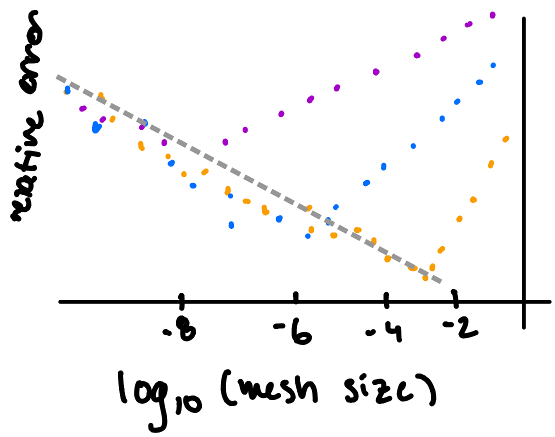
grade: check (see comments in blue to upgrade to check +)

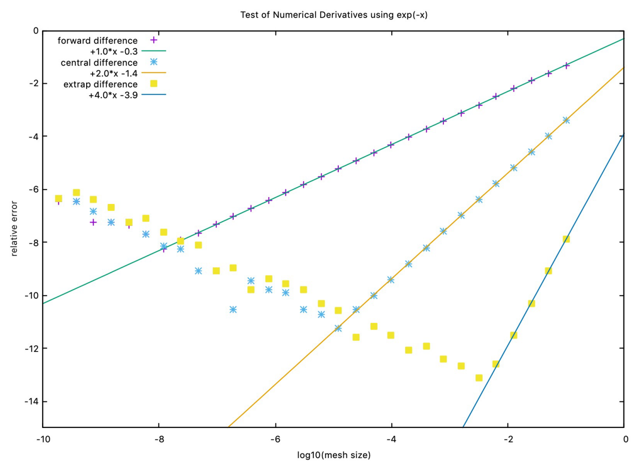
Activity 4

Numerical Derivatives and Richardson Extrap.

1) ✓

2)





3) The slope of 4.0 means that the error goes like $1/N^4$, which is what we expected.

how would you get an even higher order result?

4) The slope of the error does not depend on the method for approximating error. Instead, it depends on the use of double precision floats. which were used in all of the approximations and the error will scale the same.

You're correct that the slope will be the same for all, but it has nothing to do with whether you're using double or single precision.

(I can explain this in class/office hours if it's not clear)

Pointer Games

I created a new cpp file to do this activity, it's called "test_derivative_test.cpp"

1) ✓

2) ✓

3) ✓

4) Changed to $x=2$ and my plot file stopped working

5) Yes, the slopes are the same in each, and the intercepts are different.

Linear Algebra

1) ✓

2) Called Hilbert.py in my session-04 GitHub

3) Same answers! But my Python code gives more decimals / precision.

4) ✓

5) My only guess is that it will not be linear. This is because square matrices do not grow linearly. For example a 100×100 matrix does not have double the amount of inputs as a 50×50 matrix, it has 4x the amount.

$x \cdot x$ matrix compared to a $2x \cdot 2x$ matrix

$$\frac{2x \cdot 2x}{x \cdot x} = 4$$

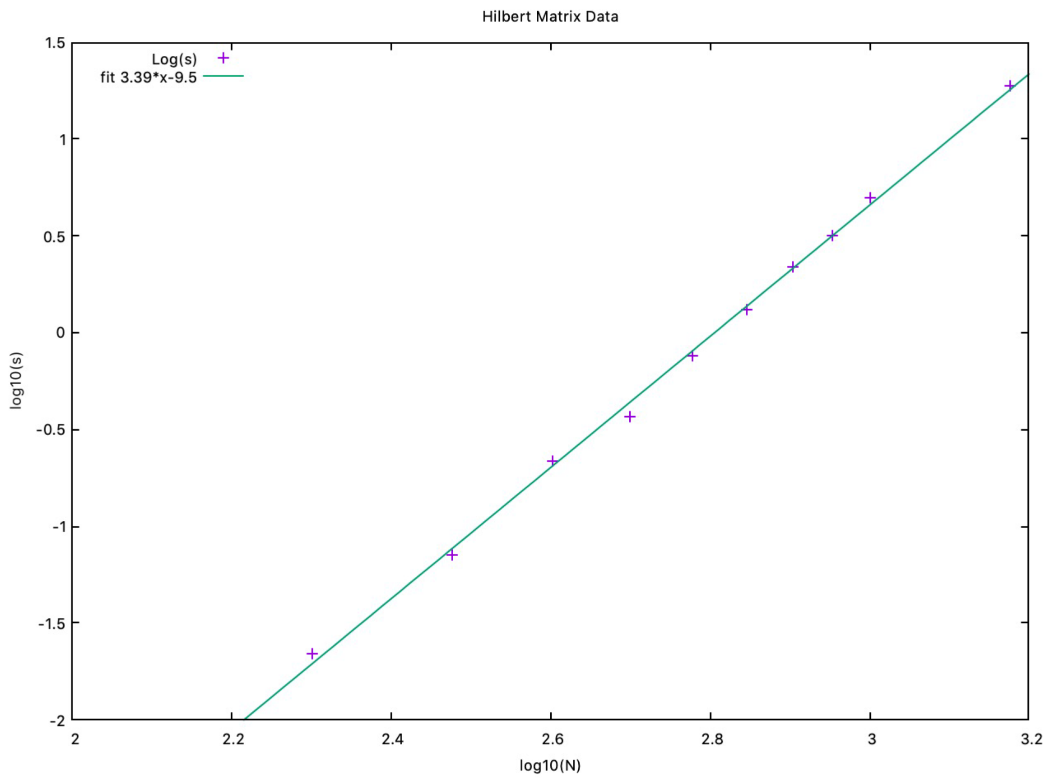
$x \cdot x$ matrix compared to a $4x \cdot 4x$ matrix

$$\frac{4x \cdot 4x}{x \cdot x} = 16$$

I'd guess 2 or 4 for the power law based on my math

6) According to my data, the power law is 3.39, which is between 2 and 4.

Between 2 and 4 is a good guess. E.g., lots of routines for diagonalization involve matrix-matrix multiplications, which should scale as n^3.



Hilbert Matrix Data
Log(s)
fit 3.39*x-9.5
1.5
1
0.5
0
-0.5
-1
-1.5
-2
2
2.2
2.4
2.6
2.8
3
3.2
log10(s)
log10(N)