INDIAN INSTITUTE OF TECHNOLOGY PALAKKAD

COMPUTER SCIENCE AND ENGINEERING
Indian Institute of Technology, Palakkad
CS5016: Computational Methods and Applications

31 Mar, 2021

Coding Assignment 5

Least-Square Function Approximations

Max points: 100

A few instructions

- Codes should be compatible with *Python3* and should run on Ubuntu.
- Code for each question should be placed in a separate stand-alone files.
- Codes should be well-commented.
- **Appropriate exceptions should be raised and handled.**

1. Write a function to compute the polynomial of degree n that is the best fit for a given set of points. It should take as its arguments an array of tuples and an integer n, and return an object of type `Polynomial` (developed in the last coding assignment). The function is also expected to produce a plot of the input points along with the best-fit polynomial together. [20]

2. Write a function to compute the polynomial of degree n that best approximates the function $\sin(x) + \cos(x)$ in the interval $[0, \pi]$. It should take as its argument an integer n, and return an object of type `Polynomial` (developed in the last coding assignment). The function is also expected to plot the actual and approximate function together. [15]

3. Write a function that uses the enhanced `Polynomial` class (developed in the last coding assignment) to compute the n^{th} *Legendre polynomial*. It should take as its argument an integer n, and return an object of type `Polynomial` that represents the n^{th} *Legendre polynomial*. [15]

4. Write a function to compute the least-square approximation of e^x in the interval $[-1, 1]$ using the first n *Legendre polynomials*. It should take as its argument an integer n, and return an object of type `Polynomial`. Also, plot the actual and approximate function together. [10]

5. Write a function that uses the enhanced `Polynomial` class (developed in the last coding assignment) to compute the n^{th} *Chebyshev polynomial*. It should take as its argument an integer n, and return an object of type `Polynomial` that represents the n^{th} *Chebyshev polynomial*. [10]

6. Write a function to compute the first 5 *Chebyshev polynomials* and numerically demonstrate that they are *orthogonal* with respect the weight function $w(x) = 1/\sqrt{1-x^2}$ in the interval $[-1, 1]$. [10]

7. Write a function to compute coefficients of the best-fit *Fourier approximation* $S_n(x)$ of function e^x in the interval $[-\pi, \pi]$. It should take as its arguments an integer n, and print the coefficients of $S_n(x)$. The function is also expected to generate a plot with the actual and approximate function together. [10]

8. Using Python's `scipy.fft` package, write a function, with time complexity of $O(n \log n)$, to multiply two large n-digit integers. [10]

End of document