Exercises for a Numerical Methods Course

Brian Heinold

Department of Mathematics and Computer Science
Mount St. Mary's University

November 18, 2017

About the class

- Mix of Math and CS students, mostly math (counts as an elective for both)

About the class

- Mix of Math and CS students, mostly math (counts as an elective for both)
- Class size: 10-20

About the class

- Mix of Math and CS students, mostly math (counts as an elective for both)
- Class size: 10-20
- Prereqs: Calc I and Intro Programming

About the class

- Mix of Math and CS students, mostly math (counts as an elective for both)
- Class size: 10-20
- Prereqs: Calc I and Intro Programming
- Topics: floating point, interpolation, numerical equation solving, numerical integration and differentiation, numerical methods for differential equations, simulations

Back in 2009

- First taught the class in 2009

Back in 2009

- First taught the class in 2009
- Used exercises from the textbook (mostly performing the algorithms, deriving things, etc.)

Back in 2009

- First taught the class in 2009
- Used exercises from the textbook (mostly performing the algorithms, deriving things, etc.)
- Wasn't sure that they really understood the methods

Back in 2009

- First taught the class in 2009
- Used exercises from the textbook (mostly performing the algorithms, deriving things, etc.)
- Wasn't sure that they really understood the methods
- Wasn't sure they understood why we were doing things either

Back in 2009

- First taught the class in 2009
- Used exercises from the textbook (mostly performing the algorithms, deriving things, etc.)
- Wasn't sure that they really understood the methods
- Wasn't sure they understood why we were doing things either
- So I started adding "Understanding Questions"

A few types of Understanding Questions

- *Why?* questions

A few types of Understanding Questions

- *Why?* questions
- Questions asking for students to explain things we talked about *in their own words*
- Questions testing understanding of the concepts in novel ways

A few types of Understanding Questions

- *Why?* questions
- Questions asking for students to explain things we talked about *in their own words*
- Questions testing understanding of the concepts in novel ways
- Here follows a large number of exercises. These slides are available now at `www.brianheinold.net` or at the section website after the meeting.

Some “why?” I ask on homeworks

- Why do people care about solving equations?

Some “why?” I ask on homeworks

- Why do people care about solving equations?
- Why are numerical methods for solving equations needed?

Some “why?” I ask on homeworks

- Why do people care about solving equations?
- Why are numerical methods for solving equations needed?
- It seems annoying that the floating point system we use can’t represent things exactly. Why don’t we use an exact system?

Some “why?” I ask on homeworks

- Why do people care about solving equations?
- Why are numerical methods for solving equations needed?
- It seems annoying that the floating point system we use can’t represent things exactly. Why don’t we use an exact system?
- Taking derivatives is generally pretty easy. The rules you learn in Calculus I are enough to take the derivative of any elementary function. So why then are numerical differentiation techniques needed at all?

“In your own words” questions

- What is Runge’s phenomenon?

“In your own words” questions

- What is Runge’s phenomenon?
- What do Chebyshev polynomials have to do with interpolation?

“In your own words” questions

- What is Runge’s phenomenon?
- What do Chebyshev polynomials have to do with interpolation?
- In class and in the notes, I mentioned that Brent’s method combines the safety of the bisection method with the speed of IQI. Explain briefly how it accomplishes that.

“In your own words” questions

- What is Runge’s phenomenon?
- What do Chebyshev polynomials have to do with interpolation?
- In class and in the notes, I mentioned that Brent’s method combines the safety of the bisection method with the speed of IQI. Explain briefly how it accomplishes that.
- In your own words, explain how adaptive quadrature works.

“In your own words” questions

- What is Runge’s phenomenon?
- What do Chebyshev polynomials have to do with interpolation?
- In class and in the notes, I mentioned that Brent’s method combines the safety of the bisection method with the speed of IQI. Explain briefly how it accomplishes that.
- In your own words, explain how adaptive quadrature works.
- Explain geometrically how Euler’s method works and describe how the formula for Euler’s method relates to your geometrical explanation.

