EE2703 Assignment 4

Nihal John George (EE19B131)

March 9, 2021

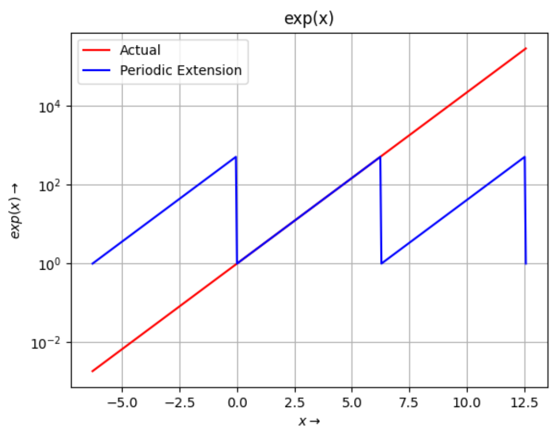
Abstract

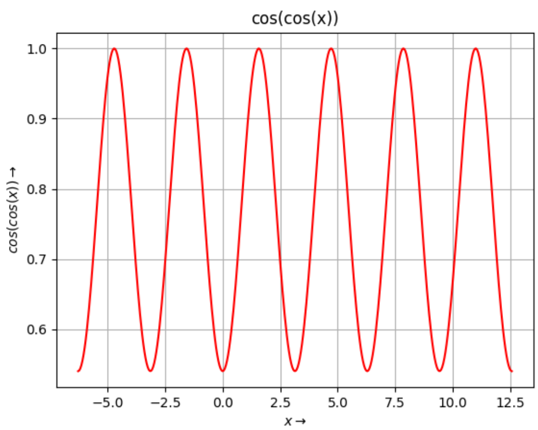
This assignment was an exploration of the Fourier Approximation method, using Numpy, Scipy, and Matplotlib in Python.
First, functions for $exp(x)$ and $cos(cos(x))$ were defined and plotted. Then the first 51 coefficients were obtained and plotted for both functions using integration. Then the least squares approach was used to find the coefficients. Its deviation from the integral method was found. The estimated functions from least squares were plotted along with the true functions, and some differences were noted.

The detailed process and observations from each of the plots is given below.

1 Visualizing The Given Functions

Firstly, vector-cum-scalar functions for $exp(x)$, $exp(x)$ by extending the $[0, 2\pi]$ period, and $cos(cos(x))$ were made. The values of the functions over $[-2\pi, 4\pi]$ were plotted. $exp(x)$ plots had only straight lines in the semilog scale, which were easier to visualise.
We notice that $exp(x)$ is not periodic, while $cos(cos(x))$ is periodic. Since Fourier series exists only for periodic functions, we can expect only a periodic extension of $exp(x)$ to be generated by our Fourier approximation. This is because the approximation is made of sines and cosines with integer multiples of a fundamental frequency, so it always has a period equal to the LCM of the individual sinusoid periods.



