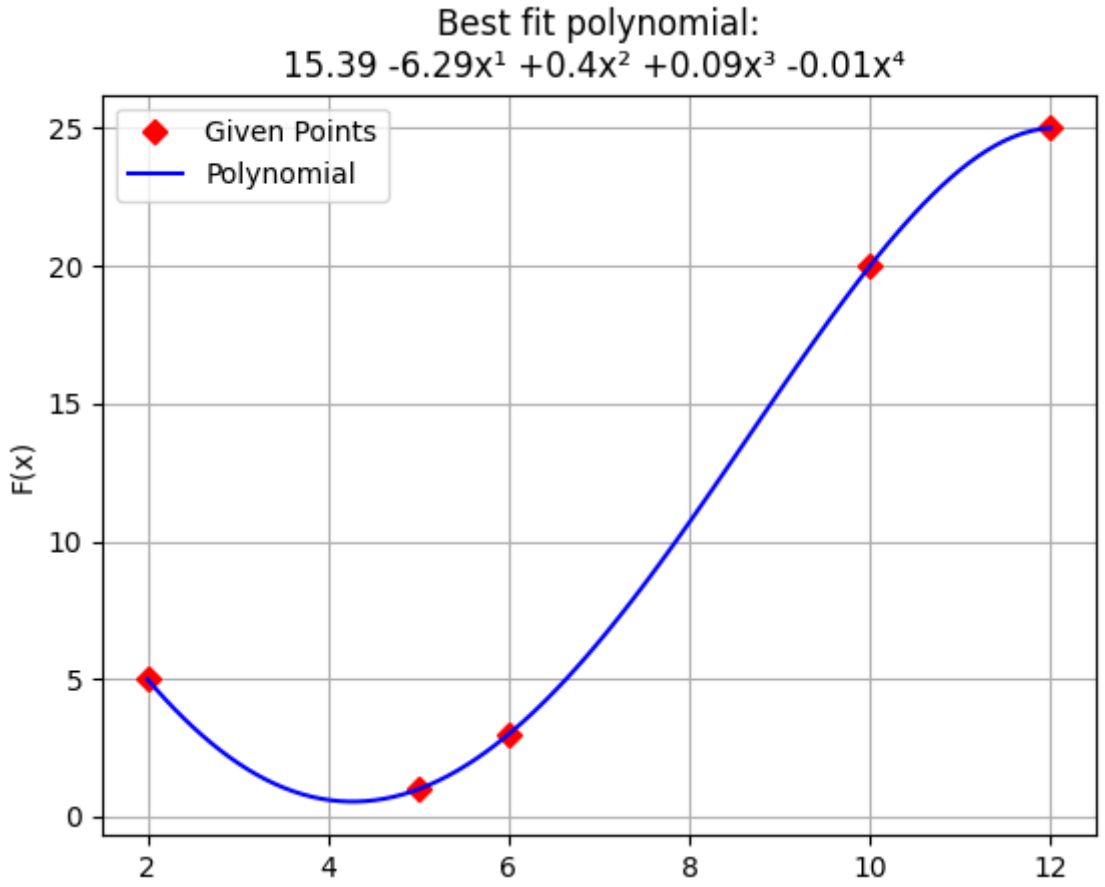
Lab Report

This assignment requires the *Polynomal()* class from the previous assignment.

Q1. The following methods are made to compute the desired results.

- *show()* : method to plot the final best fit plot and print the polynomial.
- *bestPoint()* : method to generate the best fit polynomial of degree n for the given points.

