Lab02 Report

Habiba Zaghloul(Q1, Q2, Q3, Report), Ryan Cunningham(Q1, Q2, Q3, Q4, Report)

2022.9.23

Q1 Part (a)

Pseudocode

```
# Read in the data using numpy.loadtxt
# Find how many values are in the data set (n)
# Calculate the mean by summing over all the values and dividing by n
# Subtract the mean from each value, square it, then take the sum of these differences, sqaured
# Take the square root of the sum divided by n-1 to get the standard deviation using equation (1)
# Calculate the true value using numpy.std
# Find the relative error
# To use equation (2), we calculate the square of the data and the square of the mean
# Multiply n by the square of the mean
# Use a for loop to subtract n*meansquared from the square of the data and keep only the positive va
# Take the sum of these values and use equation (2) to find the standard deviation
```

1 Q1 Part (b)

Equation 1:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{1}^{n}(x_i - \bar{x})^2} \tag{1}$$

Equation 2:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{1}^{n}(x_i^2 - n\bar{x}^2)} \tag{2}$$

With Equation 3 used to calculate Relative Error (RE):

$$RE = \frac{x-y}{y} \quad or \quad RE = \frac{\sigma_{estimate} - \sigma_{true}}{\sigma_{true}} \tag{3}$$

Using equation (1), the standard deviation of Michelsen's speed of light was calculated to be $\sigma = 0.07901054781905065 \times 10^3 kms^{-1}$. The true value came to be $\sigma_{true} = 0.07901054781905067 \times 10^3 kms^{-1}$. We can see that they are close in value and the relative error was found to be $RE_1 = -3.512894971845343 \times 10^{-16}$.

Using equation 2, we get $\sigma = 0.07901054787645388$. The relative error is $RE_2 = 7.265258724578925 \times 10^{-10}$. The relative error we get from using equation 2 is larger than that of equation 1 by a factor of 10^7.

Q1 Part (c)

For sequence 1 with mean 0, the true standard deviation is $\sigma_{true} = 0.9700155401088982$. Sequence 2 has the same variance and length as sequence 1, 1.0 and 2000, respectively. However it has a much larger mean that is 1.0×10^7. The true standard deviation for this sequence is $\sigma_{true} = 0.9908777828517911$.

We again used both methods to find the standard deviations for sequence 1 and 2 in order to find the relative error and compare. Using equation (1), $\sigma_1 = .9703495579769996$ and the relative error was $RE = 0.0003443428010069715$.

For sequence 2, $\sigma_2 = 0.9910482851258191$ and the relative error was $RE = 0.00017207195173689943$.

Using equation 2, the standard deviation for sequence 1 is $\sigma = 1.00427693835057 63$. This gives us a relative error of $5.81265753843317 \times 10^{-06}$. Using equation 2 again on sequence 2, $\sigma = 0.876575263223899$. Hence the relative error is $RE = -0.1153548112653452$

Regardless of which equation we used, the relative error for sequence 2 was always smaller than that of sequence 1. We can say that this is due to the mean being much larger in sequence 2.

Q1 Part (d)

We did not encounter an error when using Equation 2.

Q2 Part (a)

The exact value of the integral $\int_0^1 \frac{4}{1+x^2} dx = \pi$

Q2 Part (b)

Pseudocode

```
# Define the function
# Set values to the limits, number of slices, and h (width of slice) that will be used in both the t
# Let s be 0.5 times the value of the function at the lower limit + 0.5 times the value of the funct
# Create a for loop to add more of the areas of the slices to s
# Multiply h by s to get the area under the graph (our final answer)
```

The true value of the integration is π

The estimation using the trapezoid method yielded a value of 3.1311764705882354 with a relative error of -0.003315574025695356

The estimation using the Simpson method yielded a value of 3.637254901960784 with a relative error of 0.15777419386457184

It is clear to see that in this case, the trapezoid method produced a better estimate for the integral than the Simpson's method. The relative error was much smaller, by a factor of 100.

