# PHYS 512 - Problem Set 1 Solutions

Minya Bai (260856843)

September 18, 2021

## 1 Question 1

### 1.1 Estimate the first derivative at x using Taylor series expansion

Consider the following Taylor series expansions for $f(x \pm \delta)$ and $f(x \pm 2\delta)$:

$$f(x \pm \delta) = f(x) + (\pm\delta)f^{(1)}(x) + \frac{1}{2}(\pm\delta)^2 f^{(2)}(x) + \frac{1}{6}(\pm\delta)^3 f^{(3)}(x) + \ldots, \tag{1}$$

$$f(x \pm 2\delta) = f(x) + (\pm 2\delta)f^{(1)}(x) + \frac{1}{2}(\pm 2\delta)^2 f^{(2)}(x) + \frac{1}{6}(\pm 2\delta)^3 f^{(3)}(x) + \ldots. \tag{2}$$

Observe that if we take the difference between the $\pm$ terms for equations 1 and 2, the even order terms will cancel out. We can then divide by the difference between the two values to get the definition of a derivative:

$$\frac{f(x+\delta) - f(x-\delta)}{2\delta} = f^{(1)}(x) + \frac{1}{6}(\pm\delta)^2 f^{(3)}(x) + \frac{1}{120}(\pm\delta)^4 f^{(5)}(x) + \ldots, \tag{3}$$

$$\frac{f(x+2\delta) - f(x-2\delta)}{4\delta} = f^{(1)}(x) + \frac{2}{3}(\pm\delta)^2 f^{(3)}(x) + \frac{2}{15}(\pm\delta)^4 f^{(5)}(x) + \ldots. \tag{4}$$

Now we can try to find the truncation error to the fifth order expansion by eliminating the third order terms from equations by taking 4·(3) - (4):

$$4\left(\frac{f(x+\delta) - f(x-\delta)}{2\delta}\right) - \frac{f(x+2\delta) - f(x-2\delta)}{4\delta} = 3f^{(1)}(x) - \frac{1}{10}\delta^4 f^{(5)}(x)$$
$$\frac{8f(x+\delta) - 8f(x-\delta) - f(x+2\delta) + f(x-2\delta)}{12\delta} = f^{(1)}(x) - \frac{1}{30}\delta^4 f^{(5)}(x)$$

$$f^{(1)} \approx \frac{f(x-2\delta) - 8f(x-\delta) + 8f(x+\delta) - f(x+2\delta)}{12\delta} \tag{5}$$

### 1.2 Optimizing $\delta$

To calculate the total error in our calculation we need to determine the truncation and roundoff error. From part a) we can take the fifth order term to be our truncation error:

$$\epsilon_t = \frac{1}{30}\delta^4 f^{(5)}(x). \tag{6}$$

For the roundoff error, similar to what was done in class, we can estimate this to be the machine precision multiplied by the function and divided by the uncertainty in x:

$$\epsilon_r = \epsilon_m \left| \frac{f(x)}{\delta} \right|. \tag{7}$$

So the total error observed in our calculation would be the sum of these two errors. We want to fine $\delta$ such that we can minimize this error value.

$$err = \epsilon_t + \epsilon_r$$
$$err = \frac{1}{30}\delta^4 f^{(5)}(x) + \epsilon_m \left| \frac{f(x)}{\delta} \right|$$
$$0 = \frac{2}{15}\delta^3 f^{(5)} - \frac{\epsilon_m f(x)}{\delta^2}$$
$$\delta^5 = \frac{15}{2}\frac{\epsilon_m f(x)}{f^{(5)}(x)}$$

$$\delta = \left( \frac{15}{2}\frac{\epsilon_m f(x)}{f^{(5)}(x)} \right)^{\frac{1}{5}} \tag{8}$$

To verify this is the minimum delta expected, we can plot the error as a function of delta at some fixed value. The follow plots were made for x = 10 for the functions $f(x) = e^x$ and $f(x) = e^{.01x}$.