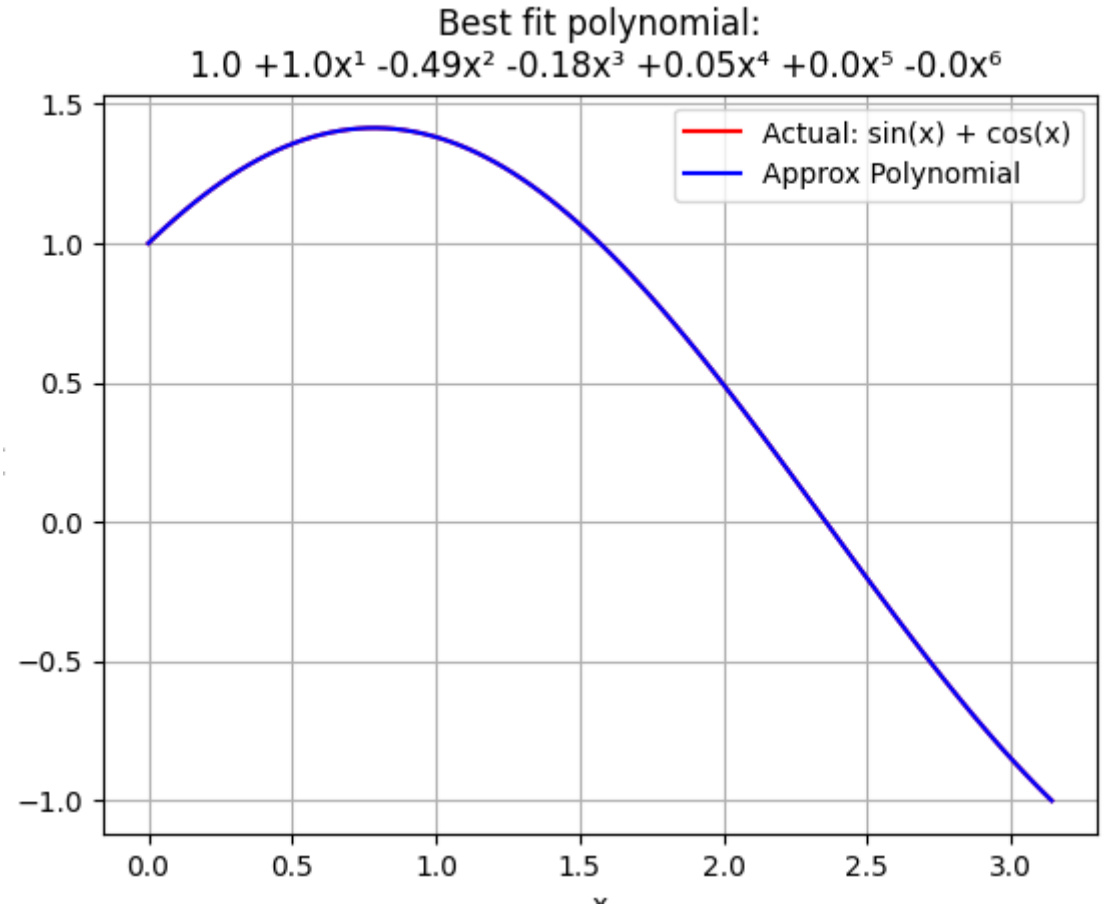
We can just call the bestPoint() function that takes the coordinates and an integer n and will plot the points and the polynomial of degree n.



Q2. The following methods are made to compute the desired results.

- ❖ *show()* : method to plot the final best fit plot and print the polynomial.
- ❖ *input()* : method to return the value of $sin(x) + cos(x)$ for a given x.
- ❖ *bestPoint()* : method to generate the best fit polynomial of degree n for the given points.
- ❖ *bestFunc()* : method to compute and graph the polynomial of degree with best approximation in the interval of [0,π].

We can just call the bestFunc() method that takes an integer n and will plot the polynomial of degree n, which will be the approximation for the $sin(x) + cos(x)$ in the interval of [0,π].



Q3. The following methods are made to compute the desired results.

- *show()* : method to plot the final best fit plot and print the polynomial.
- *__pow__()* : method to calculate the power nth power of polynomial.
- *__rpow__()* : method to solve the problem of right exponential calculation.
- *bestPoint()* : method to generate the best fit polynomial of degree n for the given points.
- *bestFunc()* : method to compute and graph the polynomial of degree with best approximation in the interval of [0,π].
- *legendrePoly()* : method to compute the Legendre polynomial of degree n.