Q2 Part (c)

The value of v for N^v to achieve an error of 10^{-9} is 12

The value of the integral using this N is 3.1415926436556836.

This had a relative value of $-3.1621252754172446 \times 10^{-9}$

The time taken for this calculation to be completed was 0.0015897750854492188 seconds

Q2 Part (d)

The value of the integral using $N_1 = 16 is 3.140941612041389$, whereas when N was equal to 32, the integral was found to be 3.141429893174975.

The relative error of the N_1 value is -0.00020723296117346922. And the relative error of the N_2 value is $-5.18082491160139 \times 10^{-5}$ When we multiply relative error for the second value by 4 we obtain -0.00020723299646405584. This is very similar to RE which is what we hoped to see according to the textbook, because when you half the width of a slice, the error decreases for a quarter.

Q2 Part (e)

Q3 Part (a)

We want to find C_1 in

$$W = \pi \int_0^\infty B dv = C_1 \int_0^\infty \frac{x^3}{e^x - 1} dx \tag{4}$$

Where:

$$B = \frac{2hc^{-2}v^3}{e^{\frac{hv}{kt}} - 1} \tag{5}$$

Where the total energy per unit area emitted by a black body follows Stephan's Law:

$$W = \sigma T^4 \tag{6}$$

$$x = \frac{hv}{kT}$$

$$\frac{dx}{dv} = \frac{h}{kT}$$

$$dv = \frac{kT}{h} dx$$

Substituting this into our integral we get:

$$\pi \int_0^\infty B dv = \frac{2hc^{-2}v^3}{e^{\frac{hv}{kT}} - 1} \times \frac{kT}{h} dx$$

$$= \pi \int_0^\infty \frac{2v^3 kT}{c^2(e^x - 1)} dx$$

Now by taking the cube of x, we have:

$$x^3 = (\frac{hv}{kT})^3$$

A new term must be added to the integral in order for us to have x^3 in the numerator. After some rearranging, the term in the integral becomes:

$$\frac{2v^3kT}{c^2(e^x-1)} \times \frac{(kT)^3}{h}$$

$$= \frac{2(kT)^4}{h^3c^2} \times \frac{x^3}{e^x-1}$$

Hence our value of C_1 is:

$$C_1 = \pi\frac{2(kT)^4}{h^3c^2}$$

Q3 Part (b)

For this part, we used Simpsons rule since we know it has higher accuracy than the trapezoidal rule. The limits could not be set to 0 and ∞ because we would get a 0 in the denominator causing errors. Therefore, we set the lower limit to 0.0001 and the upper limit to 700 (anything larger and the program would not run). The number of slices was taken to be $N = 800$. We used this many slices to offset the fact that we could not integrate to infinity so smaller sections gave us more accurate areas.

The integral was calculated to be:

$$\int_{0.0001}^{700} \frac{x^3}{e^x-1}dx = 6.503708322854249$$

We used an integral calculator (wolframalpha) to check our answer and the value of W came to be:

$$\frac{\pi^4}{15} = 6.493939402266828$$

With a relative error of:

$$RE = 0.0015043134809682268$$

The relative error suggests that our calculated value is very close to the true value of the integral (accurate to 0.15%).

Q3 Part (c)

Since we now know what C_1 is and what the value of the integral is, we can plug in the values in the formula for W given above. $W = C_1 \times 6.503708322854249$ If we split W to T^4 multiplied by some constant , we can calculate it and check to see if it agrees with the Stefan Boltzmann constant in scipy.constants. Simplifying W we get:

$$W = T^4 \times \frac{2\pi k^4}{h^3c^2} \times 6.503708322854249$$

Plugging in the values of each constant we get a final answer of $\sigma = 5.678904439865347 \times 10^{-8}Wm^{-2}K^{=4}$ Using scipy.constants, we get $\sigma = 5.670374419 \times 10^{-8}Wm^{-2}K^{=4}$.