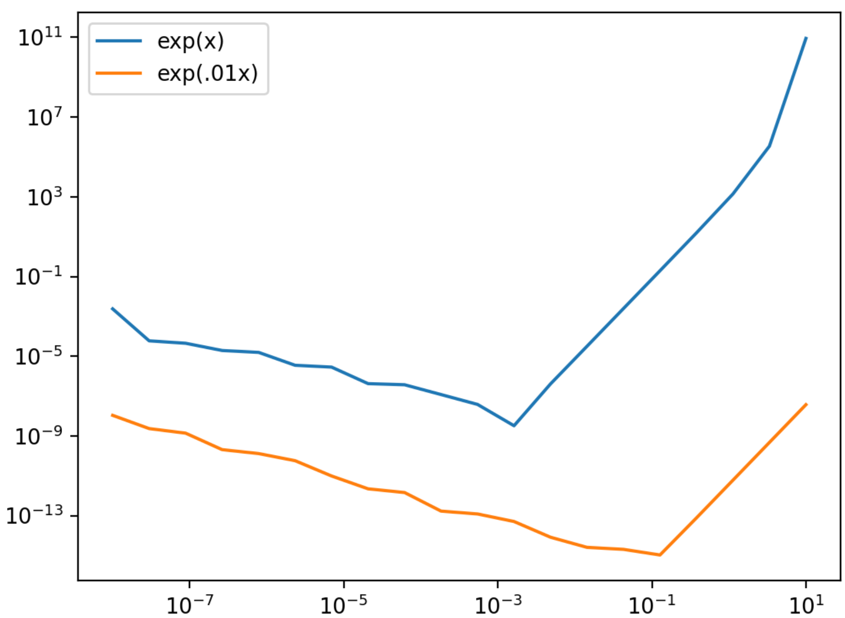


Figure 1: The optimal $\delta$ for $f(x) = e^x$ was estimated to be 0.00094 and for $f(x) = e^{.01x}$ was estimated to be 0.094.

## 2 Question 2

Similar to the previous question, we can get rid of the even order terms by taking the Taylor expansion of the function with the center derivative:

$$\frac{f(x+dx)-f(x-dx)}{2dx} = \frac{f(x)+dxf^{(1)}(x)+\frac{1}{2}dx^2f^{(2)}+\ldots-\left(f(x)-dxf^{(1)}(x)+\frac{1}{2}dx^2f^{(2)}+\ldots\right)}{2dx}$$
$$\frac{f(x+dx)-f(x-dx)}{2dx} = \frac{2dxf^{(1)}(x)+\frac{1}{3}dx^3f^{(3)}+\ldots}{2dx}$$

$$\frac{f(x+dx)-f(x-dx)}{2dx} \approx f^{(1)}(x)+\frac{1}{6}dx^2f^{(3)}(x) \tag{9}$$

From 9 we can take our truncation error to be $\frac{1}{6}dx^2f^{(3)}(x)$ and again we can add this to our roundoff error $\epsilon_m\left|\frac{f(x)}{dx}\right|$ to determine the total error in calculation.

$$err = \epsilon_t + \epsilon_r$$
$$err = \frac{1}{6}dx^2f^{(3)}(x)+\epsilon_m\left|\frac{f(x)}{dx}\right|$$
$$0 = \frac{1}{3}dxf^{(3)} - \frac{\epsilon_m f(x)}{dx^2}$$
$$dx^3 = \frac{3\epsilon_m f(x)}{f^{(3)}(x)}$$

$$dx = \left(\frac{3\epsilon_m f(x)}{f^{(3)}(x)}\right)^{\frac{1}{3}} \tag{10}$$

To calculate the fractional error, consider the following:

$$\frac{err}{f^{(1)}} = \frac{\frac{1}{6}dx^2f^{(3)}+\epsilon_m\left|\frac{f(x)}{dx}\right|}{f^{(1)}}$$
$$\frac{err}{f^{(1)}} = \frac{\frac{1}{6}\left(\left(\frac{3\epsilon_m f(x)}{f^{(3)}(x)}\right)^{\frac{1}{3}}\right)^2 f^{(3)}+\epsilon_m\frac{f(x)}{\left(\frac{3\epsilon_m f(x)}{f^{(3)}(x)}\right)^{\frac{1}{3}}}}{f^{(1)}}$$
$$\frac{err}{f^{(1)}} = \frac{\frac{1}{6}3^{\frac{2}{3}}\epsilon_m^{\frac{2}{3}}f^{\frac{2}{3}}f^{(3)\frac{1}{3}}+3^{-\frac{1}{3}}\epsilon_m^{\frac{2}{3}}f^{\frac{2}{3}}f^{(3)\frac{1}{3}}}{f^{(1)}}$$
$$\frac{err}{f^{(1)}} = \frac{\left(\frac{1}{6}3^{\frac{2}{3}}+3^{-\frac{1}{3}}\right)\epsilon_m^{\frac{2}{3}}f^{\frac{2}{3}}f^{(3)\frac{1}{3}}}{f^{(1)}}$$

$$\frac{err}{f^{(1)}} \approx \frac{3^{\frac{4}{3}}}{2}\epsilon_m^{\frac{2}{3}} \tag{11}$$

## 3 Question 3

The follow graph is the lakeshore data plotted with temperature as a function of voltage. We can observe that between two consecutive points, we can approximate the data to be linear. In order to approximate for any voltage V, we can find the closest two data points and approximate using linear interpolation.