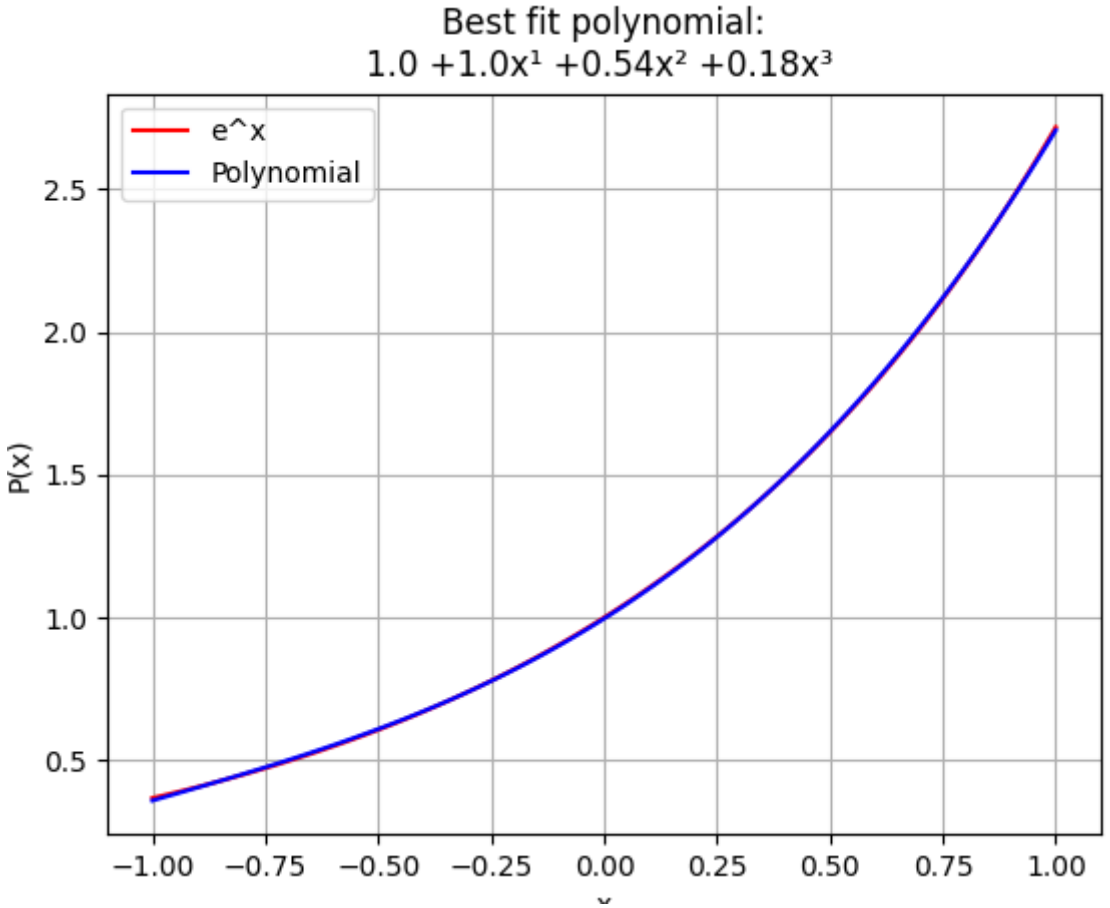
We can just call the legendrePoly() method and pass an integer n and it will return the coefficients for the legendre polynomial of degree n.

Q4.The following methods are made to compute the desired results.

- *show()* : method to plot the final best fit plot and print the polynomial.
- *__pow__()* : method to calculate the power nth power of polynomial.
- *__rpow__()* : method to solve the problem of right exponential calculation.
- *input()* : method to return the value of e^x for a given x.
- *bestPoint()* : method to generate the best fit polynomial of degree n for the given points.
- *bestFunc()* : method to compute and graph the polynomial of degree with best approximation in the interval of [0,π].

- *legendrePoly()* : method to compute the Legendre polynomial of degree n.
- *bestLegendre()* : method to compute the least square approx of degree n for the function e^x in the interval of [-1,1].

We can just call the bestLegendre() method and pass an integer n and it will plot the best fit legendre polynomial for e^x of degree n.



Q5. The following methods are made to compute the desired results.

- *chebyshevPoly()* : method to compute the nth chebyshev polynomial recursively.

We can just call the chebyshevPoly() method and pass the integer n and it will return the coefficients for the chebyshev polynomial of degree n.

Q6. The following methods are made to compute the desired results.

- *chebyshevPoly()* : method to compute the nth chebyshev polynomial.
- *wt()* : method to return the value of weight function

$$w(x) = \sqrt{1 - x^2}$$

- *orthoCheby()* : method to compute the first 5 Chebyshev polynomial and numerically demonstrate its orthogonality wrt the weight function $w(x)$.