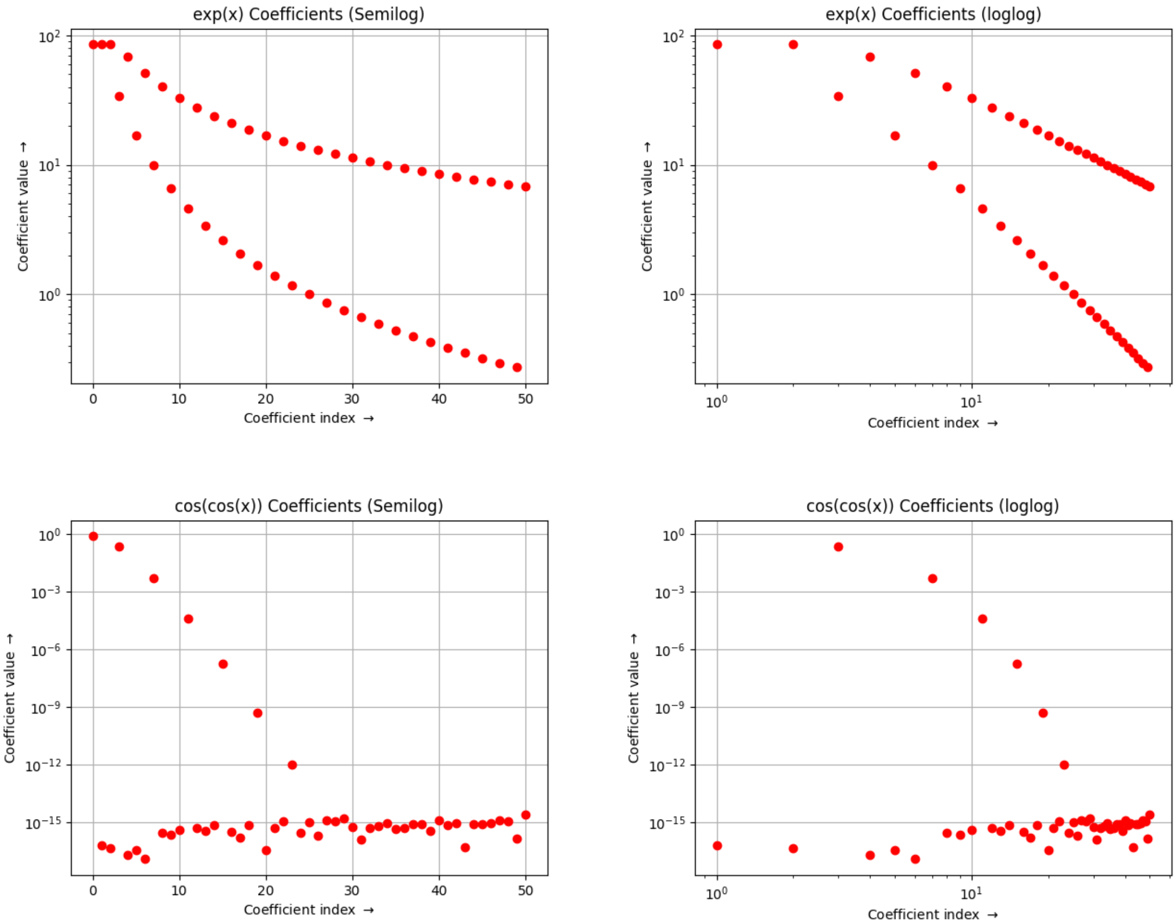


2 Fourier Integral Approach

The Fourier sine and cosine integrals were computed using $quad()$ for integration, and cosine and sine multiplication functions.

$$a_0 = \frac{1}{2\pi}\int_0^{2\pi} f(x)\,dx \qquad a_k = \frac{1}{\pi}\int_0^{2\pi} f(x)cos(kx)\,dx \qquad b_k = \frac{1}{\pi}\int_0^{2\pi} f(x)sin(kx)\,dx$$

These coefficients are plotted on the semilog and loglog scales.



(a) **The b_n are nearly zero for cos(cos(x)).** Since the function is even, the sine coefficients (b_n) are all zero (can be proved by substituting x with -x in the Fourier Sine Integral).

(b) **exp(x) coefficients do not decay as fast as cos(cos(x)) coefficients.**

On evaluating the Fourier integrals for $exp(x)$, we get

$$a_k = \frac{1}{\pi}\int_0^{2\pi} e^x cos(kx)dx = \frac{(e^{2\pi}-1)}{\pi(k^2+1)}, \qquad b_k = \frac{1}{\pi}\int_0^{2\pi} e^x sin(kx)dx = \frac{(-ke^{2\pi}+k)}{\pi(k^2+1)}$$

$$\implies log(a_k) \approx c_1 - 2log(k), \qquad log(b_k) \approx c_2 - log(k) \qquad (k >> 1)$$

So the slope of $exp(x)$ in the loglog plot is approximately a negative constant, and hence linear. The two lines in the plot are for a_k and b_k separately.