To check for accuracy we will again find the relative error to see how close our calculated value is to the true value: $RE = 0.0015043135135424423$. The relative error tells us that the method we used gave us a value accurate to 0.15%.

Q4 Part (a)

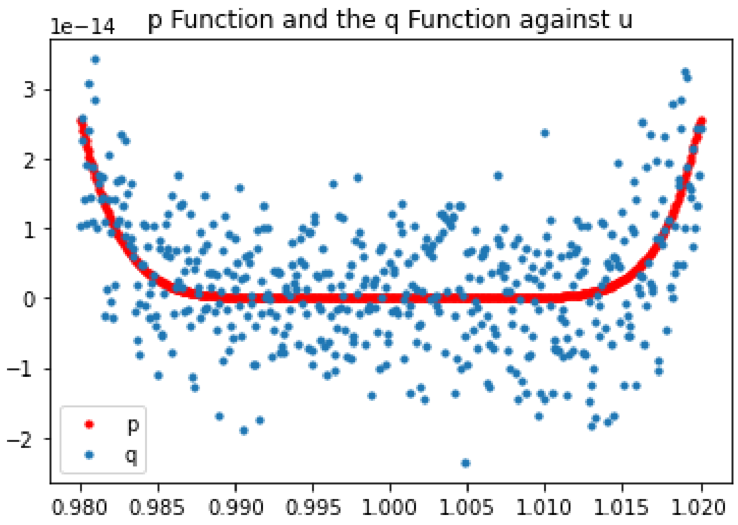
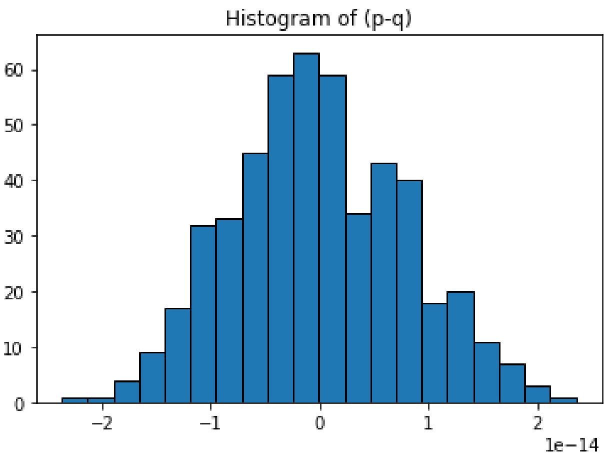


Figure 1:

Figure 1 shows the difference between the q function and the p function against u when u is very close to 1. As can be seen, the q function is a lot more noisy. This is because while the p function outputs the true value of $(1 - u)^8$, the q function is only an estimation of that expansion. It is therefore no surprise that it is less precise than its counterpart, the p function.

Q4 Part (b)



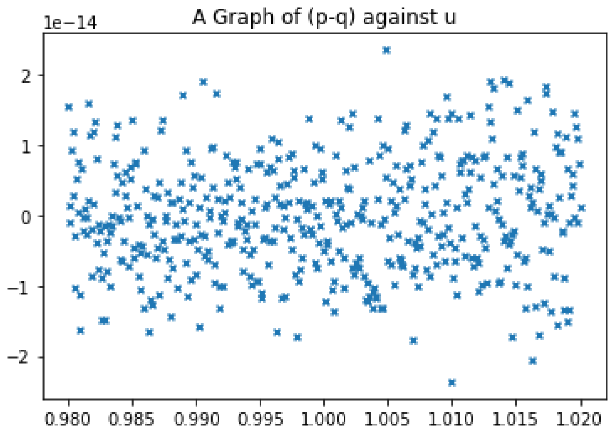


Figure 2:

Figure 3:

Where the standard deviation can be estimated with the equation:

$$\sigma_{est} = C\sqrt{N}\sqrt{\bar{x}^2} \tag{7}$$

And C is the error constant, we will be using the value of $C = 10^{-16}$. N is the number of terms, and $\bar{x}^2$ is the mean of the data set (in this case u), squared.