“In your own words” questions

- What is Runge’s phenomenon?
- What do Chebyshev polynomials have to do with interpolation?
- In class and in the notes, I mentioned that Brent’s method combines the safety of the bisection method with the speed of IQI. Explain briefly how it accomplishes that.
- In your own words, explain how adaptive quadrature works.
- Explain geometrically how Euler’s method works and describe how the formula for Euler’s method relates to your geometrical explanation.
- For numerical differentiation, there is a reason why smaller h values are desirable, and there is a reason why smaller h values can be problematic. What are the two reasons?

Practical questions testing understanding

- The calculation $123456789.0 + .000000001$ comes out to 123456789.0 and not 123456789.000000001 when using 64-bit floating point arithmetic. Why?

Practical questions testing understanding

- The calculation $123456789.0 + .000000001$ comes out to 123456789.0 and not 123456789.000000001 when using 64-bit floating point arithmetic. Why?

- The Mount's radio station WMTB 89.9 FM is running a special promotion where if you buy two items totaling exactly $89.90 using their app, then you get both for free. They programmed the app using the Python code below:

```
    if price1 + price2 == 89.9:
        print("Everything's free!!!")
```

  But someone could buy something for $57.91 and something else for $31.99 and the program will not correctly catch that as totaling $89.90. Why not? How could the problem be fixed?

More Understanding Questions

- Give two examples of numbers in the interval $(0, 1)$ that can be represented *exactly* using floating point. Explain briefly.

More Understanding Questions

- Give two examples of numbers in the interval $(0, 1)$ that can be represented *exactly* using floating point. Explain briefly.
- The IEEE 754 double precision standard allows 53 bits for the mantissa, which guarantees roughly 15 decimal digits of accuracy. Where does the 15 come from?

More Understanding Questions

- The bisection method, FPI, and Newton's method are used to find roots—that is, solutions to equations of the form $f(x) = 0$. Suppose we want to find the solution to an equation of the form $f(x) = 5$. How would we use these methods to do that?

More Understanding Questions

- Suppose you are solving $x^5 - x - 1 = 0$ using fixed-point iteration. You rewrite it as a fixed point problem, iterate, and the iterates settle down around $x = 2.516$. How can you tell that 2.516 cannot be possibly anywhere near correct?

More Understanding Questions

- If you use the midpoint rule to estimate $\int_1^4 \sin(x^3)\,dx$ with $n = 40$ rectangles, to four decimal places you will get .1973, which is off by .007 from the exact answer .2043.

  1. What is the error bound value given by the formula on page 59 of the notes)? [Hint: you will probably want to use Wolfram Alpha to find $|\max f''|$.]
  2. Why is this not equal to .007?

More Understanding Questions

- True or False (give a reason) — The RK4 method can be used to estimate $\int_0^1 \sqrt[3]{1+x^2}\,dx$.

More Understanding Questions

- The degree of precision of Boole's rule is 5. If we use it with $n = 100$ rectangles to integrate $\int_0^3 (x^4 - x + 1)\,dx$, which of the following is true about the error between the exact integral and the Boole's rule approximation?
  (a) The error is 0.
  (b) The error is on the order of $1/n^6$ but not 0.
  (c) The error term has not been theoretically determined (it is an open problem).

- We talked about two-point and three-point Gaussian quadrature. What rule (that we had also talked about) is one-point Gaussian quadrature equivalent to? Explain.

More Understanding Questions

- Suppose we want to solve $x^3 - 10 = 0$ via fixed point iteration. One way to rewrite this as a fixed point problem is as $\frac{9}{x^2+x+1} + 1 = x$. If we iterate this starting at 2 and look at the error ratios e_{i+1}/e_i, we get a sequence of values .8501, .7380, .8252, .7562, .8102, .7676, ..., eventually settling down near 0.7861370554624069. That number is a decimal approximation to a particular irrational number that is the limit of the sequence of error ratios just given. What is the exact value of that irrational number?