Since I am using linear interpolation, I would expect there to be slight deviations based on the data points I choose. Thus, I will take the uncertainty in V to be the standard deviation from different combinations of data points chosen.

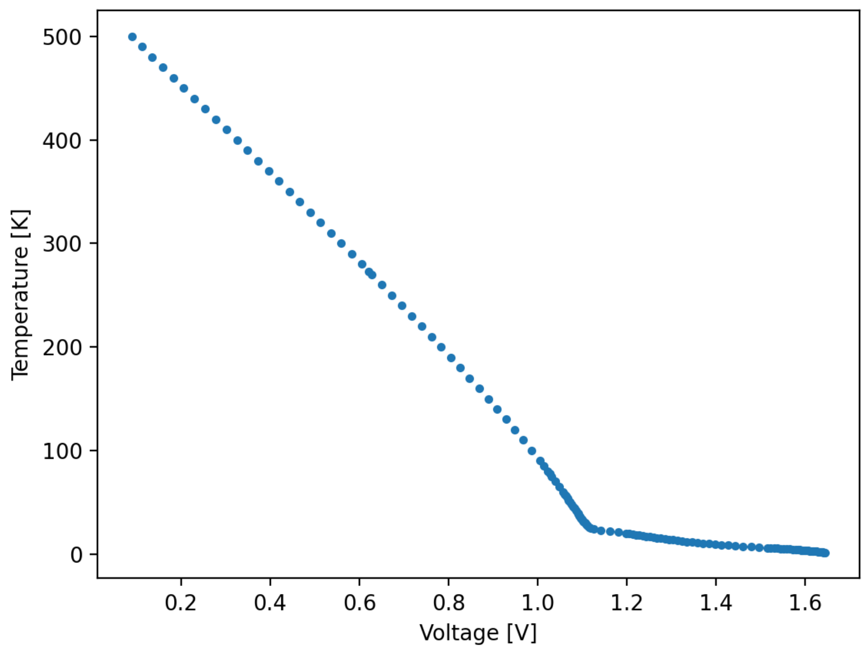


Figure 2: Temperature as a function of voltage as taken from lakeshore.txt file.

# 4 Question 4

The following plots are for the cos function and the Lorentzian using both the np.linalg.inv and np.linalg.pinv functions.

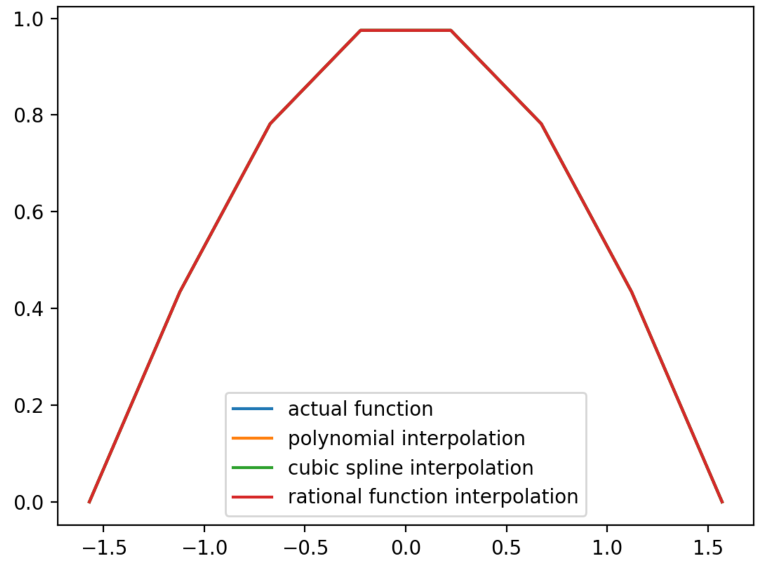


Figure 3: $f(x) = cos(x)$ for $x \in (-\pi/2 \leq x \leq \pi/2)$.