The Fourier integrals for $cos(cos(x))$ cannot be easily evaluated. However, we notice that the fundamental period is π, so fundamental angular frequency of 2 rad/s. Also due to the roughly sinusoidal shape, it suffices for a small number of low frequency sinusoids to generate the curve. Hence, we see a quick decay in higher frequencies (after coefficient index 3, which corresponds to 2 rad/s cosine)

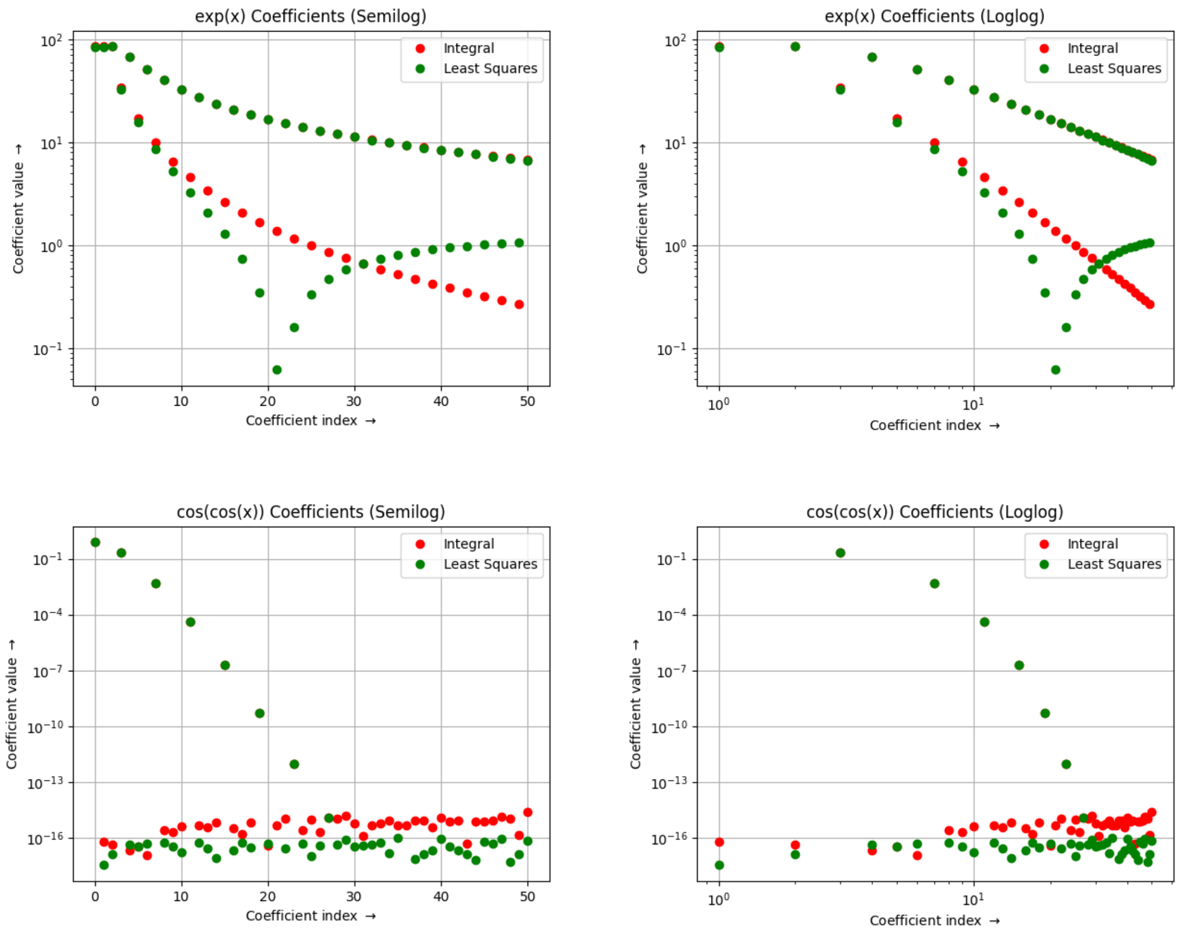
(c) **exp(x) coefficients on loglog look linear.** This is explained above

cos(cos(x)) coefficients on semilog look linear. This is because cos(cos(x)) is related to the

Bessel function $J_0(x)$ (with phase shift). This function has coefficients with exponential terms in x, which shows in the semilog plot as constant slope.