More Understanding Questions

- 1. Find the first 5 terms of the Taylor series of $f(x) = \sqrt{x}$ centered at $x = 1.5$.
- 2. Find the 5 Cheybshev points on $[0, 10]$, compute their square roots, and use NDD on these to get an interpolating polynomial for $\sqrt{x}$.
- 3. Use the Taylor polynomial as well as the Chebyshev approximation to estimate $\sqrt{.5}$, $\sqrt{1}$, $\sqrt{1.25}$, $\sqrt{2}$, $\sqrt{5}$, and $\sqrt{10}$. Find the errors in each case, and compare their accuracies.
- 4. Neither polynomial is appropriate to estimate $\sqrt{100}$, but if we write $\sqrt{100}$ as $\sqrt{2^4 \cdot (100/2^4)}$, we can use properties of square roots and the Cheybshev approximation to estimate $\sqrt{100}$. Explain how this works. How accurate is it for $\sqrt{100}$?

More Understanding Questions

- Recall that in class we talked about the "curse of dimensionality". Suppose we need to approximate $\int\int\cdots\int f(x_1, x_2, \ldots, x_{27})\, dV$, where there are 27 integral signs. Assume the region of integration is the unit 27-dimensional hypercube, and we want to divide it up using a mesh of size .01 in all dimensions. If the numerical method we use requires one function evaluation for each piece of the mesh, how many function evaluations will be needed?

More Understanding Questions

- Consider the polynomial $(x-1)(x-2)(x-3)(x-4)(x-5)(x-6)(x-7)(x-8)(x-9)(x-10)$.
  1. What are its roots?
  2. Use Wolfram Alpha or another computer algebra system to expand it. Then increase the coefficient of the x^9 term by .0001. Use Wolfram or a computer algebra system to find the roots of the new polynomial.

Why spreadsheets?

- Really handy for the types of calculations done in Numerical Methods

Why spreadsheets?

- Really handy for the types of calculations done in Numerical Methods
- Makes it easy to demonstrate some numerical concepts

Why spreadsheets?

- Really handy for the types of calculations done in Numerical Methods
- Makes it easy to demonstrate some numerical concepts
- I personally use Excel for a lot of things, and I suspect many of my students will need to as well.

Why spreadsheets?

- Really handy for the types of calculations done in Numerical Methods
- Makes it easy to demonstrate some numerical concepts
- I personally use Excel for a lot of things, and I suspect many of my students will need to as well.
- Exercises provide an opportunity for them to get comfortable with Excel and to learn some Excel formulas, functions, and tricks.

Spreadsheet Problems

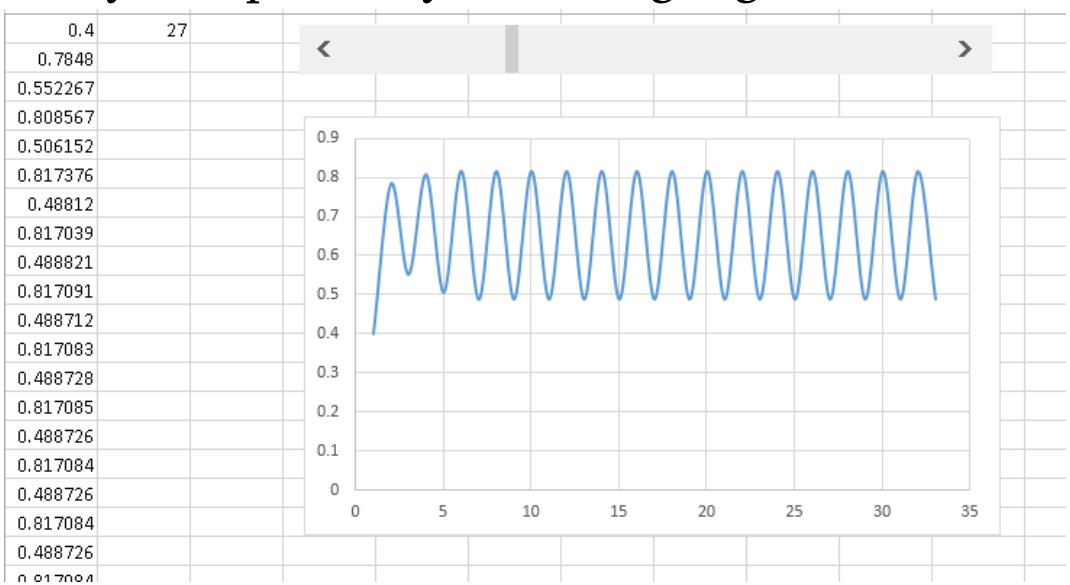
- In class, we created a spreadsheet about ratios of errors for fixed-point iteration. Column A contains the result of iterating $1/(1+x^2)$ 60 times starting at $x = .5$. Column B contains the errors $e_i = |x_i - r|$, where the x_i are the iterates in the first row and r is an estimate of the root (here it will be the entry in cell A61. Column C contains the error ratios e_{i+1}/e_i. Recreate this spreadsheet.