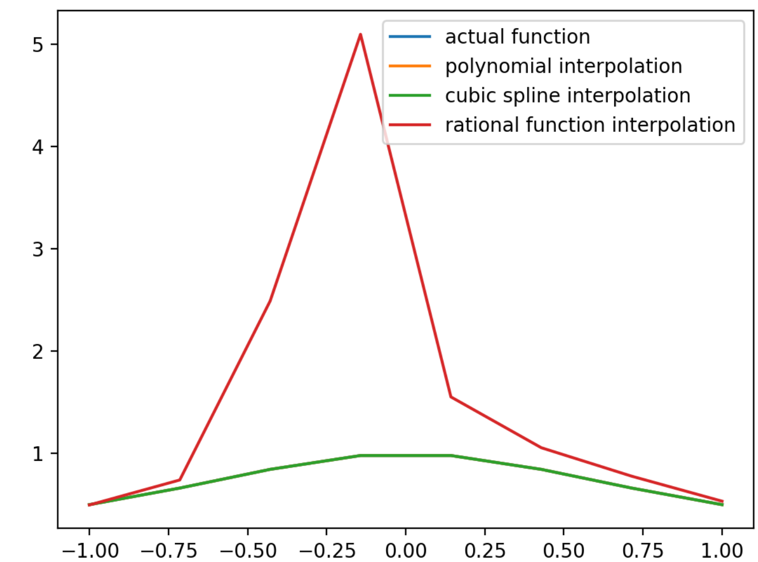


Figure 4: $f(x) = \frac{1}{1+x^2}$ for $x \in (-1 \leq x \leq 1)$ using np.linalg.inv.

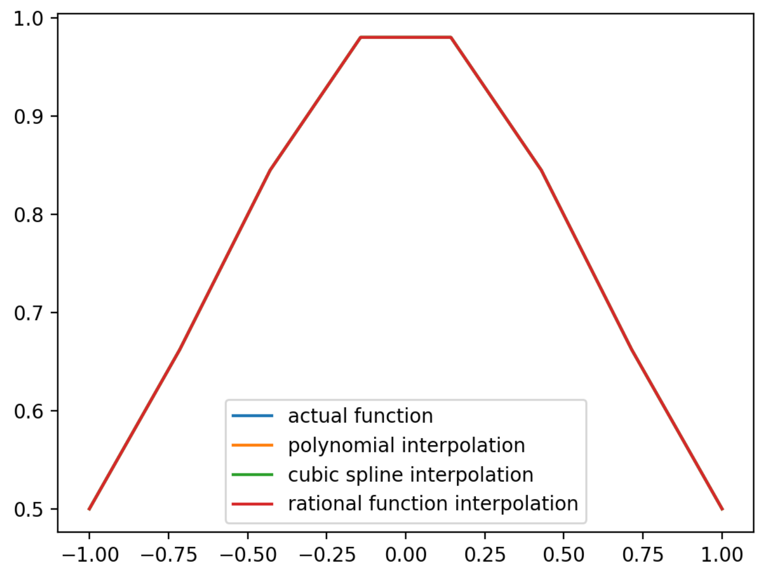


Figure 5: $f(x) = \frac{1}{1+x^2}$ for $x \in (-1 \leq x \leq 1)$ using np.linalg.pinv.

For doubles, the machine error is approximately $10^{-16}$ so we can expect errors of similar orders. The errors in the interpolated values is of order $10^{-15}$ using polynomial interpolation and is of order $10^{-16}$ using cubic spline. For rational functions, as it can be seen in the plots, the errors using np.linalg.inv varies from 1.64464828 to 2.90829617e-03 whereas it only varies by order of $10^{-15}$ using np.linalg.pinv. The errors using np.linalg.inv is much greater than anything we would expect.

When np.linalg.inv is used, the coeffs for p and q are [2.30752097, 5.0, 7.0,-3.89449214] and [6.0,9.0,-4.5,8.0] respectively. When np.linalg.pinv is used the coeffs for p and q are [1.0,1.77635684e-15,-0.333,0.0] and [0.00000000e+00,6.66666667e-01,-1.77635684e-15,-3.33333333e-01] respectively. We can see that when pinv is used instead of inv the coefficients are much smaller. This is possibly due to the fact that the values are very small in the matrix and will blow up when inverted. In pinv, since it sets small values to 0, the values will not blow up resulting in more reasonable coefficients.