This histogram shown in Figure 2 shows a normal distribution around the zero value. this is to be expected because we expect the difference between the p and q functions to be close to zero for most values, with anomalies occurring less frequently more larger differences.

The standard deviation, σ_{p-q}, of this data set was found, using the numpy function to be $\sigma_{p-q} = 7.98635142589447 \times 10^{-15}$. The estimation for this standard deviation using Equation (7) to be $\sigma_{est} = 2.23606797749979 \times 10^{-15}$. It can be seen that the order of magnitude is the same as the true value, which is within our desired margin of error.

Q4 Part (c)

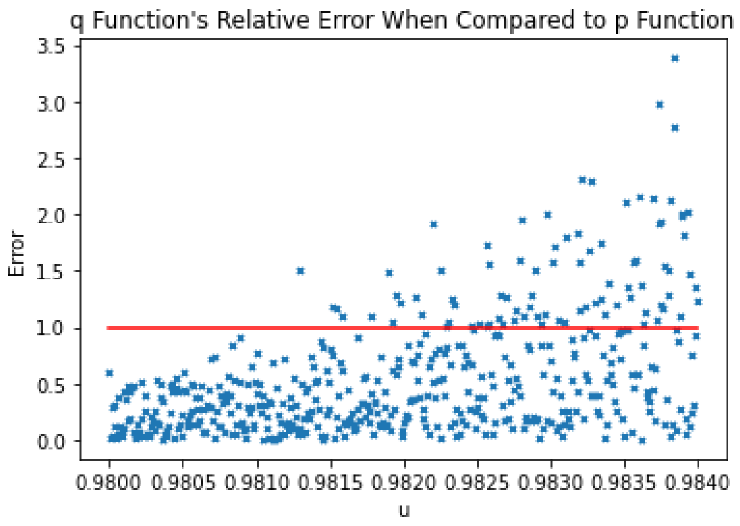


Figure 4:

It can be seen in Figure 4 that as u approaches 1, the noise increases. This results in more points with a larger error than 100 per cent, as shown on the graph with a red line.

Q4 Part (d)

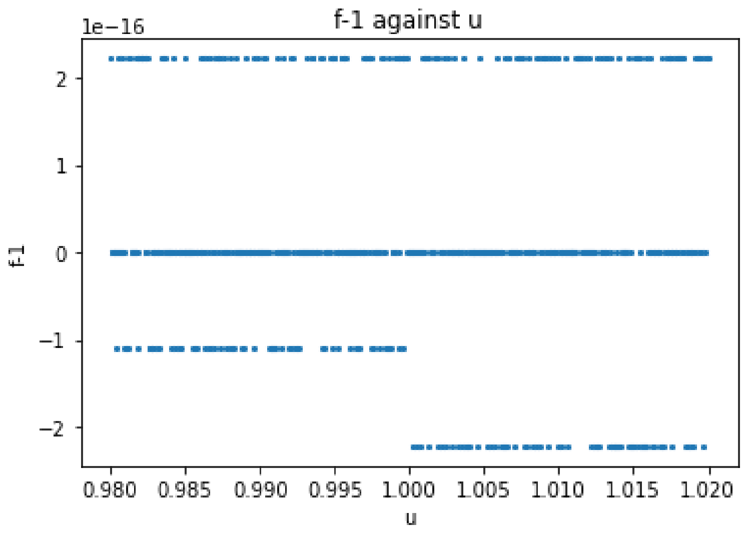


Figure 5:

Figure 5 shows the graph of f-1 vs u, and the error was found to be $\sigma = 1.4122106698406831 \times 10^{-16}$.