	A	B	C
1	**iterates**	**error**	**error ratio**
2	0.5	0.18233	0.6453881
3	0.8	0.11767	0.6167277
4	0.609756098	0.07257	0.6426762
5	0.72896791	0.04664	0.6288167

Spreadsheet Problems

- Sometimes Newton's Method can behave chaotically. One instance of Newton's Method reduces to iterating $4x(1-x)$, which behaves chaotically. To show how this, create a spreadsheet that iterates $4x(1-x)$ with starting value .4 for 50-100 iterations and plot the results with a scatter plot (and connect the dots).

  Then add a scrollbar to your program to allow the user to be able to vary the 4 in $4x(1-x)$ to values of a between 2 and 4 at intervals of .01. To add a scrollbar, you will need to enable the Developer tab and you'll probably need to google how to insert the scrollbar.



Spreadsheet Problems

- Implement Steffensen's FPI in a spreadsheet to iterate $f(x) = \cos x$. Steffensen's FPI does the following: It does 3 iterations of FPI. Then it applies Aitken's Δ^2 to that to get a value a. Then it does 2 more iterations of FPI, using a as the value to feed into the function. Then it applies Aitken's Δ^2 again, followed by 2 more iterations of FPI. This process of two iterations of FPI followed by one of Aitken's Δ^2 continues.

  Hint: Look up how to do `if()` expressions in Excel and also how to use the `MOD()` function. Your answer should be a general formula, not something that requires line-by-line editing.

	A	B	C	D	E
1	Part (a)			Part (b)	
2	1	0.72801		1	1
3	0.540302	0.733665		2	0.540302
4	0.857553	0.736906		3	0.857553
5	0.65429	0.73805		4	0.72801
6	0.79348	0.738636		5	0.7465
7	0.701369	0.738877		6	0.73407
8	0.76396	0.738992		7	0.739067
9	0.722102	0.739043		8	0.739097
10	0.750418	0.739066		9	0.739077
11	0.731404	0.739076		10	0.739085
12	0.744237	0.739081		11	0.739085

Spreadsheet Problems

- The function $\mathrm{Li}(x) = \int_2^x \frac{1}{\ln t}\,dt$ provides a remarkably good estimate for the number of primes less than x. Write a spreadsheet that uses Simpson's rule with 100 parabolic rectangles to estimate Li(10000). [Note that there are exactly 1229 primes less than 10,000.]