We can just call the orthoCheby() method will will internally call the chebyPoly() to fing the first 5 Chebyshev Polynomial and then compute:

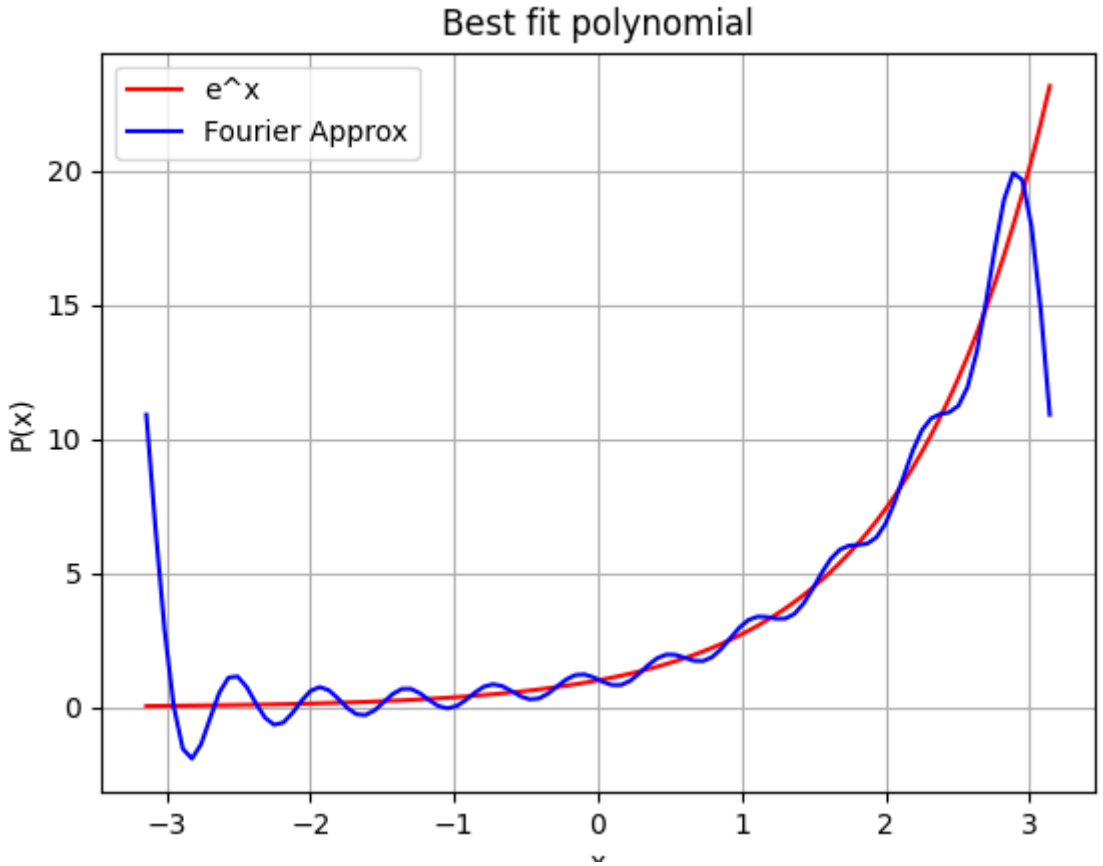
$$\int w(x) \,.\, \Phi(x).\, \Phi(y)$$

wrt to the weight function $w(x)$, for all pairs and print the results to numerically determine the orthogonality in teh interval [-1,1].

Q7. The following methods are made to compute the desired results.

- *input()* : method to return the value of $e^x,\ cos(x),\ sin(x)$ when required.
- *bestFourier()* : method to compute the best fit Fourier approx $S_n(x)$ for the function e^x in the interval [-π,π].

We can just call the bestFourier method and pass the integer n and will plot the best fit fourier approx for the function e^x and print the coefficients for the fourier series.



Q8. The following methods are made to compute the desired results.

- ❖ *calculate()* : method that will find the the multiplication for the large n digit no. using the First Fourier Transformation and its inverse.

We can just call the calculate() method and input the 2 desirable large integer and it will return the product of them in $O(nlog(n))$.