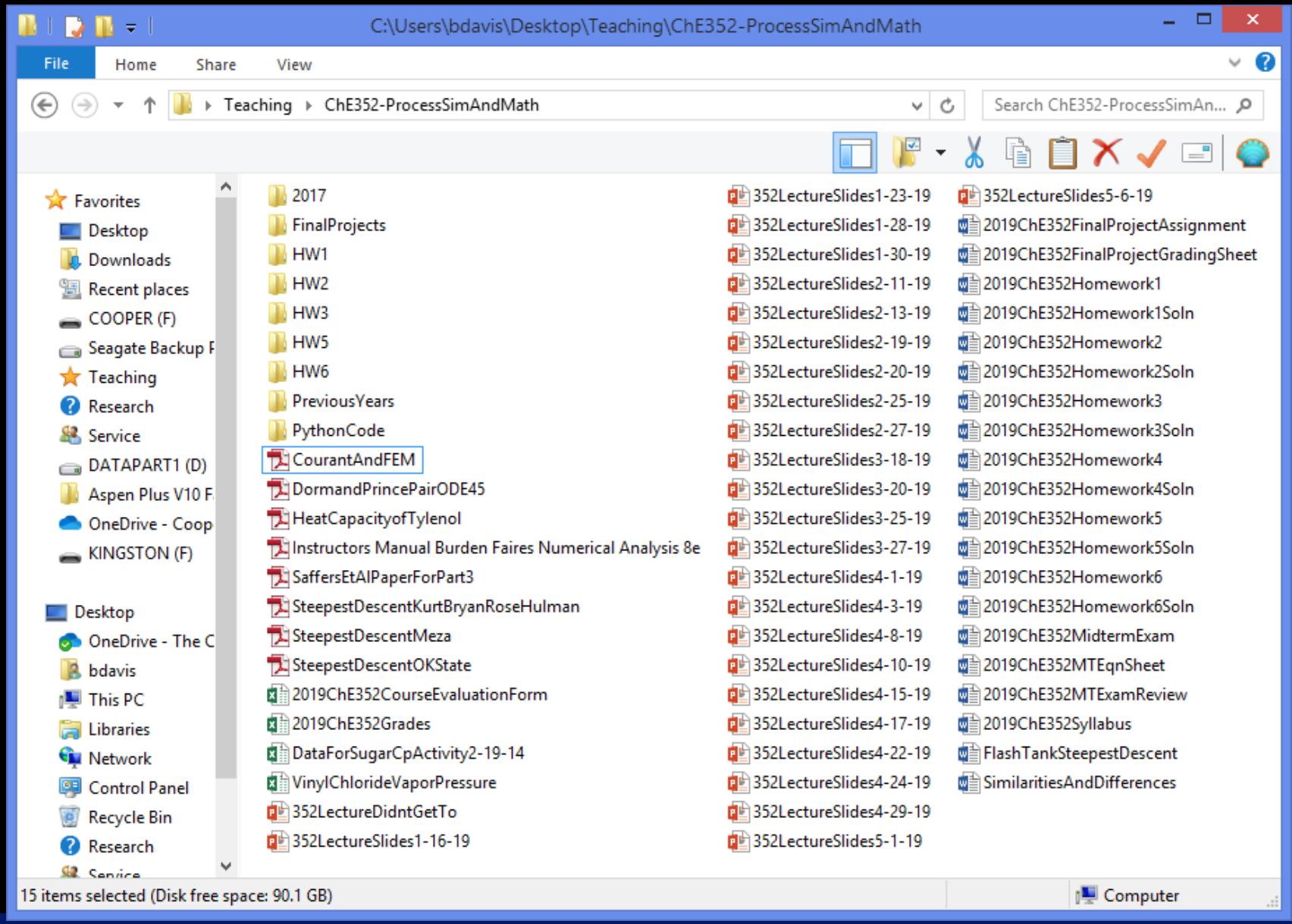
Final Project Grades Slightly Lower



Any theories?

The Cooper Union for the Advancement of Science and Art

You Can Have All Course Materials!



Why Should You Steal this Course?

- All new scientists and engineers (including ChEs) need to know how to write their own code
- Our "CS101" education is lacking and not specific to ChEs and what they need to do
- Python is open source – ensures access
- Important math plus fun problems / applications (started with a Numerical Analysis / Numerical Methods text and added ChE problems to it)