Spreadsheet Problems

- Create a spreadsheet that does the tabular form of Richardson extrapolation to estimate the derivative of $\cos x$ at $x = 1$. Formulas for this were covered in class and are on page 51 of the online notes. Your spreadsheet should use values of h starting at $h = 1$ and going down to $h = 1/64$ by halves. Create an area in your spreadsheet that computes the errors between the estimation and the exact value. The errors should look like the ones below:

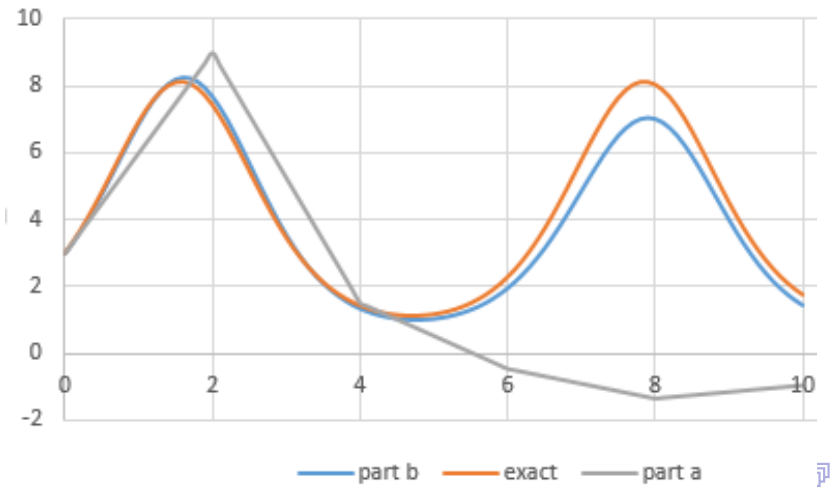
0.133398						
0.034626	0.001702					
0.008738	0.000109	2.56E-06				
0.00219	6.84E-06	4.06E-08	5.59E-10			
0.000548	4.28E-07	6.36E-10	2.2E-12	2E-14		
0.000137	2.67E-08	9.95E-12	8.99E-15	3.33E-16	3.33E-16	
3.42E-05	1.67E-09	1.6E-13	4.22E-15	4.22E-15	4.22E-15	4.22E-15

Spreadsheet Problems

- Consider the following IVP:

$$\begin{cases} y' = y\cos t \\ y(0) = 3 \\ t \in [0, 10]. \end{cases}$$

1. Use Euler's method with $h = 2$. Do this by hand (with a calculator to help with the computations). Show all your work.
2. Use a spreadsheet to implement Euler's method to numerically solve the IVP using $h = .1$.
3. The exact solution is $y = 3e^{\sin t}$. Plot your solutions from parts (a) and (b) on the same graph as the exact solution.



Why Python?

- The syntax is simple, but powerful, making it easy to demonstrate numerical concepts.

Why Python?

- The syntax is simple, but powerful, making it easy to demonstrate numerical concepts.
- My students have had a class in Python programming.

Why Python?

- The syntax is simple, but powerful, making it easy to demonstrate numerical concepts.
- My students have had a class in Python programming.
- Many of the math majors hate programming, but this gives them a chance to use it to do some math.

Why Python?

- The syntax is simple, but powerful, making it easy to demonstrate numerical concepts.
- My students have had a class in Python programming.
- Many of the math majors hate programming, but this gives them a chance to use it to do some math.
- I'm hoping this makes them more comfortable with programming and that they'll appreciate it more.

Why Python?

- The syntax is simple, but powerful, making it easy to demonstrate numerical concepts.
- My students have had a class in Python programming.
- Many of the math majors hate programming, but this gives them a chance to use it to do some math.
- I'm hoping this makes them more comfortable with programming and that they'll appreciate it more.
- To challenge the CS students, sometimes I'll give a harder problems and give students the option to choose to do either a math problem or a programming problem.