3 Least Squares Approach

This approach can be used to find the coefficients of an arbitrary relation of functions equated to a value. The function matrix of cosine and sine functions was constructed according to the question paper's needs. Then the coefficients were obtained using numpy.linalg.lstsq()
These coefficients were plotted along with those found by integration.



The deviation between the two methods was found using the following code

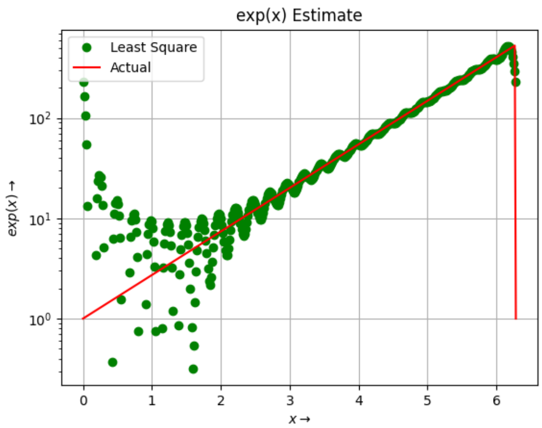
```
maxerr_e = np.max(np.abs(mixed_e-c_e))
maxerr_cc = np.max(np.abs(mixed_cc-c_cc))
```

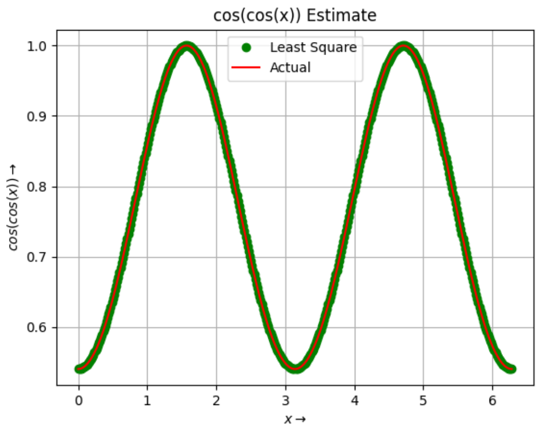
The deviation values were ≈ 1.333 for exp(x), and $\approx 2.674 \times 10^{-15}$ for cos(cos(x))

The deviation is very low for cos(cos(x)), but considerably high for exp(x). This is because the coefficients drop off very fast for cos(cos(x)) as explained earlier, and hence the lower frequencies considered in the matrix are sufficient.
In case of exp(x), the coefficients do not drop off fast enough, so the least squares matrix A does not account for many higher order sinusoids. In response to this, least squares approach may underweight or overweight certain functions (seen near coefficient 21 in exp(x)) in order to fit with the given functions. Another feature to note is the discontinuity at 0. The Fourier approximation will achieve the midpoint between the left and right hand limits if very large harmonics are included, so the deviation will be large from this neighbourhood, especially if larger harmonics were excluded during estimation.

4 Visualizing the Fourier Approximation

The plots of the least square estimates show very good approximation for cos(cos(x)) but significant deviation in exp(x) near 0. This is expected due to the discontinuity as explained earlier.





5 Conclusion

This assignment explored the method of Fourier approximation using two methods - Integrals and Least Squares. The student learnt integrating functions, and gained more experience with least squares estimation. The patterns in Fourier coefficients for $exp(x)$ and $cos(cos(x))$ were explored using various plots. The errors in estimation were explored and justified using mathematical arguments.

Through this assignment, the student has been acquainted with Numpy, Scipy and Matplotlib for the Fourier approximation task. A variety of plots like lines, semilog and loglog were used to effectively convey the key ideas.