PHYS 512 Assignment 1

Michelle Lam, SN: 261005326

September 2021

1 Problem 1

Part A

$$\begin{aligned}
f(x+\delta) &= f(x) + \delta f'(x) + \frac{\delta^2}{2} f''(x) + \frac{\delta^3}{3!} f^{(3)}(x) + \frac{\delta^4}{4!} f^{(4)}(x) + \frac{\delta^5}{5!} f^{(5)}(x) + ... \\
f(x-\delta) &= f(x) - \delta f'(x) + \frac{\delta^2}{2} f''(x) - \frac{\delta^3}{3!} f^{(3)}(x) + \frac{\delta^4}{4!} f^{(4)}(x) - \frac{\delta^5}{5!} f^{(5)}(x) + .. \\
f(x+2\delta) &= f(x) + 2\delta f'(x) + \frac{(2\delta)^2}{2} f''(x) + \frac{(2\delta)^3}{3!} f^{(3)}(x) + \frac{(2\delta)^4}{4!} f^{(4)}(x) + \frac{(2\delta)^5}{5!} f^{(5)}(x) + .. \\
f(x-2\delta) &= f(x) - 2\delta f'(x) + \frac{(2\delta)^2}{2} f''(x) - \frac{(2\delta)^3}{3!} f^{(3)}(x) + \frac{(2\delta)^4}{4!} f^{(4)}(x) - \frac{(2\delta)^5}{5!} f^{(5)}(x) + ..
\end{aligned} \tag{1}$$

Calculating 2-sided derivative for x $\pm$ δ:

$$\begin{aligned}
f'(x) &= \frac{f(x+\delta) - f(x-\delta)}{2\delta} \\
&\approx \frac{1}{2\delta}(2\delta f'(x) + \frac{2\delta^3}{3!} f^{(3)}(x) + \frac{2\delta^5}{5!} f^{(5)}(x) + \epsilon f(x)) \\
&\approx f'(x) + \frac{\delta^2}{3!} f^{(3)}(x) + \frac{\delta^4}{5!} f^{(5)}(x) + \frac{\epsilon f(x)}{2\delta}
\end{aligned} \tag{2}$$

Calculating 2-sided derivative for x $\pm$ 2δ:

$$\begin{aligned}
f'(x) &= \frac{f(x+2\delta) - f(x-2\delta)}{4\delta} \\
&= \frac{1}{4\delta}(4\delta f'(x) + \frac{2^4\delta^3}{3!} f^{(3)}(x) + \frac{2^6\delta^5}{5!} f^{(5)}(x) + \epsilon f(x)) \\
&= f'(x) + \frac{4\delta^2}{3!} f^{(3)}(x) + \frac{2^4\delta^4}{5!} f^{(5)}(x) + \frac{\epsilon f(x)}{4\delta}
\end{aligned} \tag{3}$$

Now, we want to combine these derivatives in a way that cancels out the leading error term,$\sim \frac{\delta^2}{3!} f^{(3)}(x)$

$$
\begin{aligned}
4\frac{f(x+\delta)-f(x-\delta)}{2\delta} - \frac{f(x+2\delta)-f(x-2\delta)}{4\delta} \\
= 3f'(x) - \frac{12\delta^4}{5!}f^{(5)}(x) + \frac{7\epsilon f(x)}{4\delta} \\
3f'(x) = 3f'(x) - \frac{12\delta^4}{5!}f^{(5)}(x) + \frac{7\epsilon f(x)}{4\delta} \\
f'(x) = f'(x) - \frac{4\delta^4}{5!}f^{(5)}(x) + \frac{7\epsilon f(x)}{12\delta}
\end{aligned} \tag{4}
$$

with the 4 points, the error term is now:

$$
Error = -\frac{4\delta^4}{5!}f^{(5)}(x) + \frac{7\epsilon f(x)}{12\delta} \tag{5}
$$

Note, ϵ is the machine accuracy and accounts for roundoff error, and the $\frac{\delta^4}{5!}f^{(5)}(x)$ term is the new leading error term.
Therefore, the estimate of the first derivative, f'(x), at x should be:

$$
\begin{aligned}
f'(x) &= \frac{1}{3}(4\frac{f(x+\delta)-f(x-\delta)}{2\delta} - \frac{f(x+2\delta)-f(x-2\delta)}{4\delta}) \\
f'(x) &= \frac{8(f(x+\delta)-f(x-\delta)) - (f(x+2\delta)-f(x-2\delta))}{12\delta}
\end{aligned} \tag{6}
$$

1.1 Part B

Again, our error term is:

$$
Error = -\frac{4\delta^4}{5!}f^{(5)}(x) + \frac{7\epsilon f(x)}{12\delta} \tag{7}
$$