Python Problems

- Write a Python program that implements Simpson's rule in an a manner analogous to the program we wrote in class for the trapezoid rule.

```
def trap(f, a, b, n):
    dx = (b-a)/n
    return dx/2*(f(a)+f(b) + 2*sum(f(a+i*dx) for i in range(1,n)))

print(trap(lambda x:x**3, 1, 9, 8))
```

Python Problems

- Modify the provided Python Monte Carlo program to write a Python function that estimates $\int_a^b \int_c^d f(x,y)\,dy\,dx$.

```
def monte_carlo(f, a, b, c, d, n):
    count = 0
    for i in range(n):
        x = uniform(a, b)
        y = uniform(c, d)
        if f(x) >= y >= 0:
            count += 1
        if 0 >= y >= f(x):
            count -= 1
    return count / n * (b-a)*(d-c)
```

Python Problems

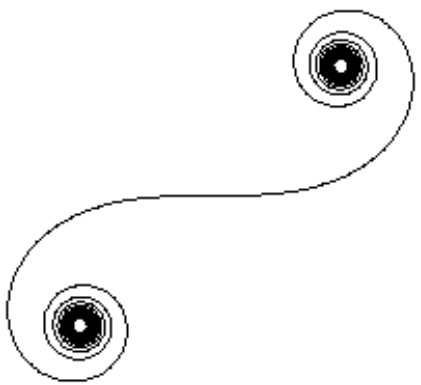
- Included with this assignment is a file that contains code to plot a function parametrically. Currently the code plots $x = \cos t$ and $y = \sin t$, which gives a circle. Change the code so that it plots $x = \int_0^t \cos(u^2)\,du$ and $y = \int_0^t \sin(u^2)\,du$. Use the trapezoid method code provided to do this. You should get a pretty interesting shape.

```
from tkinter import *
from math import *

def plot():
    t = -10
    oldx = func1(t) * 100 + 200
    oldy = func2(t) * -100 + 200
    while t < 10:
        t += .01
        x = func1(t) * 100 + 200
        y = func2(t) * -100 + 200
        canvas.create_line(oldx, oldy, x, y)
        oldx = x
        oldy = y

func1 = lambda t:cos(t)
func2 = lambda t:sin(t)

root = Tk()
canvas = Canvas(width=400, height=400, bg='white')
canvas.grid(row=0, column=0)
plot()
```



Python Problems

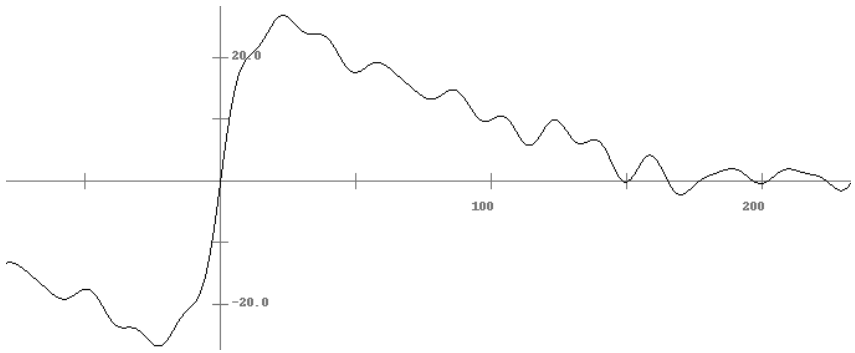
- Recall from class that complex (imaginary) numbers are built into Python. The equation $x^4 + 3x^2 - 2x + 2$ has no real roots, but it does have four complex roots. Use Newton's Method and Python to estimate one of those roots correct to 4 decimal places.

Python Problems

- Use the Python `Decimal` class to estimate the solution of $1-2x-x^5=0$ correct to 50 decimal places.

Python Problems

- Shown below is the graph of a function defined by the Python code below it. It would be painful to evaluate the derivative of this function using basic calculus. However, it is quick to use a numerical method to estimate its derivative. Use the forward difference formula with $h = 10^{-8}$ to estimate $f'(2)$.




```
from math import sin

def f(x):
    n = 27
    t = 0
    for i in range(50):
        t += sin(10*x/n)
        n = (n//2) if n%2==0 else 3*n+1
    return t
```

Python Problems

- Use one of the numerical methods we've learned to write a Python function called `my_sqrt` that computes $\sqrt{n}$ as accurately Python's own `sqrt` function (but without using the language's `sqrt` or power functions).