So to find δ, we take the derivative of our error w.r.t δ, set to 0, and isolate for δ :

$$
\begin{aligned}
Error' = -\frac{16\delta^3}{5!}f^{(5)}(x) - \frac{7\epsilon f(x)}{12\delta^2} = 0 \\
\delta^5 \sim -\frac{\epsilon f(x)}{f^{(5)}(x)} \\
|\delta| \sim (\frac{\epsilon f(x)}{f^{(5)}(x)})^{1/5}
\end{aligned} \tag{8}
$$

Therefore, in terms of machine precision ϵ and our function f(x), our δ is:

$$
|\delta| \sim (\frac{\epsilon f(x)}{f^{(5)}(x)})^{1/5} \tag{9}
$$

Run **Problem1b.py**, used x = 42 as test value. And the error from using our δ above yielded, 4.6e-12 for f(x) = e^x and -7.76e-14 for f(x) = $e^{0.01x}$. This demonstrates that our estimate of δ was roughly correct

2 Problem 2

$$f'(x) = f'(x) + \frac{\delta^2}{3!}f'''(x) + \frac{\epsilon f(x)}{2\delta} \tag{10}$$

$$Error = \frac{\delta^2}{3!}f'''(x) + \frac{\epsilon f(x)}{2\delta} \tag{11}$$

$$\begin{aligned} Error' = 0 = \frac{2\delta}{3!}f'''(x) - \frac{\epsilon f(x)}{2\delta^2} \\ \delta = (\frac{\epsilon f(x)}{f'''(x)})^{1/3} \end{aligned} \tag{12}$$

Assuming our function is a polynomial, then if the leading order of f(x) $\sim x^n$, then f"'(x) $\sim x^{n-3}$, which leaves us with..

$$\delta = x\epsilon^{1/3} \tag{13}$$

To determine the error in our function we will plug our original δ back into our Error equation:

$$\begin{aligned} Error &= ((\frac{\epsilon f(x)}{f'''(x)})^{1/3})^2 \frac{f'''(x)}{3!} + \frac{\epsilon f(x)}{2}(\frac{f'''(x)}{\epsilon f(x)})^{1/3} \\ &= \frac{\epsilon^{2/3} f(x)^{2/3} f'''(x)^{1/3}}{3!} + \frac{\epsilon^{2/3} f(x)^{2/3} f'''(x)^{1/3}}{2} \\ &\sim \epsilon^{2/3} f(x) \end{aligned} \tag{14}$$

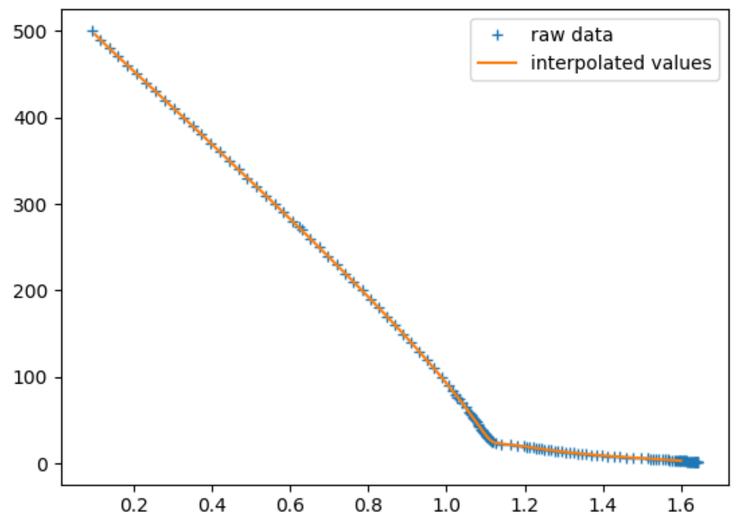
Refer to Problem2.py code for function, or you can run it with a test function (right now I have test function = cos(x))

3 Problem 3

Write routine that takes voltage, and interpolate to return a temperature. From class, we called scipy.interpolate.splrep to do a spline fit, and then found out interpolated y values from there. To estimate the error, from tutorial, we used a bootstrap resampling method, which takes a subset of our original data multiple time (with replacement) and calculates statistics from it.

Here's the interpolated values compared to the raw data:

Run **Problem3.py** for output. The estimated error was $\sim$ 0.013 as the mean standard deviation.



4 Problem 4

Run **Problem4.py** for graphs of interpolation and residuals. To compare the accuracy between the interpolation, I plotted the residuals for the cosine and lorentzian function.

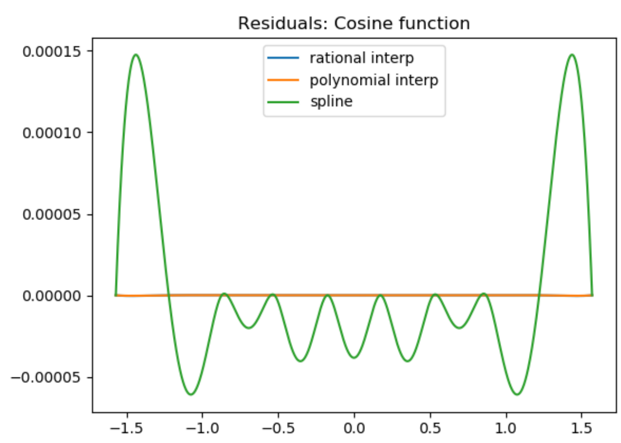


Figure 1: Residuals of cosine function. Note, for rational function 'inv' was used to calculate the inverse matrix.

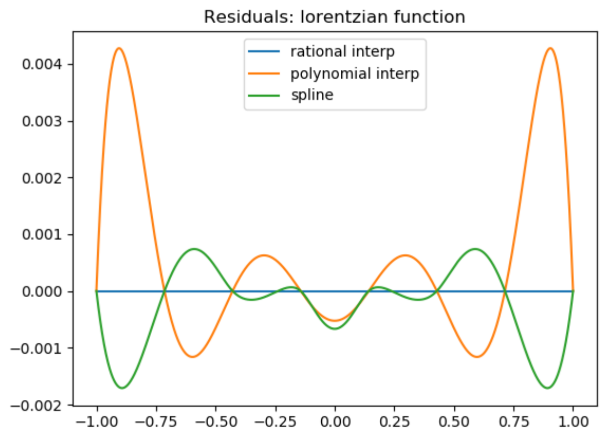


Figure 2: Residuals of Lorentzian function. Note, for rational function 'pinv' was used to calculate the inverse matrix.

To compare rational function, i plotted residuals with and without pinv on the same graph and for 6 pts, and for 10 pts (m=5, n=4)

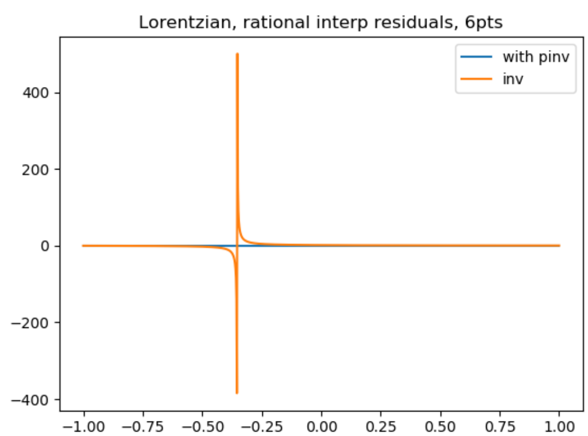


Figure 3: Residuals of Lorentzian function in rational interpolation. Using 6 points.

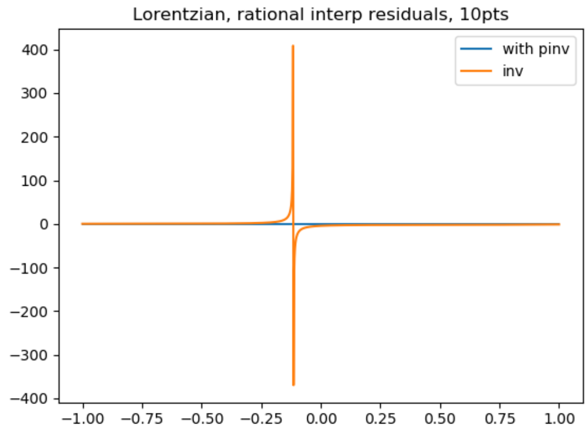


Figure 4: Residuals of Lorentzian function in rational interpolation. Using 10 points.

The higher order m =5, n=4 seemed to make the residuals slightly smaller. And from the blue line we see that pinv improved the interpolation greatly.

Looking at the p, q, values below. We see that the terms in the top rows using pinv are smaller than the bottom rows using inv. So, pinv's ability to deal with singular matrices and possibly rounding errors since the precision in the values is higher in the top row?

```
p: [1.00000000e+00 2.77555756e-16 1.55431223e-15]
q: [-1.11022302e-16  1.00000000e+00  1.11022302e-15]
p: [ 2.65837104  4.          -0.03159467]
q: [3.  1.  1.5]
(base)
```

Figure 5: p,q values from rational function, from 6pts. Top two p,q values are with pinv, and bottom two p,q values are with inv.