Python Problems

- Consider the non-elementary function $f(x) = \int_0^x \sqrt[3]{t^2 - 1}\, dt$. Use this definition along with the Python trapezoid method code to estimate a nonzero root of $f(x)$ using Newton's method. [Hints: The nonzero roots are near ± 1.87. Use FTC Part 1 to find $f'(x)$.]

Python Problems

- Consider the polynomial $(x-1)^9(x-2)$. If we expand it out, we get the polynomial below:

$$x^{10} - 11x^9 + 54x^8 - 156x^7 + 294x^6 - 378x^5 + 336x^4 - 204x^3 + 81x^2 - 19x + 2.$$

1. Use Newton's Method in Python on this with a starting value of $x = 1.1$ and perform about 30 iterations. What value do the iterates converge on?
2. The real root is of course 1, but the answer from (a) is as close as we can get. Compute the forward and backward errors of the approximation in part (a).
3. Explain in terms of the graph of the function why the forward and backward errors are the way they are for this problem.
4. If we try a starting value of 1.01, we actually get a division by zero error. In terms of floating point numbers and the functions involved, explain why we get this error.

Python Problems

- Recall that the big idea of the iterative trapezoid rule is that we cut the step size h in half at each step and that allows us to make use of the estimate from the previous step.
  There is also an iterative midpoint rule that is developed from the midpoint rule in a similar way that the iterative trapezoid rule is developed from the trapezoid rule. The major difference is that where the iterative trapezoid rule cuts the step size h in half at every step, the iterative midpoint rule has to cut the step size into a third (i.e. $h_{n+1} = h_n/3$).
  1. Using an example, explain why cutting the step size in half won't work for the extended midpoint rule, but cutting it into a third does work.
  2. Modify the included Python program for the iterative trapezoid rule to implement the iterative midpoint rule. Be careful — several things have to change.

Harder Python Problems

- Write a function in a programming language that is given a list of data points, an x-value, and uses Newton's divided differences to compute the value of the interpolating polynomial at x. It's up to you how to specify how the data points are passed to your function, but make sure that it works for any number of data points.

Harder Python Problems

- Write a program that returns the nth Chebychev polynomial, nicely formatted as a string. For instance, `cheb(5)` should return "16x^5-20x^3+5x".

Calculus review questions

- Students learn a lot in calculus, and if it is not reinforced in upper level classes, then they will forget it.

Calculus review questions

- Students learn a lot in calculus, and if it is not reinforced in upper level classes, then they will forget it.
- Some particular things I've focused on: Taylor series, FTC Part 1, Functions defined as integrals

Calculus review questions

- Students learn a lot in calculus, and if it is not reinforced in upper level classes, then they will forget it.
- Some particular things I've focused on: Taylor series, FTC Part 1, Functions defined as integrals
- Sometimes, I'll ask a question straight out of a calculus class, and other times I'll ask numerical questions that require those ideas.

A few final notes

- I do include plenty of problems where students are just doing the methods by hand.

A few final notes

- I do include plenty of problems where students are just doing the methods by hand.
- Tests consist of about half understanding questions and half doing methods by hand.

A few final notes

- I do include plenty of problems where students are just doing the methods by hand.
- Tests consist of about half understanding questions and half doing methods by hand.
- In the future I would like to add more application problems.

A few final notes

- I do include plenty of problems where students are just doing the methods by hand.
- Tests consist of about half understanding questions and half doing methods by hand.
- In the future I would like to add more application problems.
- Feel free to borrow any of these for your own classes.

A few final notes

- I do include plenty of problems where students are just doing the methods by hand.
- Tests consist of about half understanding questions and half doing methods by hand.
- In the future I would like to add more application problems.
- Feel free to borrow any of these for your own classes.
- Thank you. These slides are available now at `www.brianheinold.net` or at the section website after the meeting.